



SIM800 Series_Bluetooth _Application Note

GPRS Module

SIMCom Wireless Solutions Limited

Building B, SIM Technology Building, No.633, Jinzhong Road
Changning District, Shanghai P.R. China

Tel: 86-21-31575100

support@simcom.com

www.simcom.com

Document Title:	SIM800 Series_Bluetooth_Application Note
Version:	1.09
Date:	2020.6.15
Status:	Release

GENERAL NOTES

SIMCOM OFFERS THIS INFORMATION AS A SERVICE TO ITS CUSTOMERS, TO SUPPORT APPLICATION AND ENGINEERING EFFORTS THAT USE THE PRODUCTS DESIGNED BY SIMCOM. THE INFORMATION PROVIDED IS BASED UPON REQUIREMENTS SPECIFICALLY PROVIDED TO SIMCOM BY THE CUSTOMERS. SIMCOM HAS NOT UNDERTAKEN ANY INDEPENDENT SEARCH FOR ADDITIONAL RELEVANT INFORMATION, INCLUDING ANY INFORMATION THAT MAY BE IN THE CUSTOMER'S POSSESSION. FURTHERMORE, SYSTEM VALIDATION OF THIS PRODUCT DESIGNED BY SIMCOM WITHIN A LARGER ELECTRONIC SYSTEM REMAINS THE RESPONSIBILITY OF THE CUSTOMER OR THE CUSTOMER'S SYSTEM INTEGRATOR. ALL SPECIFICATIONS SUPPLIED HEREIN ARE SUBJECT TO CHANGE.

COPYRIGHT

THIS DOCUMENT CONTAINS PROPRIETARY TECHNICAL INFORMATION WHICH IS THE PROPERTY OF SIMCOM WIRELESS SOLUTIONS LIMITED COPYING, TO OTHERS AND USING THIS DOCUMENT, ARE FORBIDDEN WITHOUT EXPRESS AUTHORITY BY SIMCOM. OFFENDERS ARE LIABLE TO THE PAYMENT OF INDEMNIFICATIONS. ALL RIGHTS RESERVED BY SIMCOM IN THE PROPRIETARY TECHNICAL INFORMATION , INCLUDING BUT NOT LIMITED TO REGISTRATION GRANTING OF A PATENT , A UTILITY MODEL OR DESIGN. ALL SPECIFICATION SUPPLIED HEREIN ARE SUBJECT TO CHANGE WITHOUT NOTICE AT ANY TIME.

SIMCom Wireless Solutions Limited

Building B, SIM Technology Building, No.633 Jinzhong Road, Changning District, Shanghai P.R. China

Tel: +86 21 31575100

Email: _simcom@simcom.com

For more information, please visit:

<https://www.simcom.com/download/list-863-en.html>

For technical support, or to report documentation errors, please visit:

<https://www.simcom.com/ask/> or email to: support@simcom.com

Copyright © 2020 SIMCom Wireless Solutions Limited All Rights Reserved.

About Document

Version History

Version	Date	Chapter	What is new
V1.00	2013.11.07	Original	New version
V1.01.	2014.03.26	1.4	Add "power-saving mode" description
		2.6 AT+BTSCAN	add <rssi> parameter
		2.13 AT+BTSPGET	Modify parameter
		2.14 AT+BTSPSEND	Modify parameter
		2.22 AT+BTVTS	Add command
		2.23 AT+BTCIND	Add command
		2.24 AT+BTCLCC	Add command
		2.25 AT+BTPBSYNC	Add command
		2.26 AT+BTPBF	Add command
		2.27 AT+BTAIRCOP	Add command
		2.28 AT+BTVIS	Add command
		2.29 AT+BTSPCFG	Add command
		2.30 AT+BTPAIRCFG	Add command
		3	Add Error Code 1051,1056--1058,1060
4	Add 4.7--4.17		
V1.02	2014.06.30	2.13 AT+BTSPGET	Modify command and <command> description
		2.31 AT+CPBFEX	Add command
		2.32 AT+BTRING	Add command
		4.12	Modify demo
		5	Add
V1.03	2015.01.12	2.14 AT+BTSPSEND	Modify usage
		2.25	Modify description of <fail_num>
		2.29 AT+BTSPCFG	Modify command
		2.3 AT+BTPOWER	Modify command note
		2.31 AT+CPBFEX	Modify command
		2.32 AT+BTRING	Modify command
		2.33 AT+BTACI	Add command
		2.34 AT+BTHFGOP	Add command
2.35 AT+BTSPPURC	Add command		

		5.2 AT+CPBF	Modify command difference
V1.04	2015.02.09		Add SIM800C
		2.36 AT+BTCLCCS	Add command
V1.05	2015.08.06		Add SIM800A,SIM800F
		1.5	Add
		1.6	Add
		2.37 AT+BTSPPCFD	Add command
		2.38 AT+BTCOD	Add command
V1.06	2017.09.28	2.40 AT+BLESREG	Add command
		2.41 AT+BLESREG	Add command
		2.42 AT+BLESSAD	Add command
		2.43 AT+BLESSRM	Add command
		2.44 AT+BLESSC	Add command
		2.45 AT+BLESSD	Add command
		2.46 AT+BLESSTART	Add command
		2.47 AT+BLESSTOP	Add command
		2.48 AT+BLESSTART	Add command
		2.49 AT+BLESSTOP	Add command
		2.50 AT+BLEADV	Add command
		2.51 AT+BLECPU	Add command
		2.52 AT+BLESIND	Add command
		2.53 AT+BLESRSP	Add command
		2.54	Notify when connection's status change
		2.55 AT+BLEFMP	Add command
		2.56	Notify when connection's status change comes +BLEFMPCON
		2.57	Notify when a client's write request comes +BLEFMPWREQ
		2.58 AT+BLEXP	Add command
		2.59	Notify when connection's status change comes +BLEXPXCON
		2.60	Notify when a Link loss alert comes +BLEXPXLLAT
		2.61	Notify when a disconnection alert comes +BLEXPXDISAT
V1.07	2017.10.31	1.6	Add SIM868E
		Appendix	Add BLE profiles
V1.08	2018.10.31	2.59--2.85	Add BLE Client feature, module can scan\connect\read\write remote devices
		4.28	BLE client example
V1.09	2020.06.15	All	

Scope

The document can apply to all SIM800 series modules with Bluetooth function.

SIMCom
Confidential

Contents

About Document	3
Version History	3
Scope	5
Contents	6
1 Introduction	10
1.1 Purpose of the document.....	10
1.2 Related documents	10
1.3 Conventions and abbreviations.....	10
2 Bluetooth Function	13
2.1 Bluetooth Introduction	13
2.2 Bluetooth Profile.....	13
2.3 Bluetooth Device Address	13
2.4 AT Interface for Bluetooth Function.....	14
2.5 Multi Device Connection	14
2.6 Function Differences	14
3 AT Command	16
3.1 AT+BTHOST Inquiry and Set Host Device Name.....	18
3.2 AT+BTSTATUS Inquiry Current BT Device Status.....	18
3.3 AT+BTPOWER Power on/off BT Radio	20
3.4 AT+BTLPCR Modify the Bluetooth Transmit Power.....	20
3.5 AT+BTPAIR Pair BT Device	21
3.6 AT+BTUNPAIR Un-pair BT Device	22
3.7 AT+BTSCAN Scan Surrounding BT Device	22
3.8 AT+BTCONNECT Connect Paired BT Device.....	23
3.9 AT+BTDISCONN Disconnect BT Connection.....	24
3.10 AT+BTGETPROF Get Profile Provided by Paired Device	24
3.11 AT+BTACPT Accept Connecting Request.....	25
3.12 AT+BTOPPACPT Accept OPP Service	26
3.13 AT+BTOPPPUSH Push OPP Object to Paired Device.....	26
3.14 AT+BTSPGET Get Data Based on SPP Service	27
3.15 AT+BTSPSEND Send Data Based on SPP Service	28
3.16 AT+BTATA Answer Incoming Call	29
3.17 AT+BTATDL Redial Last Number	29
3.18 AT+BTATH Hung up Voice Call	30
3.19 AT+BTVGS Configure Voice Volume.....	30
3.20 AT+BTVGM Configure MIC Gain Level	31
3.21 AT+BTATD Dial Voice Call.....	31

3.22	AT+BTRSSI	Get RSSI of Connected BT Device	32
3.23	AT+BTVTS	Send DTMF Tone	32
3.24	AT+BTCIND	Get Status of Smart Phone	33
3.25	AT+BTCLCC	Get Call Status of Smart Phone	34
3.26	AT+BTPBSYNC	Sync Phonebook from Remote by BT	35
3.27	AT+BTPBF	Find Name or Number from Remote by BT	37
3.28	AT+BTAVRCOP	AVRCP Operation	38
3.29	AT+BTVIS	Set Visibility of BT	39
3.30	AT+BTSPPCFG	SPP Configuration	39
3.31	AT+BTPAIRCFG	Set BT Pairing Mode	40
3.32	AT+CPBFEX	Find Name or Number in Module Phonebook	41
3.33	AT+BTRING	Control Ring Playing Transferred from Phone	42
3.34	AT+BTACI	Set Report Mode of BT Audio Service State Change	42
3.35	AT+BTHFGOP	Set Action Mode of MS When Earphone Button is Pressed During BT Link	43
3.36	AT+BTSPPURC	Set the Report Format of Command +BTSPSEND	44
3.37	AT+BTCLCCS	Get Call Status of Smart Phone	45
3.38	AT+BTSPPCFD	Set String of SPP Switching Work Mode	46
3.39	AT+BTCOD	Set the Bluetooth Class of Device	47
3.40	AT+BLESREG	Register GATT Server	48
3.41	AT+BLESREG	Deregister GATT Server	48
3.42	AT+BLESSAD	Add a Service	49
3.43	AT+BLESSRM	Remove a Service	50
3.44	AT+BLESSC	Add a Characteristic to an Existed Service	50
3.45	AT+BLESSD	Add a Descriptor to an Existed Service	52
3.46	AT+BLESSTART	Start a Service	53
3.47	AT+BLESSTOP	Stop a Service	53
3.48	AT+BLESSTART	Start Advertising	54
3.49	AT+BLESSTOP	Stop Advertising	55
3.50	AT+BLEADV	Set Advertising Parameters	55
3.51	AT+BLESSTATUS	Inquiry Current BLE Connect Status	56
3.52	AT+BLEADDR	Inquiry Current Ble Address	57
3.53	AT+BLEDISCONN	Disconnect BLE Connection	58
3.54	AT+BLESIND	Send an Indication to a Client	58
3.55	AT+BLESRSP	Send a Response to a Client'S Read or Write Operation	59
3.56	+BLESCON	Notify When a Connection's Status Change	60
3.57	AT+BLECREG	Register GATT Client	61
3.58	AT+BLECDREG	Deregister GATT Client	61
3.59	AT+BLESCAN	Scan Surrounding BLE Device	62
3.60	+BLESCANRST	Notify When Find a BLE Device Comes	63
3.61	AT+BLECGDT	Get Device Type Request	63
3.62	AT+BLECCON	Connect GATT Client to Remote LE/Dual-mode Device	64
3.63	AT+BLECDISC	Disconnect GATT Client to Remote Device	65
3.64	AT+BLECSS	Search Peer's Service Description	65
3.65	AT+BLECGC	Search Peer's Characteristic	66
3.66	AT+BLECGD	Search Peer's Descriptor	67
3.67	AT+BLECRC	Read Peer's Characteristic	68

3.68	+BLECRC	Notify When Get a Value from Peer's Device Comes	69
3.69	AT+BLECWC	Write Peer's Characteristic	69
3.70	AT+BLECRD	Read Peer's Descriptor	70
3.71	+BLECRD	Notify When Get a Value From Peer's Device Comes	71
3.72	AT+BLECWD	Write Peer's Descriptor	71
3.73	AT+BLECRN	Register Notification Request	72
3.74	+BLECN	Notify When Get a Value from Peer's Device Comes	73
3.75	AT+BLEFMP	Deregister a FMP Service	74
3.76	+BLEFMPCON	Notify when a connection's status change comes	74
3.77	+BLEFMPWREQ	Notify When a Client's Write Request Comes	75
3.78	AT+BLEPXP	Deregister PXP Service	75
3.79	+BLEPXPCON	Notify When a Connection Comes	76
3.80	+BLEXPWREQ	Notify When a Write Request Comes	76
3.81	+BLEPXPCON	Notify When a Disconnection Alert Comes	76
3.82	AT+BLESP	Deregister a SPP Service	77
3.83	+BLESPCON	Notify When a Connection's Status Change Comes	77
3.84	+BLESPWREQ	Notify When a Client's Write Request Comes	77
3.85	AT+BLESPSIND	Send an Indication to SPP Server	78
4	CME Error Code		79
5	Bluetooth Examples		81
5.1	Accept request from other BT device		81
5.2	Send pairing request to other BT device		81
5.3	Get the profile provided by paired device		82
5.4	Connect service		83
5.5	Accept file from paired device		83
5.6	Send file to other paired BT device		83
5.7	Create SPP's link as a client		84
5.8	SPP's link be create as a server		84
5.9	Configure SPP		84
5.10	Send data as a SPP's client		85
5.11	As a SPP's server worked in AT mode		86
5.12	As a SPP's server worked in APP mode and multi-connection		86
5.13	Sync phonebook from remote by BT		88
5.14	Find name or number from remote by BT		88
5.15	Play music and so on by AVRCP		89
5.16	Add phonebook records to ME or SM phonebook from VCARD file		90
5.17	Set BT pairing mode		91
5.18	Inquiry current ble address		92
5.19	Set Adverting Parameters		92
5.20	Setup GATT server		92
5.21	Data transmission between module and client		93
5.22	Setup FMP server		94
5.23	Setup PXP server		94
5.24	Setup SPP server		95
5.25	Inquiry current ble status		95

5.26	Module disconnect with APP.....	96
5.27	Module disconnect Start or stop advertising	96
5.28	BLE client.....	96
6	Differences between bluetooth version and standard Version.....	99
6.1	ATD<str>.....	99
6.2	AT+CPBF.....	99
6.3	AT+CPBFEX.....	99
6.4	AT+CMUX.....	100
6.5	AT+CNUM.....	100
6.6	AT+CMGS.....	100
6.7	AT+CMSS.....	100
6.8	AT+CPMS.....	100
6.9	AT+CHFA.....	101
6.10	TTS function.....	101

SIMCom
Confidential

1 Introduction

1.1 Purpose of the document

This document describes how to use the AT command about Bluetooth and some application note.

Developers could understand and develop application quickly and efficiently based on this document.

1.2 Related documents

[1] SIM800 Series_AT Command Manual

1.3 Conventions and abbreviations

Profile

Profile	Introduction
SPP	Abbreviation of Serial Port Profile, to implement BT serial port function. Module a transmit data to connected BT device through AT+BTSPSEND after successfully applying this profile. The module will receive data report +BTSPDATA in automatic mode, and +BTSPMAN in manual mode.
OPP	Abbreviation of OPP Object Push Profile, to implement pushing BT object. This function is used between the two paired BT devices, AT+BTOPPPUSH to push file, AT+OPPACPT to receive the pushed file.
HFP/HSP	Abbreviation of Hands free Profile/Headset Profile, i.e. BT earphone function. HFP is the enhanced version of HSP, so even if the other BT device just supports HSP, SIM800H still can connect the BT device with HFP. Module's call voice would be displayed from BT earphone after this profile being connected. When the module play a role as smart phone, BT earphone could control the call operation (e.g. hang up, answer, redial).
A2DP	Abbreviation of Advanced Audio Distribution Profile, which is advanced protocol

	for audio frequency distribution. Earphone will activate AVRCP connection after the profile being connected. It is mainly used to for BT earphone to transmit Hi-Q audio frequency. If be suffixed with source, it means this device is audio frequency source, i.e. play a role as smart phone.
AVRCP	Abbreviation of Audio Video Remote Control Profile. It is AV remote control protocol. This profile depends on A2DP and only could be connected after the A2DP connection is established. It is mainly used for BT earphone to control the media function of smart phone. If be suffixed with target, it means this device is controlling target, i.e. play a role as smart phone.
HFP(AG)	This profile is HFP, i.e. play a role as BT earphone. After the module connected with smart phone, the call voice of smart phone could be displayed by the module's audio channel. Also the call operation of smart phone can be controlled by those commands such as AT+BTATD, AT+BTATH, AT+BTATA.
HFG	This profile is HFP, but plays a role as smart phone at this moment. After the module connected with smart phone, there will display such information indicates profile being connected successfully. If the module plays a role of earphone, then the information displayed after connection will be HFP(AG).
PBAP	Phone Book Access Profile (PBAP) is a profile that allows exchange of Phone Book Objects between devices.
BLEFMP	Find Me Profile (FMP) , The mobile terminal can send data to the module to identify the current phone calls, SMS, alarm clock or find module location.
BLEPXP	PXP Profile, Support all the functions of FMP, you can set the URC report after the disconnection.
BLESPP	To implement BLE serial port function.

Glossary and Abbreviation

Glossary	Description
EVB	Evaluation Board
BT	Blue tooth
PROFILE	Bluetooth function protocol
SPP	Serial Port Profile
OPP	OPP Object Push Profile
A2DP	Advanced Audio Distribution Profile
AVRCP	Audio Video Remote Control Profile
HSP	BT handset protocol
HFP	Hand Free application protocol
URC	Unsolicited Result Code
TE	Terminal Equipment
TA	Terminal Adapter
DTE	Data Terminal Equipment
DCE	Data Communication Equipment

ME	Mobile Equipment
MS	Mobile station
PBAP	Phone Book Access Profile

SIMCom
Confidential

2 Bluetooth Function

2.1 Bluetooth Introduction

Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength radio transmissions in the ISM band from 2400–2480 MHz) from fixed and mobile devices, creating personal area networks (PANs) with high levels of security. Bluetooth was standardized as IEEE 802.15.1.

The Bluetooth version is BT3.0(all projects) and BLE(only SIM868E).

2.2 Bluetooth Profile

To use Bluetooth wireless technology, a device has to be able to interpret certain Bluetooth profiles, which are definitions of possible applications and specify general behaviors that Bluetooth enabled devices use to communicate with other Bluetooth devices. These profiles include settings to parameterize and to control the communication from start. Adherence to profiles saves the time for transmitting the parameters anew before the bi-directional link becomes effective. There are a wide range of Bluetooth profiles that describe many different types of applications or use cases for devices.

2.3 Bluetooth Device Address

The Bluetooth device address stores the network address of a Bluetooth-enabled device. It is used to identify a particular device during operations such as connecting to, pairing with, or activating the device.

A Bluetooth-enabled device address is a unique, 48 bits address containing the following three fields:

- LAP field: lower part of the address containing 24 bits.
- UAP field: upper part of the address containing 8 bits.
- NAP field: non-significant part of the address containing 16 bits.

The LAP and the UAP represent the significant address part (SAP) of the Bluetooth device address.

2.4 AT Interface for Bluetooth Function

As module solution, we provide series of AT interface to operate Bluetooth function, including pairing, bonding, pushing or receiving file.

Also including interface for SPP service which could communicate between Bluetooth device and others via serial port.

When the module as a Bluetooth headset role, we provide a set of AT commands to control the remote smart phones, such as phone calls, turn on or hang up calls and so on.

By default, the module operates in power-saving mode, which means that the module can be simultaneously connected to a Bluetooth device. When the module to establish a connection with a device, other devices does not be scanned into the module, the module does not get profile, will not be able to establish new connections and modules. If the customer's application scenario, the module needs to be multiple Bluetooth devices (currently up to three) connection, you need to use the AT+BTSPPCFG=1 command to turn off the power saving mode. It should be noted that the power saving mode does not affect the module initiative to connect to other Bluetooth devices.

2.5 Multi Device Connection

For the MTK6260 platform module, by default, the module works in power saving mode, which means that the module can only be connected to a Bluetooth device. When the module is connected with a certain device, other devices do not scan to the module, but also unable to obtain the module's Profile and do not establish a new connection with the module. If the customer's application scenario, the need for the module is connected to a number of Bluetooth devices (currently up to three), then you need to use the AT+BTSPPCFG=1 command to shut down the power saving mode. Note that the power saving mode does not affect the module's initiative to connect to other Bluetooth devices.

2.6 Function Differences

The current Bluetooth module series can be divided into four platforms, these two platforms to support the

Bluetooth function will be different, divided as follows:

MTK6260 platforms: SIM800, SIM800M64, SIM800H.

MTK6261 platforms: SIM808, SIM800C, SIM800A, SIM800F.

MTK6261_DS platforms: SIM800C-DS.

MTK2503 platforms: SIM868, SIM868E.

➤ support Profile

All of the SIM800 series modules have four basic profiles, they are OPP, HSP/HFP, SPP.

For the MTK6260 platform module, support A2DP, AVRCP, PBAP all the roles.

For the MTK6261 and MTK2503 platform module, support PBAP all the roles and only supports A2DP, AVRCP mobile role.

For the MTK2503 platform module SIM868E, additionally support BLEFMP, BLEXP, BLESPP, Customer can also define their own GATT server.

➤ Multi-device connection

For the MTK6260 and MTK6261_DS platform module, supports simultaneous connection of multiple devices, up to 3.

For the MTK6261 and MTK2503 platform module, only supports the simultaneous connection of 1 device.

➤ The difference of the AT command

For the MTK6260 and MTK6261_DS platform module, access to the phone call status of the AT command is: AT+BTCLCC; the default SPP server mode is AT channel mode; Bluetooth open state will be saved when shutdown.

For the MTK6261 and MTK2503 platform module, access to the phone call status of the AT command is: AT+BTCLCCS; the default SPP server mode is the APP data mode; Bluetooth open state is not saved when shutdown.

AT commands of BLE are supported on MTK2503 platform module SIM868E.

3 AT Command

Command	Description
AT+BTHOST	Inquiry and set host device name
AT+BTSTATUS	Inquiry current BT device status
AT+BTPOWER	Power on or power off BT radio
AT+BTLPWR	Modify the Bluetooth transmit power
AT+BTPAIR	Pair BT device
AT+BTSCAN	Scan surrounding BT device
AT+BTUNPAIR	Un-pair BT device
AT+BTCONNECT	Connect paired BT device
AT+BTDISCONN	Disconnect BT device
AT+BTGETPROF	Get profile provided by paired device
AT+BTACPT	Accept connecting request
AT+BTOPPACPT	Accept OPP service
AT+BTOPPPUSH	Push OPP object to paired device
AT+BTSPSEND	Send data based on SPP service
AT+BTSPGET	Get data based on SPP service
AT+BTATA	Answer incoming call
AT+BTATDL	Redial last number
AT+BTATH	Hung up voice call
AT+BTVGS	Configure voice volume
AT+BTVGM	Configure MIC volume
AT+BTATD	Dial up a voice call
AT+BTRSSI	Get RSSI of connected device
AT+BTVTS	Send DTMF tone
AT+BTCIND	Get status of smart phone
AT+BTCLCC	Get call status of smart phone
AT+BTPBSYNC	Sync phonebook from remote by BT
AT+BTPBF	Find name or number from remote by BT
AT+BTAVRCOP	AVRCP operation
AT+BTVIS	Set visibility of BT
AT+BTSPPCFG	SPP configure
AT+BTPAIRCFG	Set BT pairing mode
AT+CPBFEX	Find name or number in module phonebook
AT+BTRING	Control ring playing transferred from phone
AT+BTACI	Set report mode of BT audio service state change

AT+BTHFGOP	Set action mode of MS when earphone button is pressed during BT link
AT+BTSPPURC	Set the report format of command +BTSPSEND
AT+BTCLCCS	Get call status of smart phone
AT+BTSPPCFD	Set string of SPP switching work mode
AT+BTCOD	Set the Bluetooth class of device
AT+BLESREG	Register GATT Server
AT+BLESREG	Deregister GATT Server
AT+BLESSAD	Add a service
AT+BLESSRM	Remove a service
AT+BLESSC	Add a characteristic to an existed service
AT+BLESSD	Add a descriptor to an existed service
AT+BLESSSTART	Start a service
AT+BLESSSTOP	Stop a service
AT+BLESSTART	Start advertising
AT+BLESSTOP	Stop advertising
AT+BLEADV	Set adverting parameters
AT+BLESTATUS	Inquiry current BLE connect status
AT+BLEADDR	Inquiry current BLE address
AT+BLEDISCONN	Disconnect BLE connection
AT+BLESIND	Send an indication to a client
AT+BLESRSP	Send a response to a client's read or write operation
	+BLESCON Notify when a connection's status change
AT+BLECREG	Register GATT Client
AT+BLECDREG	Deregister GATT Client
AT+BLESCAN	Scan surrounding BLE device
	Notify when find a BLE device comes +BLESCANRST
AT+BLECGDT	Get device type request
AT+BLECCON	Connect GATT client to remote LE/Dual-mode device
AT+BLECDISC	Disconnect GATT client to device
AT+BLECSS	Search peer's service Description
AT+BLECGC	Search peer's characteristic
AT+BLECGD	Search peer's descriptor
AT+BLECRC	Read peer's characteristic
	Notify when get a value from peer's device comes +BLECRC
AT+BLECWC	Write peer's characteristic
AT+BLECRD	Read peer's descriptor
	Notify when get a value from peer's device comes +BLECRD
AT+BLECWD	Write peer's descriptor
AT+BLECRN	Register notification Request
	Notify when get a value from peer's device comes +BLESCN
AT+BLEFMP	(De)Register a FMP Service

	Notify when connection's status change comes +BLEFMPCON
	Notify when a client's write request comes +BLEFMPWREQ
AT+BLEXP	(De)Register PXP Service
	Notify when connection's status change comes +BLEXPXPCON
	Notify when a Write request comes +BLEXPWREQ
	Notify when a disconnection alert comes +BLEXPXPCON
AT+BLESPP	(De)Register a SPP Service
	Notify when connection's status change comes +BLESPPCON
	Notify when a client's write request comes +BLESPPWREQ
AT+BLESPPSIND	Send an indication to SPP server

3.1 AT+BTHOST Inquiry and Set Host Device Name

AT+BTHOST Inquiry and Set Host Device Name	
Test command AT+BTHOST=?	Response +BTHOST: (1-18) OK
	Parameters See Write Command
Read command AT+BTHOST?	Response +BTHOST: <name>,<address> OK
	Parameters See Write Command
Write command AT+BTHOST=<name>	Response OK
	Parameters <name> Device name <address> Device address
Note	Max length of <name> is 18 bytes, and display in UTF-8 code.

3.2 AT+BTSTATUS Inquiry Current BT Device Status

AT+BTSTATUS Inquiry Current BT Device Status

<p>Test Command AT+BTSTATUS=?</p>	<p>Response OK</p> <hr/> <p>Parameters See Read Command</p>																																																													
<p>Read Command AT+BTSTATUS?</p>	<p>Response</p> <p>If unpaired before: +BTSTATUS: <status></p> <p>If paired before but unconnected: +BTSTATUS: <status> P: <paired id>,<name>,<address></p> <p>If paired and connected: +BTSTATUS: <status> P: <paired id>,<name>,<address> C: <connected id>,<name>,<address>,<profile name></p> <hr/> <p>OK</p> <hr/> <p>Parameters</p> <table border="0"> <tr> <td><status></td> <td>0</td> <td>Initial</td> </tr> <tr> <td></td> <td>1</td> <td>Deactivating</td> </tr> <tr> <td></td> <td>2</td> <td>Activating</td> </tr> <tr> <td></td> <td>5</td> <td>Idle</td> </tr> <tr> <td></td> <td>6</td> <td>Inquiry</td> </tr> <tr> <td></td> <td>7</td> <td>Inquiry Res Indication</td> </tr> <tr> <td></td> <td>8</td> <td>Cancelling inquiry</td> </tr> <tr> <td></td> <td>9</td> <td>Bonding</td> </tr> <tr> <td></td> <td>11</td> <td>Pairing</td> </tr> <tr> <td></td> <td>12</td> <td>Connecting</td> </tr> <tr> <td></td> <td>14</td> <td>Deleting paired device</td> </tr> <tr> <td></td> <td>15</td> <td>Deleting all paired device</td> </tr> <tr> <td></td> <td>19</td> <td>Pairing confirm while passive pairing</td> </tr> <tr> <td></td> <td>20</td> <td>Waiting for remote confirm while passive pairing</td> </tr> <tr> <td></td> <td>25</td> <td>Accepting connection</td> </tr> <tr> <td></td> <td>26</td> <td>SDC refreshing</td> </tr> <tr> <td></td> <td>29</td> <td>Setting host name</td> </tr> </table> <table border="0"> <tr> <td><paired id></td> <td>Paired device ID</td> </tr> <tr> <td><connected id></td> <td>Connected device ID</td> </tr> <tr> <td><name></td> <td>Device name</td> </tr> <tr> <td><address></td> <td>Device address</td> </tr> <tr> <td><profile name></td> <td>Profile</td> </tr> </table>	<status>	0	Initial		1	Deactivating		2	Activating		5	Idle		6	Inquiry		7	Inquiry Res Indication		8	Cancelling inquiry		9	Bonding		11	Pairing		12	Connecting		14	Deleting paired device		15	Deleting all paired device		19	Pairing confirm while passive pairing		20	Waiting for remote confirm while passive pairing		25	Accepting connection		26	SDC refreshing		29	Setting host name	<paired id>	Paired device ID	<connected id>	Connected device ID	<name>	Device name	<address>	Device address	<profile name>	Profile
<status>	0	Initial																																																												
	1	Deactivating																																																												
	2	Activating																																																												
	5	Idle																																																												
	6	Inquiry																																																												
	7	Inquiry Res Indication																																																												
	8	Cancelling inquiry																																																												
	9	Bonding																																																												
	11	Pairing																																																												
	12	Connecting																																																												
	14	Deleting paired device																																																												
	15	Deleting all paired device																																																												
	19	Pairing confirm while passive pairing																																																												
	20	Waiting for remote confirm while passive pairing																																																												
	25	Accepting connection																																																												
	26	SDC refreshing																																																												
	29	Setting host name																																																												
<paired id>	Paired device ID																																																													
<connected id>	Connected device ID																																																													
<name>	Device name																																																													
<address>	Device address																																																													
<profile name>	Profile																																																													
<p>Note</p>	<p>Max length of <name> is 18 bytes, 18 bytes in UTF-8 code</p>																																																													

3.3 AT+BTPOWER Power on/off BT Radio

AT+BTPOWER Power on/off BT Radio	
Test Command AT+BTPOWER=?	Response +BTPOWER: (list of supported <n>s) OK <hr/> Parameters See Write Command
Read Command AT+BTPOWER?	Response +BTPWR: <status> OK <hr/> Parameters See Write Command
Write Command AT+BTPOWER=<n>	Response OK <hr/> parameter <n> 0 Power off BT radio 1 Power on BT radio
Note	After turning off, the BT radio shall not be re-opened until the status of BT is changed to 0. So wait for some seconds is needed. The status can be obtained by using AT+BTSTATUS.

3.4 AT+BTLPWR Modify the Bluetooth Transmit Power

AT+BTLPWR Modify the Bluetooth Transmit Power	
Read Command AT+BTLPWR?	Response +BTPWR: <status> OK <hr/> Parameters See Write Command
Test Command AT+BTLPWR=?	Response +BTPOWER: (0-7) OK <hr/> Parameters See Write Command

Write Command AT+BTLPWR=<n>	Response OK					
	parameter <table border="0"> <tr> <td><n></td> <td>0</td> <td>Reset power status to default</td> </tr> <tr> <td></td> <td>1-7</td> <td>The class of Bluetooth transmit power</td> </tr> </table>	<n>	0	Reset power status to default		1-7
<n>	0	Reset power status to default				
	1-7	The class of Bluetooth transmit power				

3.5 AT+BTPAIR Pair BT Device

AT+BTPAIR Pair BT Device															
Test Command AT+BTPAIR=?	Response +BTPAIR: 0, (list of supported <device ID> s) +BTPAIR: 1, (list of supported <confirm> s) +BTPAIR: 2, (length of supported <passkey> s) OK														
Write Command 1) active AT+BTPAIR=0,<device ID>	Response OK If digital key exchanged +BTPAIRING: <name>,<address>,<passcode>														
2) passive with digital key request AT+BTPAIR=1,<confirm>	If passkey exchanged: +BTPAIRING: <name>,<address>														
3) passive with passkey request AT+BTPAIR=2,<passkey>	If passive mode with success: +BTPAIR: <id>,<name>,<address> If passive mode with failure: +BTPAIR: 0														
	Parameters <table border="0"> <tr> <td><device ID></td> <td>BT device ID</td> </tr> <tr> <td><confirm></td> <td>1 Accept 0 Reject</td> </tr> <tr> <td><passkey></td> <td>Passkey, length is (4-16)</td> </tr> <tr> <td><id></td> <td>0 Paired failed >=1 Paired device ID</td> </tr> <tr> <td><name></td> <td>BT device name</td> </tr> <tr> <td><address></td> <td>BT device address</td> </tr> <tr> <td><passcode></td> <td>Digital password</td> </tr> </table>	<device ID>	BT device ID	<confirm>	1 Accept 0 Reject	<passkey>	Passkey, length is (4-16)	<id>	0 Paired failed >=1 Paired device ID	<name>	BT device name	<address>	BT device address	<passcode>	Digital password
<device ID>	BT device ID														
<confirm>	1 Accept 0 Reject														
<passkey>	Passkey, length is (4-16)														
<id>	0 Paired failed >=1 Paired device ID														
<name>	BT device name														
<address>	BT device address														
<passcode>	Digital password														
	URC If there is incoming request:														

	<p>+BTPAIRING: <name>,<address>,<passcode> or +BTPAIRING: <name>,<address></p> <p>Parameters</p> <p><name> Device name <address> Device address <passcode> Digital password</p>
Note	<ul style="list-style-type: none"> ● Max length of <name> is 18 bytes, 18 bytes in UTF-8 code ● Pairing timeout is around 15s each side

3.6 AT+BTUNPAIR Un-pair BT Device

AT+BTUNPAIR Un-pair BT Device	
Test Command AT+BTUNPAIR=?	Response +BTUNPAIR: (list of supported <device ID>s) OK
	Parameter See Write Command
Write Command AT+BTUNPAIR=<device ID>	Response OK
	Parameter <device ID> Paired Device ID. 0 Delete all the paired device 1 Delete the paired device corresponding to ID

3.7 AT+BTSCAN Scan Surrounding BT Device

AT+BTSCAN Scan Surrounding BT Device	
Test Command AT+BTSCAN=?	Response +BTSCAN: (list of supported <switch>s), (list of supported <Timer>s) OK
	Parameters See Write Command
Write Command	Response

<p>AT+BTSCAN=<switch>[,<Timer>]</p>	<p>OK</p> <p>If BT device scanned: +BTSCAN: <status>,<device ID>,<name>,<address>,<rssi></p> <p>If terminate: +BTSCAN: <status></p> <hr/> <p>Parameters</p> <p><switch> 1 Start 0 Stop</p> <p><status> 0 BT device found 1 Scanning finished 2 Scanning stop 3 Scanning failed</p> <p><Timer> Scanning time 10-60s</p> <p><device ID> BT device ID scanned</p> <p><name> BT device name</p> <p><address> BT device address</p> <p><rssi> -127...0 RSSI value of BT device</p> <hr/> <p>Note</p> <ul style="list-style-type: none"> ● Max length of <name> is 18 bytes, 18 bytes in UTF-8 code ● If <timer> omitted, the default value is 30s
--	---

3.8 AT+BTCONNECT Connect Paired BT Device

AT+BTCONNECT Connect Paired BT Device	
<p>Test Command</p> <p>AT+BTCONNECT=?</p>	<p>Response</p> <p>+BTCONNECT: (list of supported <device ID>s), (list of supported <profile ID>s)</p> <p>OK</p> <hr/> <p>Parameters</p> <p>See Write Command</p>
<p>Write Command</p> <p>AT+BTCONNECT=<device ID>,<profile ID></p>	<p>Response</p> <p>OK</p> <p>If OK: +BTCONNECT: <id>,<name>,<address>,<profile name></p> <p>If failed: +BTCONNECT: 0</p> <hr/> <p>Parameters</p> <p><device ID> ID of paired BT device</p> <p><profile ID> BT profile ID</p>

	<p><id> ID of connected BT device</p> <p><name> BT device name</p> <p><address> BT device address</p> <p><profile name> BT device service name</p>
Note	<ul style="list-style-type: none"> ● Max length of <name> is 18 bytes, 18 bytes in UTF-8 code ● Connection timeout is around 20s ● If incoming request, there will be URC <p>+BTCONECING: <address>,<profile name></p>

3.9 AT+BTDISCONN Disconnect BT Connection

AT+BTDISCONN Disconnect BT Connection	
Test Command AT+BTDISCONN=?	Response + BTDISCONN: (list of supported <device ID>s) OK
	Parameter See Write Command
Write Command AT+BTDISCONN=<device ID>	Response OK +BTDISCONN: <name>,<address>,<profile name>
	Parameter <device ID> Connected device ID <name> Device name <address> Device address <profile name> Profile service
Note	<ul style="list-style-type: none"> ● Max length of <name> is 18 bytes, 18 bytes in UTF-8 code ● If disconnected by remote, there still be URC: +BTDISCONN

3.10 AT+BTGETPROF Get Profile Provided by Paired Device

AT+BTGETPROF Get Profile Provided by Paired Device	
Test Command AT+BTGETPROF=?	Response +BTGETPROF: (list of supported <device ID>s) OK
	Parameters

	See Write Command
Write Command AT+BTGETPROF=<device ID>	Response OK +BTGETPROF: <profile ID>,<profile name>
	Parameters <device ID> Paired Device ID <profile ID> Profile ID <profile name> Profile name

3.11 AT+BTACPT Accept Connecting Request

AT+BTACPT Accept Connecting Request	
Test command AT+BTACPT=?	Response +BTACPT: (list of supported <confirm> s) OK
	Parameters See Write Command
Write command AT+BTACPT=<confirm>	Response OK If connected successfully, then will report: +BTCONNECT: <id>,<name>,<address>,<profile name> If connecting failed: + BTDISCONN: <name>,<address>,<profile name>
	Parameters <confirm> 1 Accept 0 Reject <id> >0 Connected device ID <name> Device name <address> Device address <profile name> Profile name
	URC If incoming connecting request: +BTCONNECTING: <address>,<profile name>
	Parameters <address> Device address <profile name> Profile name
Note	Max length of <name> is 18 bytes, and display in UTF-8 code.

3.12 AT+BTOPPACPT Accept OPP Service

AT+BTOPPACPT Accept OPP Service																							
<p>Test Command AT+BTOPPACPT=?</p>	<p>Response +BTOPPACPT: (list of supported <confirm>s),(list of supported<drv>)</p> <p>OK</p> <hr/> <p>Parameters See Write Command</p>																						
<p>Write Command AT+BTOPPACPT=<confirm> >[,<drv>]</p>	<p>Response OK</p> <p>+BTOPPPUSH: <status></p> <hr/> <p>Parameters</p> <table border="0"> <tr> <td style="padding-right: 10px;"><confirm></td> <td style="padding-right: 10px;">1</td> <td>Accept</td> </tr> <tr> <td></td> <td>0</td> <td>Reject</td> </tr> <tr> <td style="padding-right: 10px;"><drv></td> <td style="padding-right: 10px;">0</td> <td>Internal flash memory</td> </tr> <tr> <td></td> <td>1</td> <td>External memory card</td> </tr> <tr> <td style="padding-right: 10px;"><status></td> <td style="padding-right: 10px;">0</td> <td>Failed</td> </tr> <tr> <td></td> <td>1</td> <td>Successful</td> </tr> </table> <hr/> <p>URC: If there has an incoming OPP file, there will be a URC report. +BTOPPPUSHING: <name>,<file name></p> <p>Parameters</p> <table border="0"> <tr> <td style="padding-right: 10px;"><name></td> <td>Device name</td> </tr> <tr> <td style="padding-right: 10px;"><file name></td> <td>File name</td> </tr> </table>	<confirm>	1	Accept		0	Reject	<drv>	0	Internal flash memory		1	External memory card	<status>	0	Failed		1	Successful	<name>	Device name	<file name>	File name
<confirm>	1	Accept																					
	0	Reject																					
<drv>	0	Internal flash memory																					
	1	External memory card																					
<status>	0	Failed																					
	1	Successful																					
<name>	Device name																						
<file name>	File name																						
<p>Note</p>	<ul style="list-style-type: none"> ● Max length of <name> is 18 bytes, 18 bytes in UTF-8 code ● File is stored in path: C:\User\BtReceived\ for internal memory card, D:\BtReceived\ for external memory card. At the first time to use SD card, customer must execute "AT+SD2PCM=0" and "AT&W", then reboot the module. 																						

3.13 AT+BTOPPPUSH Push OPP Object to Paired Device

AT+BTOPPPUSH Push OPP Object to Paired Device

<p>Test Command AT+BTOPPPUSH=?</p>	<p>Response +BTOPPPUSH: (list of supported <device ID>s), (length of supported <string>s)</p> <p>OK</p> <hr/> <p>Parameters See Write Command</p>
<p>Write Command AT+BTOPPPUSH=<device ID>,<string></p>	<p>Response OK</p> <p>+BTOPPPUSH: <para></p> <hr/> <p>Parameters <device ID> Paired Device ID</p> <p><string>File name include complete path, length (4-259)</p> <p><para></p> <ul style="list-style-type: none"> 0 Send failed 1 Send successfully 2 Server issue
<p>Note</p>	

3.14 AT+BTSPGET Get Data Based on SPP Service

AT+BTSPGET Get Data Based on SPP Service	
<p>Test command AT+BTSPGET=?</p>	<p>Response +BTSPGET: (list of supported <command>s), (list of supported <connectId>), (list of supported <reqLength>s), (list of supported <showWithHex>s)</p> <p>OK</p> <hr/> <p>Parameters See Write Command</p>
<p>Read command AT+BTSPGET?</p>	<p>Response +BTSPGET: <command></p> <p>OK</p> <hr/> <p>Parameters See Write Command</p>
<p>Write command 1).If AT+BTSPPCFG="MC",2 response 1(Enable multi-connect)</p>	<p>Response OK or ERROR</p>

<p>AT+BTSPGET=<command>[,<connectId>][,<reqLength>][,<showWithHex>] 2).If AT+BTSPCFG="MC",2 response 0(Disable multi-connect) AT+BTSPGET=<command>[,<reqLength>][,<showWithHex>]</p>	<p>If command value is 2,return: +BTSPGET: <connectId>,<cnfLen1> OK If command value is 3,return: +BTSPGET: <connectId>,<cnfLen1>[,<data string>] OK</p> <hr/> <p>Parameters <command> 0 Auto mode. Data will be output in decimal system. 1 Manual mode. There will be an indication when first package arrives. 2 Inquiry data length in manual mode. If multi-connect enabled, this command need parameter <connectId>. 3 Getting data in manual mode. If multi-connect enabled, this command need parameter <connectId>.You can input parameters of <reqLength> and <showWithHex> when you need. <reqLength> The length of data requested, only valid in manual mode. 1-1024. <showWithHex> 1, displayed in hex, only valid in manual mode <connectId> Connection`s ID <cnfLen1> Character length.0-1024. <data string> String printed</p>
<p>Note</p>	<p>URC When the module receives data by SPP, there will be URC report: 1. Auto mode +BTSPDATA: <connectId>,<cnfLen2>,<data string> 2. Manual mode +BTSPMAN: <connectId></p> <hr/> <p>Parameters <cnfLen2> Length of printed character.1-1024.</p>

3.15 AT+BTSPSEND Send Data Based on SPP Service

AT+BTSPSEND Send Data Based on SPP Service	
<p>Write Command 1).If AT+BTSPCFG="MC",2 response 1(Enable multi-connect) AT+BTSPSEND=<connect</p>	<p>Response > If successful, SEND OK If failed,</p>

<p>Id>,<length> 2).If AT+BTSPPCFG="MC",2 response 0(Disable multi-connect) AT+BTSPSEND=<length></p>	<p>SEND FAIL Or if this <connectId> is not allowed to send data, ERROR</p> <hr/> <p>Parameters <connectId> Connection`s ID. If disable multi-connection, this parameter is no need. <length> 1-1024. The length of data will be sent. When the length of inputting data is up to <length> specified, the package will be sent out automatically.</p>
<p>Execute Command AT+BTSPSEND</p>	<p>Response > If successful, SEND OK Or failed, SEND FAIL Or if this connect Id is not allowed to send data, ERROR</p> <hr/> <ul style="list-style-type: none"> ● If multi-connection function is enabled, this command will be disabled. ● In this mode, <Ctrl+z> will send the package immediately, and ESC will quit the process.

3.16 AT+BTATA Answer Incoming Call

AT+BTATA Answer Incoming Call	
<p>Execute Command AT+BTATA</p>	<p>Response OK URC If there is incoming Call on remote phone, will report below: BTRING</p>
<p>Note</p>	<p>When module connected with smart phone as an earphone, if here comes incoming call, the call would be answered through this command</p>

3.17 AT+BTATDL Redial Last Number

AT+BTATDL Redial Last Number

Execute Command AT+BTATDL	Response OK
Note	When module connected with smart phone as an earphone, would redial last number through this command

3.18 AT+BTATH Hung up Voice Call

AT+BTATH Hung up Voice Call

Execute Command AT+BTATH	Response OK
Note	When module connected with smart phone as an earphone, the incoming call would be hung up through this command

3.19 AT+BTVGS Configure Voice Volume

AT+BTVGS Configure Voice Volume

Test command AT+BTVGS=?	Response +BTVGS: (<gain> range) OK
	Parameters See Write Command
Read command AT+BTVGS?	Response +BTVGS: <gain> OK
	Parameters See Write Command
Write command AT+BTVGS=<gain>	Response OK
	Parameters <gain> Volume This command is used configure call volume when the module is connected with smart phone as an earphone
Note	For some smart phone, after connected with BT earphone, the current call volume may not be transmitted to earphone, thus the return value

of the read command may be 0. But after setting once, the value would be correct.

3.20 AT+BTVGM Configure MIC Gain Level

AT+BTVGM Configure MIC Gain Level	
Test command AT+BTVGM=?	Response +BTVGM: (<gain> range) OK Parameters See Write Command
Read command AT+BTVGM?	Response +BTVGM: <gain> OK Parameters See Write Command
Write command AT+BTVGM=<gain>	Response OK Parameters <gain> MIC gain level This command is used set MIC volume when the module is connected with smart phone as an earphone
Note	For some smart phone, after connected with BT earphone, the current MIC volume may not be transmitted to earphone, thus the return value of the read command may be 0. But after setting once, the value would be correct.

3.21 AT+BTATD Dial Voice Call

AT+BTATD Dial Voice Call	
Test Command AT+BTATD=?	Response +BTATD: (<number> length range) OK Parameters

	See Write Command
Write Command AT+BTATD=<number>	Response OK
	Parameters <number> Phone number Module as earphone connected to smart phone, this command could make an outgoing call
Note	

3.22 AT+BTRSSI Get RSSI of Connected BT Device

AT+BTRSSI Get RSSI of Connected BT Device	
Test Command AT+BTRSSI=?	Response +BTRSSI: (list of supported <device ID> s) OK
	Parameters See Write Command
Write Command AT+BTRSSI=<device ID>	Response +BTRSSI: <rssi> OK
	Parameters <device ID> Connected Device ID <rssi> -127...0 RSSI value of BT device
Note	RSSI value is negative, the smaller value represents the worse signal

3.23 AT+BTVTS Send DTMF Tone

AT+BTVTS Send DTMF Tone	
Test Command AT+BTVTS=?	Response +BTVTS: (<dtmf> 'scope) OK
	Parameters See Write Command
Write Command	Response

AT+BTVTS=<dtmf>	OK
	Parameters <dtmf> DTMF tone
Note	When module connected with smart phone as an earphone, would send DTMF tone through this command

3.24 AT+BTCIND Get Status of Smart Phone

AT+BTCIND Get Status of Smart Phone																																														
Test command AT+BTCIND=?	Response +BTCIND: (0,1)																																													
	OK																																													
	Parameters See Write Command																																													
Read command AT+BTCIND?	Response +BTCIND: <mode>,<service>,<call>,<call_setup>,<held>,<signal>,<roam>,< battchg>																																													
	OK																																													
	Parameters																																													
	<table border="0"> <tr> <td><service></td> <td>0</td> <td>No net service</td> </tr> <tr> <td></td> <td>1</td> <td>Net service is normal</td> </tr> <tr> <td><call></td> <td>0</td> <td>Not active</td> </tr> <tr> <td></td> <td>1</td> <td>Active</td> </tr> <tr> <td><call_setup></td> <td>0</td> <td>Set up complete</td> </tr> <tr> <td></td> <td>1</td> <td>Incoming call</td> </tr> <tr> <td></td> <td>2</td> <td>Outgoing call</td> </tr> <tr> <td></td> <td>3</td> <td>Remote alert</td> </tr> <tr> <td><held></td> <td>0</td> <td>No held call</td> </tr> <tr> <td></td> <td>1</td> <td>Active calls be placed or switched</td> </tr> <tr> <td></td> <td>2</td> <td>Active calls be placed and no active call</td> </tr> <tr> <td><signal></td> <td>0..5</td> <td>Net work signal</td> </tr> <tr> <td><roam></td> <td>0</td> <td>No roaming</td> </tr> <tr> <td></td> <td>1</td> <td>In roaming</td> </tr> <tr> <td><battchg></td> <td>0..5</td> <td>Power level</td> </tr> </table>	<service>	0	No net service		1	Net service is normal	<call>	0	Not active		1	Active	<call_setup>	0	Set up complete		1	Incoming call		2	Outgoing call		3	Remote alert	<held>	0	No held call		1	Active calls be placed or switched		2	Active calls be placed and no active call	<signal>	0..5	Net work signal	<roam>	0	No roaming		1	In roaming	<battchg>	0..5	Power level
<service>	0	No net service																																												
	1	Net service is normal																																												
<call>	0	Not active																																												
	1	Active																																												
<call_setup>	0	Set up complete																																												
	1	Incoming call																																												
	2	Outgoing call																																												
	3	Remote alert																																												
<held>	0	No held call																																												
	1	Active calls be placed or switched																																												
	2	Active calls be placed and no active call																																												
<signal>	0..5	Net work signal																																												
<roam>	0	No roaming																																												
	1	In roaming																																												
<battchg>	0..5	Power level																																												
Write command AT+BTCIND=<mode>	Response OK																																													
	Parameters <mode> 1 Auto report open																																													

	0 Auto report close
	<p>Unsolicited Result Code</p> <p>When<mode>=1,anychanged in<service>,<call>,<call_setup>,<held>,<signal>,<roam>,<battchg> , an unsolicited result code is returned:</p> <p>+BTCIND:1,<service>,<call>,<call_setup>,<held>,<signal>,<roam>,<battchg></p>
Note	When module connected with smart phone as an earphone, these statuses can be gotten.

3.25 AT+BTCLCC Get Call Status of Smart Phone

AT+BTCLCC Get Call Status of Smart Phone	
Test Command AT+BTCLCC=?	<p>Response</p> <p>OK</p> <p>Parameter</p> <p>See Write Command</p>
Read Command AT+BTCLCC?	<p>Response</p> <p>OK</p> <p>When call is active:</p> <p>+BTCLCC: <index>,<dir>,<stat>,<mode>,<mpty>,<number>,<type>...</p> <p>When no call:</p> <p>+BTCLCC:0</p> <p>Parameter</p> <p><idx> 1..7 Call identification number</p> <p><dir> 0 Mobile originated (MO) call 1 Mobile terminated (MT) call</p> <p><stat> State of the call</p> <p>0 Active 1 Held 2 Dialing(MO call) 3 Alerting (Mo call) 4 Incoming (MT call) 5 Waiting (MT call)</p> <p><mode> Bearer/tele service</p> <p>0 Voice 1 Data 2 Fax</p> <p><mpty> 0 Call is not one of multiparty (conference) call parties 1 Call is one of multiparty (conference) call parties</p>

	<p><number> String type (string should be included in quotation marks) phone number in format specified by <type>.</p> <p><type> Type of address</p>
Note	<ul style="list-style-type: none"> ● If there are multi calls, multi "+BTCLCC" will be reported, but <index> is different ● MTK_6261 platform does not support this command.

3.26 AT+BTPBSYNC Sync Phonebook from Remote by BT

AT+BTPBSYNC Sync Phonebook from Remote by BT	
<p>Test Command</p> <p>AT+BTPBSYNC=?</p>	<p>Response</p> <p>+BTPBSYNC: (0,1),(1-10),(0,1),(0,1),(0,1)</p> <p>OK</p> <p>Parameters</p> <p>See Write Command</p>
<p>Write Command</p> <p>AT+BTPBSYNC=<mode>,<storage>,<loc>[,<loc_phb>[,<loc_mode>]]</p>	<p>Response</p> <p>OK</p> <p>If sync phonebook succeed in mode 0</p> <p>+BTPBSYNC: <mode>,<result>,<length></p> <p>If sync phonebook failed in mode 0</p> <p>+BTPBSYNC: <mode>,<result></p> <p>If in mode 1</p> <p>+BTPBSYNC:</p> <p><mode>,<sync2loc_result>,<succ_num>,<fail_num></p> <p>If error is related to ME functionality:</p> <p>+CME ERROR: <err></p> <p>Parameters</p> <p><mode> Sync mode</p> <ul style="list-style-type: none"> 0 Get remote phonebook and save in file system. This file will store phonebook in VCARD format. 1 Add phonebook records to ME or SM phonebook from VCARD file. Should get remote phonebook file by mode 0 first. <p><storage> Phonebook storage to sync.</p> <ul style="list-style-type: none"> 1 Phonebook on phone storage 2 Incoming call list on phone storage 3 Outgoing call list on phone storage

4 Missed call list on phone storage
5 All call list in storage 2, 3, 4
6 Phonebook on SIM card
7 Incoming call list on SIM card
8 Outgoing call list on SIM card
9 Missed call list on SIM card
10 All call list in storage 7,8,9

<loc> File saved in ROM or SD card.

0 Saved in ROM
file will be saved in "C:\user\bt\remotePb<n>.txt"

1 Saved in SD card
file will be saved in "D:\bt\remotePb<n>.txt".The 'n' in angle brackets is corresponding with **<storage>**, from 1 to 10.

<result> Sync phonebook result

0 Sync phonebook succeed
1 Fail to get phonebook on remote phone
2 Save phonebook fail

<length> File length

<loc_phb> Save phonebook file to ME or SM. Just use in mode 1.

0 SM phonebook
1 ME phonebook

<loc_mode> Append or overwrite local phonebook. Just use in mode 1.

0 Append mode. Phonebook records in VCARD file will add in not used index of local phonebook.
1 Overwrite mode. Local phonebook records will be deleted first.

<sync2loc_result> Sync result in mode 1

0 Sync in mode 1 succeed
1 Function has already run
2 Local phonebook(ME or SM) full
3 Not enough memory
4 Error when read VCARD file.
5 Error when analyze VCARD file
6 Local phonebook not ready
7 SIM card not ready

<succ_num> num of phonebook records succeed add to local phonebook

<fail_num> num of phonebook records failed add to local phonebook.
The most common reason of add failed is name and number field of VCARD phonebook record is both empty.

Note

3.27 AT+BTPBF Find Name or Number from Remote by BT

AT+BTPBF Find Name or Number from Remote by BT	
Test command AT+BTPBF=?	Response +BTPBF: (0,1),(32,64),(1-10),(0-2) OK
Write command AT+BTPBF=<mode>,<string>[,<storage>[,<order>]]	Response OK If find name by number succeed +BTPBF:1,<phb_total> +BTPBF:1,<phb_index>,<name> ... If find number by name succeed +BTPBF:0,<phb_total> +BTPBF:0,<phb_index>,<num_total> +BTPBF:0,<phb_index>,<num_index>,<number>,<type> ... If find name by number failed or find number by name failed at get list step. +BTPBF: <mode>,<error> If find number by name failed at get entry step +BTPBF: <mode>,<phb_index>,<error> If error is related to ME functionality: +CME ERROR: <err>
	Parameters <mode> Find mode 0 Find number by name 1 Find name by number <string> String to be searched. If use mode 0, it should be alphanumeric ASCII text string up to 32 characters If use mode 1, it should be UCS2(big Indian) value form with alphanumeric ASCII text string. Max length is 64 <storage> See AT+BTPBSYNC . Default value is 1. <order> Search results order 0 Order by indexed 1 Order by alpha

2 Order by sound

<phb_total> Total number of phonebook record be found. We support max 5 phonebook records.

<phb_index> Index of phonebook record

<name> The name found by number. It will be UCS2(big Indian) value.

<num_total> Total number of <number> in one phonebook record. We support max 4 numbers in one phonebook record.

<num_index> Index of <number>

<number> The number found by name.

<type> Type of <number>

- 0 Voice
- 1 Cell
- 2 Home
- 3 Work
- 4 Fax

<error> Find error

- 255 Fail to find

Note

The support of this function on different brands of mobile phone is different.

3.28 AT+BTAVRCOP AVRCP Operation

AT+BTAVRCOP AVRCP Operation

Test Command
AT+BTAVRCOP=?

Response
+BTAVRCOP: (0-STOP,1-PLAY,2-PAUSE,3-FORWARD,4-BACKWARD,5-VOL_UP,6-VOL_DOWN)

OK

Parameters
See Write Command

Write Command
AT+BTAVRCOP=<operator>

Response
OK

If error is related to ME functionality:
+CME ERROR: <err>

Parameters
<operator>

- 0 Stop the music
- 1 Play the music
- 2 Pause the music
- 3 Play the next song

	<ul style="list-style-type: none"> 4 Play the back song 5 Increase the volume 6 Decrease the volume
Note	

3.29 AT+BTVIS Set Visibility of BT

AT+BTVIS Set Visibility of BT	
Test command AT+BTVIS=?	Response +BTVIS: (0,1) OK Parameters See Write Command
Read command AT+BTVIS?	Response +BTVIS: <visibility> OK Parameters See Write Command
Write command AT+BTVIS=<visibility>	Response OK Parameters <visibility> Visibility of BT 1 Open visibility 0 Close visibility
Note	

3.30 AT+BTSPPCFG SPP Configuration

AT+BTSPPCFG SPP Configuration	
Test command AT+BTSPPCFG=?	Response +BTSPPCFG: (list of supported <btSppCfg>s) OK
Read command AT+BTSPPCFG?	Response Every SPP's link has been connected as server,output:

	<p>+BTSPPCFG: S,<connectId>,<serverMode> Every SPP's link has been connected as client,output: +BTSPPCFG: C,<connectId></p> <p>OK</p> <hr/> <p>Parameters <connectId> Connection`s ID <serverMode> 0 AT mode 1 APP mode</p>
<p>Write command AT+BTSPPCFG=<btSppCfg> >, <mode></p>	<p>Response OK or ERROR</p> <hr/> <p>Parameters <btSppCfg> "MC" Multi-connection, enable this function to make the module support to connect double SPP's client at the same time. "TT" Transparent transmission mode, this function makes the module automatically enter the data mode after the SPP connection is established. <mode> 0 Disable 1 Enable 2 Query</p>
<p>Note</p>	<ul style="list-style-type: none"> ● In AT mode, module of server can't execute AT+BTSPSEND and AT+BTSPGET commands. ● In APP mode, module of server can execute AT+BTSPSEND and AT+BTSPGET commands.

3.31 AT+BTPAIRCFG Set BT Pairing Mode

AT+BTPAIRCFG Set BT Pairing Mode	
<p>Test command AT+BTPAIRCFG=?</p>	<p>Response +BTPAIRCFG: (list of supported <mode>s)</p> <p>OK</p> <hr/> <p>Parameters See Write Command</p>
<p>Read command AT+BTPAIRCFG?</p>	<p>Response If mode=1, the notification information is: +BTPAIRCFG: <mode>,<pin_code></p>

	<p>OK</p> <p>If mode=0 or 2, the notification information is: +BTPAIRCFG: <mode></p>
<p>Write command</p> <p>1) if PIN-Code inputted by manual while pairing AT+BTPAIRCFG=1[,<pin_code>]</p> <p>2) if using random PIN-Code while pairing AT+BTPAIRCFG=<mode></p>	<p>OK</p> <p>Parameters</p> <p>See Write Command</p> <hr/> <p>Response</p> <p>OK</p> <p>Parameters</p> <p><mode></p> <ul style="list-style-type: none"> 0 Random PIN-Code, and need confirm the pairing request 1 PIN-Code inputted by manual 2 Random PIN-Code, and response the pairing request automatic <p><pin_code> PIN-Code, the length is four. default value is 0000</p>
<p>Note</p>	<ul style="list-style-type: none"> ● When <mode> is 0 or 2, it is random PIN-Code. ● When <mode> is 2, it has no +BTPAIRING information and response the pairing request automatic. ● When <mode> is 0, it has +BTPAIRING information, and need input AT+BTPAIR=1,1 to confirm pairing request. ● The setting will be valid after reboot.

3.32 AT+CPBFEX Find Name or Number in Module Phonebook

AT+CPBFEX Find Name or Number in Module Phonebook	
<p>Test Command</p> <p>AT+CPBFEX=?</p>	<p>Response</p> <p>+CPBFEX: (0,1),40</p> <p>OK</p>
<p>Write Command</p> <p>AT+CPBFEX=<mode>,<value></p>	<p>Response</p> <p>TA returns phone book entries, which contains alphanumeric string <text>.</p> <p>[+CPBFEX: <text>]</p> <p>OK</p> <p>Parameters</p> <p><mode> Find mode</p> <ul style="list-style-type: none"> 0 Find name by number 1 Find number by name <p><value> String type field of maximum length 40. When select</p>

	<p><mode> 1, <value> should set in current TE character set specified by +CSCS.</p> <p><text> String type field. When select <mode> 0, <text> will return in current TE character set specified by +CSCS.</p>
Note	<ul style="list-style-type: none"> ● AT+CPBFEX will only return the first find result. ● AT+CPBFEX could find name or number which CPBFEX could not display when use BTPBSYNC sync PHB to ME phonebook.

3.33 AT+BTRING Control Ring Playing Transferred from Phone

AT+BTRING Control Ring Playing Transferred from Phone	
Test command AT+BTRING=?	Response +BTRING: (0,1) OK Parameters See Write Command
Read command AT+BTRING?	Response +BTRING: <mode> OK Parameters See Write Command
Write command AT+BTRING=<mode>	Response OK Parameters <mode> 0 Not play ring transferred from mobile phone 1 Play ring transferred from mobile phone
Note	<ul style="list-style-type: none"> ● This command takes effect when module acts as earphone in BT link. ● This command doesn't support power off save.

3.34 AT+BTACI Set Report Mode of BT Audio Service State Change

AT+BTACI Set Report Mode of BT Audio Service State Change	
Test command	Response

AT+BTACI=?	<p>+BTACI: (0,1)</p> <p>OK</p> <hr/> <p>Parameters See Write Command</p>
Read command AT+BTACI?	<p>Response +BTACI: <mode>,<state></p> <p>OK</p> <hr/> <p>Parameters See Write Command</p>
Write command AT+BTACI=<mode>	<p>Response OK</p> <hr/> <p>Parameters <mode> Set URC report or not when audio service state change 0 No URC report when audio service state change 1 URC report when audio service state change <state> BT audio State 0 Idle 1 SCO service 2 A2DP service</p> <hr/> <p>Unsolicited Result Code When <mode> is set to 1, URC +BTACI: <state> will report when BT audio service state change</p>
Note	This command doesn't support power off save.

3.35 AT+BTHFGOP Set Action Mode of MS When Earphone Button is Pressed During BT Link

AT+BTHFGOP Set Action Mode of MS When Earphone Button Is Pressed During BT Link	
Test command AT+BTHFGOP=?	<p>Response +BTHFGOP: (0-2)</p> <p>OK</p> <hr/> <p>Parameters See Write Command</p>
Read command AT+BTHFGOP?	<p>Response +BTHFGOP: <mode>,<event></p> <p>OK</p>

	Parameters See Write Command
Write command AT+BTHFGOP=<mode>	Response OK
	Parameters <mode> Set action mode of MS when earphone button is pressed during BT link <ul style="list-style-type: none"> 0 MS acts normally 1 URC is reported and RI pin will be pulled down for 120ms, MS will suspend earphone events and take no action. 2 Clear event to 0,mode not change
	<event> Earphone event <ul style="list-style-type: none"> 0 No event 1 Call redial 2 Answer incoming call 3 Call hang up
	Unsolicited Result Code When <mode> is set to 1, URC +BTHFGOP: <event> will report when earphone event has been changed.
Execute Command AT+BTHFGOP	Execute command will restore earphone events of MS. Execute command can't execute when no event.
	Response OK
Note	This command doesn't support power off save.

3.36 AT+BTSPPURC Set the Report Format of Command +BTSPSEND

AT+BTSPPURC Set the Report Format of Command +BTSPSEND	
Test command AT+BTSPPURC=?	Response +BTSPPURC:(0,1) OK
	Parameters See Write Command
Read command AT+BTSPPURC?	Response +BTSPPURC: <mode>,<succ_str>,<fail_str> OK
	Parameters See Write Command

Write command AT+BTSPURC=<mode>	<p>Response OK</p> <hr/> <p>Parameters</p> <p><mode> Set the report format of command +BTSPSEND</p> <ul style="list-style-type: none"> 0 Common URC of data mode 1 Special URC of Bluetooth data mode <p><succ_str></p> <ul style="list-style-type: none"> SEND OK Common URC for success BT SEND OK Special URC for success <p><fail_str></p> <ul style="list-style-type: none"> SEND FAIL Common URC for failure BT SEND FAIL Special URC for failure
Note	This command doesn't support power off save. The default value of <mode> is 0.

3.37 AT+BTCLCCS Get Call Status of Smart Phone

AT+BTCLCCS Get Call Status of Smart Phone	
Test command AT+BTCLCCS=?	<p>Response +BTCLCCS: (0,1)</p> <p>OK</p> <hr/> <p>Parameters</p> <p>See Write Command</p>
Read command AT+BTCLCCS?	<p>Response +BTCLCCS: <mode></p> <p>OK</p> <hr/> <p>Parameters</p> <p>See Write Command</p>
Write command AT+BTCLCCS=<mode>	<p>Response OK</p> <hr/> <p>Parameters</p> <p><mode> Auto report state</p> <ul style="list-style-type: none"> 1 Active 0 Deactive <hr/> <p>Unsolicited Result Code</p> <p>When <mode> is set to 1 , URC will report when call state change:</p> <p>+BTCLCCS: 1,<call_stat>,<number>,<call_id></p>
Execute Command	Response

AT+BTCLCCS

OK

When call is active:

+BTCLCCS: <mode>,<call_stat>,<number>,<call_id>

...

When no call:

+BTCLCCS: <mode>,0,,0

Parameters

<mode> Auto report state

1 Active

0 Deactive

<call_stat> state of call

0 Idle

1 Dialing(MO call)

2 Incoming (MT call)

4 Active

8 Hold

<number> String type (string should be included in quotation marks) phone number in format specified by <type>.

<call_id> 1..7 Call identification number

Note

- If there are multi calls, multi "**+BTCLCCS**" will be reported, but **<index>** is different
- Only MTK_6261 platform support this command.

3.38 AT+BTSPPCFD Set String of SPP Switching Work Mode

AT+BTSPPCFD Set String of SPP Switching Work Mode

Test command
AT+BTSPPCFD=?

Response

+BTSPPCFD: (list of supported <switchStr>)

OK

Parameters

See Write Command

Read command
AT+BTSPPCFD?

Response

+BTSPPCFD: <switchStr>

OK

Parameters

See Write Command

Write command
AT+BTSPPCFD=<switchStr>
>

Response

OK

or

	<p>ERROR</p> <hr/> <p>Parameters</p> <p><switchStr> String used to switch work mode from AT mode to data mode</p>
Note	<p>The usage of this command depends on the model of modules:</p> <ul style="list-style-type: none"> ● When any module except SIM800C acts as the SPP server, the default connection type is AT mode. User needs to input special strings in order to switch to data mode. If the string is null (AT+BTSPPCFD=""), SPP server will directly enter data mode after any data is received from client during the next connection. ● When SIM800C acts as the SPP server, the default connection type is APP data mode. User needs to input special strings in order to switch to the AT mode. If the string is null (AT+BTSPPCFD=""), SPP server will never enter into the data mode.

3.39 AT+BTCOD Set the Bluetooth Class of Device

AT+BTCOD Set the Bluetooth Class of Device	
<p>Test command</p> <p>AT+BTCOD=?</p>	<p>Response</p> <p>OK</p> <hr/> <p>Parameters</p> <p>See Write Command</p>
<p>Read command</p> <p>AT+BTCOD?</p>	<p>Response</p> <p>+BTCOD: <en>,<mjr_srv>,<mjr_cls>,<mnr_cls></p> <p>OK</p> <hr/> <p>Parameters</p> <p>See Write Command</p>
<p>Write command</p> <p>AT+BTCOD=<en>[,<mjr_srv>[,<mjr_cls>[,<mnr_cls>]]]</p>	<p>Response</p> <p>OK</p> <p>or</p> <p>ERROR</p> <hr/> <p>Parameters</p> <p><en> 0 Disable customized COD</p> <p> 1 Enable customized COD</p> <p><mjr_srv> Major service code</p> <p><mjr_cls> Major class code</p> <p><mnr_cls> Minor class code</p>
Note	<p>The setting does not support power-off preservation. This command only be used when the Bluetooth is power down.</p>

3.40 AT+BLESREG Register GATT Server

AT+BLESREG Register GATT Server	
Test command AT+BLESREG=?	Response OK
Execute Command AT+BLESREG	Response +BLESREG: <server_index>,<user_id> OK or ERROR
	Parameters <server_index> Server index <user_id> User id of GATT server, or the name of the GATT server. A Hex value string. Each char of it should in set { '0'~'9','a'~'f','A'~'F' }. Max length of it is 32.
Read Command AT+BLESREG?	Response +BLESREG: <server_index>,<user_id> OK
	Parameters See Execute Command
Note	The user id will be generated automatically.

3.41 AT+BLESREG Deregister GATT Server

AT+BLESREG Deregister GATT Server	
Test command AT+BLESREG=?	Response OK
Read command AT+BLESREG?	Response OK
	Parameters See Write Command
Write command AT+BLESREG=<server_index>	Response +BLESREG: <server_index>,<user_id>

	<p>OK</p> <p>or</p> <p>ERROR</p> <hr/> <p>Parameters</p> <p><server_index> Server index</p> <p><user_id> User id of GATT server, or the name of the GATT server. A Hex value string, each char of it should in set { '0'~'9', 'a'~'f', 'A'~'F' }. Max length of it is 32.</p>
Note	

3.42 AT+BLESSAD Add a Service

AT+BLESSAD Add a Service	
Test command AT+BLESSAD=?	<p>Response</p> <p>OK</p>
Read command AT+BLESSAD?	<p>Response</p> <p>+BLESSAD:</p> <p><service_index>,<user_id>,<uuid>,<is_primary>,<inst>,<service_handle></p> <p>OK</p> <p>Parameters</p> <p>See Write Command</p>
Write command AT+BLESSAD=<server_index>,<uuid>,<num_handles>,<is_primary>,<inst>	<p>Response</p> <p>+BLESSAD:</p> <p><service_index>,<user_id>,<uuid>,<is_primary>,<inst>,<service_handle></p> <p>OK</p> <p>or</p> <p>ERROR</p> <hr/> <p>Parameters</p> <p><server_index> Server index</p> <p><service_index> Service index</p> <p><user_id> user id of GATT server, or the name of the GATT server. A Hex value string, each char of it should in set { '0'~'9', 'a'~'f', 'A'~'F' }. Max length of it is 32.</p> <p><uuid> The UUID of the service, a string with hex value, max length is 32, min length is 4.</p> <p><num_handles> Number of handles of this service. Dec format. 1~30. Should be larger than num of services + 2* num of Chars + num</p>

of descriptor.

<is_primary> 0 Not primary service
1 Primary service

<inst> Instance id of this UUID. Dec format.

<service_handle> The handle of this service. Dec format.

Note

3.43 AT+BLESSRM Remove a Service

AT+BLESSRM Remove a Service

Test command
AT+BLESSRM=?

Response
OK

Read command
AT+BLESSRM?

Response
OK

Parameters
See Write Command

Write command
AT+BLESSRM=<service_index>

Response
+BLESSRM: <service_index>,<user_id>,<uuid>,<service_handle>

OK
or
ERROR

Parameters

<service_index> Service index

<user_id> User id of GATT server, or the name of the GATT server. A Hex value string, each char of it should in set { '0'~'9', 'a'~'f', 'A'~'F' }. Max length of it is 32.

<uuid> The UUID of the service, a string with hex value, max length is 32, min length is 4.

<service_handle> The handle of this service. Dec format.

Note

3.44 AT+BLESSC Add a Characteristic to an Existed Service

AT+BLESSC Add a Characteristic to an Existed Service

Test command
AT+BLESSC=?

Response
OK

Read command
AT+BLESSC?

Response
+BLESSC:
<char_index>,<user_id>,<service_handle>,<char_uuid>,<inst>,<prop>,<permission>,<char_handle>

OK

Parameters

See Write Command

Write command
AT+BLESSC=<service_index>,<char_uuid>,<inst>,<prop>,<permission>

Response
+BLESSC:
<char_index>,<user_id>,<service_handle>,<char_uuid>,<inst>,<char_handle>

OK

or

ERROR

Parameters

<service_index> Service index

<char_index> Characteristic index

<user_id> user id of GATT server, or the name of the GATT server. A Hex value string, each char of it should in set { '0'~'9','a'~'f','A'~'F' }. Max length of it is 32.

<service_handle> The handle of this service. Dec format.

<char_uuid> The UUID of characteristic, a string with hex value, max length is 32, min length is 4.

<inst> Instance id of this UUID. Dec format.

<prop> Properties of this characteristic. Dec format. (0 - 4294967295)

Default 0

Broadcast 1

Read 2

Write without response 4

Write 8

Notify 16

Indicate 32

Signed write 64

Extended properties 128

<permission> Permission of this characteristic. Dec format. (0 - 4294967295)

Read 1

Read with encrypted protection 2

Read with MITM protection 4

Write 8

Write with encrypted protection 16

Write with MITM protection 32

Signed write 64

	Signed write with MITM protection 128 <char_handle> The handle of this Characteristic. Dec format.
Note	

3.45 AT+BLESSD Add a Descriptor to an Existed Service

AT+BLESSD Add a Descriptor to an Existed Service	
Test command AT+BLESSD=?	Response OK
Read command AT+BLESSD?	Response +BLESSD: <desc_index>,<user_id>,<service_handle>,<desc_uuid>,<inst>,<permission>,<desc_handle> OK
Write command AT+BLESSD=<service_index>,<desc_uuid>,<inst>,<permission>	Response +BLESSD: <desc_index>,<user_id>,<service_handle>,<desc_uuid>,<inst>,<desc_handle> OK or ERROR
Note	Parameters See Write Command Parameters <service_index> Service index <desc_index> Descriptor index <user_id> User id of GATT server, or the name of the GATT server. A Hex value string, each char of it should in set { '0'~'9', 'a'~'f', 'A'~'F' }. Max length of it is 32. <service_handle> The handle of this service. Dec format. <desc_uuid> The UUID of the descriptor, a string with hex value, max length is 32, min length is 4. <inst> Instance id of this UUID. Dec format. <permission> Permission of this descriptor. Dec format. (0 - 4294967295) <desc_handle> Handle of this descriptor. Dec format.

3.46 AT+BLESSTART Start a Service

AT+BLESSTART Start a Service	
Test command AT+BLESSTART=?	Response OK
Read command AT+BLESSTART?	Response +BLESSTART: <service_index>,<user_id>,<service_handle> OK
	Parameters See Write Command
Write command AT+BLESSTART=<service_index>,<transport>	Response +BLESSTART: <service_index>,<user_id>,<service_handle> OK or ERROR
	Parameters <service_index> Service index <transport> Transport way to start service. 0 LE 1 BR/EDR 2 Dual mode <user_id> User id of GATT server, or the name of the GATT server.A Hex value string, each char of it should in set { '0'~'9','a'~'f','A'~'F' }. Max length of it is 32. <service_handle> The handle of this service. Dec format.
Note	

3.47 AT+BLESSTOP Stop a Service

AT+BLESSTOP Stop a Service	
Test command AT+BLESSTOP=?	Response OK
Read command AT+BLESSTOP?	Response OK
Write command AT+BLESSTOP=<service_index>,<user_id>,<service_handle>	Response +BLESSTOP: <service_index>,<user_id>,<service_handle>

index>	<p>OK</p> <p>or</p> <p>ERROR</p> <hr/> <p>Parameters</p> <p><service_index> Service index</p> <p><transport> Transport way to start service.</p> <p>0 LE</p> <p>1 BR/EDR</p> <p>2 Dual mode</p> <p><user_id> User id of GATT server, or the name of the GATT server.A Hex value string, each char of it should in set { '0'~'9','a'~'f','A'~'F' }. Max length of it is 32.</p> <p><service_handle> The handle of this service. Dec format.</p> <hr/>
Note	

3.48 AT+BLESSTART Start Advertising

AT+BLESSTART Start Advertising	
Test command AT+BLESSTART=?	<p>Response</p> <p>OK</p> <hr/>
Read command AT+BLESSTART?	<p>Response</p> <p>+BLESSTART: <server_index>,<user_id></p> <hr/> <p>OK</p> <p>Parameters</p> <p>See Write Command</p> <hr/>
Write command AT+BLESSTART=<server_index>	<p>Response</p> <p>+BLESSTART: <server_index>,<user_id></p> <hr/> <p>OK</p> <p>or</p> <p>ERROR</p> <hr/> <p>Parameters</p> <p><server_index> Server index</p> <p><user_id> User id of GATT server, or the name of the GATT server.A Hex value string, each char of it should in set { '0'~'9','a'~'f','A'~'F' }.Max length of it is 32.</p> <hr/>
Note	<p>The advertising is started automatically while the server registers successfully by default.</p>

3.49 AT+BLESLSLSTOP Stop Advertising

AT+BLESLSLSTOP Stop Advertising	
Test command AT+BLESLSLSTOP=?	Response OK
	Parameters See Write Command
Read command AT+BLESLSLSTOP?	Response OK
	Parameters See Write Command
Write command AT+BLESLSLSTOP=<server_index>	Response +BLESLSLSTOP: <server_index>,<user_id>
	OK or ERROR
	Parameters <server_index> Server index <user_id> User id of GATT server, or the name of the GATT server. A Hex value string, each char of it should in set { '0'~'9', 'a'~'f', 'A'~'F'}. Max length of it is 32.
Note	

3.50 AT+BLEADV Set Advertising Parameters

AT+BLEADV Set Advertising Parameters	
Test Command AT+BLEADV=?	Response OK
	Parameters See Write Command
Write Command AT+BLEADV=<server_index>,<scan_rsp>,<include_name>,<include_txpower>,<appearance>,<manufacturer_data>,<service_data>,<service_uuid>	Response +BLEADV: <user_id>
	OK or ERROR
	Parameters <server_index> Server index

<**user_id**> User id of GATT server, or the name of the GATT server.
A Hex value string.

<**scan_rsp**> Include flag parameter or not
0 Not include
1 Include

<**include_name**> Include BT name
0 Not include
1 Include

<**include_txpower**> Include Tx power Level
0 Not include
1 Include

<**appearance**> Set appearance , 0~16384.

<**manufacturer_data**> Set manufacturer, A Hex value string, each char of it should in set { '0'~'9', 'a'~'f', 'A'~'F'}. Max length of it is 56.

<**service_data**> Set service_data uuid, A Hex value string, each char of it should in set { '0'~'9', 'a'~'f', 'A'~'F'}. The length of it should be 0 or 4~32.

<**service_uuid**> Set complete services uuid, A Hex value string, each char of it should in set { '0'~'9', 'a'~'f', 'A'~'F'}. The length of it should be 0 or 4~32.

Note

AT+BLEADV will return error when broadcast packet size is over 31 bytes:

scan_rsp = 1	3 bytes
include_name = 1	characteristic number of bthost name + 2
include_txpower = 1	3 bytes
appearance = 0	0 bytes (else will take 4 bytes space)
manufacturer_data	(Hex value number+1) / 2 + 2
service_data	(Hex value number+1) / 2 + 2

3.51 AT+BLESTATUS Inquiry Current BLE Connect Status

AT+BLESTATUS Inquiry Current BLE Connect Status

Test Command
AT+BLESTATUS=?

Response
OK

Parameters
See Read Command

Read Command
AT+BLESTATUS?

Response
If unopened btpower :
+BLESTATUS: <status>

OK

If btpower opened and connected:

+BLESTATUS: <status>

+BLESTATUS: <conn_id>,<gatts_type>,<userid>,<addr>

OK

Parameters

<status>

- 0 Unopened BT power
- 1 BT power opened

<conn_id> The connection id of current connection

<gatts_type>

- 0 Custom GATT server
- 1 Custom GATT client
- 2 FMP server
- 3 PXP server
- 4 SPP server

<userid> User id of GATT server, or the name of the GATT server.

A Hex value string

<addr> Address of the peer device.

Note

3.52 AT+BLEADDR Inquiry Current Ble Address

AT+BLEADDR Inquiry Current Ble Address

Test Command
AT+BLEADDR=?

Response

OK

Parameters

See Read Command

Read Command
AT+BLEADDR?

Response

+BLEADDR: <status>,<addr>

OK

Parameters

<status>

- 0 Success
- 1 Unsuccess

<addr> Address of current device.

Note

3.53 AT+BLEDISCONN Disconnect BLE Connection

AT+BLEDISCONN Disconnect BLE Connection	
Test command AT+BLEDISCONN=?	<p>Response OK</p> <hr/> <p>Parameters See Write Command</p>
Read command AT+BLEDISCONN?	<p>Response OK</p> <hr/> <p>Parameters See Write Command</p>
Write command AT+BLEDISCONN=<conn_id>	<p>Response +BLESCON: <op>,<user_id>,<addr>,<conn_id></p> <hr/> <p>OK or ERROR</p> <hr/> <p>Parameters</p> <p><op> 0 Disconnect 1 Connect</p> <p><user_id> User id of GATT server, or the name of the GATT server. A Hex value string, each char of it should in set { '0'~'9', 'a'~'f', 'A'~'F' }. Max length of it is 32.</p> <p><addr> Address of the peer device.</p> <p><conn_id> The connection id of current connection.</p>
Note	<p>When use BLEDISCONN to disconnect server, FMP and PXP , SPP will have its own URC report (Refer to the BLEFMP and BLEXP BLESPP</p>

3.54 AT+BLESIND Send an Indication to a Client

AT+BLESIND Send an Indication to a Client	
Write command AT+BLEDISCONN=<conn_id>	<p>Response +BLESIND: <result>,<user_id>,<conn_id>,<attr_handle></p> <hr/> <p>OK or</p>

ERROR

Parameters

<char_index> Characteristic index

<user_id> User id of GATT server, or the name of the GATT server.A Hex value string, each char of it should in set { '0'~'9', 'a'~'f', 'A'~'F' }.Max length of it is 32.

<conn_id> The connection id of current connection.

<attr_handle> The handle of the characteristic value. Dec format.

<value> The value need to be notified. Hex format.

<result>

0 Success

Other Un-success

3.55 AT+BLESRSP Send a Response to a Client'S Read or Write Operation

AT+BLESRSP Send a Response to a Client's Read or Write Operation

Write Command

AT+BLESRSP=<switch>[,<value>]

Response

+BLESRSP: <result>,<user_id>,<conn_id>,<attr_handle>

OK

or

ERROR

Parameters

<switch> Read or write

0 Read

1 Write

<user_id> User id of GATT server, or the name of the GATT server.A Hex value string, each char of it should in set { '0'~'9', 'a'~'f', 'A'~'F' }.Max length of it is 32.

<conn_id> The connection id of current connection.

<attr_handle> The handle of the characteristic value. Dec format.

<value> The value need to be notified. Hex format.
If <switch> is 0, <value> is mandatory.

<result>

0 Success

Other Un-success

AT+BLESRSP will be used when read or write URC is reported.

URC

if there is incoming a read request:

+BLESRREQ:

<user_id>,<conn_id>,<trans_id>,<addr>,<attr_handle>,<is_long>,<offset>

Parameters

<user_id> User id of GATT server, or the name of the GATT server. A Hex value string, each char of it should in set { '0'~'9', 'a'~'f', 'A'~'F' }. Max length of it is 32.

<conn_id> The connection id of current connection.

<trans_id> The id of current transaction. 0~65535

<addr> Address of the peer device.

<attr_handle> Handle of attribute.

<is_long> Tell server that the request is one or several requests.

<offset> Offset of the request. 0~65535

URC

if there is incoming a write request:

+BLESWREQ:

<user_id>,<conn_id>,<trans_id>,<addr>,<attr_handle>,<value>,<need_rsp>,<is_prep>,<offset>

Parameters

<user_id> User id of GATT server, or the name of the GATT server. A Hex value string, each char of it should in set { '0'~'9', 'a'~'f', 'A'~'F' }. Max length of it is 32.

<conn_id> The connection id of current connection.

<trans_id> The id of current transaction. 0~65535

<addr> Address of the peer device.

<attr_handle> Handle of attribute.

<value> The value need to be write, Hex format

<need_rsp> Whether client need server's response

1 Yes

0 No

<is_prep> Whether or not server execute request immediately

0 No

1 Yes

<offset> Offset of the request. 0~65535

3.56 +BLESCON Notify When a Connection's Status Change

Notify When Connection's Status Change

Response

+BLESCON: <op>,<user_id>,<addr>,<conn_id>

Parameters

	<p><op></p> <p>0 Disconnect</p> <p>1 Connect</p> <p><user_id> User id of GATT server, or the name of the GATT server. A Hex value string, each char of it should in set { '0'~'9', 'a'~'f', 'A'~'F' }. Max length of it is 32.</p> <p><addr> Address of the peer device.</p> <p><conn_id> The connection id of current connection.</p>
Note	

3.57 AT+BLECREG Register GATT Client

AT+BLECREG Register GATT Client	
Test command AT+BLECREG=?	<p>Response</p> <p>OK</p> <p>Parameters</p> <p>See Write Command</p>
Read command AT+BLECREG?	<p>Response</p> <p>+BLECREG: <client_index>,<user_id></p> <p>OK</p> <p>Parameters</p> <p>See Execute Command</p>
Execute command AT+BLECREG	<p>Response</p> <p>+BLESREG: <client_index>,<user_id></p> <p>OK</p> <p>or</p> <p>ERROR</p> <p>Parameters</p> <p><client_index> Client index</p> <p><user_id> User id of GATT client, or the name of the GATT client. A Hex value string.</p>
Note	The user id will be generated automatically.

3.58 AT+BLECDREG Deregister GATT Client

AT+BLECDREG Deregister GATT Client

Test command AT+BLECDREG=?	Response OK
Write Command AT+BLECDREG=<client_index>	Response +BLECDREG: <client_index>,<user_id> OK or ERROR
Read Command AT+BLECDREG?	Response OK
Note	Parameters See Write Command

3.59 AT+BLESCAN Scan Surrounding BLE Device

AT+BLESCAN Scan Surrounding BLE Device

Test command AT+BLESCAN=?	Response OK
Read command AT+BLESCAN?	Response +BLESCAN: <scan_index><addr> OK
Write command AT+BLESCAN=<client_index>,<op>	Response If <op> is 0 +BLESREG: <op><status>,<user_id> OK If <op> is 1 OK or ERROR
	Parameters See Write Command
	Parameters <server_index> Server index

	<p><user_id> User id of GATT client, or the name of the GATT client. A Hex value string.</p> <p><op></p> <ul style="list-style-type: none"> 0 Stop scan 1 Start scan <p><status></p> <ul style="list-style-type: none"> 0 Success 1 Fail <p><scan_index> Scan index</p> <p><addr> Address of the device that have found.</p>
Note	The user id will be generated automatically.

3.60 +BLESCANRST Notify When Find a BLE Device Comes

Notify When Find a BLE Device Comes +BLESCANRST	
	<p>Response</p> <p>+BLESCANRST:</p> <p><user_id><scan_index>,<RSSI>,<addr>,<msg></p>
	<p>Parameters</p> <p><scan_index> scan index.</p> <p><user_id> User id of GATT client, or the name of the GATT client. A Hex value string.</p> <p><addr> Address of the device that have found.</p> <p><RSSI> Signal strength.0~255 means -127 ~127.</p> <p><msg> ADV data string.HEX format.</p>

3.61 AT+BLECGDT Get Device Type Request

AT+BLECGDT Get Device Type Request	
Test command AT+BLECGDT=?	Response OK
Read command AT+BLECGDT?	Response OK
	Parameters See Write Command
Write command AT+BLECGDT=<client_inde	Response +BLECGDT: <user_id> <addr>,<dev_type>

x>,<scan_index>

OK

or

ERROR

Parameters

<user_id> User id of GATT client, or the name of the GATT client.
A Hex value string.

<scan_index> Scan_index

<addr> Address of the device that have found.

<dev_type>

- 0 Unknown
- 1 BLE
- 2 BR/EDR
- 3 Dual-mode

Note

3.62 AT+BLECCON Connect GATT Client to Remote LE/Dual-mode

Device

AT+BLECCON Connect GATT Client to Remote LE/Dual-mode Device

Test command
AT+BLECCON=?

Response
OK

Read command
AT+BLECCON?

Response
OK
Parameters
See Write Command

Write command
AT+BLECCON=<client_index>,<scan_index>[,direct]

Response
OK
+BLECCON: <status>,<user_id>,<addr>,<conn_id>
or
ERROR

Parameters

<user_id> User id of GATT client, or the name of the GATT client.
A Hex value string.

<addr> Address of the device that have found.

<status>

- 0 Success
- 1 Fail

<conn_id> Conn_id.

	<p><direct></p> <p>0 Non-direct connect</p> <p>1 Direct connect.</p>
Note	

3.63 AT+BLECDISC Disconnect GATT Client to Remote Device

AT+BLECDISC Disconnect GATT Client to Remote Device	
Test command AT+BLECDISC=?	Response OK
Read command AT+BLECDISC?	Response OK
	Parameters See Write Command
Write command AT+BLECDIS=<client_index>,<conn_id>	<p>Response</p> <p>OK</p> <p>+BLECCON: <status>,<user_id>,<addr>,<conn_id></p> <p>or</p> <p>ERROR</p> <p>Parameters</p> <p><user_id> User id of GATT client, or the name of the GATT client. A Hex value string.</p> <p><addr> Address of the device that have found.</p> <p><status></p> <p>0 Success</p> <p>1 Fail</p> <p><conn_id> Conn_id.</p> <p><direct></p> <p>0 Non-direct connect</p> <p>1 Direct connect.</p>
Note	

3.64 AT+BLECSS Search Peer's Service Description

AT+BLECSS Search Peer's Service Description	
Test command	Response

AT+BLECSS=?	OK
Write Command AT+BLECSS=<client_index>,<conn_id>	Response +BLECSS: <user_id>,<service_index>,<conn_id>,<service_id>,<inst>,<is_primary>
	OK or ERROR
	Parameters <user_id> User id of GATT client, or the name of the GATT client. A Hex value string. <service_index> service index. <conn_id> conn_id. <service_id> User id of peer's service, or the name of the GATT service. A Hex value string. <is_primary> 0 Not primary service 1 Primary service <inst> Instance id of this UUID. Dec format.
Read Command AT+BLECSS?	Response +BLECSS: <service_index>,<service_id>,<inst>
	OK
	Parameters See Write Command
Note	

3.65 AT+BLECGC Search Peer's Characteristic

AT+BLECGC Search Peer's Characteristic	
Test command AT+BLECGC=?	Response OK
Write Command AT+BLECGC=<service_index>,<char_index>	Response +BLECGC: <status>,<user_id>,<conn_id>,<service_index>,<char_index>,<char_id>,<inst>,<prop>
	OK or ERROR
	Parameters

	<p><status></p> <p>0 Success</p> <p>1 Fail</p> <p><user_id> User id of GATT client, or the name of the GATT client. A Hex value string.</p> <p><char_id> User id of GATT characteristic or the name of the GATT characteristic. A Hex value string</p> <p><service_index> Service index.</p> <p><char_index> Char index</p> <p><conn_id> Conn_id</p> <p><inst> Instance id of this UUID. Dec format.</p> <p><prop> Properties of this characteristic. Dec format. (0 - 4294967295)</p> <p>Default 0</p> <p>Broadcast 1</p> <p>Read 2</p> <p>Write without response 4</p> <p>Write 8</p> <p>Notify 16</p> <p>Indicate 32</p> <p>Signed write 64</p> <p>Extended properties 128</p>
Read Command AT+BLECGC?	<p>Response</p> <p>+BLECSS: <service_index>,<char_index>,<char_id>,<prop></p> <p>OK</p> <p>Parameters</p> <p>See Write Command</p>
Note	Don't set char_index, if you only want to get the first characteristic

3.66 AT+BLECGD Search Peer's Descriptor

AT+BLECGD Search Peer's Descriptor	
Test command AT+BLECGD=?	<p>Response</p> <p>OK</p>
Write Command AT+BLECGD=<service_index>,<char_index>[,<desc_index>]	<p>Response</p> <p>+BLECGD: <status>,<user_id>,<conn_id>,<service_index>[,<char_index>,<desc_index>,<desc_id>,<inst>]</p> <p>OK</p>

	<p>or</p> <p>ERROR</p> <hr/> <p>Parameters</p> <p><status></p> <p>0 Success</p> <p>1 Fail</p> <p><user_id> User id of GATT client, or the name of the GATT client. A Hex value string.</p> <p><char_id> User id of GATT characteristic, or the name of the GATT characteristic. A Hex value string</p> <p><service_index> Service index.</p> <p><char_index> Char index</p> <p><desc_index> Descriptor index</p> <p><desc_id> User id of GATT descriptor, or the name of the GATT descriptor. A Hex value string.</p> <p><conn_id> Conn_id.</p> <p><inst> Instance id of this UUID. Dec format.</p>
<p>Read Command</p> <p>AT+BLECGD?</p>	<p>Response</p> <p>+BLECGD:</p> <p><service_index>,<char_index>,<desc_index><char_id></p> <p>OK</p> <hr/> <p>Parameters</p> <p>See Write Command</p>
Note	<p>Don't set desc_index, if you only want to get the first descriptor</p>

3.67 AT+BLECRC Read Peer's Characteristic

AT+BLECRC Read Peer's Characteristic	
<p>Test command</p> <p>AT+BLECRC=?</p>	<p>Response</p> <p>OK</p>
<p>Write Command</p> <p>AT+BLECGD=<service_index>,<char_index>[,<desc_index>]</p>	<p>Response</p> <p>OK</p> <p>or</p> <p>ERROR</p> <hr/> <p>Parameters</p> <p><char_index> char index</p> <p><auth_req></p> <p>0 Auth_req_NONE</p> <p>1 Auth_req_NO_M1TM</p> <p>2 Auth_req_SIGNED_NO_MITMs</p>

	3 Auth_req_SIGNED_M1TM
Read Command AT+BLECRC?	Response OK
	Parameters See Write Command
Note	

3.68 +BLECRC Notify When Get a Value from Peer's Device Comes

Notify When Get a Value from Peer's Device Comes +BLECRC

	Response +BLECRC: <status>,<user_id>,<conn_id>,<service_index>,<char_index>,<value>
	Parameters <status> 0 Success 1 Fail <user_id> User id of GATT client, or the name of the GATT client. A Hex value string. <service_index> Service index. <char_index> Char index <conn_id> Conn_id. <value> Value sent by the peer's device.

3.69 AT+BLECWC Write Peer's Characteristic

AT+BLECWC Write Peer's Characteristic

Test command AT+BLECRC=?	Response OK
Write Command AT+BLECWC=<service_index>,<char_index>,<auth_req>,<value>,<write_type>	Response +BLECWC: <status>,<user_id>,<conn_id>,<service_index>,<char_index> OK or ERROR

	<p>Parameters</p> <p><status></p> <ul style="list-style-type: none"> 0 Success 1 Fail <p><user_id> User id of GATT client, or the name of the GATT client. A Hex value string.</p> <p><service_index> Service index.</p> <p><char_index> Char index</p> <p><conn_id> Conn_id.</p> <p><auth_req></p> <ul style="list-style-type: none"> 0 Auth_req_NONE 1 Auth_req_NO_M1TM 2 Auth_req_SIGNED_NO_MITM 3 Auth_req_SIGNED_M1TM <p><value> value send to the peer's device.</p> <p><write_type></p> <ul style="list-style-type: none"> 1 Write without RSP 2 Write request 3 Prepare write
Read Command AT+BLECWC?	<p>Response</p> <p>OK</p> <p>Parameters</p> <p>See Write Command</p>
Note	

3.70 AT+BLECRD Read Peer's Descriptor

AT+BLECRD Read Peer's Descriptor	
Test command AT+BLECRD=?	<p>Response</p> <p>OK</p>
Write Command AT+BLECRD=<client_index>,<conn_id>,<service_index>,<char_index>,<desc_index>,<auth_req>	<p>Response</p> <p>OK</p> <p>or</p> <p>ERROR</p> <p>Parameters</p> <p><client_index> Client index.</p> <p><conn_id> Conn_id.</p> <p><service_index> Service index.</p> <p><char_index> Char index.</p> <p><desc_index> Descriptor index.</p> <p><auth_req></p>

	0 Auth_req_NONE 1 Auth_req_NO_M1TM 2 Auth_req_SIGNED_NO_MITM 3 Auth_req_SIGNED_M1TM
Read Command AT+BLECRD?	Response OK
	Parameters See Write Command
Note	

3.71 +BLECRD Notify When Get a Value From Peer's Device Comes

Notify When Get a Value from Peer's Device Comes +BLECRD

	Response +BLECRD: <status>,<user_id>,<conn_id>,<service_index>,<char_index>,<desc_index>,<value>
	Parameters <status> 0 Success 1 Fail <user_id> User id of GATT client, or the name of the GATT client. A Hex value string. <service_index> Service index. <char_index> Char index <desc_index> Descriptor index. <conn_id> Conn_id. <value> Value sent by the peer's device. <value> Value sent by the peer's device.

3.72 AT+BLECWD Write Peer's Descriptor

AT+BLECWD Write Peer's Descriptor

Test command AT+BLECWD=?	Response OK
Write Command AT+BLECWD=<client_inde	Response +BLECWD:

<p>x>,<conn_id>,<service_index>,<char_index>,<desc_index>,<auth_req>,<value>,<write_type></p>	<p><status>,<user_id>,<conn_id>,<service_index>,<char_index>,<desc_index></p> <p>OK</p> <p>or</p> <p>ERROR</p> <hr/> <p>Parameters</p> <p><status></p> <ul style="list-style-type: none"> 0 Success 1 Fail <p><user_id> User id of GATT client, or the name of the GATT client. A Hex value string.</p> <p><service_index>Service index.</p> <p><char_index> Char index.</p> <p><desc_index> Descriptor index.</p> <p><conn_id> Conn_id.</p> <p><auth_req></p> <ul style="list-style-type: none"> 0 Auth_req_NONE 1 Auth_req_NO_M1TM 2 Auth_req_SIGNED_NO_MITM 3 Auth_req_SIGNED_M1TM <p><value> Value send to the peer's device.</p> <p><write_type></p> <ul style="list-style-type: none"> 1 Write without rsp 2 Write request 3 Prepare write
<p>Read Command AT+BLECWD?</p>	<p>Response</p> <p>OK</p> <hr/> <p>Parameters</p> <p>See Write Command</p>
<p>Note</p>	

3.73 AT+BLECRN Register Notification Request

AT+BLECRN Register Notification Request	
<p>Test command AT+BLECRN=?</p>	<p>Response</p> <p>OK</p>
<p>Write Command AT+BLECRN=<op>,<client_index>,<conn_id>,<service_index>,<char_index></p>	<p>Response</p> <p>+BLECRN: <op>,<status>,<user_id>,<conn_id>,<service_index>,<char_index></p>

	<p>OK</p> <p>or</p> <p>ERROR</p> <hr/> <p>Parameters</p> <p><op></p> <ul style="list-style-type: none"> 0 Register 1 Deregister <p><status></p> <ul style="list-style-type: none"> 0 Success 1 Fail <p><user_id> User id of GATT client, or the name of the GATT client. A Hex value string.</p> <p><client_index> Client index.</p> <p><service_index>Service index.</p> <p><char_index> Char index</p> <p><conn_id> Conn_id.</p> <p><auth_req></p> <ul style="list-style-type: none"> 0 Auth_req_NONE 1 Auth_req_NO_M1TM 2 Auth_req_SIGNED_NO_MITM 3 Auth_req_SIGNED_M1TM <p><value> Value send to the peer's device.</p> <p><write_type></p> <ul style="list-style-type: none"> 1 Write without rsp 2 Write request 3 Prepare write
<p>Read Command AT+BLECRN?</p>	<p>Response</p> <p>OK</p> <hr/> <p>Parameters</p> <p>See Write Command</p>
<p>Note</p>	

3.74 +BLECN Notify When Get a Value from Peer's Device Comes

<p>Notify When Get a Value from Peer's Device Comes +BLESCN</p>	
	<p>Response</p> <p>+BLECN:</p> <p><user_id>,<conn_id>,<service_index>,<char_index>,<is_notify>,<value></p> <hr/> <p>Parameters</p>

<user_id> User id of GATT client, or the name of the GATT client.
A Hex value string.

<service_index> Service index.

<is_notify>

- 0 Indication
- 1 Notification

<char_index> Char index

<conn_id> Conn_id.

<value> Value sent by the peer's device.

3.75 AT+BLEFMP Deregister a FMP Service

AT+BLEFMP (De)Register a FMP Service	
Test command AT+BLEFMP=?	Response +BLEFMP: (0-1) OK
Execute Command AT+BLEFMP=<op>	Response OK or ERROR Parameters <op> 0 Deregister 1 Register
Read Command AT+BLEFMP?	Response +BLEFMP: <op> OK Parameters See Execute Command
Note	

3.76 +BLEFMPCON Notify when a connection's status change comes

Notify when connection's status change comes +BLEFMPCON

Response
+BLEFMPCON: <connect_state>,<addr>

Parameters
<connect_state>
 0 Disconnect
 1 Connect
<addr> Address of the peer device.

3.77 +BLEFMPWREQ Notify When a Client's Write Request Comes

Notify When a Client's Write Request Comes +BLEFMPWREQ

Response
+BLEFMPWREQ: <addr>,<alert_level>

Parameters
<addr> Address of the peer device.
<alert_level> Value of Alert Level characteristic. HEX format.

3.78 AT+BLEPXP Deregister PXP Service

AT+BLEPXP (De)Register PXP Service

Test command AT+BLEPXP=?	Response +BLEPXP:(0-1) OK
Execute Command AT+BLEPXP=<op>	Response OK or ERROR
	Parameters <op> 0 Deregister 1 Register

Read Command AT+BLEPXP?	Response +BLEPXP: <op> OK
	Parameters

See Execute Command

3.79 +BLEXPXCON Notify When a Connection Comes

Notify When Connection's Status Change comes +BLEXPXCON

Response

+BLEXPXCON: <connect_state>,<addr>

Parameters

<connect_state>

0 Disconnect

1 Connect

<addr> Address of the peer device.

3.80 +BLEXPWREQ Notify When a Write Request Comes

Notify When a Write Request Comes +BLEXPWREQ

Response

+BLEXPWREQ: <addr>,<alert_level>

Parameters

<addr> Address of the peer device.

<alert_level> Value of Alert Level characteristic. HEX format.

3.81 +BLEXPXCON Notify When a Disconnection Alert Comes

Notify When a Disconnection Alert Comes +BLEXPXCON

Response

+BLEXPXCON: <addr>,<alert_level>

Parameters

<addr> Address of the peer device.

<alert_level> Value of Alert Level characteristic. HEX format.

3.82 AT+BLESPP (De)Register a SPP Service

AT+BLESPP (De)Register a SPP Service	
Test command AT+BLESPP=?	Response +BLESPP: (0-1) OK
Execute Command AT+BLESPP=<op>	Response OK or ERROR Parameters <op> 0 Deregister 1 Register
Read Command AT+BLESPP?	Response +BLESPP: <op> OK Parameters See Execute Command

3.83 +BLESPPCON Notify When a Connection's Status Change Comes

Notify When Connection's Status Change Comes +BLESPPCON	
	Response +BLESPPCON: <connect_state>,<addr> Parameters <connect_state> 0 Disconnect 1 Connect <addr> Address of the peer device.

3.84 +BLESPPWREQ Notify When a Client's Write Request Comes

Notify When a Client's Write Request Comes +BLESPWREQ

	Response
	+BLESPWREQ: <addr>,<value>
	Parameters
	<p><addr> Address of the peer device.</p> <p><value> Value from peer device.</p>

3.85 AT+BLESPPSIND Send an Indication to SPP Server

AT+BLESPPSIND Send an Indication to SPP Server

Write Command AT+BLESPPSIND=<value>	Response
	OK or ERROR
	Parameters
	<value> The value need to be notified. Hex format,1~1024.

SIMCom Confidential

4 CME Error Code

The following error message is associated with the Bluetooth operation following format: +CME ERROR: <err>, the specific error code and error message in the following table:

Code	Description
1000	Return fail
1002	Not power on
1003	State not idle
1004	Malloc error
1010	Scan fail
1011	scan return error
1020	Out of scanning count
1021	Out of profile id count
1025	Out of pairing count
1026	Bond error
1027	Device has Bonded
1030	Debonded error
1031	Get device info error
1032	Service refresh error
1033	Profile connect error
1034	HF attach error
1040	OPP handle error
1041	OPP send error
1042	OPP received path error
1043	SD card not exist
1044	OPP file path error
1045	OPP send error by server
1046	Get index by profile error
1047	Connect not support
1048	Disconnect not support
1049	Active or address error
1050	Only connect one device
1051	Out of max connection
1055	SPP is not connect
1056	SPP server isn't work at send mode
1057	Input data length beyond

1058	SPP port is not create
1060	Pls connect A2DP first
1061	Connected device exceed max
1099	BTAUD attach error
1997	GATT server write error
1998	GATT server read error
1999	GATT server connect error
2000	GATT server register error
2001	GATT server deregister error
2002	GATT no server error
2003	GATT add service error
2004	GATT remove service error
2005	GATT add characteristic error
2006	GATT start service error
2007	GATT stop service error
2008	GATT start/stop advertising error
2009	GATT add descriptor error
2010	GATT server exceed the max number

SIM
Confidential

5 Bluetooth Examples

There are some examples to explain how to use these commands.

5.1 Accept request from other BT device

```
//Example of how to accept request from other BT device.
AT+BTPOWER=1 // Power on BT radio
OK
+BTPAIRING:"PC-NS130100361", // Incoming digital key request from other BT device
34:c7:31:aa:37:5b,763191
AT+BTPAIR=1,1 // Accept pairing request, and paired successfully
OK
+BTPAIR:1,"PC-NS130100361",
34:c7:31:aa:37:5b
+BTPAIRING:"JabraBT160", // Incoming passkey request from other BT device
00:16:8f:0d:65:82
AT+BTPAIR=2,0000 // Accept pairing request, and paired successfully.
OK // Default passkey of other BT device is 0000.If not,
// please change this value according to other device's
// passkey.
+BTPAIR: 2,"LBH505",50:5b:0b:0a:10:32
```

5.2 Send pairing request to other BT device

```
//Example of how to send HTTP GET response.
AT+BTPOWER=1 //Power on BT radio
OK
AT+BTSCAN=1,20 //Inquiring surrounding BT device
OK

+BTPAIRING:"PC-NS130100361",34:c7:31:aa:37:5b,-34

+BTPAIRING:"JabraBT160",00:16:8f:0d:65:82
```

```
0,2,"ADMIN-9A6E040AC",68:5d:43:ec:fe:72,-44
```

```
+BTSCAN:
```

```
0,3,"LIB-PC",c8:f7:33:43:48:e6,-54
```

```
+BTSCAN:
```

```
0,4,"MKFUJIANJUN",88:53:2e:e8:9d:0f,-33
```

```
+BTSCAN:
```

```
0,5,"MTKBTDEVICE",45:8c:96:3e:66:01,-56
```

```
+BTSCAN:
```

```
0,6,"MK-ZHANZHIMIN",00:1a:7d:da:71:10,-67
```

```
+BTSCAN:
```

```
0,7,"Jabra BT160",00:16:8f:0d:65:82,-55
```

```
+BTSCAN:1
```

```
AT+BTPAIR=0,6
```

```
// Try to pair the sixth BT device in the view list
```

```
OK
```

```
+BTPAIRING:
```

```
// Answer to the pairing request in digital key mode
```

```
"MK-ZHANZHIMIN",00:1a:7d:da:71:10,763191
```

5.3 Get the profile provided by paired device

```
//Example of how to get the profile provided by paired device
```

```
// Configure based on example 4.2
```

```
AT+BTGETPROF=1
```

```
// Get the profile of first paired device in list
```

```
+BTGETPROF: 1,"A2DP(Source)"
```

```
+BTGETPROF: 2,"HFP(AG)"
```

```
+BTGETPROF: 8,"AVRCP(Target)"
```

```
+BTGETPROF: 3,"A2DP"
```

```
+BTGETPROF: 4,"SPP"
```

```
+BTGETPROF: 6,"HFP"
```

```
+BTGETPROF: 5,"HSP"
```

```
OK
```

5.4 Connect service

//Example of how to connect service

AT+BTCONNECT=1,2

OK

+BTCONNECT:

1,"MK-ZHANZHIMIN",00:1a:7d:da:71:10,
"HFP(AG)"

// Get Profile based on example 4.3

// Connect with the second profile service of first
paired device, "HFP(AG)"

5.5 Accept file from paired device

//Example of how to accept file from paired device

+BTOPPPUSHING:

"MK-ZHANZHIMIN","link.txt"

AT+BTOPPACPT=1

OK

+BTOPPPUSH:1

// Pairing device based on example 4.2

// Incoming OPP pushing service from paired device

// Accept file(stored in internal memory card by
default, input "AT+BTOPPACPT=1,1" if want it stored
in external memory

5.6 Send file to other paired BT device

//Example of how to send file to other paired BT device

AT+BTOPPPUSH=1, c:\User\BtReceived

link.txt

OK

+BTOPPPUSH: 1

// Pairing device based on example 4.2

// Sending file and waiting for response

5.7 Create SPP's link as a client

```
//Example of how to create SPP's link as a client

// Suppose this device's ID is 12:34:56:78:90:12,name is IT;Another ID is 34:c7:31:aa:37:5b,name is ME. they make pair successfully.

AT+BTCONNECT=1,4
OK

// Try to build a SPP's connection to server.

+BTCONNECT:
1,"IT",12:34:56:78:90:12,"SPP"

// If successfully, output these URC.
```

5.8 SPP's link be create as a server

```
//Example of SPP's link be create as a server

// Suppose this device's ID is 12:34:56:78:90:12, name is IT; The other ID is 34:c7:31:aa:37:5b, name is ME. they make pair successfully.

+BTCONNECTING:
"34:c7:31:aa:37:5b", "SPP"
AT+BTACPT=1
OK

// Receive a request from client which builds a connection.
// Accept it.

// Build success.

+BTCONNECT:
1,"ME",34:c7:31:aa:37:5b,"SPP"
```

5.9 Configure SPP

```
//Example of how to configure SPP

// Get Profile based on example 4.3. Suppose this device's ID is 12:34:56:78:90:12, and name is IT; The other ID is 34:c7:31:aa:37:5b, and name is ME. This module has had a server-type link of SPP.

AT+BTSPPCFG?
+BTSPPCFG:S,1,0

//There is a link. It's a server; Connection's ID is 1;
```

```

OK
AT
OK
AT
OK
AT+BTSPPCFG="MC",1
OK
AT+BTSPPCFG="MC",2
+BTSPPCFG: MC,1

OK
+BTCONNECTING:
"0c:c5:95:09:62:60","SPP"
AT+BTACPT=1
OK
// There is a request that tries to build a SPP's
connection.

+BTCONNECT:
1,"THIRD",0c:c5:95:09:62:60,"SPP"
+BTSPPDATA: 2,15,SIMCOMSPPFORAPP
AT
OK
// Receive the message of switching mode to APP
mode from the second client's link.
AT+BTSPPCFG?
+BTSPPCFG: S,1,0
+BTSPPCFG: S,2,1

OK
// Allow to send data to second client's link.

```

5.10 Send data as a SPP's client

A SPP connection has two modules. One is client, and the other is server. Let us see the demo with client module.

```

//Example of how to send data as a SPP's client
// Based on example 4.7, as a client.

AT+BTSPPCFG?
+BTSPPCFG: C,1
// There is a link, client-type, and allowed to send
data to the server.

OK
AT+BTSPSEND
>AT+CREG?
SEND OK
// If the client sends AT command to the server, this

```

```

command and its response will output to client.

+BTSPPDATA: 19,1,A

+BTSPPDATA: 19,3,T+C

+BTSPPDATA: 19,25,REG?
+CREG: 0,0

OK
AT+BTSPSEND=10
>1234567890
SEND OK

// "AT+CREG?" are input characters.
// "+CREG: 0,0" and "OK" are responses.

// If the multi-connection function is disabled, we
// don't need to input connection's ID. Input
// data(1234567890) and press Ctrl+Z keys, the data
// will be sent.

```

5.11 As a SPP's server worked in AT mode

SPP's connection as a server has two mode. One is AT mode. In this mode, we can't use AT+BTSPSEND/BTSPGET commands to send data to the client or get data from the client. We can only receive data from the client.

```

//Example of as a SPP's server worked in AT mode

// Based on example 4.8, as a server.

AT+BTSPPCFG?
+BTSPPCFG: S,1,0

// There is a link. Server-type; connection's ID is 1;
// It's not allowed to send data to the client

OK
AT+BTSPSEND=10

// Fail to send.
ERROR
AT+BTSPSEND

// Fail to send.
ERROR

```

5.12 As a SPP's server worked in APP mode and multi-connection

Another SPP's link mode as a server is the APP mode. In this mode, we can execute AT+BTSPSEND and AT+BTSPGET commands.

```

//Example of as a SPP's server worked in APP mode and multi-connection

// Based on example 4.7, as a server.

```

```

+BTSPpdata: 1,15,SIMCOMSPPFORAPP // Receive the specified data package from the first
AT // client's link which means switching the mode to APP
OK // mode(This data package must be the first package
AT // received).After executing AT+BTSPPCFD="",client
OK // will enter APP mode when sending data package
AT+BTSPPCFG? // without specified strings.
+BTSPPCFG: S,1,1

OK
AT+BTSPSEND // Allow to send data to the client.
>12345
SEND OK // Send successfully.
AT+BTDISCONN=1
OK

+BTDISCONN: // Disconnect this link of client.
"SIM800H",34:c7:31:aa:37:5b,"SPP"
AT+BTSPGET=1 // Switch to manual mode.
OK

+BTCONNECTING: // Receive the connecting request from the client.
"34:c7:31:aa:37:5b","SPP"
AT+BTACPT=1
OK

+BTCONNECT: // Build link successfully.
1,"SIM800H",34:c7:31:aa:37:5b,"SPP"

+BTSPPMAN: 1 // Receive the data from the client whose
AT // connection's ID is 1.
OK
AT+BTSPGET=2,1
+BTSPGET: 1,15 // Connection's ID is 1, and the data length is 15.

OK
AT+BTSPGET=3,1,15
+BTSPGET: 1,15,SIMCOMSPPFORAPP // Get data, length is 15(This data package means
// switching the mode to APP mode).

OK
AT+BTSPSEND // Send data to the client.
> 1234567890
SEND OK // Send successfully.
AT+BTSPGET=?
+BTSPGET: (0-3),(1-6),(1-1024),1

OK

```

5.13 Sync phonebook from remote by BT

```
//Example of how to sync phonebook from remote by BT
// Based on example 4.2
AT+BTGETPROF=1 // Connect with the second profile service of first
+BTGETPROF: 10,"PBAP"
+BTGETPROF: 1,"A2DP(Source)"
+BTGETPROF: 2,"HFP(AG)"
+BTGETPROF: 8,"AVRCP(Target)"

OK
AT+BTCONNECT=1,10 // Connect server
OK

+BTCONNECT: // Report automatically once ready
1,"LG-P705",00:aa:70:23:7d:06,"PBAP(C)"
AT+BTPBSYNC=0,1,0 // Sync phonebook
OK

+BTBPSYNC: 0,0,53786 // Sync succeeds. File size is 53786 bytes.
```

5.14 Find name or number from remote by BT

```
//Example of how to find name or number from remote by BT
// Based on example 4.2
AT+BTGETPROF=1 // Get the profile of first paired device in list
+BTGETPROF: 10,"PBAP"
+BTGETPROF: 1,"A2DP(Source)"
+BTGETPROF: 2,"HFP(AG)"
+BTGETPROF: 8,"AVRCP(Target)"

OK
AT+BTCONNECT=1,10 // Connect server
OK

+BTCONNECT: // Report automatically once ready
1,"LG-P705",00:aa:70:23:7d:06,"PBAP(C)"
```

```

AT+BTPBF=1,"135",1 // Find name whose number contain "135".
OK

+BTPBF: 1,5 // Find succeed. Five names found.

+BTPBF: 1,1,003100330035003800350
03800380037003700370035

+BTPBF: 1,2,5170621056FD

+BTPBF: 1,3,521800206587660E

+BTPBF: 1,4,52186021

+BTPBF: 1,5,5362592A592A
AT+BTPBF=0,"0063",1 //Find number which owner's name contains char "c"
OK (format with usc2 value is "0063").

+BTPBF: 0,1 //Find succeed. One phonebook record found.

+BTPBF: 0,1,1 //First phonebook record contain one number

+BTPBF: 0,1,1,*****,,1

```

5.15 Play music and so on by AVRCP

```

//Example of how to play music and so on by AVRCP
// Get Profile based on example 4.3
AT+BTGETPROF=1 // Get the profile of first paired device in list
+BTPROF: 1,"A2DP(Source)"
+BTPROF: 2,"HFP(AG)"
+BTPROF: 8,"AVRCP(Target)"

OK
AT+BTCONNECT=1,1 // Connect with the first profile service of first paired
OK device, "A2DP", For the service of "AVRCP" depends
on the "A2DP". After connected with "A2DP"
+BTPROF: 1,"Lenovo successfully, the modem will connect to the service
of "AVRCP" automatically.
A780",d8:71:57:2b:02:66,"A2DP"

+BTPROF: 2,"Lenovo // Report automatically once ready.
A780",d8:71:57:2b:02:66,"AVRCP"

```

```

+BTCONNECT:          3,"Lenovo
A780",d8:71:57:2b:02:66,"HFP(AG)"
AT+BTAVRCOP=1        // Play music
OK                    //The sound can be heard from the modem

AT+BTAVRCOP=2        //Pause music
OK                    //The music will be paused

AT+BTAVRCOP=1        //Play music again
OK                    //The music will be played

AT+BTAVRCOP=3        //Play the next song
OK                    //The next song will be played

AT+BTAVRCOP=4        //Play the back song
OK                    //The back song will be played

AT+BTAVRCOP=5        //Increase the volume
OK                    //The volume of the music will be increased

AT+BTAVRCOP=6        //Decrease the volume
OK                    //The volume of the music will be Decreased

AT+BTAVRCOP=0        //Stop music
OK                    //The music will be stopped

```

5.16 Add phonebook records to ME or SM phonebook from VCARD file

```

//Example of how to add phonebook records to ME or SM phonebook from VCARD file
// Based on example 4.13
AT+BTPBSYNC=1,1,0,0,1 // Sync file "c:\user\bt\remotePb1.txt" to SM
OK                    phonebook with overwrite mode

+BTPBSYNC: 1,0,214,67 //Sync finished. 214 phonebook records add succeed
                        and 67 records failed.

AT+CPBR=1,250        //Read phonebook records.
+CPBR: 1,"",129,"Me"
...
OK

```

5.17 Set BT pairing mode

```
//Example of how to set BT pairing mode

// Get Profile based on example 4.3
// Power on BT radio

AT+BTPOWER=1
OK

//Set pairing mode is PIN-Code inputted by manual
( mode=1 ) , and the default PIN-Code value is
0000, if you want to set other PIN-Code, follow it :
AT+BTPAIRCFG=1,<pin_code>
//BT reboot

AT+BTSCAN=1
OK

// Inquiring surrounding BT device and pair, input
PIN-Code by opposite side, the default value is 0000

+BTSCAN:                0,1,"XT615
",00:11:94:cb:20:d2,-34

+BTSCAN:
0,2,"LIB-PC",c8:f7:33:43:48:e6,-45
AT+BTPAIR=0,1
OK

+BTSCAN: 2

+BTPAIR: 1,"XT615 ",00:11:94:cb:20:d2
AT+BTPAIRCFG=2
OK
// Set pairing mode is random PIN-Code(mode=2).
(mode=0, reference 4.2 section)
// BT reboot

AT+BTSCAN=1
OK
// Inquiring surrounding BT device and pair, and wait
to confirm pairing request by opposite side.

+BTSCAN:                0,1,"XT615
",00:11:94:cb:20:d2,-44

+BTSCAN:
0,2,"MK-ZHANZHIMIN",00:1a:7d:da:71:10,-55
AT+BTPAIR=0,1
OK

+BTSCAN: 2
```

```
+BTPAIR: 1,"XT615 ",00:11:94:cb:20:d2
```

5.18 Inquiry current ble address

```
//Example of how to nquiry current ble address
```

```
AT+BTPOWER=1
```

```
// Power on BT radio
```

```
OK
```

```
AT+BLEADDR?
```

```
+BLEADDR: 0,d4:d9:f9:30:88:33
```

```
// Inquiry current BLE address.
```

```
OK
```

5.19 Set Adverting Parameters

```
//Example of how to set Adverting Parameters
```

```
AT+BTPOWER=1
```

```
// Power on BT radio
```

```
OK
```

```
AT+BLESREG
```

```
// Register GATT Server.
```

```
+BLESREG: 1,ABCDEFF0
```

```
OK
```

```
AT+BLEADV=1,0,0,0,0,"",""
```

```
// Set Adverting Parameters.
```

```
+BLEADV: ABCDEFF0
```

```
Gradually add parameters to see the changes through the APP.
```

```
OK
```

```
AT+BLEADV=1,1,1,0,25,"4c00","02291234",  
"2902"
```

```
+BLEADV: ABCDEFF0
```

```
OK
```

5.20 Setup GATT server

```
//Example of how to setup GATT server
```

```

AT+BTPOWER=1 // Power on BT radio
OK
AT+BLESREG // Register GATT Server.
+BLESREG: 1,ABCDEFF0

OK
AT+BLESSAD=1,"123456",15,1,1 // Add a service.
+BLESAD: 1,ABCDEFF0,123456,1,1,256

OK
AT+BLESSC=1,"ABCDEF",1, 10,17 // Add a R/W characteristic.
+BLESSC: 1,ABCDEFF0,256,ABCDEF,1,258

OK
AT+BLESSC=1,"ABCDEF",1, 16,17 // Add a Notify characteristic.
+BLESSC: 2,ABCDEFF0,256,ABCDEF,1,260

OK
AT+BLESSD=1,"0229",1,0 // Add a descriptor.
+BLESSD: 1,ABCDEFF0,256,0229,1,261

OK
AT+BLESSTART=1,0 // Setup service.
+BLESSTART: 1,ABCDEFF0,256

OK
AT+BLESSTART=1 // Start advertising.
+BLESSTART: 1,ABCDEFF0

OK

```

5.21 Data transmission between module and client

```

//Example of data transmission between module and client
// Start the GATT service as shown in example 4.20.
+BLESSTART: 1,ABCDEFF0,256
// APP connects with module.
+BLESSTART: 1,ABCDEFF0,256
//APP read data.
+BLESRRQ: 1,ABCDEFF0,256,0,0
AT+BLESRSP=0,A1B2 // Answer with A1B2.
+BLESRSP: 0,ABCDEFF0,1,258

```

```
OK
+BLESWREQ:                                // Write with ABCD.
ABCDEFF0,1,100,7a:16:fc:60:72:40,258,
ABCD,1,0,0
AT+BLESRSP=1                             // Answer the write request.
+BLESRSP: 0,ABCDEFF0,1,258

OK
AT+BLESIND=2,"9876"                     // Module sends 9876 to Notify characteristic.
+BLESIND: 0,ABCDEFF0,1,260

OK
+BLESCON:                                // APP disconnect white module.
0,ABCDEFF0,7a:16:fc:60:72:40,1
```

5.22 Setup FMP server

```
//Example of how to setup FMP server
AT+BTPOWER=1                             // Power on BT radio.
OK
AT+BLEFMP=1                             // Setup FMP server.

OK
+BLEFMPCON: 1,69:e9:06:60:7a:e7           // APP connects with module.
+BLEFMPWREQ: 69:e9:06:60:7a:e7,87        // APP writes data.
+BLEFMPCON: 0,69:e9:06:60:7a:e7         // APP disconnect white module.
```

5.23 Setup PXP server

```
//Example of how to setup PXP server
AT+BTPOWER=1                             // Power on BT radio.
OK
AT+BLEPXP=1                             // Setup PXP server.

OK
+BLEPXPCON: 1,6f:53:17:18:56:15          // APP connects with module.
+BLEPXPWREQ: 6f:53:17:18:56:15,78       // APP writes data.
```

```
+BLEFMPCON: 0,69:e9:06:60:7a:e7 // APP disconnect write module.
```

5.24 Setup SPP server

```
//Example of how to setup SPP server
AT+BTPOWER=1 // Power on BT radio.
OK
AT+BLESPP=1 // Setup SPP server.
OK
+BLESPPCON: 1,6f:53:17:18:56:15 // APP connects with module.
+BLESPPWREQ: 6f:53:17:18:56:15,78 // Module sent data to APP.
AT+BLESPPSIND="ABCD" // APP writes data.
OK
+BLESPPCON: 0,66:ee:48:40:e0:64 // APP disconnect write module.
```

5.25 Inquiry current ble status

```
//Example of how to inquiry current ble status
// Setup GATT , FMP,PXP,SPP.
APP connects with module.
// Inquiry current BLE status.
AT+BLESTATUS?
+BLESTATUS: 1
+BLESTATUS:
1,0,ABCDEF0,66:ee:48:40:e0:64
+BLESTATUS:
2,1,ABCDEF1,66:ee:48:40:e0:64
+BLESTATUS:
3,2,ABCDEF2,66:ee:48:40:e0:64
+BLESTATUS:
4,3,ABCDEF3,66:ee:48:40:e0:64
OK
```

5.26 Module disconnect with APP

```
//Example of module disconnect with APP
// Inquiry current BLE status first.
AT+BLEDISCONN=1
+BLESCON: 0,ABCDEF0,49:bb:c7:48:4d:87,1 // Module disconnect with APP.

OK
AT+BLEDISCONN=2
+BLEFMPCON: 0,49:bb:c7:48:4d:87

OK
```

5.27 Module disconnect Start or stop advertising

```
//Example of module disconnect Start or stop advertising
AT+BTPOWER=1 // Power on BT radio.
OK
AT+BLESREG // Register GATT Server.
+BLESREG: 1,ABCDEF0

OK
AT+BLESSTART=1 // Start advertising.
+BLESSTART: 1,ABCDEF0

OK
AT+BLESSTOP=1 // Stop advertising.
+BLESSTOP: 1,ABCDEF0

OK
```

5.28 BLE client

```
//Example of BLE client
AT+BTPOWER=1 // Power on BT radio.
```

OK

AT+BLECREG

+BLECREG: 1,ABCDEF50

// Register BLE Server.

OK

AT+BLESCAN=1,1

+BLESCANRST:

ABCDEF50,1,51,6a:20:bb:39:fa:04,

02011A1AFF4C000C0E004E3A69

54DE2CC36C8174C7229210050B

10AA4A6C

// Start scan remote devices

+BLESCANRST:

ABCDEF50,2,45,0c:6d:72:41:9f:49,

1EFF060001092002BC7B6805FDC

B7E78F927ABEF3A1E073DC24BD

DD973177A

+BLESCANRST:

ABCDEF50,3,48,2d:d7:07:ae:65:35,

1EFF0600010920028314B9EB5351

81695B37E3BC4AD9F5F060DE53

8EEB87DD

...

AT+BLESCAN=1,0

// Stop scan and find valid index.

+BLESCAN: 0,0,ABCDEF50

OK

AT+BLECGDT=1,17

+BLECGDT: ABCDEF50,fb:a4:27:39:d9:ab,1

// Check the device type of 17th scan index, it's BLE device.

OK

AT+BLECCON=1,17,1

// Connect to 17th directly

OK

+BLECCON:

1,ABCDEF50,fb:a4:27:39:d9:ab,1

AT+BLECSS=1,1

// Get all services from connected devices

+BLECSS: ABCDEF50,1,1,0018,0,1

+BLECSS: ABCDEF50,2,1,0118,0,1

+BLECSS: ABCDEF50,3,1,E7FE,0,1

+BLECSS: ABCDEF50,4,1,C0FC,0,1

+BLECSS: ABCDEF50,5,1,00A5,0,1

+BLECSS: ABCDEF50,6,1,0A18,0,1

OK

AT+BLECGC=6

// Get 6th service characteristics.

+BLECGC: 0,ABCDEF50,1,6,1,292A,0,2

OK

AT+BLECGD=6,1

// Get description for above characteristic.

+BLECGD: 1,ABCDEF50,1,6

OK

AT+BLECRC=6,1,0

// Read characteristics.

OK

+BLECRC:

0,ABCDEF50,1,6,1,6C69666573656E7365

// "6C69666573656E7365" is in hex format, it means "lifesense".

AT+BLECWC=6,1,0,313233,1

// Write characteristics.

OK

+BLECWC: 0,ABCDEF50,1,6,1

Confidential

6 Differences between bluetooth version and standard Version

In this chapter, SIM800 BT indicates SIM800 series BT version, SIM800 indicates SIM800 series standard version. Differences among SIM800 series standard version, please refer to chapter 22 for details in doc "SIM800 Series AT Command Manual".

6.1 ATD<str>

SIM800 BT does not support finding number by name.

6.2 AT+CPBF

SIM800 BT	SIM800
Max length of <findtext> is always 40 bytes.	Max length of <findtext> depends on AT+CSCS
Results will order by phonebook index when select "SM" or "ME" phonebook, from small to large.	Results will order by the order user inputs phonebooks.
<findtext> must match <text> from the leftmost side, when select "SM" or "ME" phonebook	No this limit
Difference: There are multi difference of AT+CPBF between SIM800 BT and SIM800.	

6.3 AT+CPBFEX

MTK MMI version can support this command and modem version is the opposite. That is to say, MTK6260 and MTK 6260A platform without BT version and MTK6261A platform cannot support this command.

6.4 AT+CMUX

SIM800 BT does not support MUX function.

6.5 AT+CNUM

SIM800 BT	SIM800
+CNUM: [<alpha>],<number>,<type>,,<service>	+CNUM: <alpha>,<number>,<type>,<speed>,<service>
Difference: <alpha> of SIM800 BT does not display if length of <alpha> is 0. SIM800 BT does not support <speed> field and left blank.	

6.6 AT+CMGS

SIM800 BT does not support sending message by phonebook index or name.

6.7 AT+CMSS

SIM800 BT does not support sending message from storage.

6.8 AT+CPMS

SIM800 BT	SIM800
AT+CPMS=? +CPMS: ("SM","ME","MT"),("SM","ME","MT"), ("SM","ME","MT")	AT+CPMS=? +CPMS: ("SM","ME","SM_P","ME_P","MT"), ("SM","ME","SM_P","ME_P","MT"),

OK	("SM","ME","SM_P","ME_P","MT") OK
<p>Difference: SIM800 BT supports three modes: "SM","ME","MT". SIM800 supports "SM","ME","SM_P","ME_P","MT" modes.</p>	

6.9 AT+CHFA

SIM800 BT	SIM800
AT+CHFA=? +CHFA: (0=NORMAL_AUDIO, 1=AUX_AUDIO,2=HANDFREE_AUDIO, 3=AUX_HANDFREE_AUDIO, 4=PCM_AUDIO,5=BT_CHANNEL)	AT+CHFA=? +CHFA:(0=NORMAL_AUDIO, 1=AUX_AUDIO,2=HANDFREE_AUDIO, 3=AUX_HANDFREE_AUDIO, 4=PCM_AUDIO)
OK	OK
<p>Difference: Value of parameter <n> has BT audio channel in SIM800 BT. BT channel can be set when BT link is established and module acts as mobile phone. After switch to BT channel, local sound can be transferred to BT earphone. If BT link is disconnected, audio channel will restore to the original channel and URC +CHFA: <n> is reported. Because the audio service is always on after switch to BT channel, consumption current is bigger than normal.</p>	

6.10 TTS function

SIM800 BT which module memory is 32M does not support TTS function.