

ECx00U&EG915U Series

BT Application Note

LTE Standard Module Series

Version: 1.0.0

Date: 2022-09-08

Status: Preliminary



At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:

Quectel Wireless Solutions Co., Ltd.

Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: info@quectel.com

Or our local offices. For more information, please visit:

<http://www.quectel.com/support/sales.htm>.

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

<http://www.quectel.com/support/technical.htm>.

Or email us at: support@quectel.com.

Legal Notices

We offer information as a service to you. The provided information is based on your requirements and we make every effort to ensure its quality. You agree that you are responsible for using independent analysis and evaluation in designing intended products, and we provide reference designs for illustrative purposes only. Before using any hardware, software or service guided by this document, please read this notice carefully. Even though we employ commercially reasonable efforts to provide the best possible experience, you hereby acknowledge and agree that this document and related services hereunder are provided to you on an “as available” basis. We may revise or restate this document from time to time at our sole discretion without any prior notice to you.

Use and Disclosure Restrictions

License Agreements

Documents and information provided by us shall be kept confidential, unless specific permission is granted. They shall not be accessed or used for any purpose except as expressly provided herein.

Copyright

Our and third-party products hereunder may contain copyrighted material. Such copyrighted material shall not be copied, reproduced, distributed, merged, published, translated, or modified without prior written consent. We and the third party have exclusive rights over copyrighted material. No license shall be granted or conveyed under any patents, copyrights, trademarks, or service mark rights. To avoid ambiguities, purchasing in any form cannot be deemed as granting a license other than the normal non-exclusive, royalty-free license to use the material. We reserve the right to take legal action for noncompliance with abovementioned requirements, unauthorized use, or other illegal or malicious use of the material.

Trademarks

Except as otherwise set forth herein, nothing in this document shall be construed as conferring any rights to use any trademark, trade name or name, abbreviation, or counterfeit product thereof owned by Quectel or any third party in advertising, publicity, or other aspects.

Third-Party Rights

This document may refer to hardware, software and/or documentation owned by one or more third parties (“third-party materials”). Use of such third-party materials shall be governed by all restrictions and obligations applicable thereto.

We make no warranty or representation, either express or implied, regarding the third-party materials, including but not limited to any implied or statutory, warranties of merchantability or fitness for a particular purpose, quiet enjoyment, system integration, information accuracy, and non-infringement of any third-party intellectual property rights with regard to the licensed technology or use thereof. Nothing herein constitutes a representation or warranty by us to either develop, enhance, modify, distribute, market, sell, offer for sale, or otherwise maintain production of any our products or any other hardware, software, device, tool, information, or product. We moreover disclaim any and all warranties arising from the course of dealing or usage of trade.

Privacy Policy

To implement module functionality, certain device data are uploaded to Quectel’s or third-party’s servers, including carriers, chipset suppliers or customer-designated servers. Quectel, strictly abiding by the relevant laws and regulations, shall retain, use, disclose or otherwise process relevant data for the purpose of performing the service only or as permitted by applicable laws. Before data interaction with third parties, please be informed of their privacy and data security policy.

Disclaimer

- a) We acknowledge no liability for any injury or damage arising from the reliance upon the information.
- b) We shall bear no liability resulting from any inaccuracies or omissions, or from the use of the information contained herein.
- c) While we have made every effort to ensure that the functions and features under development are free from errors, it is possible that they could contain errors, inaccuracies, and omissions. Unless otherwise provided by valid agreement, we make no warranties of any kind, either implied or express, and exclude all liability for any loss or damage suffered in connection with the use of features and functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage may have been foreseeable.
- d) We are not responsible for the accessibility, safety, accuracy, availability, legality, or completeness of information, advertising, commercial offers, products, services, and materials on third-party websites and third-party resources.

Copyright © Quectel Wireless Solutions Co., Ltd. 2022. All rights reserved.

About the Document

Revision History

Version	Date	Author	Description
-	2021-08-25	Ryan YI/ Chaos HUANG	Creation of the document
1.0.0	2022-09-08	Ryan YI/ Chaos HUANG/ Evan MENG	Preliminary

Contents

About the Document.....	3
Contents.....	4
Table Index.....	8
1 Introduction	9
1.1. Applicable Modules.....	9
2 Description of BT AT Commands	11
2.1. AT Command Introduction.....	11
2.1.1. Definitions.....	11
2.1.2. AT Command Syntax.....	11
2.2. Declaration of AT Command Examples	12
2.3. General BT AT Commands.....	12
2.3.1. AT+QBTPWR Turn On/Off BT	12
2.3.2. AT+ QBTNAME Set BT Device Name	13
2.3.3. AT+ QBSEND Send Data	14
2.4. Description of BLE AT Commands.....	15
2.4.1. General AT Commands	15
2.4.1.1. AT+QBTLEADDR Get BLE Device Address.....	15
2.4.1.2. AT+QBTLERANADDR Get Random Address of BLE Device	15
2.4.1.3. AT+QBTGATDISCONN Disconnect the Connected BLE Device.....	16
2.4.1.4. AT+QBTGATCONNP Update Connection Parameters	17
2.4.1.5. AT+QBTLEADDWHL Add a BLE Device to White List.....	18
2.4.1.6. AT+QBTLEDELWHL Delete a BLE Device from White List	18
2.4.1.7. AT+QBTLEWHLINFO Get the Information of BLE Device in White List	19
2.4.1.8. AT+QBTLEEXMTU Exchange MTU	20
2.4.2. AT Commands of GATT Server.....	21
2.4.2.1. AT+QBTGATSS Add a Service.....	21
2.4.2.2. AT+QBTGATSC Add a Characteristic	22
2.4.2.3. AT+QBTGATSCV Configure Characteristic Value	23
2.4.2.4. AT+QBTGATSCD Add Characteristic Descriptors	24
2.4.2.5. AT+QBTGATSSC Finish Adding Service or Clear All Services	25
2.4.2.6. AT+QBTGATCHSCV Change a Characteristic Value	26
2.4.2.7. AT+QBTGATSIND Send an Indication	27
2.4.2.8. AT+QBTGATSNOD Send a Notification	28
2.4.2.9. AT+QBTGATADV Set Advertising Parameters	29
2.4.2.10. AT+QBTADVDATA Set Advertising Data	31
2.4.2.11. AT+QBTADVSTR Set Advertising Data Format	32
2.4.2.12. AT+QBTADVRSPPDATA Set Scan Response Data	33
2.4.2.13. AT+QBTADV Start/Stop Advertising	34
2.4.2.14. AT+QBTLEIBEA Set iBeacon Data.....	34
2.4.2.15. AT+QBTLEIBEACFG Configure iBeacon Data to NVM	35

2.4.2.16.	AT+QBTLESTATE	Query Connection State.....	36
2.4.2.17.	AT+QBTLESEND	Send Data to GATT Client.....	37
2.4.2.18.	AT+QBTLEGSND	Get Buffer Size of the Channel	37
2.4.2.19.	AT+QBTLERCVM	Set Data Receiving Mode	38
2.4.2.20.	AT+QBTLEREAD	Read Buffer Data.....	39
2.4.3.	Description of GATT Client AT Commands.....		40
2.4.3.1.	AT+QBTSCANPARA	Set Scan Parameters	40
2.4.3.2.	AT+QBTFILTER	Configure Filtering Feature for Scanning Device	41
2.4.3.3.	AT+QBTGATSCAN	Start/Stop Scanning Device	43
2.4.3.4.	AT+QBTGATCONN	Connect a Device.....	43
2.4.3.5.	AT+QBTGATSERV	Scan Services	44
2.4.3.6.	AT+QBTGATINC	Scan an Include Service.....	45
2.4.3.7.	AT+QBTGATCHAR	Scan Characteristics	46
2.4.3.8.	AT+QBTGATDESC	Scan Characteristic Descriptors	47
2.4.3.9.	AT+QBTWRCHAR	Write Characteristic Value	47
2.4.3.10.	AT+QBTWRCHARNORSP	Write Characteristic Value Without Response ...	48
2.4.3.11.	AT+QBTRDCHARUUID	Read Characteristic Value by UUID	49
2.4.3.12.	AT+QBTRDCHARHAND	Read Characteristic Value by Handle	50
2.4.3.13.	AT+QBTGATWRDESC	Write Characteristic Descriptor.....	51
2.4.3.14.	AT+QBTGATRDESC	Read Characteristic Descriptor	52
2.5.	Description of BT AT Commands		53
2.5.1.	General AT Commands		53
2.5.1.1.	AT+QBTSCANMODE	Set Scan Mode.....	53
2.5.2.	Description of HFP AT Commands.....		54
2.5.2.1.	AT+QBTHFPCONN	Connect a Device.....	54
2.5.2.2.	AT+QBTHFPDISCONN	Disconnect a Device	54
2.5.2.3.	AT+QBTHFPVOI	Set the Volume	55
2.5.2.4.	AT+QBTHFPCALL	Control a Voice Call	56
2.5.2.5.	AT+QBTHFPDIAL	Dial a Number	57
2.5.2.6.	AT+QBTHFPVOLR	Turn On/Off Voice Assistant	57
2.5.2.7.	AT+QBTHFPAGDIAL	Make a Call.....	58
2.5.3.	Description of A2DP AVRCP AT Commands.....		59
2.5.3.1.	AT+QBTA2DPDISCONN	Disconnect a Device	59
2.5.3.2.	AT+QBTAVRCPVOL	Set the Volume	60
2.5.3.3.	AT+QBTAVRCPCTRL	Control Audio Playback.....	60
2.5.3.4.	AT+QBTAVRCPSTATE	Get the Audio Playing State.....	61
2.5.4.	Description of SPP AT Commands.....		62
2.5.4.1.	AT+QBTINQUIRY	Search a Device.....	62
2.5.4.2.	AT+QBTSPPCONN	Connect a Device.....	63
2.5.4.3.	AT+QBTSPPDISCONN	Disconnect a Device	64
2.5.4.4.	AT+QBTSPPSSENDATA	Send Data	64
3	Description of URCs		66
3.1.	BLE Related URCs		66
3.1.1.	+QBTGATSCON	Connect GATT	66

3.1.2.	+QBTGATSDCON	Disconnect GATT Connection	66
3.1.3.	+QBTGATMTU	Report MTU During Connection	67
3.1.4.	+QBTGATCONN	Connection Parameters Update.....	67
3.1.5.	+QBTGATDESCDATA	GATT Server Report Characteristic Descriptor.....	67
3.1.6.	+QBTSCANDATAIND	GATT Client Report Scanning Data	68
3.1.7.	+QBTSERVDATA	GATT Client Report Service Data	69
3.1.8.	+QBTCHARDATA	GATT Client Report Characteristic Value.....	69
3.1.9.	+QBTDESCDATA	GATT Client Report Characteristic Descriptor.....	70
3.1.10.	+QBTATTERR	GATT Client Report Attribute Error	70
3.1.11.	+QBTGATNOD	GATT Client Report Receiving Notification.....	71
3.1.12.	+QBTGATIND	GATT Client Reports Receiving Indication	71
3.1.13.	+QBTGATWRCHAR	GATT Client Report the State of Writing Characteristic Value. 72	
3.1.14.	+QBTGATWRCHARNORSP	GATT Client Report the State of Writing Characteristic Value Without Response	72
3.1.15.	+QBTGATRDCHAR	GATT Client Report Reading Characteristic Value by Handle.. 73	
3.1.16.	+QBTGATRDCHARUUID	GATT Client Report Reading Characteristic Value by UUID	73
3.1.17.	+QBTGATWRDESC	GATT Client Report the State of Writing Characteristic Descriptor.....	74
3.1.18.	+QBTGATRDDESC	GATT Client Report Reading Characteristic Descriptor	74
3.1.19.	+QBTGATRDDATAIND	GATT Server Report GATT Client Reading Data.....	75
3.1.20.	+QBTLESTATE	GATT Server Report Connection State Update	75
3.1.21.	+QBTLEVALDATA	GATT Server Report Receiving Data	76
3.1.22.	+QBTLEVALDATI	GATT Server Report Receiving Buffer Data.....	76
3.2.	BT HFP Related URCs		77
3.2.1.	+QBTHFPSCON	Connect HFP.....	77
3.2.2.	+QBTHFPSDCON	Disconnect HFP Connection	77
3.2.3.	+QBTHFPCALL	Change of Call State	77
3.2.4.	+QBTHFPCALS	Change of Call Setting State.....	78
3.2.5.	+QBTHFPNET	Change of Network State	78
3.2.6.	+QBTHFPNETS	Change of Network Signal Strength	79
3.2.7.	+QBTHFPBAT	Change of Battery Level.....	79
3.2.8.	+QBTHFPCALH	Change of Call Holding State	80
3.2.9.	+QBTHFPAUD	Change of Voice State.....	80
3.2.10.	+QBTHFPVOL	Change of Voice Type.....	81
3.2.11.	+QBTHFPNETT	Change of Network Type	81
3.2.12.	+QBTHFPRING	Change of Ring Indication	81
3.2.13.	+QBTHFPCOD	Change of Code Type	82
3.3.	BT A2DP AVRCP Related URCs		82
3.3.1.	+QBTA2DPSCON	Connect A2DP	82
3.3.2.	+QBTA2DPSDCON	Disconnect A2DP Connection.....	83
3.3.3.	+QBTA2DPAUDIOCFG	Configure A2DP Audio	83
3.3.4.	+QBTA2DPAUDIOSTART	Start Playing Audio	84
3.3.5.	+QBTA2DPAUDIOSTOPPED	Stop Playing Audio	84

3.3.6.	+QBTAVRCPSCON	Connect AVRCP	85
3.3.7.	+QBTAVRCPSDCON	Disconnect AVRCP Connection	85
3.3.8.	+QBTAVRCPVOLCHANGE	Change of AVRCP Volume	86
3.4.	BT SPP Related URCS		86
3.4.1.	+QBTSPPSCON	Connect SPP	86
3.4.2.	+QBTSPPSDCON	Disconnect SPP Connection	86
3.4.3.	+QBTSPPREVDATA	Report Received Data	87
3.4.4.	+QBTINQUIRY	Report Device Search	87
3.4.5.	+QBTINQSTATE	Report Device Search Status	88
3.4.6.	+QBTBOND	Report Binding Information	88
3.4.7.	+QBTBONDSTATE	Report Binding Status	88
4	Examples		90
4.1.	BLE Communication		90
4.2.	BT HFP Use Process		97
4.3.	BT A2DP AVRCP Use Process		98
5	Appendix Reference		99

Table Index

Table 1: Applicable Modules	9
Table 2: Types of AT Commands	11
Table 3: Terms and Abbreviations	99

1 Introduction

Bluetooth (BT) technology is an open global specification for wireless data and voice communication. It is a special short-range wireless technology connection based on the low-cost short-range wireless connection, which establishes a communication environment for fixed devices and mobile devices. Bluetooth includes classic Bluetooth and BLE (Bluetooth Low Energy). This document mainly introduces BT function of the Quectel LTE standard EC200U series, EC600U series, and EG915U series modules that can be used in combination with Quectel FC20 series & FC21 modules so as to realize device interconnection through current wireless technology with the lowest power consumption.

The development of classical Bluetooth is based on SPP and HFP protocols. SPP intends to establish a transmission channel between local Bluetooth devices and remote Bluetooth devices to realize data interaction. HFP controls the Bluetooth device to make a voice call in the Bluetooth protocol stack, such as answer, hang up, reject and perform voice calls. HFP* defines two roles: the audio gateway role (AG) and the hands-free component role (HF). HF is the remote audio input and output mechanism of the audio gateway and provides several remote-control functions. It is generally used as the car Bluetooth. AG is the input and output gateway of audio devices. It is usually used on mobile phones. This document only introduces AT commands related to HFP-AG.

Bluetooth Low Energy (Bluetooth LE or BLE) is a wireless personal area network technology designed and marketed by the Bluetooth Special Interest Group aimed at novel applications in the healthcare, fitness, beacons, security, and home entertainment industries. Compared to Classic Bluetooth, Bluetooth Low Energy is intended to provide considerably reduced power consumption. The Bluetooth Low Energy is based on the GATT (Generic Attribute Profile) protocol, which is a general specification for sending and receiving very short data segments called attributes over a Bluetooth connection.

1.1. Applicable Modules

Table 1: Applicable Modules

Module Series	Module
ECx00U	EC200U series
	EC600U series
EG915U	EG915U Series

2 Description of BT AT Commands

2.1. AT Command Introduction

2.1.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on the command line.
- **[...]** Optional parameter of a command or an optional part of TA information response. Square brackets do not appear on the command line. When an optional parameter is not given in a command, the new value equals to its previous value or the default settings, unless otherwise specified.
- **Underline** Default setting of a parameter.

2.1.2. AT Command Syntax

All command lines must start with AT or at and end with **<CR>**. Information responses and result codes always start and end with a carriage return character and a line feed character: **<CR><LF><response><CR><LF>**. In tables presenting commands and responses throughout this document, only the commands and responses are presented, and **<CR>** and **<LF>** are deliberately omitted.

Table 2: Types of AT Commands

Command Type	Syntax	Description
Test Command	AT+<cmd>=?	Test the existence of corresponding Write Command and return information about the type, value, or range of its parameter.
Read Command	AT+<cmd>?	Check the current parameter value of a corresponding Write Command.
Write Command	AT+<cmd>=<p1>[,<p2>[,<p3>[...]]]	Set user-definable parameter value.
Execution Command	AT+<cmd>	Return a specific information parameter or perform a specific action.

2.2. Declaration of AT Command Examples

The AT command examples in this document are provided to help you learn about how to use the AT commands introduced herein. The examples, however, should not be taken as Quectel's recommendation or suggestions about how you should design a program flow or what status you should set the module into. Sometimes multiple examples may be provided for one AT command. However, this does not mean that there exists a correlation among these examples and that they should be executed in a given sequence.

2.3. General BT AT Commands

2.3.1. AT+QBTPWR Turn On/Off BT

This command turns on or turns off BT.

AT+QBTPWR Turn On/Off BT	
Test Command AT+QBTPWR=?	Response +QBTPWR: (range of supported <enable>) OK
Read Command AT+QBTPWR?	Response +QBTPWR: <enable> OK
Write Command AT+QBTPWR=<enable>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<enable>	Integer type. Turn on or off BT. Range: 0–127 (in ORed).
0	Turn off Bluetooth
1	Turn on BLE GATT server
2	Turn on BLE GATT client
4	SPP

- 8 Turn on BT HFP protocol
- 16 Turn on BT A2DP and AVRCP protocol
- 32 Turn on BT A2DP SOURCE and AVRCP TARGET protocol (Currently not supported)
- 64 Turn on BT HFP AG protocol

Example

```
AT+QBTPWR=1
OK
```

2.3.2. AT+ QBTNAME Set BT Device Name

This command sets the name of BT device.

AT+QBTNAME Set BT Device Name	
Test Command AT+QBTNAME=?	Response OK
Read Command AT+QBTNAME?	Response +QBTNAME: <code_type>,<device_name> OK
Write Command AT+QBTNAME=<code_type>,<device_name>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<code_type>	Integer type. Code type. 0 UTF8 1 GBK
<device_name>	String type. BT device name. Maximum: 29 bytes.

Example

```
AT+QBTNAME?
+QBTNAME: 0,"MYBTDEVICE"
```

OK

2.3.3. AT+ QBSEND Send Data

This command sends data.

AT+QBSEND Send Data	
Test Command AT+QBSEND=?	Response +QBSEND: (range of supported <data_type>s), (list of supported <send_mode>s), <string> OK
Write Command AT+QBSEND=<data_type>,<send_mode>[,<string>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<data_type>	Integer type. The type of data to be sent. <ul style="list-style-type: none"> 0 BLE GATT SERVER sends notification 1 BLE GATT SERVER sends indication 2 BT SPP sends data
<send_mode>	Integer type. Sending mode. <ul style="list-style-type: none"> 0 Direct mode 2 Transparent transmission mode
<string>	String type. The data to be sent. Maximum length is 512 bytes. When <send_mode>=2 , this parameter is omitted.

2.4. Description of BLE AT Commands

2.4.1. General AT Commands

2.4.1.1. AT+QBTLEADDR Get BLE Device Address

This command gets BLE device address.

AT+QBTLEADDR Get BLE Device Address	
Test Command AT+QBTLEADDR=?	Response OK
Read Command AT+QBTLEADDR?	Response +QBTLEADDR: <BLE_addr> OK
Maximum Response Time	10 s
Characteristics	/

Parameter

<BLE_addr>	String type. BLE device address, for example, "A662616202C3".
-------------------------	---

Example

```
AT+QBTLEADDR?
+QBTLEADDR: "A662616202C3"
OK
```

2.4.1.2. AT+QBTLERANADDR Get Random Address of BLE Device

This command gets the random address of BLE device.

AT+QBTLERANADDR Get Random Address of BLE Device	
Test Command AT+QBTLERANADDR=?	Response OK
Read Command AT+QBTLERANADDR?	Response +QBTLERANADDR: <random_address>

	OK
Maximum Response Time	10 s
Characteristics	/

Parameter

<random_address> String type. Random address of BLE device, for example, "A662616202C3".

Example

```
AT+QBTLERANADDR?
+QBTLERANADDR: "A662616202C3"
OK
```

2.4.1.3. AT+QBTGATDISCONN Disconnect the Connected BLE Device

This command disconnects the connected BLE device.

AT+QBTGATDISCONN Disconnect the Connected BLE Device

Test Command AT+QBTGATDISCONN=?	Response +QBTGATDISCONN: (range of supported <connID> s) OK
Write Command AT+QBTGATDISCONN=<connID>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<connID> Integer type. Connection ID. Range: 0–65535.

Example

```
AT+QBTGATDISCONN=0
OK
```

2.4.1.4. AT+QBTGATCONNP Update Connection Parameters

This command updates connection parameters.

AT+QBTGATCONNP Update Connection Parameters	
Test Command AT+QBTGATCONNP=?	Response +QBTGATCONNP: (range of supported <connID>s),(range of supported <min_interval>s),(range of supported <max_interval>s),(range of supported <latency>s),(range of supported <timeout>s) OK
Write Command AT+QBTGATCONNP=<connID>,<min_interval>,<max_interval>,<latency>,<timeout>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<connID>	Integer type. Connection ID. Range: 0–65535.
<min_interval>	Integer type. Minimum interval time. Range: 6–3200. Interval: 1.25 ms. Corresponding time range: 7.5 ms–4 s.
<max_interval>	Integer type. Maximum interval time. Range: 6–3200. Interval: 1.25 ms. Corresponding time range: 7.5 ms–4 s.
<latency>	Integer type. Latency, the number of ignored connection events. Range: 0–499.
<timeout>	Integer type. Disconnect timeout. Range: 10–3200. Interval: 10 ms. Corresponding time range: 100 ms–32 s.

Example

```
AT+QBTGATCONNP=0,6,6,0,2000
OK
```

2.4.1.5. AT+QBTLEADDWHL Add a BLE Device to White List

This command adds a BLE device to White List.

AT+QBTLEADDWHL Add a BLE Device to White List	
Test Command AT+QBTLEADDWHL=?	Response +QBTLEADDWHL: (list of supported <addr_type>s),<address> OK
Write Command AT+QBTLEADDWHL=<addr_type>,<address>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<addr_type>	Integer type. Address type. 0 Public address 1 Random address
<address>	String type. BLE device address.

Example

```
AT+QBTLEADDWHL=0,"112233da8040"
OK
```

2.4.1.6. AT+QBTLEDELWHL Delete a BLE Device from White List

This command deletes a BLE device from White List.

AT+QBTLEDELWHL Delete a BLE Device from White List	
Test Command AT+QBTLEDELWHL=?	Response +QBTLEDELWHL: (list of supported <op>s),(list of supported <addr_type>s),<address> OK

Write Command AT+QBTLEDELWHL=<op>[,<addr_type>,<address>]	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration are not saved.

Parameter

<op>	Integer type. Operation type. Omit <addr_type> and <address> when <op>=0 . 0 Delete all BLE devices from the White List 1 Delete a specified BLE device from the White List
<addr_type>	Integer type. Address type. 0 Public address 1 Random address
<address>	String type. BLE device address.

Example

```
AT+QBTLEDELWHL=0
OK
```

2.4.1.7.AT+QBTLEWHLINFO Get the Information of BLE Device in White List

This command gets the information of BLE device in white list

AT+QBTLEWHLINFO Get the Information of BLE Device in White List	
Test Command AT+QBTLEWHLINFO=?	Response OK
Read Command AT+QBTLEWHLINFO?	Response +QBTLEWHLINFO: <addr_type>,<address> OK Or ERROR
Maximum Response Time	10 s
Characteristics	/

Parameter

<addr_type>	Integer type. Address type. 0 Public address 1 Random address
<address>	String type. BLE device address.

Example

```
AT+QBTLEWHLINFO?
+QBTLEWHLINFO: 1,"112233da8040"

OK
```

2.4.1.8.AT+QBTLEEXMTU Exchange MTU

This command exchanges MTU.

AT+QBTLEEXMTU Exchange MTU	
Test Command AT+QBTLEEXMTU?	Response +QBTLEEXMTU: (range of supported <connID>s),(range of supported <MTU>s) OK
Write Command AT+QBTLEEXMTU=<connID>,<MTU>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration are not saved.

Parameter

<connID>	Integer type. Connection ID. Range: 0–65535.
<MTU>	Integer type. Maximum transmission unit. Range: 23–247. Unit: byte.

Example

```
AT+QBTLEEXMTU=0,220

OK
```

2.4.2. AT Commands of GATT Server

2.4.2.1. AT+QBTGATSS Add a Service

This command adds a service.

AT+QBTGATSS Add a Service	
Test Command AT+QBTGATSS=?	Response +QBTGATSS: (range of supported <servID>s),(list of supported <UUID_type>s),<serv_UUID_l>,(range of supported <serv_UUID_s>s),(list of supported <primary>s) OK
Write Command AT+QBTGATSS=<servID>,<UUID_type>[,<serv_UUID_l>][,<serv_UUID_s>],<primary>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration are not saved.

Parameter

<servID>	Integer type. Service ID. Range: 0–65535.
<UUID_type>	Integer type. UUID type. Omit <serv_UUID_s> when <UUID_type>=0. Omit <serv_UUID_l> when <UUID_type>=1. 0 Long 128-bit UUID 1 Short 16-bit UUID
<serv_UUID_l>	String type. 128-bit service UUID.
<serv_UUID_s>	Integer type. 16-bit service UUID. Range: 0–65535.
<primary>	Integer type. Whether it is the primary service. 0 Secondary service 1 Primary service

Example

```
AT+QBTGATSS=0,1,6159,1
OK
```

2.4.2.2. AT+QBTGATSC Add a Characteristic

This command adds a characteristic to an existing service.

AT+QBTGATSC Add a Characteristic	
Test Command AT+QBTGATSC=?	Response +QBTGATSC: (range of supported <servID>s),(range of supported <charaID>s),(range of supported <properties>s),(list of supported <UUID_type>s),<serv_UUID_I>,(range of supported <serv_UUID_s>s) OK
Write Command AT+QBTGATSC=<servID>,<charaID>,<properties>,<UUID_type>[,<serv_UUID_I>][,<serv_UUID_s>]	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<servID>	Integer type. Service ID. Range: 0–65535.
<charaID>	Integer type. Characteristic ID. Range: 0–65535.
<properties>	Integer type. Characteristic properties. Set Bit as 1 to enable the corresponding property. Range: 0–255 (in ORed). 0 Disable all the properties Bit0 Broadcast Bit1 Read Bit2 Write without confirmation of link layer Bit3 Write Bit4 Notify Bit5 Indicate Bit6 Authenticated signed Writes Bit7 Extended properties
<UUID_type>	Integer type. UUID type. Omit <serv_UUID_s> when <UUID_type>=0. Omit <serv_UUID_I> when <UUID_type>=1. 0 Long 128-bit UUID 1 Short 16-bit UUID
<serv_UUID_I>	String type. 128-bit service UUID.
<serv_UUID_s>	Integer type. 16-bit service UUID. Range: 0–65535.

Example

```
AT+QBTGATSC=0,0,58,1,10777
OK
```

2.4.2.3. AT+QBTGATSCV Configure Characteristic Value

This command configures characteristic value.

AT+QBTGATSCV Configure Characteristic Value	
Test Command AT+QBTGATSCV=?	Response +QBTGATSCV: (range of supported <servID>s),(range of supported <charaID>s),(range of supported <permission>s),(list of supported <UUID_type>s), <serv_UUID_l> ,(range of supported <serv_UUID_s>s),(range of supported <value_length>s), <value> OK
Write Command AT+QBTGATSCV=<servID>,<charaID>,<permission>,<UUID_type>[,<serv_UUID_l>][,<serv_UUID_s>],<value_length>,<value>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<servID>	Integer type. Service ID. Range: 0–65535.
<charaID>	Integer type. Characteristic ID. Range: 0–65535.
<permission>	Integer type. The permission of characteristic value. Set Bit as 1 to enable the corresponding permission. Range: 0–1023 (in ORed). 0 Disable all the permissions Bit0 Read only Bit1 Write only Bit2 Read, authentication required Bit3 Read, authorization required Bit4 Read, encryption required Bit5 Read, authorization and authentication required Bit6 Write, authentication required Bit7 Write, authorization required

	Bit8	Write, encryption required
	Bit9	Write, authorization and authentication required
<UUID_type>	Integer type. UUID type. Omit <serv_UUID_s> when <UUID_type>=0. Omit <serv_UUID_l> when <UUID_type>=1.	
	0	Long 128-bit UUID
	1	Short 16-bit UUID
<serv_UUID_l>	String type. 128-bit service UUID.	
<serv_UUID_s>	Integer type. 16-bit service UUID. Range: 0–65535.	
<value_length>	Integer type. Length of characteristic value. Range: 0–512. Unit: byte.	
<value>	Integer type. Characteristic value.	

Example

```
AT+QBTGATSCV=1,1,3,1,10777,244,"1234"
OK
```

2.4.2.4. AT+QBTGATSCD Add Characteristic Descriptors

This command adds characteristic descriptors.

AT+QBTGATSCD Add Characteristic Descriptors	
Test Command AT+QBTGATSCD=?	Response +QBTGATSCD: (range of supported <servID>s),(range of supported <charalD>s),(range of supported <permission>s),(list of supported <UUID_type>s),<serv_UUID_l>,(range of supported <serv_UUID_s>s),(range of supported <value_length>s),<value> OK
Write Command AT+QBTGATSCD=<servID>,<charalD>,<permission>,<UUID_type>[,<serv_UUID_l>][,<serv_UUID_s>],<value_length>,<value>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<servID>	Integer type. Service ID. Range: 0–65535.
<charalD>	Integer type. Characteristic ID. Range: 0–65535.

<permission>	Integer type. Characteristic value permission. Set Bit as 1 to enable the corresponding permission. Range: 0–1023 (in ORed). 0 Disable all the permissions Bit0 Read only Bit1 Write only Bit2 Read, authentication required Bit3 Read, authorization required Bit4 Read, encryption required Bit5 Read, authorization and authentication required Bit6 Write, authentication required Bit7 Write, authorization required Bit8 Write, encryption required Bit9 Write, authorization and authentication required
<UUID_type>	Integer type. UUID type. Omit <serv_UUID_s> when <UUID_type>=0 . Omit <serv_UUID_l> when <UUID_type>=1 . 0 Long 128-bit UUID 1 Short 16-bit UUID
<serv_UUID_l>	String type. 128-bit service UUID.
<serv_UUID_s>	Integer type. 16-bit service UUID. Range: 0–65535.
<value_length>	Integer type. Length of characteristic descriptor value. Range: 0–65535. Unit: byte.
<value>	Integer type. Characteristic descriptor value.

Example

```
AT+QBTGATSCD=0,0,3,1,10498,2,"1234"
OK
```

2.4.2.5. AT+QBTGATSSC Finish Adding Service or Clear All Services

This command finishes adding service or clears all services.

AT+QBTGATSSC Finish Adding Service or Clear All Services

Test Command AT+QBTGATSSC=?	Response +QBTGATSSC: (list of supported <type>s),(list of supported <op>s) OK
Write Command AT+QBTGATSSC=<type>[,<op>]	Response OK Or ERROR

Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<type>	Integer type. Finish adding service or clear all services. Omit <op> when <type>=0 . 0 Clear all services and characteristics, including the default GAP and GATT services. And the default GAP and GATT services cannot be restored the next time a service is added. 1 Finish adding service.
<op>	Integer type. Whether to retain the default GAP and GATT services. <u>0</u> The default GAP and GATT services are not reserved. The GAP and GATT services reserved by the system are deleted. In some cases, the peer BLE device does not initiate a service update request. 1 Reserve the default GAP and GATT services.

Example

```
AT+QBTGATSSC=1,1
OK
```

2.4.2.6. AT+QBTGATCHSCV Change a Characteristic Value

This command changes a characteristic value.

AT+QBTGATCHSCV Change a Characteristic Value	
Test Command AT+QBTGATCHSCV=?	Response +QBTGATCHSCV: (range of supported <servID>s),(range of supported <charaID>s),(range of supported <value_length>s),<value> OK
Write Command AT+QBTGATCHSCV=<servID>,<charaID>,<value_length>,<value>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<servID>	Integer type. Service ID. Range: 0–65535.
<charalD>	Integer type. Characteristic ID. Range: 0–65535.
<value_length>	Integer type. Length of changed characteristic value, starting from 0. The value of this parameter cannot exceed the <length> configured in AT+QBTGATSCV . The characteristic value remains unchanged if the length of changed characteristic value is less than the length of the original characteristic value. Unit: byte.
<value>	Integer type. Characteristic value.

Example

```
AT+QBTGATCHSCV=0,0,2,"0012"
OK
```

2.4.2.7.AT+QBTGATSIND Send an Indication

This command sends an indication.

AT+QBTGATSIND Send an Indication	
Test Command AT+QBTGATSIND=?	Response +QBTGATSIND: (range of supported <op> s),(range of supported <connID> s),(range of supported <att_handle> s), <value_length> , <value> OK
Write Command AT+QBTGATSIND=<op>,<connID>,<att_handle>[,<value_length>,<value>]	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<op>	Integer type. Omit <value_length> and <value> when <op> is either 1 or 2. 0 Direct mode 1 Echo mode 2 Transparent transmission mode
<connID>	Integer type. Connection ID. Range: 0–65535.

<att_handle>	Integer type. Attribute handle. Range: 1–65535.
<value_length>	Integer type. The length of indication content. Range: 0 to <MTU>-3. Unit: byte.
<value>	Integer type. Indication content.

Example

```
AT+QBTGATSIND=0,0,18,4,"1111"
OK
```

2.4.2.8. AT+QBTGATSNO D Send a Notification

This command sends a notification.

AT+QBTGATSNO D Send a Notification	
Test Command AT+QBTGATSNO D=?	Response +QBTGATSNO D: (range of supported <op>s),(range of supported <connID>s),(range of supported <att_handle>s), <value_length>,<value> OK
Write Command AT+QBTGATSNO D=<op>,<connID>,<att_handle>[,<value_length>,<value>]	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<op>	Integer type. Omit <value_length> and <value> when <op> is either 1 or 2. 0 Direct mode 1 Echo mode 2 Transparent transmission mode
<connID>	Integer type. Connection ID. Range: 0–65535.
<att_handle>	Integer type. Attribute handle. Range: 1–65535.
<value_length>	Integer type. The length of notification content. Range: 0 to <MTU>-3. Unit: byte.
<value>	Integer type. Notification content.

Example

```
AT+QBTGATSNOD=0,0,18,4,"1111"
OK
```

2.4.2.9.AT+QBTGATADV Set Advertising Parameters

This command sets advertising parameters.

AT+QBTGATADV Set Advertising Parameters

Test Command AT+QBTGATADV=?	Response +QBTGATADV: (list of supported <op> s),(range of supported <min_interval> s),(range of supported <max_interval> s),(range of supported <adv_type> s),(list of supported <own_addrtype> s),(range of supported <channel_map> s),(range of supported <filter> s),(list of supported <remote_addrtype> s), <remote_addr> OK
Write Command AT+QBTGATADV=<op>,<min_interval>,<max_interval>,<adv_type>,<own_addrtype>,<channel_map>,<filter>[[,<remote_addrtype>][,<remote_addr>]]	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration are not saved.

Parameter

<op>	Integer type. Set advertising parameters of all address or the specified address. 0 Specified address 1 All address. Omit <remote_addrtype> and <remote_addr> when <op>=1
<min_interval>	Integer type. Minimum advertising interval, which should be less than <max_interval> . Range: 32–16384. Interval: 0.625 ms. Corresponding time range: 20 ms–10.24 s.
<max_interval>	Integer type. Maximum advertising interval. Range: 32–16384. Interval: 0.625 ms. Corresponding time range: 20 ms–10.24 s.
<adv_type>	Integer type. Advertising type. 0 Connectable undirected advertising

	<ol style="list-style-type: none"> 1 Connectable high duty cycle directed advertising. <min_interval> and <max_interval> do not work for connectable high duty cycle directed advertising, and the fixed advertising interval is 3.75 ms. The advertising time of directed advertising is 1.28 s. 2 Non-connectable undirected advertising. For this type, advertising interval parameter should be more than 160. 3 Scannable undirected advertising. For this type, the advertising interval should be more than 160. 4 Connectable low duty cycle directed advertising. <min_interval> and <max_interval> do not work for connectable low duty cycle directed advertising, and the fixed advertising interval is 3.75 ms. The advertising time of directed advertising is 1.28 s.
<own_addrtype>	<p>Integer type. Address type of local device.</p> <ol style="list-style-type: none"> 0 Public address 1 Random address
<channel_map>	<p>Integer type. Channels on which the advertising message is transmitted. There are three channels: 37, 38, 39, which can be represented by 3 bit. A number from 1 to 7 is used to represent the combination of the three channels.</p> <ol style="list-style-type: none"> 1 Channel 37 2 Channel 38 3 Channel 37 and Channel 38 4 Channel 39 5 Channel 37 and Channel 39 6 Channel 38 and Channel 39 7 Channel 37, Channel 38 and Channel 39
<filter>	<p>Integer type. Advertising filter strategy.</p> <ol style="list-style-type: none"> 0 Process scan and connection requests from all devices 1 Process connection requests from all devices and only process scan requests from devices that are in the White List 2 Process scan requests from all devices and only process connection requests from devices that are in the White List 3 Process scan and connection requests only from devices in the White List
<remote_addrtype>	<p>Integer type. The address type of remote device.</p> <ol style="list-style-type: none"> 0 Public address 1 Random address
<remote_addr>	<p>String type. Remote device address.</p>

Example

```
AT+QBTGATADV=0,128,160,0,1,7,0,1,"b17e431d1c5f"
OK
```

2.4.2.10.AT+QBTADVDATA Set Advertising Data

This command sets advertising data. The advertising message format is shown in the following figure:

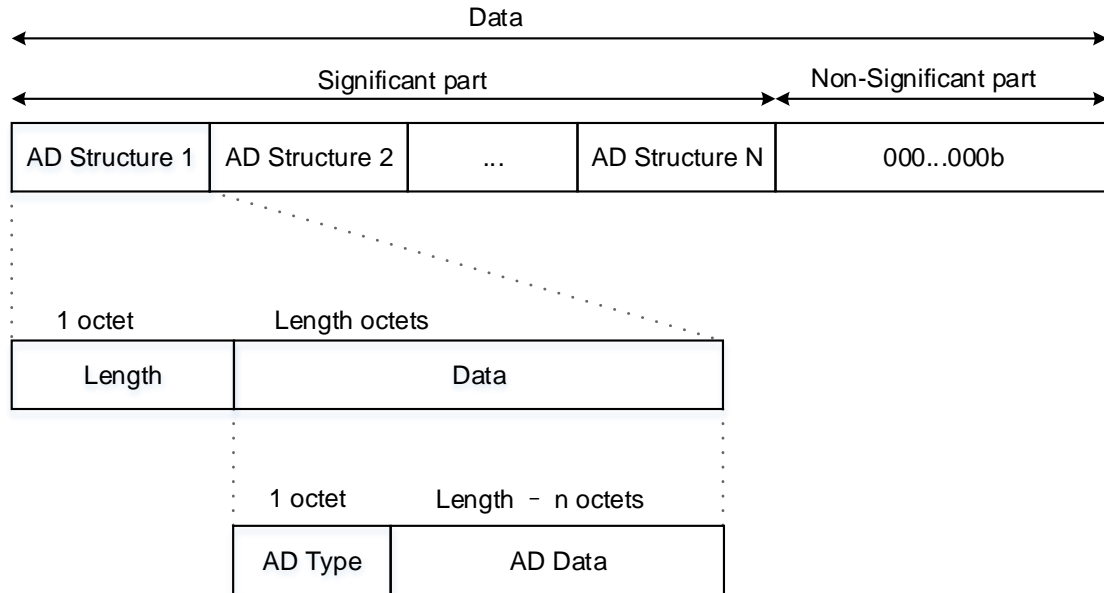


Figure 1: Advertising Message Format

AT+QBTADVDATA Set Advertising Data	
Test Command AT+QBTADVDATA=?	Response OK
Write Command AT+QBTADVDATA=<data_length>,<data>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<data_length>	Integer type. The length of advertising data. Maximum: 31 octets.
<data>	String type. Advertising data, which consist of the following three fields (that is, multiple AD Structures), and whose composition mode conforms to the message format in the above figure, and the content must be a hexadecimal string.
Length	Length of AD Structure, including the lengths of AD Type and AD Data, except from one octet of Length itself. Maximum: 0x1e, which means the

	maximum length of a data field is 30 octets
AD Type	1 octet, the type to which the advertising data belongs, such as TX Power Level(0x0A), Local Name(0x09), Le Role(0x1C), Service UUIDs(0x16). The peer device scans the adverting data and figures out the meaning of the data according to the AD Type. For the value and meaning of AD Type, please refer to the official BLE document <i>Core_v5.2</i> .
AD Data	Advertising data content, organized in big-endian format.

Example

```
AT+QBTADVDATA=3,"020105"
OK
```

2.4.2.11.AT+QBTADVSTR Set Advertising Data Format

This command sets the advertising data format.

AT+QBTGATADVSTR Set Advertising Data Format

Test Command AT+QBTADVSTR=?	Response OK
Write Command AT+QBTADVSTR=<advdata_type>,<code_type1>,<advdata1>[,<code_type2>,<advdata2>[,...]]	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

- <advdata_type>** Integer type. Advertising data type in ORed. The total length of all AD Structures cannot exceed 31 octets. If the length of AD Data is 6 octets, 2 octets need to be added to **<data_length>**. 1 octet is the length of Length itself and 1 octet is the length of AD Type, totally 8 octets. Set Bit as 1 to enable to advertise the corresponding data type.
- Bit0 Advertising BLE device name.
 - Bit1 Flag. **<advdata>** is required to be included, which only supports hexadecimal format.
 - Bit2 Manufacturer data. **<advdata>** is required to be included.
 - Bit3 Transmission power level. **<advdata>** is required to be included, which only

	supports hexadecimal format.
Bit4	Complete list of 16-bit service class UUIDs. <advdata> is required to be included, which only supports hexadecimal format.
Bit5	Service data. <advdata> is required to be included.
<code_type>	Integer type. Code type.
0	Hexadecimal string format.
1	GBK encoding
2	UTF8 encoding
<advdata>	String type. Advertising data.

Example

```
AT+QBTADVSTR=63,1,"quec",0,"06",1,"quectel",0,"0a",0,"1803",1,"bata"
OK
```

2.4.2.12.AT+QBTADVRSPDATA Set Scan Response Data

This command sets the scan response data.

AT+QBTADVRSPDATA Set Scan Response Data	
Test Command AT+QBTADVRSPDATA=?	Response OK
Write Command AT+QBTADVRSPDATA=<scan_rspdata_length>,<scan_rspdata>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<scan_rspdata_length>	Integer type. The length of scan response data. Maximum: 31 octets.
<scan_rspdata>	String type. Scan response data. The data format is the same as advertising data.

Example

```
AT+QBTADVRSPDATA=7,"06094138393130"
OK
```

2.4.2.13.AT+QBTADV Start/Stop Advertising

This command starts or stops advertising.

AT+QBTADV Start/Stop Advertising	
Test Command AT+QBTADV=?	Response +QBTADV: (list of supported <enable>s) OK
Write Command AT+QBTADV=<enable>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<enable>	Integer type. Start/Stop advertising. 0 Stop advertising 1 Start advertising
-----------------------	--

Example

```
AT+QBTADV=1
OK
```

2.4.2.14.AT+QBTLEIBEA Set iBeacon Data

This command sets iBeacon data.

AT+QBTLEIBEA Set iBeacon Data	
Test Command AT+QBTLEIBEA=?	Response +QBTLEIBEA: <UUID_I>,(range of supported <major>s), (range of supported <minor>s) OK
Write Command AT+QBTLEIBEA=<UUID_I>,<major>,<minor>	Response OK

<minor>	Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration are not saved.

Parameter

<UUID_I>	String type. 128-bit UUID.
<major>	Integer type. Major. Range: 0–65535.
<minor>	Integer type. Minor. Range: 0–65535.

Example

```
AT+QBTLEIBEA="f5899b5f8000008000100000FE180000",20,25
OK
```

2.4.2.15. AT+QBTLEIBEACFG Configure iBeacon Data to NVM

This command configures iBeacon data to NVM.

AT+QBTLEIBEACFG Configure iBeacon Data to NVM	
Test Command AT+QBTLEIBEACFG=?	Response +QBTLEIBEACFG: <UUID_I>,(range of supported <major>),<major>),(range of supported <minor>),<minor>) OK
Write Command AT+QBTLEIBEACFG=<UUID_I>,<major>,<major>,<minor>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration are not saved.

Parameter

<UUID_I>	String type. 128-bit UUID.
<major>	Integer type. Major. Range: 0–65535.

<minor> Integer type. Minor. Range: 0–65535.

Example

```
AT+QBTLEIBACFG="f5899b5f800000800010000FE180000",20,25
OK
```

2.4.2.16.AT+QBTLESTATE Query Connection State

This command queries connection state.

AT+QBTLESTATE Query Connection State	
Test Command AT+QBTLESTATE=?	Response +QBTLESTATE: (range of supported <cid>s),(range of supported <connID>s), <address> ,(list of supported <conn_state>s),(range of supported <att_handle>s) OK
Read Command AT+QBTLESTATE?	Response +QBTLESTATE: <cid> , <connID> , <address> , <conn_state> , <att_handle> OK
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<cid>	Integer type. Channel ID. Range: 0–65535.
<connID>	Integer type. Connection ID. Range: 0–65535.
<address>	String type. Address of the connected device.
<conn_state>	Integer type. Connection state. 0 Disconnected 1 Connected
<att_handle>	Integer type. Attribute handle. Range: 1–65535.

2.4.2.17. AT+QBTLESEND Send Data to GATT Client

This command sends data to GATT client.

AT+QBTLESEND Send Data to GATT Client	
Test Command AT+QBTLESEND=?	Response +QBTLESEND: (range of supported <cid>s),(list of supported <type>s),(range of supported <data_length>s),<data> OK
Write Command AT+QBTLESEND=<cid>,<type>,<data_length>,<data>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<cid>	Integer type. Channel ID. Range: 0–65535.
<type>	Integer type. Data type. 0 Notification 1 Indication
<data_length>	Integer type. Data length. Range: 0–1000. Unit: byte.
<data>	Hexadecimal string type. The sent Data.

2.4.2.18. AT+QBTLEGSND Get Buffer Size of the Channel

This command gets buffer size of the channel.

AT+QBTLEGSND Get Buffer Size of the Channel	
Test Command AT+QBTLEGSND=?	Response +QBTLEGSND: (range of supported <cid>s) OK
Read Command AT+QBTLEGSND?	Response +QBTLEGSND: <cid>,<size>,<nsend>

	OK
Write Command AT+QBTLEGSND=<cid>	Response +QBTLEGSND: <cid>,<size>,<nsend> OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<cid>	Integer type. Channel ID. Range: 0–65535.
<size>	Integer type. The allowable size of buffer data in the channel. Maximum: 1000 bytes.
<nsend>	Integer type. Size of data not sent in the buffer. Range: 0–1000. Unit: byte.

Example

```
AT+QBTLEGSND=1
+QBTLEGSND: 1,1000,0
OK
```

2.4.2.19.AT+QBTLERCVM Set Data Receiving Mode

This command sets the data receiving mode of the server.

AT+QBTLERCVM Set Data Receiving Mode	
Test Command AT+QBTLERCVM=?	Response +QBTLERCVM: (list of supported <type>s) OK
Read Command AT+QBTLERCVM?	Response +QBTLERCVM: <type>,<time> OK
Write Command AT+QBTLERCVM=<type>,<time>	Response OK Or

	ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<type>	Integer type. The data receiving mode for the server. Omit <time> when <type>=0 . 0 Direct mode 1 Buffer mode
<time>	Integer type. Time to receive buffer data from the channel and report URC. No report of URC when <time>=0 . Range: 0–3600000. Unit: ms.

Example

```
AT+QBTLERCVM=1,2000
OK
```

2.4.2.20.AT+QBTLERead Read Buffer Data

This command reads buffer data.

AT+QBTLERead Read Buffer Data	
Test Command AT+QBTLERead=?	Response +QBTLERead: (range of supported <cid>s),(range of supported <data_length>s) OK
Write Command AT+QBTLERead=<cid>,<data_length>	Response +QBTLERead: <cid>,<data_length>,<data> OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<cid>	Integer type. Channel ID. Range: 0–65535.
<data_length>	Integer type. The length of read buffer data. Range: 0–2048. Unit: byte.
<data>	Hexadecimal string type. The read data.

2.4.3. Description of GATT Client AT Commands

2.4.3.1. AT+QBTSCANPARA Set Scan Parameters

This command sets scan parameters.

AT+QBTSCANPARA Set Scan Parameters

Test Command AT+QBTSCANPARA=?	Response +QBTSCANPARA: (list of supported <scan_mode>s),(range of supported <scan_interval>s),(range of supported <scan_window>s),(range of supported <scan_type>s),(list of supported <own_addrtype>s) OK
Write Command AT+QBTSCANPARA=<scan_mode>,<scan_interval>,<scan_window>,<scan_type>,<own_addrtype>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration are not saved.

Parameter

<scan_mode>	Integer type. Scanning mode. 0 Passive scan 1 Active scan
<scan_interval>	Integer type. Scan interval. Range: 4–16384. Interval: 0.625 ms. Corresponding time range: 2.5 ms–10.24 s.
<scan_window>	Integer type. Scan window, that is the duration of the scan, which should be less than <interval> . Range: 4–16384. Interval: 0.625 ms. Corresponding time range: 2.5 ms–10.24 s.
<scan_type>	Integer type. Scanning filter strategy. 0 Accept all advertising packets except the directed advertising packets 1 Accept advertising packets from devices in the White List except the directional

- advertising of this device
- 2 Accept undirected advertising packets, directed advertising packets addressed to this device or directed advertising packets where the initiator address is a resolvable private address (Currently not supported)
- 3 Accept Undirected advertising packets from devices in the White List, directed advertising packets addressed to this device or directed advertising packets where the initiator address is a resolvable private address (Currently not supported)

<own_addrtype> Integer type. The address type of local device.

- 0 Public address
- 1 Random address

Example

```
AT+QBTSCANPARA=0,96,48,0,0
OK
```

2.4.3.2. AT+QBTFILTER Configure Filtering Feature for Scanning Device

This command configures the filtering feature for scanning device.

AT+QBTFILTER Configure Filtering Feature for Scanning Device

Test Command AT+QBTFILTER=?	Response +QBTFILTER: (list of supported <RSSI_filter>s),(range of supported <RSSI>s) +QBTFILTER: (list of supported <name_filter>s),(range of supported <name_type>s),<filtername> +QBTFILTER: (list of supported <same_dev_filter>s),(range of supported <same_dev_filter_switch>s) OK
Read Command AT+QBTFILTER?	Response +QBTFILTER: 1,<RSSI> +QBTFILTER: 2,<name_type>,<filtername> +QBTFILTER: 3,<same_dev_filter_switch> OK
Write Command Configure to filter device by RSSI AT+QBTFILTER=<RSSI_filter>[,<RSSI>]	Response OK Or ERROR
Write Command	Response

Configure to filter device by device name AT+QBTFILTER=<name_filter>[,<name_type>[,<filtername>]]	OK Or ERROR
Write Command Configure deduplication function AT+QBTFILTER=<same_dev_filter>[,<same_dev_filter_switch>]	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration are not saved.

Parameter

<RSSI_filter>	Integer type. Configure whether to filter device by RSSI. 0 Restore the default configuration, that is do not filter device by RSSI. Omit <RSSI> when <RSSI_filter>=0 . 1 Configure to filter device by RSSI, and <RSSI> cannot be omitted.
<RSSI>	Integer type. Received signal strength indication. Range: 0–255. The smaller the parameter value is, the higher the signal strength is.
<name_filter>	Integer type. Configure whether to filter device by device name. 0 Restore the default configuration, that is do not filter device by device name. Omit <name_type> and <filtername> when <name_filter>=0 . 2 Configure to filter device by device name. <name_type> and <filtername> cannot be omitted.
<name_type>	Integer type. The filtering type of device name. 0 Disable to filter device by device name. <filtername> is omitted. 1 Filter device by full device name 2 Filter device by key characters of device name
<filtername>	String type. The characters to be filtered.
<same_dev_filter>	Integer type. Configure deduplication function. 0 Restore the default configuration, that is enable deduplication function. Omit <same_dev_filter_switch> when <same_dev_filter>=0 . 3 Enable or disable the deduplication function
<same_dev_filter_switch>	Integer type. Enable or disable the deduplication function. 0 Enable the deduplication function and the same device will not be reported repeatedly. 1 Disable the deduplication function and the same device will be reported repeatedly.

2.4.3.3. AT+QBTGATSCAN Start/Stop Scanning Device

This command starts or stops scanning device.

AT+QBTGATSCAN Start/Stop Scanning Device	
Test Command AT+QBTGATSCAN=?	Response +QBTGATSCAN: (list of supported <activate>s) OK
Write Command AT+QBTGATSCAN=<activate>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<activate>	Integer type. Start/stop scanning device. 0 Stop scanning 1 Start scanning
-------------------------	--

Example

```
AT+QBTGATSCAN=1
OK
```

2.4.3.4. AT+QBTGATCONN Connect a Device

This command connects a device.

AT+QBTGATCONN Connect a Device	
Test Command AT+QBTGATCONN=?	Response +QBTGATCONN: (list of supported <conn_type>s),(list of supported <addr_type>s),<address> OK
Write Command AT+QBTGATCONN=<conn_type>[,<	Response OK

addr_type>],<address>	Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration are not saved.

Parameter

<conn_type>	Integer type. Omit <type> when <conn_type>=0. 0 Cancel connection 1 Connect
<type>	Integer type. Address type of the device to be connected. 0 Public address 1 Random address
<address>	String type. Device address to be connected.

Example

```
AT+QBTGATCONN=1,1,"554dd0dc5854"
OK
```

2.4.3.5.AT+QBTGATSERV Scan Services

This command scans services.

AT+QBTGATSERV Scan Services	
Test Command AT+QBTGATSERV=?	Response +QBTGATSERV: (list of supported <type>s),(range of supported <connID>s),(list of supported <UUID_type>s),<UUID_I>,(range of supported <UUID_s>s) OK
Write Command AT+QBTGATSERV=<type>,<connID>[,<UUID_type>][,<serv_UUID_I>][,<UUID_s>]]	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<type>	Integer type. Omit <UUID_type> , <UUID_I> and <UUID_s> when <type>=0 . 0 Scan all services 1 Scan a specified service
<connID>	Integer type. Connection ID. Range: 0–65535.
<UUID_type>	Integer type. Service UUID type. 0 128-bit UUID. Omit <UUID_s> when <UUID_type>=0 . 1 16-bit UUID. Omit <UUID_I> when <UUID_type>=1 .
<UUID_I>	String type. 128-bit UUID.
<UUID_s>	Integer type. 16-bit UUID. Range: 0–65535.

Example

```
AT+QBTGATSERV=0,0
OK
```

2.4.3.6. AT+QBTGATINC Scan an Include Service

This command scans an include service.

AT+QBTGATINC Scan an Include Service	
Test Command AT+QBTGATINC=?	Response +QBTGATINC: (range of supported <connID>s),(range of supported <start_handle>s),(range of supported <end_handle>s) OK
Write Command AT+QBTGATINC=<connID><start_handle>,<end_handle>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration are not saved.

Parameter

<connID>	Integer type. Connection ID. Range: 0–65535.
<start_handle>	Integer type. Start handle. Range: 1–65535.
<end_handle>	Integer type. End handle. Range: 1–65535.

Example

```
AT+QBTGATINC=0,16,65535
OK
```

2.4.3.7. AT+QBTGATCHAR Scan Characteristics

This command scans characteristics.

AT+QBTGATCHAR Scan a Characteristics	
Test Command AT+QBTGATCHAR=?	Response +QBTGATCHAR: (range of supported <connID>s),(range of supported <start_handle>),(range of supported <end_handle>s) OK
Write Command AT+QBTGATCHAR=<connID>,<start_handle>,<end_handle>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration are not saved.

Parameter

- <connID> Integer type. Connection ID. Range: 0–65535.
- <start_handle> Integer type. Characteristic start handle. Range: 1–65535.
- <end_handle> Integer type. Characteristic end handle. Range: 1–65535.

Example

```
AT+QBTGATCHAR=0,16,65535
OK
```

2.4.3.8.AT+QBTGATDESC Scan Characteristic Descriptors

This command scans characteristic descriptors.

AT+QBTGATDESC Scan a Characteristic Descriptors	
Test Command AT+QBTGATDESC=?	Response +QBTGATDESC: (range of supported <connID>s),(range of supported <start_handle>s),(range of supported <end_handle>s) OK
Write Command AT+QBTGATDESC=<connID>,<start_handle>,<end_handle>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

- <connID> Integer type. Connection ID. Range: 0–65535.
- <start_handle> Integer type. Descriptor start handle. Range: 1–65535.
- <end_handle> Integer type. Descriptor end handle. Range: 1–65535.

Example

```
AT+QBTGATDESC=0,19,65535
OK
```

2.4.3.9.AT+QBTWRCHAR Write Characteristic Value

This command writes characteristic value.

AT+QBTWRCHAR Write Characteristic Value	
Test Command AT+QBTWRCHAR=?	Response +QBTWRCHAR: (range of supported <connID>s),(range of supported <att_handle>s),(range of supported <value_length>s),<value>,(list of supported <islong>s),(range of supported <offset>s)

	OK
Write Command AT+QBTCHAR=<connID>,<att_handle>,<value_length>,<value>,<islong>,<offset>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<connID>	Integer type. Connection ID. Range: 0–65535.
<att_handle>	Integer type. Characteristic value handle. Range: 1–65535.
<length>	Integer type. Data length. Range: 1–65535. Unit: byte.
<value>	String type. Written data.
<islong>	Integer type. Whether multiple data packets are required. 0 Multiple data packets are not required. 1 Multiple data packets are required.
<offset>	Integer type. Offset address required after the data is separated in multiple data packets. Range: 0–65535. Default: 0.

Example

```
AT+QBTWRCHAR= 0,41,2,"1234",0,0
OK
```

2.4.3.10.AT+QBTWRCHARNORSP Write Characteristic Value Without Response

This command writes characteristic value without response.

AT+QBTWRCHARNORSP Write Characteristic Value Without Response	
Test Command AT+QBTWRCHARNORSP=?	Response +QBTWRCHARNORSP: (range of supported <connID>s),(range of supported <att_handle>s),(range of supported <value_length>s),<value> OK
Write Command AT+QBTWRCHARNORSP=<connID>,<	Response OK

att_handle>,<value_length>,<value>	Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<connID>	Integer type. Connection ID. Range: 0–65535.
<att_handle>	Integer type. Characteristic handle. Range: 1–65535.
<value_length>	Integer type. The length of data. Range: 1–65535. Unit: byte.
<value>	String type. The written data.

Example

```
AT+QBTRDCHARUID=0,41,2,"1234"
OK
```

2.4.3.11.AT+QBTRDCHARUID Read Characteristic Value by UUID

This command reads characteristic value by UUID.

AT+QBTRDCHARUID Read Characteristic Value by UUID	
Test Command AT+QBTRDCHARUID=?	Response +QBTRDCHARUID: (range of supported <connID>s),(list of supported <UUID_type>s),<UUID_l>,(range of supported <UUID_s>s),(range of supported <start_handle>s),(range of supported <end_handle>s) OK
Write Command AT+QBTRDCHARUID=<connID>,<UUID_type>[,<UUID_l>][,<UUID_s>],<start_handle>,<end_handle>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<connID>	Integer type. Connection ID. Range: 0–65535.
<UUID_type>	Integer type. Service UUID. 0 128bit UUID. Omit <UUID_s> when <UUID_type>=0 . 1 16bit UUID. Omit <UUID_l> when <UUID_type>=1 .
<UUID_l>	String type. 128-bit UUID.
<UUID_s>	Integer type. 16-bit UUID. Range: 0–65535.
<start_handle>	Integer type. Characteristic start handle. Range: 1–65535.
<end_handle>	Integer type. Characteristic end handle. Range: 1–65535.

Example

```
AT+QBTRDCHARUUID=0,1,6159,18,18
OK
```

2.4.3.12. AT+QBTRDCHARHAND Read Characteristic Value by Handle

This command reads characteristic value by handle.

AT+QBTRDCHARHAND Read Characteristic Value by Handle	
Test Command AT+QBTRDCHARHAND=?	Response +QBTRDCHARHAND: (range of supported <connID>s),(range of supported <att_handle>s),(list of supported <islong>s),(range of supported <offset>s) OK
Write Command AT+QBTRDCHARHAND=<connID>,<att_handle>,<islong>,<offset>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<connID>	Integer type. Connection ID. Range: 0–65535.
<att_handle>	Integer type. Characteristic handle. Range: 1–65535.
<islong>	Integer type. Whether multiple data packets are required. <u>0</u> Multiple data packets are not required

<offset>	1 Multiple data packets are required Integer type. Offset address required after the data is separated in multiple data packets. Range: 0–65535.
-----------------------	---

Example

```
AT+QBTRDCHARHAND= 0,18,0,0
OK
```

2.4.3.13.AT+QBTGATWRDESC Write Characteristic Descriptor

This command writes characteristic descriptor.

AT+QBTGATWRDESC Write Characteristic Descriptor

Test Command AT+QBTGATWRDESC=?	Response +QBTGATWRDESC: (range of supported <connID> s),(range of supported <att_handle> s),(range of supported <value_length> s), <value> OK
Write Command AT+QBTGATWRDESC=<connID>,<att_handle>,<value_length>,<value>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<connID>	Integer type. Connection ID. Range: 0–65535.
<att_handle>	Integer type. Descriptor handle. Range: 1–65535.
<length>	Integer type. The length of data. Range: 1–65535. Unit: byte.
<value>	String type. Written data.

Example

```
AT+QBTGATWRDESC= 0,19,2,"1234"
OK
```

2.4.3.14.AT+QBTGATRDESC Read Characteristic Descriptor

This command reads characteristic descriptor.

AT+QBTGATRDESC Read Characteristic Descriptor	
Test Command AT+QBTGATRDESC=?	Response +QBTGATRDESC: (range of supported <connID>s),(range of supported <att_handle>s),(list of supported <islong>s) OK
Write Command AT+QBTGATRDESC=<connID>,<att_handle>,<islong>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<connID>	Integer type. Connection ID. Range: 0–65535.
<att_handle>	Integer type. Description handle. Range: 1–65535.
<islong>	Integer type. Whether multiple data packets are required. 0 Multiple data packets are not required. 1 Multiple data packets are required.

Example

```
AT+QBTGATRDESC=0,19,0
OK
```

2.5. Description of BT AT Commands

2.5.1. General AT Commands

2.5.1.1. AT+QBTSCANMODE Set Scan Mode

This command sets scan mode.

AT+QBTSCANMODE Set Scan Mode	
Test Command AT+QBTSCANMODE=?	Response +QBTSCANMODE: (list of supported <scan_mode>s) OK
Read Command AT+QBTSCANMODE?	Response +QBTSCANMODE: <scan_mode> OK Or ERROR
Write Command AT+QBTSCANMODE=<scan_mode>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<scan_mode>	Integer type. Scan mode. 0 Non-discoverable and non-connectable 1 Non-connectable but discoverable 2 Non-discoverable but connectable 3 Discoverable and connectable
--------------------------	--

Example

```
AT+QBTSCANMODE=3
OK
```

2.5.2. Description of HFP AT Commands

2.5.2.1. AT+QBTHFPCONN Connect a Device

This command connects a device.

AT+QBTHFPCONN Connect a Device	
Test Command AT+QBTHFPCONN=?	Response +QBTHFPCONN: <address> OK
Write Command AT+QBTHFPCONN=<address>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<address> String type. Address of the device to be connected.

Example

```
AT+QBTHFPCONN="66cac9a26e38"
OK
```

2.5.2.2. AT+QBTHFPDISCONN Disconnect a Device

This command disconnects a device.

AT+QBTHFPDISCONN Disconnect a Device	
Test Command AT+QBTHFPDISCONN=?	Response +QBTHFPDISCONN: <address> OK
Write Command AT+QBTHFPDISCONN=<address>	Response OK

	Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<address> String type. Address of the connected device.

Example

```
AT+QBTHFPDISCONN="66cac9a26e38"
OK
```

2.5.2.3.AT+QBTHFPVOL Set the Volume

This command sets the volume.

AT+QBTHFPVOL Set the Volume

Test Command AT+QBTHFPVOL=?	Response +QBTHFPVOL: (range of supported <volume>s),<address> OK
Write Command AT+QBTHFPVOL=<volume>,<address>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration are not saved.

Parameter

<volume> Integer type. Volume. Range: 1–15.

<address> String type. Address of the connected device.

Example

```
AT+QBTHFPVOL=10,"66cac9a26e38"
OK
```

2.5.2.4. AT+QBTHFPCALL Control a Voice Call

This command controls a voice call.

AT+QBTHFPCALL Control a Voice Call	
Test Command AT+QBTHFPCALL=?	Response +QBTHFPCALL: (range of supported <op>s),<address> OK
Write Command AT+QBTHFPCALL=<op>,<address>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration are not saved.

Parameter

<op>	Integer type. Control a voice call. 0 Reject a call 1 Answer a call 2 Hang up after answering a call
<address>	String type. Address of the connected device.

Example

```
AT+QBTHFPCALL=0,"66cac9a26e38"
OK
```

2.5.2.5.AT+QBTHFPDIAL Dial a Number

This command dials a Number.

AT+QBTHFPDIAL Dial a Number	
Test Command AT+QBTHFPDIAL=?	Response +QBTHFPDIAL: (list of supported <type>s),<address>,<number> OK
Write Command AT+QBTHFPDIAL=<type>,<address>[,<number>]	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<type>	Integer type. Dialing type. Omit <number> when <type>=1 . 0 Dial 1 Redial
<address>	String type. Address of the connected device.
<number>	String type. Dialed number.

Example

```
AT+QBTHFPDIAL=0,"66cac9a26e38","13249166530"
OK
```

2.5.2.6.AT+QBTHFPVOLR Turn On/Off Voice Assistant

This command turns on or turns off voice assistant.

AT+QBTHFPVOLR Turn On/Off Voice Assistant	
Test Command AT+QBTHFPVOLR=?	Response +QBTHFPVOLR: (支持的<op>列表),<address> OK

Write Command AT+QBTHFPVOLR=<op>,<address>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<op>	Integer type. Turn off or turn on voice assistant. 0 Turn off voice assistant 1 Turn on voice assistant
<address>	String type. Address of the connected device.

Example

```
AT+QBTHFPVOLR=0,"66cac9a26e38"
OK
```

2.5.2.7.AT+QBTHFPAGDIAL Make a Call

This command makes a call on HFP AG side.

AT+QBTHFPAGDIAL Make a Call	
Test Command AT+QBTHFPAGDIAL=?	Response +QBTHFPAGDIAL: <number> OK
Write Command AT+QBTHFPAGDIAL=<number>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<number> String type. The dialed number.

Example

```
AT+QBTHFPAGDIAL="13034562345"
OK
```

2.5.3. Description of A2DP AVRCP AT Commands

2.5.3.1. AT+QBTA2DPDISCONN Disconnect a Device

This command disconnects a device.

AT+QBTA2DPDISCONN Disconnect a Device

Test Command AT+QBTA2DPDISCONN=?	Response +QBTA2DPDISCONN: <address> OK
Write Command AT+QBTA2DPDISCONN=<address>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<address> String type. Address of the connected device.

Example

```
AT+QBTA2DPDISCONN="66cac9a26e38"
OK
```

2.5.3.2. AT+QBTAVRCPVOL Set the Volume

This command sets the volume.

AT+QBTAVRCPVOL Set the Volume	
Test Command AT+QBTAVRCPVOL=?	Response +QBTAVRCPVOL: (list of supported <volume>s) OK
Read Command AT+QBTAVRCPVOL?	Response +QBTAVRCPVOL: <volume> OK Or ERROR
Write Command AT+QBTAVRCPVOL=<volume>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<volume> Integer type. Volume. Range: 0–127.

Example

```
AT+QBTAVRCPVOL=80
OK
```

2.5.3.3. AT+QBTAVRCPCTRL Control Audio Playback

This command controls audio playback.

AT+QBTAVRCPCTRL Control Audio Playback	
Test Command AT+QBTAVRCPCTRL=?	Response +QBTAVRCPCTRL: (list of supported <op>s)

	OK
Write Command AT+QBTAVRCPCTRL=<op>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<op>	Integer type. Control audio playback.
0	Stop playing
1	Start playing
2	Switch to the last audio
3	Switch to the next audio

Example

```
AT+QBTAVRCPCTRL=0
OK
```

2.5.3.4.AT+QBTAVRCPSTATE Get the Audio Playing State

This command gets the audio playing state.

AT+QBTAVRCPSTATE Get the Audio Playing State	
Test Command AT+QBTAVRCPSTATE=?	Response +QBTAVRCPSTATE: (list of supported <state>s) OK
Read Command AT+QBTAVRCPSTATE?	Response +QBTAVRCPSTATE: <state> OK Or ERROR
Maximum Response Time	10 s

Characteristics	/
-----------------	---

Parameter

<state>	Integer type. Audio playing state.
	0 Stop playing
	1 Playing
	2 Pause
	255 Error

Example

```

AT+QBTAVRCPSTATE=?
AT+QBTAVRCPSTATE: 1

OK
    
```

2.5.4. Description of SPP AT Commands

2.5.4.1. AT+QBTINQUIRY Search a Device

This command searches a device.

AT+QBTINQUIRY Search a Device	
Test Command AT+QBTINQUIRY=?	Response +QBTINQUIRY: <on_off> OK
Write Command AT+QBTINQUIRY=<on_off>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<on_off>	String type. Turn on/off the device search.
-----------------------	---

- 0 Turn off
- 1 Turn on

Example

```
AT+QBTINQUIRY=1
OK
```

2.5.4.2.AT+QBTSPPCONN Connect a Device

This command connects a device.

AT+QBTSPPCONN Connect a Device	
Test Command AT+QBTSPPCONN=?	Response +QBTSPPCONN: <address> OK
Write Command AT+QBTSPPCONN=<address>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<address> String type. Address of the device to be connected.

Example

```
AT+QBTSPPCONN="66cac9a26e38"
OK
```


2.5.4.3. AT+QBTSPDISCONN Disconnect a Device

This command disconnects a device.

AT+QBTSPDISCONN Disconnect a Device	
Execution Command AT+QBTSPDISCONN	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Example

```
AT+QBTSPDISCONN
OK
```

2.5.4.4. AT+QBTSPSENDDATA Send Data

This commands sends data.

AT+QBTSPSENDDATA Send Data	
Test Command AT+QBTSPSENDDATA=?	Response +QBTSPSENDDATA: <string> OK
Write Command AT+QBTSPSENDDATA=<string>	Response OK Or ERROR
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<string> String type. The data to be sent. Maximum length is 255 bytes.

Example

```
AT+QBTSPSENDATA="123456789000"
```

```
OK
```

3 Description of URCs

3.1. BLE Related URCs

3.1.1. +QBTGATSCON Connect GATT

+QBTGATSCON Connect GATT

+QBTGATSCON: <connID>,<address>

Parameter

<connID>	Integer type. Connection ID.
<address>	String type. Address of the connected device.

Example

+QBTGATSCON: 0,"69b4:67:55370a"

3.1.2. +QBTGATSDCON Disconnect GATT Connection

+QBTGATSDCON Disconnect GATT Connection

+QBTGATSDCON: <connID>,<address>

Parameter

<connID>	Integer type. Connection ID.
<address>	String type. Address of the disconnected device.

Example

+QBTGATSDCON: 0,"69b4:67:55370a"

3.1.3. +QBTGATMTU Report MTU During Connection

+QBTGATMTU Report MTU During Connection

+QBTGATMTU: <conn_handle>,<MTU>

Parameter

<conn_handle> Integer type. Connection handle.
 <MTU> Integer type. Maximum transmission unit. Unit: byte.

Example

+QBTGATMTU: 0,200

3.1.4. +QBTGATCONNP Connection Parameters Update

+QBTGATCONNP Connection Parameters Update

+QBTGATCONNP: <connID>,<min_interval>,<max_interval>,<latency>,<timeout> Connection parameters update

Parameter

<connID> Integer type. Connection ID.
 <min_interval> Integer type. Minimum interval. Time interval: 1.25 ms.
 <max_interval> Integer type. Maximum interval. Time interval: 1.25 ms.
 <latency> Integer type. Latency.
 <timeout> Integer type. Timeout period. Time interval: 10 ms.

Example

+QBTGATCONNP: 0,36,36,0,2000

3.1.5. +QBTGATDESCDATA GATT Server Report Characteristic Descriptor

+QBTGATDESCDATA GATT Server Report Characteristic Descriptor

+QBTGATDESCDATA: <connID>,<att_handle>,<value_length>,<value>

Parameter

<connID>	Integer type. Connection ID.
<att_handle>	Integer type. Attribute handle.
<length>	Integer type. Length of characteristic descriptor. Unit: byte.
<value>	String type. Received characteristic descriptor.

Example

+QBTGATDESCDATA: 0,19,2,"1234"

3.1.6. +QBTSCANDATAIND GATT Client Report Scanning Data

+QBTSCANDATAIND GATT Client Report Scanning Data

+QBTSCANDATAIND:<name>,<addr_type>,<address>,<RSSI>,<event_type>,<raw_data>

Parameter

<name>	String type without double quotation marks. Name of the scanned device.
<addr_type>	Integer type. Address type of the device 0 Public address 1 Random address
<address>	String type. Device address.
<RSSI>	Integer type. Received signal strength indication.
<event_type>	Integer type. The type of the scanned advertising. 0 Connectable and scannable undirected advertising 1 Connectable directional advertising 2 Scannable and undirected advertising 3 Non-connectable undirected advertising 4 Scan response data
<raw_data>	String type. The scanned raw advertising data.

Example

+QBTSCANDATAIND: FiiO LC-BT2,0,"58061a98ed40",67,0,"04ff04a5d20c094669694f204c432d425432"

3.1.7. +QBTSERVDATA GATT Client Report Service Data

+QBTSERVDATA GATT Client Report Service Data

+QBTSERVDATA: <serv_UUID><start_handle>,<end_handle>

Parameter

- <serv_UUID> Integer type. Current service UUID.
- <start_handle> Integer type. Start handle of the service.
- <end_handle> Integer type. End handle of the service.

Example

+QBTSERVDATA: 6159,16,65535

3.1.8. +QBTHARDATA GATT Client Report Characteristic Value

+QBTHARDATA GATT Client Report Characteristic Value

+QBTHARDATA: <chara_count>[,<chara_UUID1>,<chara_handle1>,<value_handle1>,<properties1>[,<chara_UUID2>,<chara_handle2>,<value_handle2>,<properties2>[,...]]]

Parameter

- <chara_count> Integer type. The number of characteristics. Every characteristic consists of <chara_handle>, <properties>, <value_handle> and <chara_UUID>.
- <chara_UUID> Integer type. UUID of the current characteristic.
- <chara_handle> Integer type. Handle of characteristic.
- <value_handle> Integer type. Handle of characteristics value.
- <properties> Integer type. Characteristics properties. Set Bit as 1 to enable the corresponding property. Range: 0–255 (in ORed).
 - 0 Disable all the properties
 - Bit0 Broadcast
 - Bit1 Read
 - Bit2 Write without confirmation of link layer
 - Bit3 Write
 - Bit4 Notify
 - Bit5 Indicate

Bit6	Authenticated signed Writes
Bit7	Extended properties

Example

+QBTCHARDATA: 2,10877,41,42,10,10868,45,46,58

3.1.9. +QBTDESCDATA GATT Client Report Characteristic Descriptor

+QBTDESCDATA GATT Client Report Characteristic Descriptor

+QBTDESCDATA: <des_count>[,<des_UUID>,<des_handle>[,<des_UUID>,<des_handle>[,...]]]

Parameter

<des_count>	Integer type. The number of characteristic descriptors. Every descriptor includes <des_handle> and <des_UUID> .
<des_UUID>	Integer type. UUID of the characteristic descriptor.
<des_handle>	Integer type. Handle of characteristic descriptor.

Example

+QBTDESCDATA: 2,10506,43,10497,44

3.1.10. +QBTATTERR GATT Client Report Attribute Error

+QBTATTERR GATT Client Report Attribute Error

+QBTATTERR: <att_err>

Parameter

<att_err>	Integer type. Attribute error types.
0x01	Invalid handle value
0x02	No read permission
0x03	No write permission
0x04	Invalid PDU
0x05	Invalid authentication
0x06	Unsupported request
0x07	Invalid offset

0x08	Invalid authority
0x09	Full parameter queue
0x0A	Attribute not found
0x0B	Attributes with inconsistent lengths
0x0C	Invalid EK size
0x0D	Invalid attribute value length
0x0E	Invalid error
0x0F	Invalid encrypted data
0x10	Unsupported group type
0x11	Invalid resource
0x80	Application error

Example

+QBTATTERR: 0x0A

3.1.11. +QBTGATNOD GATT Client Report Receiving Notification

+QBTGATNOD GATT Client Report Receiving Notification

+QBTGATNOD: <att_handle>,<value_length>,<value>

Parameter

<att_handle>	Integer type. The attribute handle of the received data.
<length>	Integer type. Length of the received data. Unit: byte.
<value>	String type. Received data.

Example

+QBTGATNOD: 24,6,"180000110011"

3.1.12. +QBTGATIND GATT Client Reports Receiving Indication

+QBTGATIND GATT Client Reports Receiving Indication

+QBTGATIND: <att_handle>,<lenth>,<value>

Parameter

<att_handle>	Integer type. The attribute handle of the received data.
<length>	Integer type. Length of the received data. Unit: byte.
<value>	String type. Received data.

Example

+QBTGATIND: 24,6,"180011111111"

3.1.13. +QBTGATWRCHAR GATT Client Report the State of Writing Characteristic

Value

+QBTGATWRCHAR GATT Client Report the State of Writing Characteristic Value

+QBTGATWRCHAR: <status>

Parameter

< status>	String type without double quotation marks. OK FAILED
-----------	---

Example

+QBTGATWRCHAR: OK

3.1.14. +QBTGATWRCHARNORSP GATT Client Report the State of Writing

Characteristic Value Without Response

+QBTGATWRCHARNORSP GATT Client Report the State of Writing Characteristic Value Without Response

+QBTGATWRCHARNORSP: <status>

Parameter

<status>	String type without double quotation marks. OK FAILED
-----------------------	---

Example

```
+QBTGATWRCHARNORSP: OK
```

3.1.15. +QBTGATRDCHAR GATT Client Report Reading Characteristic Value by Handle

+QBTGATRDCHAR GATT Client Report Reading Characteristic Value by Handle

```
+QBTGATRDCHAR: <value_length>,<value>
```

Parameter

<value_length>	Integer type. Length of received data. Unit: byte.
<value>	String type. Received data.

Example

```
+QBTGATRDCHAR: 6,"180011111111"
```

3.1.16. +QBTGATRDCHARUUID GATT Client Report Reading Characteristic Value by UUID

+QBTGATRDCHARUUID GATT Client Report Reading Characteristic Value by UUID

```
+QBTGATRDCHARUUID: <handle>,<value_length>,<value>
```

Parameter

<handle>	Integer type. Received data handle.
<length>	Integer type. Length of received data. Unit: byte.
<value>	String type. Received data.

Example

```
+QBTGATRDCHARUUID: 18,6,"180011111111"
```

3.1.17. +QBTGATWRDESC GATT Client Report the State of Writing Characteristic Descriptor**+QBTGATWRDESC GATT Client Report the State of Writing Characteristic Descriptor**

```
+QBTGATWRDESC: <status>
```

Parameter

<status>	String type without double quotation marks. OK FAILED
-----------------------	---

Example

```
+QBTGATWRDESC: OK
```

3.1.18. +QBTGATRDESC GATT Client Report Reading Characteristic Descriptor**+QBTGATRDESC GATT Client Report Reading Characteristic descriptor**

```
+QBTGATRDESC: <value_length>,<value>
```

Parameter

<value_length>	Integer type. Length of received data. Unit: byte.
<value>	String type. Received data.

Example

```
+QBTGATRDESC: 6,"180011111111"
```

3.1.19. +QBTGATRDDATAIND GATT Server Report GATT Client Reading Data

+QBTGATRDDATAIND GATT Server Report GATT Client Reading Data

+QBTGATRDDATAIND: <connID>,<att_handle>,<length>,<value>

Parameter

<connID>	Integer type. Connection ID.
<att_handle>	Integer type. Handle value for the operation.
<length>	Integer type. Length of the received data. Unit: byte.
<value>	String type. Received data.

Example

+QBTGATRDDATAIND: 0,18,4,00000000

3.1.20. +QBTLESTATE GATT Server Report Connection State Update

+QBTLESTATE GATT Server Report Connection State Update

+QBTLESTATE: <cid>,<connID>,<address>,<conn_state>,<att_handle>

Parameter

<cid>	Integer type. Channel ID.
<connID>	Integer type. Connection ID.
<address>	String type. Address of the connected device.
<conn_state>	Integer type. Connection state. 0 Disconnected 1 Connected
<att_handle>	Integer type. Attribute handle.

Example

+QBTLESTATE: 0,0,"112233da8048",1,3

+QBTLESTATE: 1,0,"112233da8048",1,4

3.1.21. +QBTLEVALDATA GATT Server Report Receiving Data

+QBTLEVALDATA GATT Server Report Receiving Data

+QBTLEVALDATA: <cid>,<address>,<value_length>,<value>

Parameter

<cid>	Integer type. Channel ID.
<address>	String type. Address of the connected device.
<value_length>	Integer type. Received data length. Unit: byte.
<value>	Hexadecimal string type. Received data.

Example

+QBTLEVALDATA: 0,"bd30d6c64bc9",2,"1234"

3.1.22. +QBTLEVALDATI GATT Server Report Receiving Buffer Data

+QBTLEVALDATI GATT Server Report Receiving Buffer Data

+QBTLEVALDATI: <cid>,<address>,<value_length>

Parameter

<cid>	Integer type. Channel ID.
<address>	String type. Address of the connected device.
<value_length>	Integer type. Received data length. Unit: byte.

Example

+QBTLEVALDATI: 0,"3af3f58716f9",3

3.2. BT HFP Related URCs

3.2.1. +QBTHFPSCON Connect HFP

+QBTHFPSCON Connect HFP

+QBTHFPSCON: <address>

Parameter

<address> String type. Address of the connected device.

Example

+QBTHFPSCON: "66cac9a26e38"

3.2.2. +QBTHFPSDCON Disconnect HFP Connection

+QBTHFPSDCON Disconnect HFP Connection

+QBTHFPSDCON: <address>

Parameter

<address> String type. Address of the disconnected device.

Example

+QBTHFPSDCON: "66cac9a26e38"

3.2.3. +QBTHFPCALL Change of Call State

+QBTHFPCALL Change of Call State

+QBTHFPCALL: <call_state>,<address>

Parameter

<call_state>	Integer type. Call state. 0 No ongoing call 1 A call is ongoing
<address>	String type. Address of the disconnected device.

Example

```
+QBTHFPCALL: 0,"66cac9a26e38"
```

3.2.4. +QBTHFPCALS Change of Call Setting State

+QBTHFPCALS Change of Call Setting State

```
+QBTHFPCALS: <call_setting_state>,<address>
```

Parameter

<state>	Integer type. Call setting state. 0 No call settings are currently available. 1 Far-end is calling in. 2 Calling out 3 Local device is calling out and the far-end device is ringing.
<address>	String type. Address of the disconnected device.

Example

```
+QBTHFPCALS: 0,"66cac9a26e38"
```

3.2.5. +QBTHFPNET Change of Network State

+QBTHFPNET Change of Internet State

```
+QBTHFPNET: <network_state>,<address>
```

Parameter

<state>	Integer type. Network state. 0 Current network is unavailable. 1 Current network is available.
----------------------	--

<address> String type. Address of the disconnected device.

Example

```
+QBTHFPNET: 0,"66cac9a26e38"
```

3.2.6. +QBTHFPNETS Change of Network Signal Strength

+QBTHFPNETS Change of Network Signal Strength

```
+QBTHFPNETS: <signal_strength>,<address>
```

Parameter

<signal_strength> Integer type. Signal Strength. Range: 0–255.

<address> String type. Address of the disconnected device.

Example

```
+QBTHFPNETS: 0,"66cac9a26e38"
```

3.2.7. +QBTHFPBAT Change of Battery Level

+QBTHFPBAT Change of Battery Level

```
+QBTHFPBAT: <level>,<address>
```

Parameter

<level> Integer type. Battery level. Range: 0–255.

<address> String type. Address of the disconnected device.

Example

```
+QBTHFPBAT: 0,"66cac9a26e38"
```


3.2.8. +QBTHFPCALH Change of Call Holding State

+QBTHFPCALH Change of Call Holding State

+QBTHFPCALH: <call_hold_state>,<address>

Parameter

<state> Integer type. Call holding state.
 0 No call is holding.
 1 Activate a call and hold a call.
 2 Call is holding.
 <address> String type. Address of the disconnected device.

Example

+QBTHFPCALH: 0,"66cac9a26e38"

3.2.9. +QBTHFPAUD Change of Voice State

+QBTHFPAUD Change of Voice State

+QBTHFPAUD: <state>,<address>

参数

<state> Integer type. Voice state.
 0 Disconnected
 1 Connecting
 2 Connected
 3 Disconnecting
 <address> String type. Address of the disconnected device.

Example

+QBTHFPAUD: 0,"66cac9a26e38"

3.2.10. +QBTHFPVOL Change of Voice Type

+QBTHFPVOL Change of Voice Type

+QBTHFPVOL: <voice_type>,<volume>

Parameter

<voice_type>	Integer type. Voice type. 0 Speaker 1 Microphone
<volume>	Integer type. Volume. Range: 1–255.

Example

+QBTHFPVOL: 0,5

3.2.11. +QBTHFPNETT Change of Network Type

+QBTHFPNETT Change of Network Type

+QBTHFPNETT: <network_type>,<address>

Parameter

<type>	Integer type. Network type. 0 HPLMN 1 Roaming
<address>	String type. Address of the disconnected device.

Example

+QBTHFPNETT: 0,"66cac9a26e38"

3.2.12. +QBTHFPRING Change of Ring Indication

+QBTHFPRING Change of Ring Indication

+QBTHFPRING: <address>

Parameter

<address> String type. Address of the disconnected device.

Example

+QBTHFPRING: "66cac9a26e38"

3.2.13. +QBTHFPCOD Change of Code Type

+QBTHFPCOD Change of Code Type

+QBTHFPCOD: <code_type>,<address>

Parameter

<code_type> Integer type. Code type.
 0 CVSD
 1 mSBC
<address> String type. Address of the disconnected device.

Example

+QBTHFPCOD: 0,"66cac9a26e38"

3.3. BT A2DP AVRCP Related URCs

3.3.1. +QBTA2DPSCON Connect A2DP

+QBTA2DPSCON Connect A2DP

+QBTA2DPSCON: <state>,<address>

Parameter

<state> Integer type.
 0 Disconnected
 1 Connecting

	2	Connected
	3	Disconnecting
<address>	String type. Address of the connected device.	

Example

```
+QBTA2DPSCON: 2,"66cac9a26e38"
```

3.3.2. +QBTA2DPSCON Disconnect A2DP Connection

+QBTA2DPSCON Disconnect A2DP Connection

```
+QBTA2DPSCON: <state>,<address>
```

Parameter

<state>	Integer type.
	0 Disconnected
	1 Connecting
	2 Connected
	3 Disconnecting
<address>	String type. Address of the disconnected device.

Example

```
+QBTA2DPSCON: 0,"66cac9a26e38"
```

3.3.3. +QBTA2DPAUDIOCFG Configure A2DP Audio

+QBTA2DPAUDIOCFG Configure A2DP Audio

```
+QBTA2DPAUDIOCFG: <address>
```

Parameter

<address>	String type. Address of the connected device.
-----------	---

Example

```
+QBTA2DPAUDIOCFG: "66cac9a26e38"
```

3.3.4. +QBTA2DPAUDIOSTART Start Playing Audio

+QBTA2DPAUDIOSTART Start Playing Audio

+QBTA2DPAUDIOSTART: <state>,<address>

Parameter

<state>	Integer type. Audio playing state. 0 Stop playing 1 Pause 2 Start playing
<address>	String type. Address of the connected device.

Example

+QBTA2DPAUDIOSTART: 2,"66cac9a26e38"

3.3.5. +QBTA2DPAUDIOSTOPPED Stop Playing Audio

+QBTA2DPAUDIOSTOPPED Stop Playing Audio

+QBTA2DPAUDIOSTOPPED: <state>,<address>

Parameter

<state>	Integer type. Audio playing state. 0 Stop playing 1 Pause 2 Start playing
<address>	String type. Address of the connected device.

Example

+QBTA2DPAUDIOSTOPPED: 1,"66cac9a26e38"

3.3.6. +QBTAVRCPSCON Connect AVRCP

+QBTAVRCPSCON Connect AVRCP

+QBTAVRCPSCON: <state>,<address>

Parameter

<state>	Integer type. Connection state. 0 Disconnected 1 Connecting 2 Connected 3 Disconnecting
<address>	String type. Address of the connected device.

Example

+QBTAVRCPSCON: 2,"66cac9a26e38"

3.3.7. +QBTAVRCPSDCON Disconnect AVRCP Connection

+QBTAVRCPSDCON Disconnect AVRCP Connection

+QBTAVRCPSDCON: <state>,<address>

Parameter

<state>	Integer type. Connection state. 0 Disconnected 1 Connecting 2 Connected 3 Disconnecting
<address>	String type. Address of the connected device.

Example

+QBTAVRCPSDCON: 0,"66cac9a26e38"

3.3.8. +QBTAVRCPVOLCHANGE Change of AVRCP Volume

+QBTAVRCPVOLCHANGE Change of AVRCP Volume

+QBTAVRCPVOLCHANGE: <volume>,<address>

Parameter

<volume>	Integer type. Volume. Range: 0–127.
<address>	String type. Address of the connected device.

Example

+QBTAVRCPVOLCHANGE: 80,"66cac9a26e38"

3.4. BT SPP Related URCs

3.4.1. +QBTSPPSCON Connect SPP

+QBTSPPSCON Connect SPP

+QBTSPPSCON: <address>

Parameter

<address>	String type. Address of the connected device.
-----------	---

Example

+QBTSPPSCON: "66cac9a26e38"

3.4.2. +QBTSPPSDCON Disconnect SPP Connection

+QBTSPPSDCON Disconnect SPP Connection

+QBTSPPSDCON: <address>

Parameter

<address> String type. Address of the disconnected device.

Example

+QBTSPPSDCON: "66cac9a26e38"

3.4.3. +QBTSPPREVDATA Report Received Data

+QBTSPPREVDARA Report Received Data

+QBTSPPREVDATA: <data>

Parameter

<data> String type. Received data.

Example

+QBTSPPREVDATA: "1234abcd"

3.4.4. +QBTINQUIRY Report Device Search

+QBTINQUIRY Report Device Search

+QBTINQUIRY: <address>,<rssi>,<device_name>

Parameter

<address> Device address.
 <rssi> Signal value.
 <device_name> Device name.

Example

+QBTINQUIRY: "66cac9a26e38",50,8910

3.4.5. +QBTINQSTATE Report Device Search Status

+QBTINQSTATE Report Device Search Status

+QBTINQSTATE: <state>

Parameter

<state>	Device search status report
0	Search completion

Example

+QBTINQSTATE: 0

3.4.6. +QBTBOND Report Binding Information

+QBTBOND Report Binding Information

+QBTBOND: <address>,<device_name>,<name_len>

Parameter

<address>	Device address.
<device_name>	Device name.
<name_len>	The length of the device name

Example

+QBTBOND: "66cac9a26e38",8910,4

3.4.7. +QBTBONDSTATE Report Binding Status

+QBTBONDSTATE Report Binding Status

+QBTBONDSTATE: <address>,<state>

Parameter

<address>	Device address.
<state>	The binding status.
	0 Not Bound
	1 Binging
	2 The binding is successful

Example

```
+QBTBONDSTATE: "66cac9a26e38",2
```

4 Examples

4.1. BLE Communication

The AT commands provided in this document support basic BLE operation, including scanning, advertising, and connection. Quectel ECx00U series and EG915U series modules support communication with other BLE devices.

The following describes the BLE GATT SERVER process when the module serves as a BLE server.

AT+QBTPWR=1

//Turn BLE GATT server on.

OK

AT+QBTGATADV=1,128,160,0,0,7,0

//Set advertising parameters of all address. Minimum advertising interval is 80 ms; maximum advertising interval is 100 ms; advertising type is Connectable undirected advertising; the address type of local device is public address; advertising message is transmitted on channel 37, 38 and 39; process scan and connection requests from all devices.

OK

AT+QBTADVDATA=9,"020106050938393130"

//Set 9-byte advertising data.

OK

AT+QBTADVRSPDATA=6,"050938393130"

//Set 6-byte scan response data. This step can be omitted according to actual needs.

OK

AT+QBTGATSS=0,1,6159,1

//Add the first primary service whose ID is 0 and UUID is 6159 in 16-bit.

OK

AT+QBTGATSC=0,0,58,1,10777

//Add characteristics in the first service. The characteristic ID is 0; characteristic properties are Read, Write, Notify and Indicate. UUID is 10777 in 16-bit.

OK

AT+QBTGATSCV=0,0,3,1,10777,244,"1234"

//Configure characteristic value: Service ID is 0; characteristic is 0; characteristic properties are Read and Write; UUID is 10777 in 16-bit; the length of characteristic value is 244 bytes; characteristic value is 1234.

OK

AT+QBTGATSCD=0,0,3,1,10498,2,"0300"

//Add 2-byte descriptor with value 0300 in characteristic whose service ID is 0, characteristic ID is 0, characteristic properties are Read and Write, UUID is 10498 in 16-bit.

OK

AT+QBTGATSS=1,0,"f5899b5f8000008000100000FE180000",1

//Add the second secondary service whose ID is 1 and UUID is f5899b5f8000008000100000FE180000 in 128-bit.

OK

AT+QBTGATSC=1,0,58,0,"f5899b5f8000008000100000FEFF1111"

//Add the first characteristic in the second service whose ID is 1. The characteristic ID is 0; characteristic properties are Read, Write, Notify and Indicate. UUID is f5899b5f8000008000100000FEFF1111 in 128-bit.

OK

AT+QBTGATSCV=1,0,3,0,"f5899b5f8000008000100000FEFF1111",244,"1234"

//Configure characteristic value: Service ID is 1; characteristic is 0; characteristic properties are Read and Write; UUID is f5899b5f8000008000100000FEFF1111 in 128-bit; the length of characteristic value is 244 bytes; characteristic value is 1234.

OK

AT+QBTGATSCD=1,0,3,1,10498,2,"0300"

//Add 2-byte descriptor with value 0300 in characteristic whose service ID is 1, characteristic ID is 0, characteristic properties are Read and Write, UUID is 10498 in 16-bit.

OK

AT+QBTGATSC=1,1,16,0,"f5899b5f8000008000100000FDFF1111"

//Add the second characteristic in the second service whose ID is 1. The characteristic ID is 1; characteristic property is Notify. UUID is f5899b5f8000008000100000FDFF1111 in 128-bit.

OK

AT+QBTGATSCV=1,1,3,0,"f5899b5f8000008000100000FDFF1111",244,"1234"

//Configure characteristic value: Service ID is 1; characteristic is 1; characteristic properties are Read and Write; UUID is f5899b5f8000008000100000FDFF1111 in 128-bit; the length of characteristic value is 244 bytes; characteristic value is 1234.

OK

AT+QBTGATSCD=1,1,3,1,10498,2,"0300"

//Add 2-byte descriptor with value 0300 in characteristic whose service ID is 1, characteristic ID is 1, characteristic properties are Read and Write, UUID is 10498 in 16-bit.

OK

AT+QBTGATSSC=1,1

//Finish adding service, and reserve the default GAP and GATT services.

OK

AT+QBTADV=1

//Start advertising, waiting for the BLE GATT client to connect.

OK

+QBTGATSCON: 0,"3af3f58716f9"

//Connect and get the connection ID.

+QBTLESTATE: 0,0,"3af3f58716f9",1,18

//Connection state updates.

+QBTLESTATE: 1,0,"3af3f58716f9",1,22

+QBTLESTATE: 2,0,"3af3f58716f9",1,25

+QBTGATMTU: 0,200

+QBTGATMTU: 0,247

//Get latest MTU.

+QBTLEVALDATA: 0,"3af3f58716f9",1,"12"

//Receiving data sent from the client.

+QBTLEVALDATA: 0,"3af3f58716f9",1,"12"

+QBTGATDESCDATA: 0,19,2,"1234"

//The client rewrites characteristic descriptor.

+QBTGATRDDATAIND: 0,18,1,"12"

//The client reads characteristic value

+QBTGATRDDATAIND: 0,19,2,"1234"

//The client reads descriptor.

AT+QBTGATSNOD=0,0,18,4,"00110011"

//Send 4-byte notification in direct mode.

OK

AT+QBTGATSIND=0,0,18,4,"11111111"

//Send 4-byte indication in direct mode.

OK

AT+QBTLESTATE?

//Query connection state.

+QBTLESTATE: 0,0,"3af3f58716f9",1,18

+QBTLESTATE: 1,0,"3af3f58716f9",1,22

+QBTLESTATE: 2,0,"3af3f58716f9",1,25

OK

AT+QBTLERCVM=1,10000

//Set the server receiving mode is buffer mode and the time to receive buffer data from the channel and report URC is 10000 ms.

OK

```

+QBTLEVALDATI: 0,"3af3f58716f9",3
//Receive data sent by the client.

+QBTLEVALDATI: 0,"3af3f58716f9",3

+QBTLEVALDATI: 0,"3af3f58716f9",3
AT+QBTLEREAD=0,3
//The server reads 3-byte buffer data from channel 0.
+QBTLEREAD: 0,3,"125678"

OK
AT+QBTLEEXMTU=0,220
//Exchange 220-byte MTU.
OK

+QBTGATMTU: 0,220
AT+QBTGATCONNP=0,39,39,0,2000
//Update connection parameters of connection 0: minimum interval is 48.75 ms; maximum interval is
48.75 ms; latency is 0 and timeout is 20 s.
OK

+QBTGATCONNP: 0,39,39,0,2000

+QBTGATSDCON: 0,"3af3f58716f9"
//Disconnect the client.

+QBTLESTATE: 0,0,"3af3f58716f9",0,18

+QBTLESTATE: 1,0,"3af3f58716f9",0,22

+QBTLESTATE: 2,0,"3af3f58716f9",0,25
    
```

The following describes the BLE GATT CLIENT process when the module serves as a BLE server.

```

AT+QBTPWR=2
//Turn on BLE GATT client.
OK
AT+QBTSCANPARA=0,96,48,0,1
//Set passive scan, scan interval is 60 ms. scan window is 30 ms; accept sll advertising packets except
the directed advertising packets; the address type of local device is public address.
OK
AT+QBTGATSCAN=1
//Start scanning and determine whether to connect according to the name of the scanned device.
OK
    
```

```

+QBTSCANDATAIND: FiiO LC-BT2,0,"58061a98ed40",67,0,"04ff04a5d20c094669694f204c432d42
5432"

+QBTSCANDATAIND: ,1,"314610f3534c",79,0,"02011a020a0c0bff4c0010061e1aef5603ee"

+QBTSCANDATAIND: ,1,"43876796aa4e",81,3,"1eff060001092002b577191ae1058dda892fba49a1d
700a7e4e1b0dacfa3ee"

+QBTSCANDATAIND: ,1,"97b5ddcfba5e",96,3,"1eff060001092002352498fc9768d50b1deef475477d
67e04cf5b063fd7cc5"

+QBTSCANDATAIND: ,1,"67e905ab5179",88,3,"1eff06000109200224d446268c0bf082f8790559b005
31a2de579d9ab7c982"

+QBTSCANDATAIND: 8910,0,"112233da8045",19,0,"020106050938393130"

+QBTSCANDATAIND: ,1,"f3113809e578",75,3,"1eff060001092002d3120f8f0cdb439c81b623299ae5
4590558b53c3a8751e"

+QBTSCANDATAIND: ,1,"868bd73e0e5d",85,3,"1eff060001092002c12c38b446a8e62911f6e70d779
c8c2af32af010409a00"

+QBTSCANDATAIND: ,0,"d3fe7c433968",86,3,""
AT+QBTGATSCAN=0
//Stop scanning.
OK
AT+QBTGATCONN=1,0,"112233da8045"
//Initiate a connection with the server.
OK

+QBTGATSCON: 0,"112233da8045"
//Connect successfully.

+QBTGATMTU: 0,200

+QBTGATMTU: 0,247
//Get latest MTU.
AT+QBTGATSERV=0,0
//Scan all services.
OK

+QBTSERVDATA: 6144,1,11

+QBTSERVDATA: 6145,12,15
    
```

+QBTSERVDATA: 6159,16,19

+QBTSERVDATA: 6398,20,65535

AT+QBTGATCHAR=0,16,19

//Scan characteristics with start handle 16 and end handle 19.

OK

+QBTCHARDATA: 1,10777,17,18,58

+QBTATTERR: 0x0A

AT+QBTGATDESC=0,19,19

//Scan characteristic descriptors with start handle 19 and end handle 19.

OK

+QBTDESCDATA: 1,10498,19

AT+QBTGATCHAR=0,20,65535

//Scan characteristics with start handle 20 and end handle 65535.

OK

+QBTCHARDATA: 2,f5899b5f8000008000100000feff1111,21,22,58,f5899b5f8000008000100000dff1111,24,25,16

+QBTATTERR: 0x0A

AT+QBTGATDESC=0,22,65535

//Scan characteristic descriptors with start handle 22 and end handle 65535.

OK

+QBTDESCDATA: 0

AT+QBTGATDESC=0,23,65535

//Scan characteristic descriptors with start handle 23 and end handle 65535.

OK

+QBTDESCDATA: 3,10498,23,10243,24,10498,26

AT+QBTWRCHAR=0,18,1,"12",0,0

//Write 1-byte characteristic value 12 to characteristic whose handle is 18 and do not require multiple data packets, offset address is 0.

OK

+QBTGATWRCHAR: OK

AT+QBTGATWRDESC=0,19,2,"1234"

//Write 2-byte characteristic descriptor 1234 to characteristic whose handle is 19.

OK

4.2. BT HFP Use Process

The following describes the process when the module serves as BT HFP device.

```

AT+QBTPWR=4 //Turn BT HFP Protocol on.
OK
AT+QBTNAME="quec_headset" //Set BT device name as "quec_headset".
OK
AT+QBTSCANMODE=3 //Set to be searchable and connectable, waiting for the
mobile phone to connect.
OK
+QBTHFPCALL: 0,"9a697d241368"
+QBTHFPCALS: 0,"9a697d241368"
+QBTHFPPNET: 1,"9a697d241368"
+QBTHFPPNETS: 5,"9a697d241368"
+QBTHFPPBAT: 2,"9a697d241368"
+QBTHFPPCALH: 0,"9a697d241368"
+QBTHFPPSCON: "9a697d241368" //Connect with a mobile phone successfully.
+QBTHFPPCALS: 1,"9a697d241368"
+QBTHFPPAUD: 2,"9a697d241368"
+QBTHFPPCALS: 1,"9a697d241368"
+QBTHFPPRING: "9a697d241368" //The mobile phone rings.
+QBTHFPPRING: "9a697d241368"
AT+QBTHFPCALL=1,"9a697d241368"
OK
+QBTHFPCALL: 1,"9a697d241368" //Answer the call.
+QBTHFPCALS: 0,"9a697d241368"
AT+QBTHFPPVOL=10,"9a697d241368" //Set the volume of mobile phone to 10.
OK
    
```

```

+QBTHFPAUD: 0,"9a697d241368"

+QBTHFPCALL: 0,"9a697d241368"
AT+QBTHFPDISCONN="9a697d241368" //The mobile phone disconnects with the module.
OK

+QBTHFPSDCON: "9a697d241368"
    
```

4.3. BT A2DP AVRCP Use Process

The following describes the process when the module serves as BT A2DP AVRCP device.

```

AT+QBTPWR=5 //Turn BT A2DP and AVRCP Protocol on.
OK
AT+QBTNAME="quec_headset" //Set BT device name as "quec_headset".
OK
AT+QBTSCANMODE=3 //Set scan mode to be searchable and connectable, waiting
for the mobile phone to connect.
OK

+QBTAVRCPSCON: 2,"9a697d241368"

+QBTA2DPAUDIOCFG:"9a697d241368"

+QBTA2DPSCON: 2,"9a697d241368"

+QBTAVRCPVOLCHANGE: 38,"9a697d241368"

+QBTAVRCPVOLCHANGE: 127,"9a697d241368"

+QBTAVRCPVOLCHANGE: 38,"9a697d241368"

+QBTA2DPAUDIOSTART: 2,"9a697d241368"

+QBTA2DPAUDIOSTOPPED: 1,"9a697d241368"

+QBTAVRCPSCON: 0,"9a697d241368"

+QBTA2DPSCON: 0,"9a697d241368"
    
```

5 Appendix Reference

Table 3: Terms and Abbreviations

Abbreviation	Description
A2DP	Advanced Audio Distribution Profile
AG	Audio Gateway
AVRCP	Audio/Video Remote Control Profile
BLE	Bluetooth Low Energy
BT	Bluetooth
CVSD	Continuous Variable Slope Delta Modulation
GAP	Generic Access Profile
GATT	Generic Attribute Profile
GBK	Chinese Internal Code Specification
HF	Hands Free
HFP	Hands-free Profile
ID	Mostly refers to Identifier in terms of software
mSBC	Modified Sub-Band Coding
MTU	Maximum Transmission Unit
NVM	Non-Volatile Memory
PDU	Protocol Data Unit
RSSI	Received Signal Strength Indication
SPP	Serial Port Profile
TA	Terminal Adapter

URC	Unsolicited Result Code
UTF	Unicode Transformation Format
UUID	Universally Unique Identifier
