

# ECx00U&EGx00U&EG915U Series BT Application Note

**LTE Standard Module Series**

Version: 1.0.0

Date: 2021-08-25

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](mailto: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](mailto: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.

## 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. 2021. 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	2021-08-25	Ryan YI/ Chaos HUANG	Preliminary

## Contents

<b>About the Document</b> .....	<b>3</b>
<b>Contents</b> .....	<b>4</b>
<b>Table Index</b> .....	<b>7</b>
<b>1 Introduction</b> .....	<b>8</b>
1.1. Applicable Modules.....	8
1.2. Special Symbols.....	9
<b>2 Description of BT AT Commands</b> .....	<b>10</b>
2.1. AT Command Introduction .....	10
2.1.1. Definitions.....	10
2.1.2. AT Command Syntax.....	10
2.2. Declaration of AT Command Examples .....	11
2.3. General BT AT Commands.....	11
2.3.1. AT+QBTPWR Turn On/Off BT .....	11
2.3.2. AT+ QBNAME Set Bluetooth Device Name .....	12
2.4. Description of BLE AT Commands .....	13
2.4.1. General BLE AT Commands .....	13
2.4.2. AT Commands of BLE GATT Service .....	19
2.5. Description of GATT Client AT Commands.....	38
2.5.1. AT+QBTSCANPARA Set Scan Parameters.....	38
2.5.2. AT+QBTGATSCAN Start/Stop Scanning Device .....	39
2.5.3. AT+QBTGATCONN Connect a Device.....	40
2.5.4. AT+QBTGATSERV Scan Service.....	41
2.5.5. AT+QBTGATINC Scan Include.....	42
2.5.6. AT+QBTGATCHAR Scan Characteristic .....	42
2.5.7. AT+QBTGATDESC Scan Characteristic Descriptor .....	43
2.5.8. AT+QBTWRCHAR Write Characteristic Value .....	44
2.5.9. AT+QBTWRCHARNORSP Write Characteristic Value Without Response .....	45
2.5.10. AT+QBTRDCHARUUID Read Characteristic Value by UUID .....	46
2.5.11. AT+QBTRDCHARHAND Read Characteristic Value by Handle .....	47
2.5.12. AT+QBTGATWRDESC Write Characteristic Descriptor.....	47
2.5.13. AT+QBTGATRDESC Read Characteristic descriptor.....	48
2.6. Description of General BT AT Commands .....	49
2.6.1. AT+QBTSCANMODE Set Scan Mode.....	49
2.7. Description of BT HFP AT Commands.....	50
2.7.1. AT+QBTHFPCONN Connect a Device.....	50
2.7.2. AT+QBTHFPDISCONN Disconnect a Device .....	51
2.7.3. AT+QBTHFPVOI Set the Volume .....	51
2.7.4. AT+QBTHFPCALL Control Voice Call .....	52
2.7.5. AT+QBTHFPDIAL* Dial.....	53
2.7.6. AT+QBTHFPVOLR* Turn On/Off Voice Assistant.....	54

2.7.7.	AT+QBTHFPTRC* Three-way Calling .....	54
2.8.	Description of BT A2DP AVRCP AT Commands .....	55
2.8.1.	AT+QBTAVRCPDISCONN Disconnect a Device .....	55
2.8.2.	AT+QBTAVRCPVOL Set the Volume .....	56
2.8.3.	AT+QBTAVRCPCTRL Control Audio Playback.....	57
2.8.4.	AT+QBTAVRCPSTATE Get the Audio Playing State.....	57
<b>3</b>	<b>Description of URCs .....</b>	<b>59</b>
3.1.	BLE Related URCs .....	59
3.1.1.	+QBTGATSCON GATT Connection .....	59
3.1.2.	+QBTGATSDCON GATT Disconnection .....	59
3.1.3.	+QBTGATMTU MTU During Connection.....	60
3.1.4.	+QBTGATCONNP Connection Parameters Update.....	60
3.1.5.	+QBTGATDESCDATA GATT Client Report Characteristic Descriptor .....	60
3.1.6.	+QBTSCANDATAIND GATT Client Report Scanning Data .....	61
3.1.7.	+QBTSERVDATA GATT Client Report Service Data.....	62
3.1.8.	+QBTCHARDATA GATT Client Report Characteristic Value.....	62
3.1.9.	+QBTDESCDATA GATT Client Report Characteristic Descriptor Value .....	63
3.1.10.	+QBTATTERR GATT Client Report Attribute Error .....	63
3.1.11.	+QBTGATNOD GATT Client Report Receiving Notification.....	64
3.1.12.	+QBTGATIND GATT Client Reports Receiving Indication .....	64
3.1.13.	+QBTGATWRCHAR GATT Client Report the State of Writing Characteristic Value .....	65
3.1.14.	+QBTGATWRCHARNORSP GATT Client Report the State of Writing Characteristic Value Without Response .....	65
3.1.15.	+QBTGATRDCHAR GATT Client Report Reading Characteristic Value by Handle..	66
3.1.16.	+QBTGATRDCHARUUID GATT Client Report Reading Characteristic Value by UUID	66
3.1.17.	+QBTGATWRDESC GATT Client Report the State of Writing Characteristic Descriptor.....	67
3.1.18.	+QBTGATRDDESC GATT Client Report Reading Characteristic Descriptor .....	67
3.1.19.	+QBTGATRDDATAIND GATT Client Read Data .....	67
3.1.20.	+QBTLESTATE GATT Server Report Connection State Update.....	68
3.1.21.	+QBTLEVALDATA GATT Server Report Receiving Data .....	68
3.1.22.	+QBTLEVALDATI GATT Server Report Receiving Buffer Data.....	69
3.2.	BT HFP Related URCs .....	69
3.2.1.	+QBTHFPSCON HFP Connection.....	69
3.2.2.	+QBTHFPSDCON HFP Disconnection.....	70
3.2.3.	+QBTHFPCALL Change of Call State .....	70
3.2.4.	+QBTHFPCALS Change of Call Setting State.....	70
3.2.5.	+QBTHFPNET Change of Network State .....	71
3.2.6.	+QBTHFPNETS Change of Network Signal Strength .....	71
3.2.7.	+QBTHFPBAT Change of Battery Level.....	72
3.2.8.	+QBTHFPCALH Change of Call Holding State .....	72
3.2.9.	+QBTHFPAUD Change of Voice State.....	73
3.2.10.	+QBTHFPVOL Change of Voice Type.....	73

3.2.11.	+QBTHFPNETT	Change of Network Type .....	74
3.2.12.	+QBTHFPRING	Change of Ring Indication .....	74
3.2.13.	+QBTHFPCOD	Change of Code Type .....	74
3.3.	BT A2DP AVRCP Related URCs .....		75
3.3.1.	+QBTA2DPSCON	A2DP Connection .....	75
3.3.2.	+QBTA2DPSDCON	A2DP Disconnection .....	75
3.3.3.	+QBTA2DPAUDIOCFG	A2DP Audio Configuration .....	76
3.3.4.	+QBTA2DPAUDIOSTART	Start Playing.....	76
3.3.5.	+QBTA2DPAUDIOSTOPPED	Stop Playing .....	77
3.3.6.	+QBTAVRCPSCON	AVRCP Connection .....	77
3.3.7.	+QBTAVRCPSDCON	AVRCP Disconnection .....	78
3.3.8.	+QBTAVRCPVOLCHANGE	Change of AVRCP Volume .....	78
<b>4</b>	<b>Examples .....</b>		<b>79</b>
4.1.	BLE Communication .....		79
4.2.	BT HFP Use Process .....		85
4.3.	BT A2DP AVRCP Use Process .....		86
<b>5</b>	<b>Appendix Terms and Abbreviations .....</b>		<b>88</b>

## Table Index

Table 1: Applicable Modules .....	8
Table 2: Special Symbols.....	9
Table 3: Types of AT Commands .....	10
Table 4: Terms and Abbreviations .....	88



# 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, EG500U-CN, EG700U-CN 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

EGx00U	EG500U-CN
	EG700U-CN
EG915U	EG915U series

## 1.2. Special Symbols

Table 2: Special Symbols

Symbol	Definition
*	Unless otherwise specified, the asterisk (*) marked after the function, feature, interface, pin name, AT command or parameter of the module indicates that the function, feature, interface, pin name, AT command or parameter is under development and therefore is not supported temporarily.

# 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 3: Types of AT Commands**

Command Type	Syntax	Description
Test Command	<b>AT+&lt;cmd&gt;=?</b>	Test the existence of corresponding Write Command and return information about the type, value, or range of its parameter.
Read Command	<b>AT+&lt;cmd&gt;?</b>	Check the current parameter value of a corresponding Write Command.
Write Command	<b>AT+&lt;cmd&gt;=&lt;p1&gt;[,&lt;p2&gt;[,&lt;p3&gt;[...]]]</b>	Set user-definable parameter value.
Execution Command	<b>AT+&lt;cmd&gt;</b>	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 <b>AT+QBTPWR=?</b>	Response <b>+QBTPWR: (range of supported &lt;enable&gt;)</b>  <b>OK</b>
Read Command <b>AT+QBTPWR?</b>	Response <b>+QBTPWR: &lt;enable&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTPWR=&lt;enable&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration will not be saved.

### Parameter

<b>&lt;enable&gt;</b>	Integer type. Turn on/off BT. <ul style="list-style-type: none"> <li><u>0</u> Turn off BT</li> <li>1 Turn on BLE GATT server</li> <li>2 Turn on BLE GATT client</li> <li>3 SPP*</li> </ul>
-----------------------	---

- 4 Turn on BT HFP protocol
- 5 Turn on BT A2DP and AVRCP protocol

**Example**

```
AT+QBTPWR=1 //Turn on BLE GATT server.
OK
```

**2.3.2. AT+ QBTNAME Set Bluetooth Device Name**

This command sets Bluetooth device name.

AT+QBTNAME Set Bluetooth Device Name	
Test Command <b>AT+QBTNAME=?</b>	Response <b>OK</b>
Read Command <b>AT+QBTNAME?</b>	Response <b>+QBTNAME: &lt;code_type&gt;,&lt;device_name&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTNAME=&lt;code_type&gt;,&lt;device_name&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;code_type&gt;</b>	Integer type. Code type. 0 UTF8 code type. 1 GBK code type.
<b>&lt;device_name&gt;</b>	String type. Bluetooth device name. Maximum: 29 bytes.

**Example**

```
AT+QBTNAME? //Query Bluetooth device name.
+QBTNAME: 0,"MYBTDEVICE"
OK
```

## 2.4. Description of BLE AT Commands

### 2.4.1. General BLE AT Commands

#### 2.4.1.1. AT+QBTLEADDR Get BLE Device Address

This command gets BLE device address.

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

#### Parameter

**<BLE\_addr>** String type. BLE device address, for example, "A662616202C3".

#### Example

```
AT+QBTLEADDR? //Query BLE device address.
+QBTLEADDR: "A662616202C3"
OK
```

#### 2.4.1.2. AT+QBTLERANADDR Get Random Address of BLE Device

This command gets the random address of BLE device.

<b>AT+QBTLERANADDR Get Random Address of BLE Device</b>	
Test Command <b>AT+QBTLERANADDR=?</b>	Response <b>OK</b>
Read Command <b>AT+QBTLERANADDR?</b>	Response <b>+QBTLERANADDR: &lt;BLE_addr&gt;</b>

	<b>OK</b>
Maximum Response Time	10 s
Characteristics	/

**Parameter**

**<BLE\_addr>** String type. Random address of BLE device, for example, "A662616202C3".

**Example**

```
AT+QBTLERANADDR? //Query the random address of the BLE device.
+QBTLERANADDR: "A662616202C3"
OK
```

**2.4.1.3.AT+QBTGATDISCONN Disconnect the Connected Device**

This command disconnects the connected device.

<b>AT+QBTGATDISCONN Disconnect the Connected Device</b>	
Test Command <b>AT+QBTGATDISCONN=?</b>	Response <b>+QBTGATDISCONN: (range of supported &lt;connID&gt;s)</b>  <b>OK</b>
Write Command <b>AT+QBTGATDISCONN=&lt;connID&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration will not be saved.

**Parameter**

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

**Example**

```
AT+QBTGATDISCONN=0 //Disconnect the connected device.
OK
```

**2.4.1.4. AT+QBTGATCONNP Update Connection Parameters**

This parameter updates connection parameters.

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

**Parameter**

<b>&lt;connID&gt;</b>	Integer type. Connection ID. Range: 0–65535.
<b>&lt;min_interval&gt;</b>	Integer type. Minimum interval time. Range: 6–3200. Interval: 1.25 ms. Corresponding time range: 7.5 ms–4 s.
<b>&lt;max_interval&gt;</b>	Integer type. Maximum interval time. Range: 6–3200. Interval: 1.25 ms. Corresponding time range: 7.5 ms–4 s.
<b>&lt;latency&gt;</b>	Integer type. Latency, the number of ignored connection events. Range: 0–499.
<b>&lt;timeout&gt;</b>	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 //Update connection parameters.
OK
```



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

This command adds a device to White List.

<b>AT+QBTLEADDWHL Add a Device to White List</b>	
Test Command <b>AT+QBTLEADDWHL=?</b>	Response <b>+QBTLEADDWHL: (list of supported &lt;addr_type&gt;s),&lt;address&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTLEADDWHL=&lt;addr_type&gt;,&lt;address&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration will not be saved.

**Parameter**

<b>&lt;addr_type&gt;</b>	Integer type. Address type. 0 Public address 1 Random address
<b>&lt;address&gt;</b>	String type. BLE device address.

**Example**

```
AT+QBTLEADDWHL=0,"112233da8040" //Add a device to White List.
OK
```

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

This command deletes a device from White List.

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

Write Command <b>AT+QBTLEDELWHL=&lt;op&gt;[,&lt;addr_type&gt;,&lt;address&gt;]</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

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

**Example**

```
AT+QBTLEDELWHL=0 //Delete all devices from the White List.  
OK
```

**2.4.1.7.AT+QBTLEWHLINFO Get White List Information**

This command gets White List information.

<b>AT+QBTLEWHLINFO Get White List Information</b>	
Test Command <b>AT+QBTLEWHLINFO=?</b>	Response <b>OK</b>
Read Command <b>AT+QBTLEWHLINFO?</b>	Response <b>+QBTLEWHLINFO: &lt;addr_type&gt;,&lt;address&gt;</b>  <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	/

**Parameter**

<b>&lt;addr_type&gt;</b>	Integer type. Address type. 0 Public address 1 Random address
<b>&lt;address&gt;</b>	String type. BLE device address.

**Example**

```
AT+QBTLEWHLINFO? //Get White List information.
+QBTLEWHLINFO: 1,"112233da8040"
OK
```

**2.4.1.8.AT+QBTLEEXMTU Exchange MTU**

This command exchanges MTU.

<b>AT+QBTLEEXMTU Exchange MTU</b>	
Test Command <b>AT+QBTLEEXMTU?</b>	Response <b>+QBTLEEXMTU:</b> (range of supported <connID>s),(range of supported <MTU>s)  <b>OK</b>
Write Command <b>AT+QBTLEEXMTU=&lt;connID&gt;,&lt;MTU&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;connID&gt;</b>	Integer type. Connection ID. Range: 0–65535.
<b>&lt;MTU&gt;</b>	Integer type. Maximum transmission unit. Rang: 23–247. Unit: byte.

**Example**

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

## 2.4.2. AT Commands of BLE GATT Service

### 2.4.2.1. AT+QBTGATSS Add a Service

This command adds a service.

<b>AT+QBTGATSS Add a Service</b>	
Test Command <b>AT+QBTGATSS=?</b>	Response <b>+QBTGATSS:</b> (range of supported <servID>s),(list of supported <UUID_type>s),<serv_UUID_I>,(range of supported <serv_UUID_s>s),(list of supported <primary>s)  <b>OK</b>
Write Command <b>AT+QBTGATSS=&lt;servID&gt;,&lt;UUID_type&gt;[,&lt;serv_UUID_I&gt;][,&lt;serv_UUID_s&gt;],&lt;primary&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be 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_I> when <UUID_type>=1. 0 Long 128bit UUID 1 Short 16bit UUID
<serv_UUID_I>	String type. 128bit service UUID.
<serv_UUID_s>	String type. 16bit 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 //Add a service.
OK
```

**2.4.2.2. AT+QBTGATSC Add a Characteristic**

This command adds a characteristic to an existing service.

<b>AT+QBTGATSC Add a Characteristic</b>	
Test Command <b>AT+QBTGATSC=?</b>	Response <b>+QBTGATSC:</b> (range of supported <servID>s),(range of supported <charalD>s),(range of supported <prop>s),(list of supported <UUID_type>s),<serv_UUID_l>,(range of supported <serv_UUID_s>s)  <b>OK</b>
Write Command <b>AT+QBTGATSC=&lt;servID&gt;,&lt;charalD&gt;,&lt;prop&gt;,&lt;UUID_type&gt;[,&lt;serv_UUID_l&gt;][,&lt;serv_UUID_s&gt;]</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;servID&gt;</b>	Integer type. Service ID. Range: 0–65535.
<b>&lt;charalD&gt;</b>	Integer type. Characteristic ID. Range: 0–65535.
<b>&lt;prop&gt;</b>	Integer type. Characteristic property. Range: 0–255. Bit0: Broadcast Bit1: Read Bit2: Write Without Response Bit3: Write Bit4: Notify Bit5: Indicate Bit6: Authenticated signed Writes Bit7: Extended properties
<b>&lt;UUID_type&gt;</b>	Integer type. UUID type. Omit <serv_UUID_s> when <UUID_type>=0. Omit <serv_UUID_l> when <UUID_type>=1. 0 Long 128bit UUID 1 Short 16bit UUID
<b>&lt;serv_UUID_l&gt;</b>	String type. 128bit service UUID.
<b>&lt;serv_UUID_s&gt;</b>	Integer type. 16bit service UUID. Range: 0–65535.

**Example**

```
AT+QBTGATSC=0,0,58,1,10777 //Add a characteristic.
OK
```

**2.4.2.3. AT+QBTGATSCV Configure Characteristic Value Parameters**

This command configures characteristic value parameters.

<b>AT+QBTGATSCV Configure Characteristic Value Parameters</b>	
Test Command <b>AT+QBTGATSCV=?</b>	Response <b>+QBTGATSCV:</b> (range of supported <servID>s),(range of supported <charaID>s),(range of supports <permission>s),(list of supported <UUID_type>s),<serv_UUID_l>,(range of supported <serv_UUID_s>s),(range of supported <length>s),<value>  <b>OK</b>
Write Command <b>AT+QBTGATSCV=&lt;servID&gt;,&lt;charaID&gt;,&lt;permission&gt;,&lt;UUID_type&gt;[,&lt;serv_UUID_l&gt;][,&lt;serv_UUID_s&gt;],&lt;length&gt;,&lt;value&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<servID>	Integer type. Service ID. Range: 0–65535.
<charaID>	Integer type. Characteristic ID. Range: 0–65535.
<permission>	Integer type. Characteristic value permission. Range: 0–1023. Bit0: Read only, no security Bit1: Write only, no security 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 128bit UUID
	1 Short 16bit UUID
<serv_UUID_l>	String type. 128bit service UUID.
<serv_UUID_s>	Integer type. 16bit service UUID. Range: 0–65535.
<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" //Add characteristic values.
OK
```

**2.4.2.4. AT+QBTGATSCD Add Characteristic Descriptor**

This command adds characteristic descriptors.

AT+QBTGATSCD Add Characteristic Descriptor	
Test Command <b>AT+QBTGATSCD=?</b>	Response <b>+QBTGATSCD:</b> (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 <length>s),<value>  <b>OK</b>
Write Command <b>AT+QBTGATSCD=&lt;servID&gt;,&lt;charalD&gt;,&lt;permission&gt;,&lt;UUID_type&gt;[,&lt;serv_UUID_l&gt;][,&lt;serv_UUID_s&gt;],&lt;length&gt;,&lt;value&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<servID>	Integer type. Service ID. Range: 0–65535.
<charalD>	Integer type. Characteristic ID. Range: 0–65535.
<permission>	Integer type. Characteristic value permission. Range: 0–1023. Bit0: Read only, no security Bit1: Write only, no security

	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
<b>&lt;UUID_type&gt;</b>	Integer type. UUID type. Omit <b>&lt;serv_UUID_s&gt;</b> when <b>&lt;UUID_type&gt;=0</b> . Omit <b>&lt;serv_UUID_l&gt;</b> when <b>&lt;UUID_type&gt;=1</b> .
	0 Long 128bit UUID
	1 Short 16bit UUID
<b>&lt;serv_UUID_l&gt;</b>	String type. 128bit service UUID.
<b>&lt;serv_UUID_s&gt;</b>	Integer type. 16bit service UUID. Range: 0–65535.
<b>&lt;length&gt;</b>	Integer type. Length of characteristic descriptor value. Range: 0–65535. Unit: byte.
<b>&lt;value&gt;</b>	Integer type. Characteristic descriptor value.

**Example**

```
AT+QBTGATSCD=0,0,3,1,10498,2,"1234" //Add characteristic descriptor.
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 <b>AT+QBTGATSSC=?</b>	Response <b>+QBTGATSSC: (list of supported &lt;type&gt;s),(list of supported &lt;op&gt;s)</b>  <b>OK</b>
Write Command <b>AT+QBTGATSSC=&lt;type&gt;[,&lt;op&gt;]</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.



**Parameter**

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

**Example**

```
AT+QBTGATSSC=1,1 //Finish adding service.
OK
```

**2.4.2.6.AT+QBTGATCHSCV Change a Characteristic Value**

This command changes a characteristic value.

<b>AT+QBTGATCHSCV Change a Characteristic Value</b>	
Test Command <b>AT+QBTGATCHSCV=?</b>	Response <b>+QBTGATCHSCV:</b> (range of supported <b>&lt;servID&gt;s</b> ),(range of supported <b>&lt;charalD&gt;s</b> ),(range of supported <b>&lt;length&gt;s</b> ), <b>&lt;value&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTGATCHSCV=&lt;servID&gt;,&lt;charalD&gt;,&lt;length&gt;,&lt;value&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;servID&gt;</b>	Integer type. Service ID. Range: 0–65535.
<b>&lt;charalD&gt;</b>	Integer type. Characteristic ID. Range: 0–65535.
<b>&lt;length&gt;</b>	Integer type. Length of changed characteristic value, starting from 0. 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 configured characteristic value. Unit: byte.

**<value>** Integer type. Characteristic value.

**Example**

```
AT+QBTGATCHSCV=0,0,2,"0012" //Change the characteristic value.
OK
```

**2.4.2.7.AT+QBTGATSIND Send an Indication**

This command sends an indication.

**AT+QBTGATSIND Send an Indication**

Test Command <b>AT+QBTGATSIND=?</b>	Response <b>+QBTGATSIND:</b> (range of supported <b>&lt;op&gt;</b> s),(range of supported <b>&lt;connID&gt;</b> s),(range of supported <b>&lt;att_handle&gt;</b> s), <b>&lt;value_length&gt;</b> , <b>&lt;value&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTGATSIND=&lt;op&gt;,&lt;connID&gt;,&lt;att_handle&gt;[,&lt;value_length&gt;,&lt;value&gt;]</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;op&gt;</b>	Integer type. Omit <b>&lt;value_length&gt;</b> and <b>&lt;value&gt;</b> when <b>&lt;op&gt;</b> is either 1 or 2. 0 Direct mode 1 Echo mode 2 Passthrough mode
<b>&lt;connID&gt;</b>	Integer type. Connection ID. Range: 0–65535.
<b>&lt;att_handle&gt;</b>	Integer type. Attribute handle. Range: 1–65535.
<b>&lt;value_length&gt;</b>	Integer type. Indication length. Range: 0 to <b>&lt;MTU&gt;</b> -3. Unit: byte.
<b>&lt;value&gt;</b>	Integer type. Indication content.

**Example**

```
AT+QBTGATSIND=0,0,18,4,"1111" //Send an indication.
OK
```

**2.4.2.8. AT+QBTGATSNO D Send a Notification**

This command sends a notification.

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

**Parameter**

<b>&lt;op&gt;</b>	Integer type. Omit <b>&lt;value_length&gt;</b> and <b>&lt;value&gt;</b> when <b>&lt;op&gt;</b> is either 1 or 2. 0 Direct mode 1 Echo mode 2 Passthrough mode
<b>&lt;connID&gt;</b>	Integer type. Connection ID. Range: 0–65535.
<b>&lt;att_handle&gt;</b>	Integer type. Attribute handle. Range: 1–65535.
<b>&lt;value_length&gt;</b>	Integer type. Notification length. Range: 0 to <b>&lt;MTU&gt;-3</b> . Unit: byte.
<b>&lt;value&gt;</b>	Integer type. Notification content.

**Example**

```
AT+QBTGATSNO D=0,0,18,4,"1111" //Send a notification.
OK
```

**2.4.2.9.AT+QBTGATADV Set Advertising Parameters**

This command sets advertising parameters.

<b>AT+QBTGATADV Set Advertising Parameters</b>	
Test Command <b>AT+QBTGATADV=?</b>	Response <b>+QBTGATADV:</b> (list of supported <b>&lt;op&gt;</b> s), (range of supported <b>&lt;min_interval&gt;</b> s),(range of supported <b>&lt;max_interval&gt;</b> s),(range of supported <b>&lt;adv_type&gt;</b> s),(list of supported <b>&lt;own_addrtype&gt;</b> s),(range of supported <b>&lt;channel_map&gt;</b> s),(range of supported <b>&lt;filter&gt;</b> s),(list of supported <b>&lt;remote_addrtype&gt;</b> s), <b>&lt;remote_addr&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTGATADV=&lt;op&gt;,&lt;min_interval&gt;,&lt;max_interval&gt;,&lt;adv_type&gt;,&lt;own_addrtype&gt;,&lt;channel_map&gt;,&lt;filter&gt;[[,&lt;remote_addrtype&gt;][,&lt;remote_addr&gt;]]</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;op&gt;</b>	Integer type. Set advertising parameters of all address or the specified address. 0 Specified advertising address 1 All advertising address. Omit <b>&lt;remote_addrtype&gt;</b> and <b>&lt;remote_addr&gt;</b> when <b>&lt;op&gt;</b> =1
<b>&lt;min_interval&gt;</b>	Integer type. Minimum advertising interval, which should be less than <b>&lt;max_interval&gt;</b> . Range: 32–16384. Interval: 0.625 ms. Corresponding time range: 20 ms–10.24 s.
<b>&lt;max_interval&gt;</b>	Integer type. Maximum advertising interval. Range: 32–16384. Interval: 0.625 ms. Corresponding time range: 20 ms–10.24 s.
<b>&lt;adv_type&gt;</b>	Integer type. Advertising type. 0 Connectable undirected advertising 1 Connectable high duty cycle directed advertising. <b>&lt;min_interval&gt;</b> and <b>&lt;max_interval&gt;</b> do not work for connectable high duty cycle directed advertising, and the advertising interval is 3.75 ms. The advertising time of directed advertising is 1.28 s. 2 Non-connectable undirected advertising. Under this type, advertising

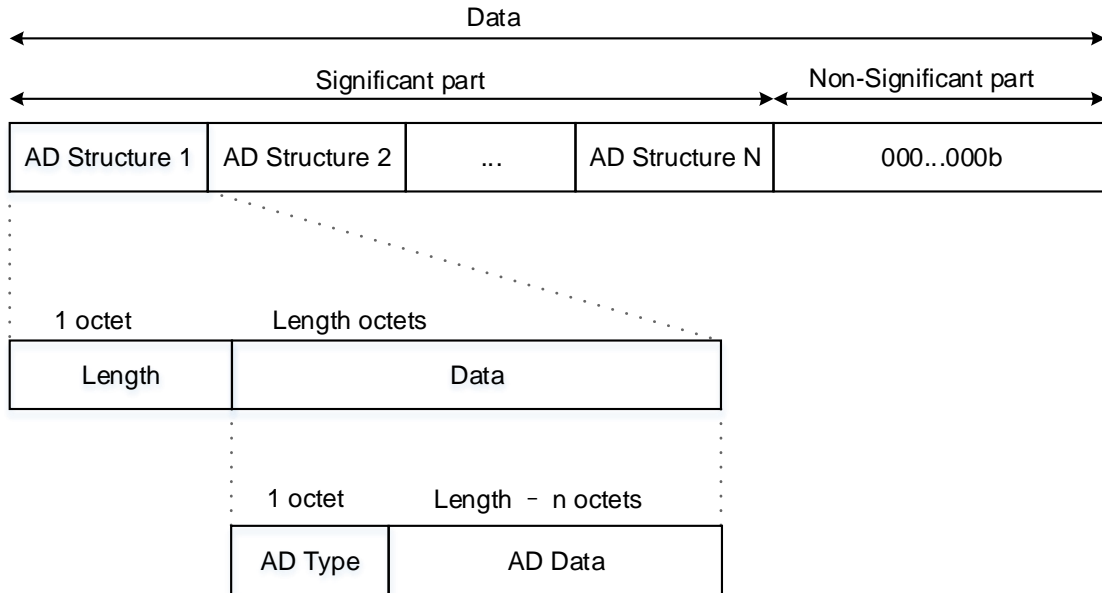
	interval parameter should be more than 160.
	3 Scannable undirected advertising. Under this type, the advertising interval should be more than 160.
	4 Connectable low duty cycle directed advertising. Under this type, the interval will no longer work, and the interval is 3.75 ms. Directed advertising time: 1.28 s.
<b>&lt;own_addrtype&gt;</b>	Integer type. Local address type.
	0 Public address
	1 Random address
<b>&lt;channel_map&gt;</b>	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.
	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
<b>&lt;filter&gt;</b>	Integer type. Advertising filter strategy.
	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
<b>&lt;remote_addrtype&gt;</b>	Integer type. Remote device address type.
	0 Public address
	1 Random address
<b>&lt;remote_addr&gt;</b>	String type. Remote device address.

**Example**

```
AT+QBTGATADV=0,128,160,0,1,7,0,1,"b17e431d1c5f" //Set advertising parameters.
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**

<b>AT+QBTADVDATA Set Advertising Data</b>	
Test Command <b>AT+QBTADVDATA=?</b>	Response <b>OK</b>
Write Command <b>AT+QBTADVDATA=&lt;length&gt;,&lt;data&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;length&gt;</b>	Integer type. The length of advertising data. Maximum: 31 octets.
<b>&lt;data&gt;</b>	String type. Advertising data, which consist of the following three fields (that is, multiple AD Structure). 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 <b>Core_v5.2</b> .
AD Data	Advertising data content, organized in big-endian format.

**Example**

**AT+QBTADVDATA=3,"020105"** Set 3 octets advertising data, which is scannable and connectable  
OK

**2.4.2.11.AT+QBTADVSTR Set Advertising Data Format**

This command sets the advertising data format.

<b>AT+QBTGATADVSTR Set Advertising Data Format</b>	
Test Command <b>AT+QBTADVSTR=?</b>	Response <b>OK</b>
Write Command <b>AT+QBTADVSTR=&lt;type&gt;,&lt;code_type1&gt;,&lt;data1&gt;[,&lt;code_type2&gt;,&lt;data2&gt;[,...]]</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;type&gt;</b>	Integer type. Advertising data type. The total length of all AD Structures cannot exceed 31 octets. If the AD data length of Bit1 is 6 octets, 2 octets need to be added to the length of advertising data. 1 octet is the length of Length itself and 1 octet is the length of AD type., totally 8 octets. Bit0: Advertising BLE device name. Bit1: Flag. <b>&lt;data&gt;</b> is required to be included, which only supports the hexadecimal format. Bit2: Manufacturer data. <b>&lt;data&gt;</b> is required to be included. Bit3: Transmission power level. <b>&lt;data&gt;</b> is required to be included, which only supports hexadecimal format. Bit4: Complete list of 16-bit service class UUIDs. <b>&lt;data&gt;</b> is required to be included,
---------------------	---

which only supports hexadecimal format.  
 Bit5: Service data. **<data>** is required to be included.

<b>&lt;code_type&gt;</b>	Integer type.
	0 hexadecimal string format.
	1 GBK encoding
	2 UTF8 encoding
<b>&lt;data&gt;</b>	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 <b>AT+QBTADVRSPDATA=?</b>	Response <b>OK</b>
Write Command <b>AT+QBTADVRSPDATA=&lt;length&gt;,&lt;data&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;length&gt;</b>	Integer type. The length of scan response data. Maximum: 31 octets.
<b>&lt;data&gt;</b>	String type. Scan response data. The data format is the same as advertising data.

**Example**

```
AT+QBTADVRSPDATA=7,"06094138393130" //Set the scan response data
OK
```



**2.4.2.13.AT+QBTADV Start/Stop Advertising**

This command starts or stops advertising.

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

**Parameter**

<b>&lt;enable&gt;</b>	Integer type. Start/Stop advertising. 0 Stop advertising 1 Start advertising
-----------------------	--

**Example**

```
AT+QBTADV=1 //Start advertising.
OK
```

**2.4.2.14.AT+ AT+QBTLEIBEA Set ibeacon Data**

This command sets ibeacon data.

<b>AT+QBTLEIBEA Set ibeacon Data</b>	
Test Command <b>AT+QBTLEIBEA=?</b>	Response <b>+QBTLEIBEA:</b> <UUID_I>,(range of supported <b>&lt;major&gt;</b> s), (range of supported <b>&lt;minor&gt;</b> s)  <b>OK</b>
Write Command <b>AT+QBTLEIBEA=&lt;UUID_I&gt;,&lt;major&gt;,&lt;minor&gt;</b>	Response <b>OK</b>

<minor>	Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

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

**Example**

```
AT+QBTLEIBEA="f5899b5f8000008000100000FE180000",20,25 //Set ibeacon data.
OK
```

**2.4.2.15.AT+QBTLEIBEACFG Set ibeacon Data to NVM**

This command sets ibeacon data to NVM.

**AT+QBTLEIBEACFG Set ibeacon Data to NVM**

Test Command <b>AT+QBTLEIBEACFG=?</b>	Response <b>+QBTLEIBEACFG: &lt;UUID_I&gt;,(range of supported &lt;major&gt;s),(range of supported &lt;minor&gt;s)</b>  <b>OK</b>
Write Command <b>AT+QBTLEIBEACFG=&lt;UUID_I&gt;,&lt;major&gt;,&lt;minor&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

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

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

**Example**

```
AT+QBTLEIBEACFG="f5899b5f8000008000100000FE180000",20,25 //Set ibeacon Data to NVM.
OK
```

**2.4.2.16.AT+QBTLESTATE Query Connection State**

This command queries connection state.

<b>AT+QBTLESTATE Query Connection State</b>	
Test Command <b>AT+QBTLESTATE=?</b>	Response <b>+QBTLESTATE:</b> (range of supported <b>&lt;cid&gt;s</b> ),(range of supported <b>&lt;connID&gt;s</b> ), <b>&lt;address&gt;</b> ,(list of supported <b>&lt;state&gt;s</b> ),(range of supported <b>&lt;att_handle&gt;s</b> )  <b>OK</b>
Read Command <b>AT+QBTLESTATE?</b>	Response <b>+QBTLESTATE:</b> <b>&lt;cid&gt;</b> , <b>&lt;connID&gt;</b> , <b>&lt;address&gt;</b> , <b>&lt;state&gt;</b> , <b>&lt;att_handle&gt;</b>  <b>OK</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;cid&gt;</b>	Integer type. Channel ID. Range: 0–65535.
<b>&lt;connID&gt;</b>	Integer type. Connection ID. Range: 0–65535.
<b>&lt;address&gt;</b>	String type. Address of the connected device.
<b>&lt;state&gt;</b>	Integer type. 0 Disconnected 1 Connected
<b>&lt;att_handle&gt;</b>	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.

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

**Parameter**

<cid>	Integer type. Channel ID. Range: 0–65535.
<type>	Integer type. Data type. 0 Send notification 1 Send indication
<length>	Integer type. Data length. Range: 0–2048. Unit: byte.
<data>	String type. Hexadecimal data.

**2.4.2.18. AT+QBTLEGSND Get Buffer Size of the Channel**

This command gets buffer size of the channel.

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

Write Command <b>AT+QBTLEGSND=&lt;cid&gt;</b>	Response <b>+QBTLEGSND: &lt;cid&gt;,&lt;size&gt;,&lt;nsend&gt;</b>  <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;cid&gt;</b>	Integer type. Channel ID. Range: 0–65535.
<b>&lt;size&gt;</b>	Integer type. The size of buffer data in the channel. Maximum: 2048 bytes.
<b>&lt;nsend&gt;</b>	Integer type. Size of data not sent in buffer. Range: 0–2028. Unit: byte.

**Example**

```
AT+QBTLEGSND=1 //Get buffer size of the channel.
+QBTLEGSND: 1,2048,0
OK
```

**2.4.2.19.AT+QBTLERCVM Set Data Receiving Mode of Server**

This command sets data receiving mode of server.

<b>AT+QBTLERCVM Set Data Receiving Mode of Server</b>	
Test Command <b>AT+QBTLERCVM=?</b>	Response <b>+QBTLERCVM: (range of supported &lt;type&gt;s)</b>  <b>OK</b>
Read Command <b>AT+QBTLERCVM?</b>	Response <b>+QBTLERCVM: &lt;type&gt;,&lt;time&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTLERCVM=&lt;type&gt;,&lt;time&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>

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

**Parameter**

<b>&lt;type&gt;</b>	Integer type. Omit <b>&lt;time&gt;</b> when <b>&lt;type&gt;=0</b> . 0 Direct mode 1 Buffer mode
<b>&lt;time&gt;</b>	Integer type. Time to receive buffer data from the channel and report it to URC. No report when <b>&lt;time&gt;=0</b> . Range: 0–3600000. Unit: ms.

**Example**

```
AT+QBTLERCVM=1,2000 //Set the sever receiving mode.
OK
```

**2.4.2.20.AT+QBTLERead Read Buffer Data**

This command reads buffer data.

<b>AT+QBTLERead Read Buffer Data</b>	
Test Command <b>AT+QBTLERead=?</b>	Response <b>+QBTLERead: (range of supported &lt;cid&gt;s),(range of supported &lt;length&gt;s)</b>  <b>OK</b>
Write Command <b>AT+QBTLERead=&lt;cid&gt;,&lt;length&gt;</b>	Response <b>+QBTLERead: &lt;cid&gt;,&lt;length&gt;,&lt;data&gt;</b>  <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

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

**Example**

```
AT+QBTLEREAD=1,3 //Reads buffer data.
OK
```

**2.5. Description of GATT Client AT Commands**

**2.5.1. AT+QBTSCANPARA Set Scan Parameters**

This command sets scan parameters.

<b>AT+QBTSCANPARA Set Scan Parameters</b>	
Test Command <b>AT+QBTSCANPARA=?</b>	Response <b>+QBTSCANPARA:</b> (list of supported <scan_mode>s),(range of supported <interval>s),(range of supported<window>s),(range of supported <scan_type>s),(list of supported <own_addrtype>s)  <b>OK</b>
Write Command <b>AT+QBTSCANPARA=&lt;scan_mode&gt;,&lt;interval&gt;,&lt;window&gt;,&lt;scan_type&gt;,&lt;own_addrtype&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<scan_mode>	Integer type. Scanning mode 0 Passive scan 1 Active scan
<interval>	Integer type. Scan interval. Range: 4–16384. Interval: 0.625 ms. Corresponding time range: 2.5 ms–10.24 s.

<b>&lt;window&gt;</b>	Integer type. Scan window. The duration of the scan, which should be less than <b>&lt;interval&gt;</b> . Range: 4–16384. Interval: 0.625 ms. Corresponding time range: 2.5 ms–10.24 s.
<b>&lt;scan_type&gt;</b>	Integer type. Scanning filter strategy. 0 All advertising packets except the directed advertising packets 1 Advertising packets from devices in the White List except the directional advertising of this device 2 Undirected advertising packets, directed advertising packets addressed to this device or directed advertising packets where the initiator address is a resolvable private address. 3 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
<b>&lt;own_addrtype&gt;</b>	Integer type. Local address type. 0 Public address 1 Random address

**Example**

```
AT+QBTSCANPARA=0,96,48,0,0 //Set scan parameters.
OK
```

**2.5.2. AT+QBTGATSCAN Start/Stop Scanning Device**

This command starts/stops scanning device.

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



**Parameter**

<b>&lt;activate&gt;</b>	Integer type. Start/stop scanning device.
0	Stop scanning
1	Start scanning

**Example**

```
AT+QBTGATSCAN=1 //Start scanning device.
OK
```

**2.5.3. AT+QBTGATCONN Connect a Device**

This command connects a device.

**AT+QBTGATCONN Connect a Device**

Test Command <b>AT+QBTGATCONN=?</b>	Response <b>+QBTGATCONN:</b> (list of supported <b>&lt;conn_type&gt;</b> s),(list of supported <b>&lt;type&gt;</b> s), <b>&lt;address&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTGATCONN=&lt;conn_type&gt;[,&lt;type&gt;],&lt;address&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;conn_type&gt;</b>	Integer type. Omit <b>&lt;type&gt;</b> when <b>&lt;conn_type&gt;</b> =0.
0	Cancel connection
1	Connect
<b>&lt;type&gt;</b>	Integer type. Address type.
0	Public address
1	Random address
<b>&lt;address&gt;</b>	String type. Device address to be connected.

**Example**

```
AT+QBTGATCONN=1,1,"554dd0dc5854" //Connect device.
OK
```

**2.5.4. AT+QBTGATSERV Scan Service**

This command scans service.

<b>AT+QBTGATSERV Scan Service</b>	
Test Command <b>AT+QBTGATSERV=?</b>	Response <b>+QBTGATSERV:</b> (list of supported <b>&lt;type&gt;s</b> ),(range of supported <b>&lt;connID&gt;s</b> ),(list of supported <b>&lt;UUID_type&gt;s</b> ), <b>&lt;UUID_I&gt;</b> ,(range of supported <b>&lt;UUID_s&gt;s</b> )  <b>OK</b>
Write Command <b>AT+QBTGATSERV=&lt;type&gt;,&lt;connID&gt;[,&lt;UUID_type&gt;][,&lt;UUID_I&gt;][,&lt;UUID_s&gt;]</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

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

**Example**

```
AT+QBTGATSERV=0,0 //Scan all services.
OK
```

### 2.5.5. AT+QBTGATINC Scan Include

This command scans Include.

<b>AT+QBTGATINC Scan Include</b>	
Test Command <b>AT+QBTGATINC=?</b>	Response <b>+QBTGATINC:</b> (range of supported <connID>s),(range of supported <start_handle>s),(range of supported <end_handle>s)  <b>OK</b>
Write Command <b>AT+QBTGATINC=&lt;connID&gt;&lt;start_handle&gt;,&lt;end_handle &gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

#### Parameter

<b>&lt;connID&gt;</b>	Integer type. Connection ID. Range: 0–65535.
<b>&lt;start_handle&gt;</b>	Integer type. Start handle. Range: 1–65535.
<b>&lt;end_handle&gt;</b>	Integer type. End handle. Range: 1–65535.

#### Example

```
AT+QBTGATINC=0,16,65535 //Scan Include.
OK
```

### 2.5.6. AT+QBTGATCHAR Scan Characteristic

This command scans characteristic.

<b>AT+QBTGATCHAR Scan Characteristic</b>	
Test Command <b>AT+QBTGATCHAR=?</b>	Response <b>+QBTGATCHAR:</b> (range of supported <connID>s),(range of supported <start_handle>),(range of supported <end_handle>s)  <b>OK</b>

Write Command <b>AT+QBTGATCHAR=&lt;connID&gt;,&lt;start_handle&gt;,&lt;end_handle&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;connID&gt;</b>	Integer type. Connection ID. Range: 0–65535.
<b>&lt;start_handle&gt;</b>	Integer type. Characteristic start handle. Range: 1–65535.
<b>&lt;end_handle&gt;</b>	Integer type. Characteristic end handle. Range: 1–65535.

**Example**

```
AT+QBTGATCHAR=0,16,65535 //Scan Characteristic
OK
```

**2.5.7. AT+QBTGATDESC Scan Characteristic Descriptor**

This command scans characteristic descriptor.

<b>AT+QBTGATDESC Scan Characteristic Descriptor</b>	
Test Command <b>AT+QBTGATDESC=?</b>	Response <b>+QBTGATDESC: (range of supported &lt;connID&gt;s),(range of supported &lt;start_handle&gt;s),(range of supported &lt;end_handle&gt;s)</b>  <b>OK</b>
Write Command <b>AT+QBTGATDESC=&lt;connID&gt;,&lt;start_handle&gt;,&lt;end_handle&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;connID&gt;</b>	Integer type. Connection ID. Range: 0–65535.
<b>&lt;start_handle&gt;</b>	Integer type. Descriptor start handle. Range: 1–65535.
<b>&lt;end_handle&gt;</b>	Integer type. Descriptor end handle. Range: 1–65535.

**Example**

```
AT+QBTGATDESC= 0,19,65535 //Scan a characteristic descriptor.
OK
```

**2.5.8. AT+QBTWRCHAR Write Characteristic Value**

This command writes characteristic value.

<b>AT+QBTWRCHAR Write Characteristic Value</b>	
Test Command <b>AT+QBTWRCHAR=?</b>	Response <b>+QBTWRCHAR:</b> (range of supported <b>&lt;connID&gt;</b> s),(range of supported <b>&lt;att_handle&gt;</b> s),(range of supported <b>&lt;length&gt;</b> s), <b>&lt;value&gt;</b> ,(list of supported <b>&lt;islong&gt;</b> s),(range of supported <b>&lt;offset&gt;</b> s)  <b>OK</b>
Write Command <b>AT+QBTCHAR=&lt;connID&gt;,&lt;att_handle&gt;,&lt;length&gt;,&lt;value&gt;,&lt;islong&gt;,&lt;offset&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;connID&gt;</b>	Integer type. Connection ID. Range: 0–65535.
<b>&lt;att_handle&gt;</b>	Integer type. Characteristic value handle. Range: 1–65535.
<b>&lt;length&gt;</b>	Integer type. Data length. Range: 1–65535. Unit: byte.
<b>&lt;value&gt;</b>	String type. Written data.
<b>&lt;islong&gt;</b>	Integer type. 0 Multiple data packets are not required. 1 Multiple subcontracting is required.
<b>&lt;offset&gt;</b>	Integer type. Offset address required after subcontracting. Range: 0–65535. Default: 0.

**Example**

```
AT+QBTWRCHAR= 0,41,2,"1234",0,0 //Write characteristic value.
OK
```

**2.5.9. AT+QBTWRCHARNORSP Write Characteristic Value Without Response**

This command writes characteristic value without response.

<b>AT+QBTWRCHARNORSP Write Characteristic Value Without Response</b>	
Test Command <b>AT+QBTWRCHARNORSP=?</b>	Response <b>+QBTWRCHARNORSP:</b> (range of supported <connID>s), (range of supported <att_handle>s),(range of supported <length>s),<value>  <b>OK</b>
Write Command <b>AT+QBTWRCHARNORSP=&lt;connID&gt; ,&lt;att_handle&gt;,&lt;length&gt;,&lt;value&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

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

**Example**

```
AT+QBTWRCHARNORSP=0,41,2,"1234" //Write characteristic value without response.
OK
```

### 2.5.10. AT+QBTRDCHARUUID Read Characteristic Value by UUID

This command reads characteristic value by UUID.

AT+QBTRDCHARUUID Read Characteristic Value by UUID	
Test Command <b>AT+QBTRDCHARUUID=?</b>	Response <b>+QBTRDCHARUUID:</b> (range of supported <connID>s),(list of supported <UUID_type>s),<UUID_I>,(range of supported <UUID_s>s),(range of supported <start_handle>s),(range of supported <end_handle>s)  <b>OK</b>
Write Command <b>AT+QBTRDCHARUUID=&lt;connID&gt;,&lt;UUID_type&gt;[,&lt;UUID_I&gt;][,&lt;UUID_s&gt;],&lt;start_handle&gt;,&lt;end_handle&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

#### Parameter

<b>&lt;connID&gt;</b>	Integer type. Connection ID. Range: 0–65535.
<b>&lt;UUID_type&gt;</b>	Integer type. Service UUID. 0 128bit UUID. Omit <UUID_s> when <UUID_type>=0. 1 16bit UUID. Omit <UUID_I> when <UUID_type>=1.
<b>&lt;UUID_I&gt;</b>	String type. 128bit UUID.
<b>&lt;UUID_s&gt;</b>	Integer type. 16bit UUID. Range: 0–65535.
<b>&lt;start_handle&gt;</b>	Integer type. Characteristic start handle. Range: 0–65535.
<b>&lt;end_handle&gt;</b>	Integer type. Characteristic end handle. Range: 0–65535.

#### Example

```
AT+QBTRDCHARUUID=0,1,6159,18,18 //Read characteristic value by UUID.
OK
```

### 2.5.11. AT+QBTRDCHARHAND Read Characteristic Value by Handle

This command reads characteristic value by handles.

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

#### Parameter

<connID>	Integer type. Connection ID. Range: 0–65535.
<att_handle>	Integer type. Characteristic handle. Range: 1–65535.
<islong>	Integer type. 0 Multiple data packets are not required. 1 Multiple data packets are required.
<offset>	Integer type. Offset address required after subpackaging. Range: 0–65535.

#### Example

```
AT+QBTRDCHARHAND= 0,18,0,0 //Read characteristic value by the handle.
OK
```

### 2.5.12. AT+QBTGATWRDESC Write Characteristic Descriptor

This command writes characteristic descriptor.

<b>AT+QBTGATWRDESC Write Characteristic Descriptor</b>	
Test Command <b>AT+QBTGATWRDESC=?</b>	Response <b>+QBTGATWRDESC:</b> (range of supported <connID>s),(range of supported <att_handle>s),(range of supported <length>s),<value>



	<b>OK</b>
Write Command <b>AT+QBTGATWRDESC=&lt;connID&gt;,&lt;att_handle&gt;,&lt;length&gt;,&lt;value&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;connID&gt;</b>	Integer type. Connection ID. Range: 0–65535.
<b>&lt;att_handle&gt;</b>	Integer type. Descriptor handle. Range: 1–65535.
<b>&lt;length&gt;</b>	Integer type. Data length. Range: 1–65535. Unit: byte.
<b>&lt;value&gt;</b>	String type. Written data.

**Example**

```
AT+QBTGATWRDESC= 0,19,2,"1234" //Write characteristic descriptor.
OK
```

**2.5.13. AT+QBTGATRDESC Read Characteristic descriptor**

This command reads characteristic descriptor.

<b>AT+QBTGATRDESC Read Characteristic descriptor</b>	
Test Command <b>AT+QBTGATRDESC=?</b>	Response <b>+QBTGATRDESC: (range of supported &lt;connID&gt;s),(range of supported &lt;att_handle&gt;s),(list of supported &lt;islong&gt;s)</b>  <b>OK</b>
Write Command <b>AT+QBTGATRDESC=&lt;connID&gt;,&lt;att_handle&gt;,&lt;islong&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;connID&gt;</b>	Integer type. Connection ID. Range: 0–65535.
<b>&lt;att_handle&gt;</b>	Integer type. Description handle. Range: 0–65535.
<b>&lt;islong&gt;</b>	Integer type. 0 Multiple data packets are not required. 1 Multiple data packets are required.

**Example**

```
AT+QBTGATRDESC=0,19,0 //Read Characteristic descriptor.
OK
```

**2.6. Description of General BT AT Commands**

**2.6.1. AT+QBTSCANMODE Set Scan Mode**

This command sets scan mode.

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

**Parameter**

---

<b>&lt;scan_mode&gt;</b>	Integer type. Scan mode.
0	Not discoverable or connectable
1	Not connectable but discoverable
2	Not discoverable but connectable
3	Discoverable and connectable

---

**Example**

```
AT+QBTHFPCONN=3 //Set scan mode.
OK
```

**2.7. Description of BT HFP AT Commands**

**2.7.1. AT+QBTHFPCONN Connect a Device**

This command connects a device.

**AT+QBTHFPCONN Connect a Device**

Test Command <b>AT+QBTHFPCONN=?</b>	Response <b>+QBTHFPCONN: &lt;address&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTHFPCONN=&lt;address&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration will not be saved.

**Parameter**

---

<b>&lt;address&gt;</b>	String type. Address of the device to be connected
------------------------	--

---

**Example**

```
AT+QBTHFPCONN="66cac9a26e38" //Connect a device.
OK
```

### 2.7.2. AT+QBTHFPDISCONN Disconnect a Device

This command disconnects a device.

<b>AT+QBTHFPDISCONN Disconnect a Device</b>	
Test Command <b>AT+QBTHFPDISCONN=?</b>	Response <b>+QBTHFPDISCONN: &lt;address&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTHFPDISCONN=&lt;address&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration will not be saved.

#### Parameter

<b>&lt;address&gt;</b>	String type. Address of the connected device.
------------------------	---

#### Example

```
AT+QBTHFPDISCONN="66cac9a26e38" //Disconnect a device.
OK
```

### 2.7.3. AT+QBTHFPVOLI Set the Volume

This command sets the volume.

<b>AT+QBTHFPVOL Set the Volume</b>	
Test Command <b>AT+QBTHFPVOL=?</b>	Response <b>+QBTHFPVOL: (list of supported &lt;volume&gt;s),&lt;address&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTHFPVOL=&lt;volume&gt;,&lt;address&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s

Characteristics	The command takes effect immediately. The configurations will not be saved.
-----------------	--

**Parameter**

<volume>	Integer type. Volume. Range: 1–15.
<address>	String type. Address of the connected device.

**Example**

```
AT+QBTHFPVOL=10,"66cac9a26e38" //Set the volume.
OK
```

**2.7.4. AT+QBTHFPCALL Control Voice Call**

This command controls voice call.

AT+QBTHFPCALL Control Voice Call	
Test Command <b>AT+QBTHFPCALL=?</b>	Response <b>+QBTHFPCALL:</b> (list of supported <op>s),<address>  <b>OK</b>
Write Command <b>AT+QBTHFPCALL=&lt;op&gt;,&lt;address&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<op>	Integer type. 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" //Operate a voice call.
OK
```

**2.7.5. AT+QBTHFPDIAL\* Dial**

This command dials.

<b>AT+QBTHFPDIAL* Dial</b>	
Test Command <b>AT+QBTHFPDIAL=?</b>	Response <b>+QBTHFPDIAL:</b> (list of supported <type>s),<address>,<number>  <b>OK</b>
Write Command <b>AT+QBTHFPDIAL=&lt;type&gt;,&lt;address&gt; [,&lt;number&gt;]</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;type&gt;</b>	Integer type. Omit <number> when <type>=1. 0 Dial 1 Redial
<b>&lt;address&gt;</b>	String type. Address of the connected device.
<b>&lt;number&gt;</b>	String type. Dialed number.

**Example**

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

### 2.7.6. AT+QBTHFPVOLR\* Turn On/Off Voice Assistant

This command turns on and turns off voice assistants.

<b>AT+QBTHFPVOLR Turn On/Off Voice Assistant</b>	
Test Command <b>AT+QBTHFPVOLR=?</b>	Response <b>+QBTHFPVOLR: (支持的&lt;op&gt;列表),&lt;address&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTHFPVOLR=&lt;op&gt;,&lt;address&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

#### Parameter

<b>&lt;op&gt;</b>	Integer type. Turn off/turn on voice assistant. 0 Turn off voice assistant 1 Turn on voice assistant
<b>&lt;address&gt;</b>	String type. Address of the connected device.

#### Example

```
AT+QBTHFPVOLR=0,"66cac9a26e38" //Turn off voice assistant.
OK
```

### 2.7.7. AT+QBTHFPTHRC\* Three-way Calling

This command makes three-way calling.

<b>AT+QBTHFPTHRC Three-way Calling</b>	
Test Command <b>AT+QBTHFPTHRC=?</b>	Response <b>+QBTHFPTHRC: (list of supported &lt;cmd&gt;s),&lt;address&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTHFPTHRC=&lt;cmd&gt;,&lt;address&gt;</b>	Response <b>OK</b> Or

	<b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configurations will not be saved.

**Parameter**

<b>&lt;cmd&gt;</b>	Integer type. Range: 0–255.
<b>&lt;address&gt;</b>	String type. Address of the connected device.

**Example**

```
AT+QBTHFPTSRC=0,"66cac9a26e38" //Three-way calling.
OK
```

**2.8. Description of BT A2DP AVRCP AT Commands**

**2.8.1. AT+QBTAVRCPDISCONN Disconnect a Device**

This command disconnects a device.

<b>AT+QBTAVRCPDISCONN Disconnect a Device</b>	
Test Command <b>AT+QBTAVRCPDISCONN=?</b>	Response <b>+QBTAVRCPDISCONN: &lt;address&gt;</b>  <b>OK</b>
Write Command <b>AT+QBTAVRCPDISCONN=&lt;address&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration will not be saved.

**Parameter**

<b>&lt;address&gt;</b>	String type. Address of the connected device.
------------------------	---



**Example**

```
AT+QBTAVRCPDISCONN="66cac9a26e38" //Disconnect a device.
OK
```

**2.8.2. AT+QBTAVRCPVOL Set the Volume**

This command sets the volume.

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

**Parameter**

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

**Example**

```
AT+QBTAVRCPVOL=80 //Set the volume to 80.
OK
```

### 2.8.3. AT+QBTAVRCPCTRL Control Audio Playback

This command controls audio playback.

<b>AT+QBTAVRCPCTRL Control Audio Playback</b>	
Test Command <b>AT+QBTAVRCPCTRL=?</b>	Response <b>+QBTAVRCPCTRL:</b> (list of supported <op>s)  <b>OK</b>
Write Command <b>AT+QBTAVRCPCTRL=&lt;op&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	10 s
Characteristics	The command takes effect immediately. The configuration will not be saved.

#### Parameter

<b>&lt;op&gt;</b>	Integer type. 0 Stop playing 1 Start playing 2 Switch to the last song 3 Switch to the next song
-------------------	--

#### Example

```
AT+QBTAVRCPCTRL=0 //Stop playing.
OK
```

### 2.8.4. AT+QBTAVRCPSTATE Get the Audio Playing State

This command gets the audio playing state.

<b>AT+QBTAVRCPSTATE Get the Audio Playing State</b>	
Test Command <b>AT+QBTAVRCPSTATE=?</b>	Response <b>+QBTAVRCPSTATE:</b> (list of supported <state>s)  <b>OK</b>
Read Command <b>AT+QBTAVRCPSTATE?</b>	Response <b>+QBTAVRCPSTATE:</b> <state>

	OK Or ERROR
Maximum Response Time	10 s
Characteristics	/

**Parameter**

<b>&lt;state&gt;</b>	Integer type. Device playing state.
0	Stop playing
1	Playing
2	Pause
3	Switching to the next song
4	Switching to the last song
255	Error

**Example**

```

AT+QBTAVRCPSTATE=? //Get the audio playing state.
AT+QBTAVRCPSTATE: 1 //Playing.

OK
    
```

# 3 Description of URCs

## 3.1. BLE Related URCs

### 3.1.1. +QBTGATSCON GATT Connection

**+QBTGATSCON GATT Connection**

+QBTGATSCON: <connID>,<address> GATT connection.

**Parameter**

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

**Example**

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

### 3.1.2. +QBTGATSDCON GATT Disconnection

**+QBTGATSDCON GATT Disconnection**

+QBTGATSDCON: <connID>,<address> GATT disconnection.

**Parameter**

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

**Example**

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

**3.1.3. +QBTGATMTU MTU During Connection**

**+QBTGATMTU MTU During Connection**

+QBTGATMTU: <handle>,<MTU> MTU during connection.

**Parameter**

<handle> Integer type. Connection handle.  
 <MTU> Integer type. MTU.

**Example**

+QBTGATMTU: 0,200

**3.1.4. +QBTGATCONNP Connection Parameters Update**

**+QBTGATCONNP Connection Parameters Update**

+QBTGATCONNP: <connID>,<min\_interval>,<max\_i  
 nterval>,<latency>,<timeout> Connection parameters update

**Parameter**

<connID> Integer type. Connection ID.  
 <min\_interval> Integer type. Minimum interval.  
 <max\_interval> Integer type. Maximum interval.  
 <latency> Integer type. Latency.  
 <timeout> Integer type. Timeout period.

**Example**

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

**3.1.5. +QBTGATDESCDATA GATT Client Report Characteristic Descriptor**

**+QBTGATDESCDATA GATT Client Report Characteristic Descriptor**

+QBTGATDESCDATA: <connID>,<att\_handl  
 e>,<length>,<value> GATT client reports the characteristic descriptor.

**Parameter**

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

**Example**

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

**3.1.6. +QBTSCANDATAIND GATT Client Report Scanning Data**

**+QBTSCANDATAIND GATT Client Report Scanning Data**

```
+QBTSCANDATAIND:<name>,<type>,<address>,<RSSI>  
>,<event_type>,<raw_data>
```

GATT client reports the scanning data.

**Parameter**

<name>	String type without double quotation marks. Name of the scanned device.
<type>	Integer type. Address type. 0 Public address 1 Random address
<address>	String type. Device address.
<RSSI>	Integer type. RSSI.
<event_type>	Integer type. The type of the scanned advertising. 0 Scanned connectable and scannable undirected advertising 1 Scanned connectable directional advertising 2 Scanned scannable and undirected advertising 3 Scanned non-connectable undirected advertising 4 Scanned response data
<raw_data>	String type. The scanned raw advertising data.

**Example**

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

### 3.1.7. +QBTSERVDATA GATT Client Report Service Data

#### +QBTSERVDATA GATT Client Report Service Data

+QBTSERVDATA: <UUID><start\_handle>,<end\_handle> GATT client reports service data.

#### Parameter

<UUID> Integer type. Current UUID.  
 <start\_handle> Integer type. Start handle.  
 <end\_handle> Integer type. End handle.

#### Example

+QBTSERVDATA: 6159,16,65535

### 3.1.8. +QBTCHARDATA GATT Client Report Characteristic Value

#### +QBTCHARDATA GATT Client Report Characteristic Value

+QBTCHARDATA: <count>[,<UUID1>,<handle1>,<value\_handle1>,<properties1>[,<UUID2>,<handle2>,<value\_handle2>,<properties2>[,...]]] GATT client reports characteristic value.

#### Parameter

<count> Integer type. The number of characteristics. Every characteristic consists of <handle>, <properties>, <value\_handle> and <UUID>.  
 <UUID> Integer type. UUID of the current characteristic.  
 <handle> Integer type. Handle to define the characteristic.  
 <value\_handle> Integer type. Characteristics value handle.  
 <properties> Integer type. Characteristics properties.

#### Example

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

### 3.1.9. +QBTDESCDATA GATT Client Report Characteristic Descriptor Value

**+QBTDESCDATA GATT Client Report Characteristic Descriptor Value**

**+QBTDESCDATA:** <count>[,<UUID>,<handle>[,<UUID>,<handle>[,...]]] GATT client reports characteristic descriptor value.

**Parameter**

<count> Integer type. Characteristics number. Every description includes <handle> and <UUID>.  
 <UUID> Integer type. UUID of the characteristic descriptor.  
 <handle> Integer type. Descriptor value handle.

**Example**

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

### 3.1.10. +QBTATTERR GATT Client Report Attribute Error

**+QBTATTERR GATT Client Report Attribute Error**

**+QBTATTERR:** <err> GATT client report attribute error.

**Parameter**

<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>,<lenth>,<value>** GATT client reports receiving notification.

**Parameter**

<b>&lt;att_handle&gt;</b>	Integer type. The characteristic value attribute handle of the received data.
<b>&lt;length&gt;</b>	Integer type. Length of received data. Unit: byte.
<b>&lt;value&gt;</b>	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>** GATT client reports receiving indication.

**Parameter**

<b>&lt;att_handle&gt;</b>	Integer type. The characteristic value attribute handle of the received data.
<b>&lt;length&gt;</b>	Integer type. Length of received data. Unit: byte.
<b>&lt;value&gt;</b>	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>** GATT client reports the state of writing characteristic value.

**Parameter**

<b>&lt;status&gt;</b>	String type without double quotation marks.
OK	Characteristic value is written successfully.
FAILED	Failed to write characteristic value.

**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>** GATT client report the state of writing characteristic value without response

**Parameter**

<b>&lt;status&gt;</b>	String type without double quotation marks.
OK	Characteristic value is written successfully.
FAILED	Failed to write characteristic value.

**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: <length>,<value>**

GATT client reports reading characteristic value by handle.

**Parameter**

**<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>,<length>,<value>**

GATT client reports reading characteristic value by UUID.

**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>

GATT client reports the state of writing characteristic descriptor.

**Parameter**

<status>	String type without double quotation marks.
OK	Characteristic value is written successfully.
FAILED	Failed to write characteristic value.

**Example**

+QBTGATWRDESC: OK

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

**+QBTGATRDESC GATT Client Report Reading Characteristic descriptor**

+QBTGATRDESC: <length>,<value>

GATT client report reading characteristic descriptor.

**Parameter**

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

**Example**

+QBTGATRDESC: 6,"180011111111"

**3.1.19. +QBTGATRDATAIND GATT Client Read Data**

**+QBTGATRDATAIND GATT Client Read Data**

+QBTGATRDATAIND: <connID>,<att\_handle>,<length>,<value>

GATT server reports that GATT client reads data.

**Parameter**

<connID>	Integer type. Connection ID.
<att_handle>	Integer type. Handle value for the operation.
<length>	Integer type. Length of received data.
<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**

<b>+QBTLESTATE: &lt;cid&gt;,&lt;connID&gt;,&lt;address&gt;,&lt;state&gt;,&lt;att_handle&gt;</b>	GATT server reports connection state update.
---	--

**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.
<state>	Integer type. 0 Disconnected 1 Connected
<att_handle>	Integer type. Attribute handle. Range: 1–65535.

**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**

<b>+QBTLEVALDATA: &lt;cid&gt;,&lt;address&gt;,&lt;length&gt;,&lt;value&gt;</b>	GATT server report receiving data.
--	------------------------------------

**Parameter**

<cid>	Integer type. Channel ID. Range: 0–65535.
<address>	String type. Address of the connected device.
<length>	Integer type. Received data length. Range: 0–512. Unit: byte.
<value>	String type. Hexadecimal 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>,<length>** GATT server reports receiving buffer data.

**Parameter**

<cid>	Integer type. Channel ID. Range: 0–65535.
<address>	String type. Address of the connected device.
<length>	Integer type. Received data length. Range: 0–512. Unit: byte.

**Example**

```
+QBTLEVALDATI: 0,"3af3f58716f9",3
```

**3.2. BT HFP Related URCs**

**3.2.1. +QBTHFPSCON HFP Connection**

**+QBTHFPSCON HFP Connection**

**+QBTHFPSCON: <address>** HFP connection.

**Parameter**

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

**Example**

+QBTHFPSCON: "66cac9a26e38"

**3.2.2. +QBTHFPSDCON HFP Disconnection**

**+QBTHFPSDCON HFP Disconnection**

+QBTHFPSDCON: <address> HFP disconnection.

**Parameter**

<address> String type. Device address that is disconnected.

**Example**

+QBTHFPSDCON: "66cac9a26e38"

**3.2.3. +QBTHFPCALL Change of Call State**

**+QBTHFPCALL Change of Call State**

+QBTHFPCALL: <state>,<address> Change of call state.

**Parameter**

<state> Integer type.  
 0 No ongoing call  
 1 Ongoing call  
 <address> String type. Address of a disconnected device.

**Example**

+QBTHFPCALL: 0,"66cac9a26e38"

**3.2.4. +QBTHFPCALS Change of Call Setting State**

**+QBTHFPCALS Change of Call Setting State**

+QBTHFPCALS: <state>,<address> Change of call setting state.

**Parameter**

<b>&lt;state&gt;</b>	Integer type. 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.
<b>&lt;address&gt;</b>	String type. Address of a disconnected device.

**Example**

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

**3.2.5. +QBTHFPNET Change of Network State**

**+QBTHFPNET Change of Internet State**

**+QBTHFPNET: <state>,<address>** Change of Internet state.

**Parameter**

<b>&lt;state&gt;</b>	Integer type. 0 Current network is unavailable. 1 Current network is available.
<b>&lt;address&gt;</b>	String type. Address of a 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>** Change of network signal strength.

**Parameter**

<b>&lt;signal_strength&gt;</b>	Integer type. Signal Strength. Range: 0–255.
<b>&lt;address&gt;</b>	String type. Address of a disconnected device.



**Example**

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

**3.2.7. +QBTHFPBAT Change of Battery Level**

**+QBTHFPBAT Change of Battery Level**

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

Change of battery level.

**Parameter**

<b>&lt;level&gt;</b>	Integer type. Battery level. Range: 0–255.
<b>&lt;address&gt;</b>	String type. Address of a disconnected device.

**Example**

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

**3.2.8. +QBTHFPCALH Change of Call Holding State**

**+QBTHFPCALH Change of Call Holding State**

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

Change of call holding state.

**Parameter**

<b>&lt;state&gt;</b>	Integer type. Call hold state.
	0 No call is holding.
	1 Activate a call and hold a call.
	2 Call is holding.
<b>&lt;address&gt;</b>	String type. Address of a disconnected device.

**Example**

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

### 3.2.9. +QBTHFPAUD Change of Voice State

**+QBTHFPAUD Change of Voice State**

+QBTHFPAUD: <state>,<address>

Change of voice state.

参数

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

Example

+QBTHFPAUD: 0,"66cac9a26e38"

### 3.2.10. +QBTHFPVOL Change of Voice Type

**+QBTHFPVOL Change of Voice Type**

+QBTHFPVOL: <type>,<volume>

Change of voice type.

Parameter

<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: <type>,<address>** Change of network type.

**Parameter**

<b>&lt;type&gt;</b>	Integer type. Network type. 0 Local 1 Roaming
<b>&lt;address&gt;</b>	String type. Address of a disconnected device.

**Example**

**+QBTHFPNETT: 0,"66cac9a26e38"**

### 3.2.12. +QBTHFPRING Change of Ring Indication

**+QBTHFPRING Change of Ring Indication**

**+QBTHFPRING: <address>** Change of ring indication.

**Parameter**

<b>&lt;address&gt;</b>	String type. Address of a disconnected device.
------------------------	--

**Example**

**+QBTHFPRING: "66cac9a26e38"**

### 3.2.13. +QBTHFPCOD Change of Code Type

**+QBTHFPCOD Change of Code Type**

**+QBTHFPCOD: <type>,<address>** Change of code type.

**Parameter**

<b>&lt;type&gt;</b>	Integer type. Code type. 0 CVSD 1 mSBC
<b>&lt;address&gt;</b>	String type. Address of a disconnected device.

**Example**

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

### 3.3. BT A2DP AVRCP Related URCs

#### 3.3.1. +QBTA2DPSCON A2DP Connection

**+QBTA2DPSCON A2DP Connection**

**+QBTA2DPSCON: <state>,<address>** A2DP connection.

**Parameter**

<b>&lt;state&gt;</b>	Integer type. 0 Disconnected 1 Connecting 2 Connected 3 Disconnecting
<b>&lt;address&gt;</b>	String type. Address of the connected device.

**Example**

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

#### 3.3.2. +QBTA2DPSDCON A2DP Disconnection

**+QBTA2DPSDCON A2DP Disconnection**

**+QBTA2DPSDCON: <state>,<address>** A2DP disconnection.

**Parameter**

<b>&lt;state&gt;</b>	Integer type. 0 Disconnected 1 Connecting 2 Connected 3 Disconnecting
<b>&lt;address&gt;</b>	String type. Address of the connected device.

**Example**

**+QBTA2DPSPDCON: 0,"66cac9a26e38"**

**3.3.3. +QBTA2DPAUDIOCFG A2DP Audio Configuration**

**+QBTA2DPAUDIOCFG A2DP Audio Configuration**

**+QBTA2DPAUDIOCFG: <address>** A2DP audio configuration.

**Parameter**

<b>&lt;address&gt;</b>	String type. Address of the connected device.
------------------------	---

**Example**

**+QBTA2DPAUDIOCFG: "66cac9a26e38"**

**3.3.4. +QBTA2DPAUDIOSTART Start Playing**

**+QBTA2DPAUDIOSTART Start Playing**

**+QBTA2DPAUDIOSTART: <state>,<address>** Start playing.

**Parameter**

<b>&lt;state&gt;</b>	Integer type. 0 Stop playing 1 Pause 2 Start playing
<b>&lt;address&gt;</b>	String type. Address of the connected device.

**Example**

```
+QBTA2DPAUDIOSTART: 2,"66cac9a26e38"
```

**3.3.5. +QBTA2DPAUDIOSTOPPED Stop Playing**

**+QBTA2DPAUDIOSTOPPED Stop Playing**

**+QBTA2DPAUDIOSTOPPED: <state>,<address>** Stop playing.

**Parameter**

<b>&lt;state&gt;</b>	Integer type. 0 Stop playing 1 Pause 2 Start playing
<b>&lt;address&gt;</b>	String type. Address of the connected device.

**Example**

```
+QBTA2DPAUDIOSTOPPED: 1,"66cac9a26e38"
```

**3.3.6. +QBTAVRCPSCON AVRCP Connection**

**+QBTAVRCPSCON AVRCP Connection**

**+QBTAVRCPSCON: <state>,<address>** AVRCP connection.

**Parameter**

<b>&lt;state&gt;</b>	Integer type. 0 Disconnected 1 Connecting 2 Connected 3 Disconnecting
<b>&lt;address&gt;</b>	String type. Address of the connected device.

**Example**

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

### 3.3.7. +QBTAVRCPSPDCON AVRCP Disconnection

**+QBTAVRCPSPDCON AVRCP Disconnection**

**+QBTAVRCPSPDCON: <state>,<address>** AVRCP disconnection.

**Parameter**

<b>&lt;state&gt;</b>	Integer type. 0 Disconnected 1 Connecting 2 Connected 3 Disconnecting
<b>&lt;address&gt;</b>	String type. Address of the connected device.

**Example**

**+QBTAVRCPSPDCON: 0,"66cac9a26e38"**

### 3.3.8. +QBTAVRCPVOLCHANGE Change of AVRCP Volume

**+QBTAVRCPVOLCHANGE Change of AVRCP Volume**

**+QBTAVRCPVOLCHANGE: <volume>,<address>** Change of AVRCP volume.

**Parameter**

<b>&lt;volume&gt;</b>	Integer type. Volume. Range: 0–127.
<b>&lt;address&gt;</b>	String type. Address of the connected device.

**Example**

**+QBTAVRCPVOLCHANGE: 80,"66cac9a26e38"**

# 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, EGx00U 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.

<b>AT+QBTPWR=1</b>	//Turn BLE GATT server on.
OK	
<b>AT+QBTGATADV=1,128,160,0,0,7,0</b>	//Set advertising parameters.
OK	
<b>AT+QBTADVDATA=9,"020106050938393130"</b>	//Set advertising data.
OK	
<b>AT+QBTADVRSPDATA=6,"050938393130"</b>	//This step can be omitted according to actual needs.
OK	
<b>AT+QBTGATSS=0,1,6159,1</b>	//Add the first service.
OK	
<b>AT+QBTGATSC=0,0,58,1,10777</b>	//Add characteristics in the first service.
OK	
<b>AT+QBTGATSCV=0,0,3,1,10777,244,"1234"</b>	//Configure characteristic value parameter.
OK	
<b>AT+QBTGATSCD=0,0,3,1,10498,2,"0300"</b>	//Add descriptors in characteristics.



```

OK
AT+QBTGATSS=1,0,"f5899b5f8000008000100000FE180000",1 //Add the second
service.

OK
AT+QBTGATSC=1,0,58,0,"f5899b5f8000008000100000FEFF1111" //Add the first
characteristic in
the second
service.

OK
AT+QBTGATSCV=1,0,3,0,"f5899b5f8000008000100000FEFF1111",244,"1234" //Configure
characteristic
value parameter.

OK
AT+QBTGATSCD=1,0,3,1,10498,2,"0300" //Add characteristic
descriptor

OK
AT+QBTGATSC=1,1,16,0,"f5899b5f8000008000100000FDFF1111" //Add the second
characteristic in
the second
service.

OK
AT+QBTGATSCV=1,1,3,0,"f5899b5f8000008000100000FDFF1111",244,"1234" //Configure
characteristic
value parameter.

OK
AT+QBTGATSCD=1,1,3,1,10498,2,"0300" //Add characteristic
descriptor

OK
AT+QBTGATSSC=1,1 //Finish adding
service.

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 by 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 notification.
OK
AT+QBTGATSIND=0,0,18,4,"11111111" //Send indication.
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.

OK

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

+QBTLEVALDATI: 0,"3af3f58716f9",3

```

```

+QBTLEVALDATI: 0,"3af3f58716f9",3
AT+QBTLEEREAD=0,3 //The server reads
buffer data.

+QBTLEEREAD: 0,3,"125678"

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

+QBTGATMTU: 0,220
AT+QBTGATCONNP=0,39,39,0,2000 //Update
connection
parameters.

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 scan parameters.
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 a service.
OK

+QBTSERVDATA: 6144,1,11

+QBTSERVDATA: 6145,12,15

+QBTSERVDATA: 6159,16,19

+QBTSERVDATA: 6398,20,65535
AT+QBTGATCHAR=0,16,19 //Scan characteristics.
OK

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

+QBTATTERR: 0x0A
AT+QBTGATDESC=0,19,19 //Scan characteristic descriptors.
OK

```



```

AT+QBTWRCHAR=0,18,3,"125678",0,0
OK

+QBTGATWRCHAR: OK

+QBTGATMTU: 0,220 //Get latest MTU.

+QBTGATCONNP: 0,39,39,0,2000 //Get latest connection parameters.
AT+QBTGATDISCONN=0 //Actively disconnect from the server.
OK

+QBTGATSDCON: 0,"112233da8045" //Disconnected from the server.
    
```

## 4.2. BT HFP Use Process

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

```

AT+QBTPWR=4 //Turn BT HFP Protocol on.
OK
AT+QBTNAME="quec_headset" //Set BT device name.
OK
AT+QBTSCANMODE=3 //Set to be searchable and connectable, waiting for the
mobile phone to connect.
OK

+QBTHFPCALL: 0,"9a697d241368"
+QBTHFPCALS: 0,"9a697d241368"
+QBTHFPNET: 1,"9a697d241368"
+QBTHFPNETS: 5,"9a697d241368"
+QBTHFPBAT: 2,"9a697d241368"
+QBTHFPCALH: 0,"9a697d241368"
+QBTHFPSCON: "9a697d241368" //Connect with a mobile phone successfully.
+QBTHFPCALS: 1,"9a697d241368"
+QBTHFPAUD: 2,"9a697d241368"
    
```

```

+QBTHFPCALS: 1,"9a697d241368"

+QBTHFPRING: "9a697d241368"           //The mobile phone rings.

+QBTHFPRING: "9a697d241368"
AT+QBTHFPCALL=1,"9a697d241368"
OK

+QBTHFPCALL: 1,"9a697d241368"         //Answer the call.

+QBTHFPCALS: 0,"9a697d241368"
AT+QBTHFPVOL=10,"9a697d241368"
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.

```

AT+QBTPWR=5           //Turn BT A2DP and AVRCP Protocol on.
OK
AT+QBTNAME="quec_headset" //Set BT name.
OK
AT+QBTSCANMODE=3     //Set 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"

+QBTAVRCPSPDCON: 0,"9a697d241368"

+QBTA2DPSPDCON: 0,"9a697d241368"



# 5 Appendix Terms and Abbreviations

**Table 4: 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
GAP	Generic Access Profile
GATT	Generic Attribute Profile
GBK	Chinese Internal Code Specification
HF	Hands Free
HFP	Hands-free Profile
ID	Identity
MTU	Maximum Transmission Unit
NVM	Non-Volatile Memory
PDU	Protocol Data Unit
SPP	Serial Port Profile
URC	Unsolicited Result Code
UTF	Unicode Transformation Format
UUID	Universally Unique Identifier