

# **MC60&M66&M66 R2.0&M66-DS**

## **BT Application Note**

**GSM/GPRS/GNSS Module Series**

Rev. MC60&M66&M66 R2.0&M66-DS\_BT\_Application\_Note\_V1.3

Date: 2019-11-07

Status: Released



**Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:**

**Quectel Wireless Solutions Co., Ltd.**

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

Tel: +86 21 5108 6236

Email: [info@quectel.com](mailto:info@quectel.com)

**Or our local office. 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 to: [support@quectel.com](mailto:support@quectel.com)

**GENERAL NOTES**

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

**COPYRIGHT**

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL WIRELESS SOLUTIONS CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

***Copyright © Quectel Wireless Solutions Co., Ltd. 2019. All rights reserved.***

# About the Document

## History

Revision	Date	Author	Description
1.0	2015-01-13	Victor CHEN	Initial
1.1	2015-03-13	Victor CHEN	<ol style="list-style-type: none"><li>1. Added independence description of Bluetooth function in Chapter 1.5</li><li>2. Added notes for operation in Chapter 2.1</li><li>3. Added pairing mode in Chapter 2.5</li><li>4. Added AT+QBTCNNND in Chapter 3.2.14</li></ol>
1.2	2015-06-25	Hyman DING/ Ramos ZHANG	<ol style="list-style-type: none"><li>1. Added description of HFG in Chapter 3.2.12-3.2.18</li><li>2. Added AT+QBTHFGAUDIO in Chapter 3.2.27</li><li>3. Added Transparent Access Mode in SPP profile</li><li>4. Added example for Transparent Access Mode</li><li>5. Added HFP AT command and example</li></ol>
1.3	2019-11-07	Miles MA/ Sharon LI	<ol style="list-style-type: none"><li>1. Updated related description of AT+QBTPAIR and AT+QBTPAIRCNF in Chapter 3.2.7 and 3.2.8.</li><li>2. Updated the example of AT+QBTCNNND in Chapter 3.2.13.</li><li>3. Added command AT+QBTOPPPUSH, AT+QBTOPPACPT and AT+QBTCFG in Chapter 3.2.23, 3.2.24 and 3.2.25.</li><li>4. Added URC +QBTVGS, +QHFCME, +QBTOPPPUSH, +QBTIND: "OPPrecv" and +QBTOPPACPT in Chapter 3.3.6~3.3.10.</li></ol>

## Contents

<b>About the Document</b> .....	<b>2</b>
<b>Contents</b> .....	<b>3</b>
<b>Table Index</b> .....	<b>6</b>
<b>1 Introduction</b> .....	<b>7</b>
1.1. General Introduction.....	7
1.2. BT Profiles.....	7
1.3. BT Address.....	8
1.4. AT Interface for BT Function .....	8
1.5. Independence of BT Function .....	8
<b>2 BT Operation</b> .....	<b>9</b>
2.1. Notes for BT Operation .....	10
2.1.1. Quantity of the Searched Devices .....	10
2.1.2. Quantity of the Paired Devices .....	10
2.1.3. Special Function .....	10
2.2. Enable/Disable BT .....	10
2.3. Configure BT .....	11
2.3.1. Change BT Name.....	11
2.3.2. Set BT Visibility.....	11
2.3.3. Read BT Address .....	11
2.4. Scan BT Devices.....	11
2.4.1. Scan until Timeout.....	12
2.4.2. Cancel Scanning .....	12
2.5. Pair/Unpair BT Device.....	12
2.5.1. Originate SSP Pairing .....	13
2.5.2. Originated PIN Code Pairing.....	13
2.5.3. Received SSP Pairing .....	14
2.5.4. Received PIN Code Pairing .....	14
2.5.5. Unpair BT Device .....	14
2.6. Check Supported Profiles .....	14
2.7. Connection/Disconnection .....	15
2.7.1. SPP Connection/Disconnection .....	15
2.7.1.1. Initiate SPP Connection (Buffer Access Mode).....	15
2.7.1.2. Initiate SPP Connection (AT Command Mode).....	15
2.7.1.3. Initiate SPP Connection (Transparent Access Mode) .....	15
2.7.1.4. Accept SPP Connection (Buffer Access Mode) .....	16
2.7.1.5. Accept SPP Connection (AT Command Mode) .....	16
2.7.1.6. Accept SPP Connection (Transparent Access Mode).....	16
2.7.1.7. Initiate SPP Disconnection (Buffer Access Mode and AT Command Mode) ..	16
2.7.1.8. Initiate SPP Disconnection (Transparent Access Mode).....	17
2.7.1.9. Indicate SPP Disconnection (Buffer Access Mode and AT Command Mode)	17
2.7.1.10. Indicate SPP Disconnection (Transparent Access Mode) .....	17

2.7.2.	HFP Connection/Disconnection .....	17
2.7.2.1.	Initiate HFP Connection with HF Device .....	17
2.7.2.2.	Initiate HFP Connection with AG Device .....	17
2.7.2.3.	Accept HFP Connection .....	18
2.7.2.4.	Initiate HFP Disconnection with HF Device.....	18
2.7.2.5.	Indication of HFP Disconnection with HF Device.....	18
2.7.2.6.	Indication of HFP Disconnection with AG device .....	18
2.7.3.	Connect to SPP Directly Using BT Device Address .....	18
2.8.	Query Information of Device Status .....	20
2.9.	SPP/HFP Connection.....	20
2.9.1.	SPP Connection .....	20
2.9.1.1.	SPP Connection (AT Command Mode).....	20
2.9.1.2.	SPP Connection (Buffer Access Mode).....	21
2.9.1.2.1	Read Data.....	21
2.9.1.2.2	Send Data.....	21
2.9.1.3.	SPP Connection (Transparent Access Mode).....	22
2.9.2.	HFP Connection .....	23
2.9.2.1.	Module Acts as an AG Device .....	23
2.9.2.2.	Module Acts as an HF Device .....	24
<b>3</b>	<b>AT Commands for BT .....</b>	<b>26</b>
3.1.	AT Command Syntax .....	26
3.2.	Description of AT Commands.....	26
3.2.1.	AT+QBTPWR Enable/Disable BT .....	26
3.2.2.	AT+QBTNAME Read/Write Local Name of BT Device .....	27
3.2.3.	AT+QBTADDR Read Local Address of BT Device.....	28
3.2.4.	AT+QBTVISB Set Visibility of BT .....	28
3.2.5.	AT+QBTSCAN Scan BT Devices .....	29
3.2.6.	AT+QBTSCANC Stop Scanning.....	31
3.2.7.	AT+QBTPAIR Pair BT Device.....	31
3.2.8.	AT+QBTPAIRCNF Confirm Pairing .....	32
3.2.9.	AT+QBTUNPAIR Delete Paired BT Device.....	33
3.2.10.	AT+QBTSTATE Query Device Status.....	33
3.2.11.	AT+QBTGPROF Check Supported Profiles.....	34
3.2.12.	AT+QBTCNN Connect with BT Device.....	35
3.2.13.	AT+QBTCNNND Connect to SPP Directly Using BT Device Address .....	37
3.2.14.	AT+QBTAAPT Accept Connection .....	38
3.2.15.	AT+QBTDISCONN Disconnection .....	39
3.2.16.	AT+QSPSEND Send Data .....	40
3.2.17.	AT+QSPREAD Read Data .....	41
3.2.18.	AT+QBTA Answer Incoming Call.....	42
3.2.19.	AT+QBTAH Hung up Voice Call .....	42
3.2.20.	AT+QBTAHD Dial Voice Call.....	43
3.2.21.	AT+QBTAHDL Redial Last Phone Number.....	43
3.2.22.	AT+QBTHFGAUDIO Select Audio Output Path While Connecting with an HF Device	

.....	43
3.2.23. AT+QBTOPPPUSH OPP Client Pushes Files .....	44
3.2.24. AT+QBTOPPACPT OPP Server Decides Whether to Accept Push Request from OPP Client.....	44
3.2.25. AT+QBTCFG Configure BT Pairing Mode .....	45
3.3. Description of URCs.....	47
3.3.1. +QBTIND: "pair" Notify Pairing.....	47
3.3.2. +QBTIND: "conn" Notify Connection.....	47
3.3.3. +QBTIND: "disc" Notify Disconnection.....	48
3.3.4. +QBTIND: "recv" Notify Incoming Data .....	48
3.3.5. +QBTIND: "ring" Notify Incoming Call .....	49
3.3.6. +QBTVGS Notify When Volume of AG Speaker is Changed.....	49
3.3.7. +QHFCME Notify When Changing of Call State is Finished.....	49
3.3.8. +QBTOPPPUSH Notify When File Sending is Finished .....	50
3.3.9. +QBTIND: "OPPPrecv" Notify When OPP Server Receives Send Request from OPP Client.....	50
3.3.10. +QBTOPPACPT Notify When OPP Server Finishes Receiving File.....	51
<b>4 Appendix A Reference.....</b>	<b>52</b>
<b>5 Appendix B Summary of &lt;errorcode&gt; .....</b>	<b>53</b>

## Table Index

TABLE 1: RELATED DOCUMENTS.....	52
TABLE 2: TERMS AND ABBREVIATIONS.....	52
TABLE 3: DIFFERENT CODING SCHEMES OF +CME ERROR: <ERRORCODE> .....	53

# 1 Introduction

This document mainly introduces how to use the BT (Bluetooth) function on Quectel MC60, M66, M66 R2.0, and M66-DS modules.

## 1.1. General Introduction

BT is a proprietary open wireless technology standard for exchanging data over short distances (using short-wavelength radio transmissions in ISM band ranging from 2400MHz to 2480MHz) among fixed and mobile devices, creating personal area networks with a high level of security. BT was standardized as IEEE802.15.1.

Module	BT 3.0	BT 4.0
M66	✓	
M66 R2.0	✓ (Optional)	
M66-DS	✓	
MC60	✓	✓ (Optional)

## 1.2. BT Profiles

To use BT wireless technology, the device must be able to interpret certain BT profiles. BT profiles define possible applications and specify general behaviors that enable BT devices to communicate with each other. There is a wide range of BT profiles describing various types of applications or use cases for devices. Under the guidance provided by the BT specification, developers can create applications so as to work with other BT devices. Among all profiles, there are four basic ones: GAP, SDAP, SPP, and GOEP. MC60, M66, M66 R2.0 and M66-DS modules support SPP, HFP, and OPP.

### 1.3. BT Address

Each BT transceiver is allocated with a unique 48-bit BT device address. This address is derived from the IEEE802 standard. This 48-bit address is divided into three fields:

- LAP field: Lower Address Part consisting of 24 bits
- UAP field: Upper Address Part consisting of 8 bits
- NAP field: Non-significant Address Part consisting of 16 bits

LAP and UAP form the significant address part of the BT device address. Quectel module with the BT function will specify an address before it leaves the factory.

### 1.4. AT Interface for BT Function

Quectel provides a set of AT commands to support basic BT operations including scanning, pairing, connection and so on. SPP, OPP and HFP profiles are supported on Quectel modules. SPP profile can be used for communication between the module and other BT devices via serial port. OPP profile defines the requirements for the protocols and procedures that shall be used by the applications providing the Object Push usage model. While through HFP profile, the module can act as an HF (Hands-Free Unit) device or AG (Audio Gateway) device.

### 1.5. Independence of BT Function

BT function could work without (U)SIM card or in airplane mode (**AT+CFUN=0** or **4**).

## 2 BT Operation

Before using the BT function, it is necessary to turn on BT via AT commands and then connect the module to a BT device. When using a BT device with the Quectel module for the first time, it is necessary to scan and pair the device with the module to ensure that they are successfully connected.

The following figure shows the detailed BT operation based on the SPP profile via AT commands.

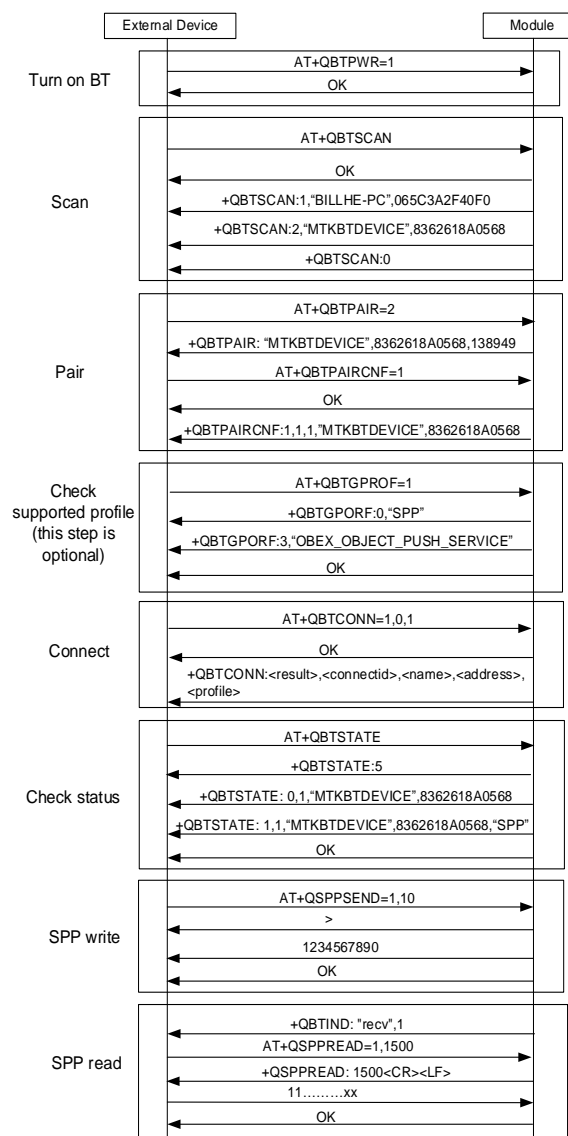


Figure 1: BT Operation Flow Chart (SPP Profile)

## 2.1. Notes for BT Operation

### 2.1.1. Quantity of the Searched Devices

In a single scanning process, at most 20 devices could be searched.

### 2.1.2. Quantity of the Paired Devices

At most 10 paired devices could be stored in users' device. If users want to continue to pair with the new device even if space is full, it is needed to remove some paired devices using "AT+QBTUNPAIR=<paired\_device\_id>" so as to free the storage space.

### 2.1.3. Special Function

If users want to use the address and the corresponding password (PIN Code pairing) of other BT devices to connect to SPP directly without scanning (pairing process will work internally), AT+QBTCONND=<addr>[,<mode>[,<password>]] can be used. For more details, please refer to *Chapter 2.7.3*.

## 2.2. Enable/Disable BT

BT function should be enabled first before use.

```
AT+QBTPWR=1           //Enable BT function
OK
AT+QBTPWR?             //Query power status
+QBTPWR: 1             //BT function is enabled
OK
```

Power off BT if it is not used.

```
AT+QBTPWR=0           //Power off BT function
OK
```

#### NOTE

If BT function is enabled, the module is visible to other BT devices. Please set the BT function of the module invisible if it is required to be hidden.

## 2.3. Configure BT

### 2.3.1. Change BT Name

BT name can be changed via **AT+QBTNAME**.

```
AT+QBTNAME="MYBTDEVICE"  
OK  
AT+QBTNAME?  
+QBTNAME: "MYBTDEVICE"  
OK
```

### 2.3.2. Set BT Visibility

BT visibility can be set via **AT+QBTVISB**.

```
AT+QBTVISB=0 //BT is invisible  
OK  
AT+QBTVISB=1 //BT is visible forever  
OK  
AT+QBTVISB=2,60 //BT is visible for 60 seconds  
OK  
  
+QBTVISB: 0 //Time of being visible comes to an end and BT is invisible now
```

### 2.3.3. Read BT Address

BT address can be read via **AT+QBTADDR**.

```
AT+QBTADDR?  
+QBTADDR: A662616202C3  
  
OK
```

## 2.4. Scan BT Devices

The module scans and displays the device ID, name and address of all available BT devices around it.

If the device that users want to pair with is not shown in the list, please check if the BT function of the target device is enabled or set as visible. **AT+QBTSCANC** can be used to cancel the scanning process, and **AT+QBTSCAN** can be used to restart the process.

### 2.4.1. Scan until Timeout

```
AT+QBTSCAN //Start to scan
OK

+QBTSCAN: 1,"BILLHE-PC",065C3A2F40F0

+QBTSCAN: 2,"GAOSHENGHAI",8E1F49F4402C

+QBTSCAN: 3,"BTDEVICE",8362618A0568

+QBTSCAN: 4,"QUECTEL-RAMOS",0AB6F27D001A

+QBTSCAN: 0 //Complete scanning
```

### 2.4.2. Cancel Scanning

```
AT+QBTSCAN //Start to scan
OK

+QBTSCAN: 1,"M66",8362618A0568

+QBTSCAN: 2,"QUECTEL-RAMOS",0AB6F27D001A

AT+QBTSCANC //Cancel scanning
+QBTSCAN: 0

OK
```

## 2.5. Pair/Unpair BT Device

Visible devices will be shown in the list with the corresponding device ID, name and address. The scanning ID of the device need to be selected, and **AT+QBTPAIR=<scan\_id>** as well as **AT+QBTPAIRCNF=1** need to be used to conduct pairing and confirmation, then the pairing result will be shown.

If pairing is successfully finished, the module will show the paired device ID. **AT+QBTSTATE** can be used to query the pairing status. Meanwhile, pairing information is saved as non-volatile, hence the device only needs to be paired once.

Currently, the following two pairing modes are supported:

### **PIN Code Pairing Mode**

For PIN Code pairing mode, matching password is required to be input by both sides of BT devices (if both of them have input devices), or one side inputs matching password while the other confirms it automatically.

### **SSP (Simple Safe Pair) Mode - Numeric Comparison Mode (6-bit random password comparison)**

In SSP mode, the protocol will generate a digital code, which needs to be confirmed by both sides. This mode is used between devices that have display function and “Yes” or “No” option, such as mobile phone or module.

In a pairing scene, the pairing mode between A and B depends on the mode supported by both sides. If both A and B support SSP, then the pairing mode between the two sides is SSP. If both A and B support PIN Code pairing only, or one supports SSP while the other supports PIN Code pairing, then the pairing mode will be PIN Code pairing.

**AT+QBTUNPAIR=<paired\_device\_id>** can be used to unpair the paired BT device with the device ID.  
**AT+QBTSTATE** can be used to query the pairing status.

#### **2.5.1. Originate SSP Pairing**

<b>AT+QBTPAIR=2</b>	//Use device ID
OK	
<b>+QBTPAIR: "M66",8362618A0568,138949</b>	//Indicate pairing status
<b>AT+QBTPAIRCNF=1</b>	//Accept pairing
OK	
<b>+QBTPAIRCNF: 1,1,1,"M66",8362618A0568</b>	//Pairing result

#### **2.5.2. Originated PIN Code Pairing**

<b>AT+QBTPAIR=2</b>	
OK	
<b>+QBTPAIR: "M66",8362618A0568</b>	//Indicate pairing status
<b>AT+QBTPAIRCNF=1,"1234"</b>	//Accept pairing with password (double quotes are required)
OK	

```
+QBTPAIRCNF: 1,1,1,"M66",8362618A0568 //Pairing result
```

### 2.5.3. Received SSP Pairing

```
+QBTIND: "pair","btd_left",656261222B10,138949 //Indicate pairing status
AT+QBTPAIRCNF=1 //Accept pairing
OK
```

```
+QBTPAIRCNF: 1,1,1,"btd_left",656261222B10 //Pairing result
```

### 2.5.4. Received PIN Code Pairing

```
+QBTIND: "pair","btd_left",656261222B10 //Indicate pairing status
AT+QBTPAIRCNF=1,"1234" //Accept pairing with password (double quotes are
                        //required)
OK
+QBTPAIRCNF: 1,1,1,"btd_left",656261222B10 //Pairing result
```

### 2.5.5. Unpair BT Device

```
AT+QBTUNPAIR=1 //Unpair the paired BT device with the device ID
OK
```

## 2.6. Check Supported Profiles

After pairing is completed, **AT+QBTGPROF=<paired\_device\_id>** can be used to check profiles supported by the device.

```
AT+QBTGPROF=1 //Paired device ID
+QBTGPROF: 0,"SPP"
+QBTGPROF: 3,"OBEX_OBJECT_PUSH_SERVICE"
+QBTGPROF: 5,"HF_PROFILE"
OK
```

## 2.7. Connection/Disconnection

**AT+QBTCONN=<paired\_device\_id>,<profile\_id>[,<mode>]** can be used to connect the module with other BT device using the paired device ID, while **AT+QBTDISCONN=<connected\_id>** can be used to disconnect the module with other BT device using the connected device ID. And **AT+QBTSTATE** can be applied to query the connection status.

### 2.7.1. SPP Connection/Disconnection

The following examples mainly introduce how to connect to/disconnect SPP.

#### 2.7.1.1. Initiate SPP Connection (Buffer Access Mode)

```
AT+QBTCONN=1,0,1 //Initiate SPP connection
OK
+QBTCONN: 1,1,"btd_left",656261222B10,"SPP" //After successful connection, MCU can send
and receive data via AT+QSPSEND or
AT+QSPREAD
```

#### 2.7.1.2. Initiate SPP Connection (AT Command Mode)

```
AT+QBTCONN=1,0,0 //Initiate SPP connection
OK
+QBTCONN: 1,1,"btd_left",656261222B10,"SPP" //After successful connection, the data sent by
the device on the other side will be treated as
AT Command (remote AT)
```

#### 2.7.1.3. Initiate SPP Connection (Transparent Access Mode)

```
AT+QBTCONN=1,0,2 //Initiate SPP connection
OK
+QBTCONN: 1,1,"btd_left",656261222B10,"SPP"
CONNECT
//Enter data transparent access mode now, and read/write data directly
```

#### 2.7.1.4. Accept SPP Connection (Buffer Access Mode)

```
+QBTIND: "conn","btd_left",656261222B10,"SPP" //Indicate connection status
AT+QBTACPT=1,1 //Accept connection
OK

+QBTACPT: 1,2,"btd_left",656261222B10,"SPP" //After successful connection, MCU can send
and receive data via AT+QSPSEND or
AT+QSPREAD
```

#### 2.7.1.5. Accept SPP Connection (AT Command Mode)

```
+QBTIND: "conn","btd_left",656261222B10,"SPP" //Indicate connection status
AT+QBTACPT=1,0 //Accept connection
OK

+QBTACPT: 1,2,"btd_left",656261222B10,"SPP" //After successful connection, the data sent by
the device on the other side will be treated as
AT command (remote AT)
```

#### 2.7.1.6. Accept SPP Connection (Transparent Access Mode)

```
+QBTIND: "conn","btd_left",656261222B10,"SPP" //Indicate connection status
AT+QBTACPT=1,2 //Accept connection
OK

+QBTACPT: 1,2,"btd_left",656261222B10,"SPP"

CONNECT

//Enter data transparent access mode now, and read/write data directly
```

#### 2.7.1.7. Initiate SPP Disconnection (Buffer Access Mode and AT Command Mode)

```
AT+QBTDISCONN=2 //Disconnect the connection with connected
device ID 2
OK

+QBTDISCONN: 2,1,"btd_left",656261222B10,"SPP" //Disconnection result
```

#### 2.7.1.8. Initiate SPP Disconnection (Transparent Access Mode)

```
AT+QBTDISCONN=2 //Disconnect the connection with connected
                    device ID 2
OK
CLOSED //Exit from transparent access mode first
+QBTDISCONN: 2,1,"btd_left",656261222B10,"SPP" //Disconnection result
```

#### 2.7.1.9. Indicate SPP Disconnection (Buffer Access Mode and AT Command Mode)

```
+QBTIND: "disc",1,1,"btd_left",656261222B10,"SPP" //URC indicating the disconnection status
```

#### 2.7.1.10. Indicate SPP Disconnection (Transparent Access Mode)

```
CLOSED //Exit from transparent access mode
+QBTIND: "disc",1,1,"btd_left",656261222B10,"SPP" //URC indicating the disconnection status
```

### 2.7.2. HFP Connection/Disconnection

The following examples mainly introduce how to connect to/disconnect HFP.

#### 2.7.2.1. Initiate HFP Connection with HF Device

```
AT+QBTCNN=4,6 //Module serving as HF actively initiates HFP
                connection to the device with paired ID 4.
OK
+QBTCNN: 1,0,"Motorola H270",00241C6743E9,"HFG_PROFILE"
```

#### 2.7.2.2. Initiate HFP Connection with AG Device

```
AT+QBTCNN=5,5 //The module serving as AG actively initiates
                HFP connection to the device with paired ID 5.
OK
```

```
+QBTCNN: 1,0,"HUAWEI P6-C00",80717A48820B,"HF_PROFILE"
```

### 2.7.2.3. Accept HFP Connection

When the module works as an AG or an HF device, the other BT device can connect to the module directly without the need of any accepting command.

```
+QBTCNN: 1,0,"HUAWEI P6-C00",80717A48820B,"HF_PROFILE" //AG device is connected to the module
```

```
+QBTCNN: 1,0,"Motorola H270",00241C6743E9,"HFG_PROFILE" //HF device is connected to the module
```

### 2.7.2.4. Initiate HFP Disconnection with HF Device

```
AT+QBTDISCONN=0 //Initiate HFP disconnection with the connected device with ID 0.
```

OK

```
+QBTDISCONN: 0,4,"Motorola H270", 00241C6743E9,"HFG_PROFILE" //Disconnection result
```

### 2.7.2.5. Indication of HFP Disconnection with HF Device

```
+QBTIND: "disc",0,4,"Motorola H270",00241C6743E9,"HFG_PROFILE"
```

//URC indicating disconnection status. 0 indicates connected device ID; 4 indicates paired device ID.

### 2.7.2.6. Indication of HFP Disconnection with AG device

```
+QBTIND: "disc",0,5,"HUAWEI P6-C00",80717A48820B,"HF_PROFILE"
```

//URC indicating disconnection status. 0 indicates connected device ID; 5 indicates paired device ID.

## 2.7.3. Connect to SPP Directly Using BT Device Address

If connecting to SPP directly is required using the address and the corresponding password (PIN Code pairing) of the other BT device, **AT+QBTCNN=<addr>[,<mode>[,<password>]]** can be used.

//After BT is powered on, the AT command can be sent directly.

```

AT+QBTCONND=00025BB3A978,1,"0000" //Connect to SPP in buffer access mode.
OK

+QBTPAIR: 00025BB3A978,"MPT-II",0 //Connect to SPP for the first time

+QBTPAIRCNF: 1,2,1,"MPT-II",00025BB3A978

+QBTCONN: 1,1,"MPT-II",00025BB3A978,"SPP" //Connected successfully
AT+QSPSEND=1,94 //Send data to print
>
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
OK
AT+QBTDISCONN //In this condition, this AT command can be used
without connected device ID

OK

+QBTDISCONN: 1,1,"MPT-II",B3A9785B0002,"SPP"

//Connect to SPP in transparent access mode directly

AT+QBTCONND=F0728C0D50AE,2
OK

+QBTCONN: 1,1,"GT-e",F0728C0D50AE,"SPP"

CONNECT //Enter transparent access mode
XXXXXXXXXXXXXXXXXXXXXXXXX //Receive data from remote BT device

+++ //Send "+++" to exit from transparent access mode
OK //Exit from transparent access mode, but do not
disconnect

ATO //Use ATO command to enter transparent mode
CONNECT //Enter transparent mode again

```

## NOTES

1. There is no need to perform scanning and pairing.
2. The connection method mentioned in this chapter can only be used for the SPP connection.

## 2.8. Query Information of Device Status

**AT+QBTSTATE** can be used to get the working, pairing and connection information. Pairing information is non-volatile. This command can also be used to query whether the device has been paired before; if so, the paired information can be directly used to connect with the device. Meanwhile, this command can be used to check the connection status.

**AT+QBTSTATE**

**+QBTSTATE: 5**

**+QBTSTATE: 0,1,"M66",8362618A0568**

**+QBTSTATE: 1,1,"M66",8362618A0568,"SPP"**

**OK**

## 2.9. SPP/HFP Connection

### 2.9.1. SPP Connection

#### 2.9.1.1. SPP Connection (AT Command Mode)

In AT command mode, SPP data sent to the module will be treated as AT command (remote AT).

**+QBTIND: "conn","GT-eeeeee",F0728C0D50AE,"SPP"**

**AT+QBTACPT=1,0**

//Connect to SPP in AT command mode

**OK**

**+QBTCONN: 1,2,"GT-eeeeee",F0728C0D50AE,"SPP"**

//Remote device:

**ATI**

**ATI**

**Quectel\_Ltd**

**Quectel\_M66**

**Revision: M66FAR01A05**

**OK**

### 2.9.1.2. SPP Connection (Buffer Access Mode)

In buffer access mode, SPP data will be buffered. URC will be reported to notify the arrival of data. AT commands are also provided to send and read SPP data.

#### 2.9.1.2.1 Read Data

In buffer access mode, after receiving data, the module will report URC **+QBTIND: "recv",<connected\_id>** to notify the host. Then the host can retrieve data via **AT+QSPPREAD=<connected\_id>,<readlength>**. Please note that if the buffer is not empty while the module receives data again, a new URC will not be reported until all the received data has been retrieved via **AT+QSPPREAD=<connected\_id>,<readlength>** from buffer.

```
+QBTIND: "recv",1           //Receive data notification via <connected_id>=1
AT+QSPPREAD=1,1500          //Read data via <connected_id>=1. The max length of the
                             data to be read is 1500 bytes
+QSPPREAD: 1500             //Read 1500 bytes
11.....xx

OK
AT+QSPPREAD=1,1500          //Continue to read
+QSPPREAD: 500              //Read 500 bytes only
11.....xx

OK
AT+QSPPREAD=1,1500          //Continue to read
+QSPPREAD: 0                //The buffer is empty, all data has been read out

OK
```

#### 2.9.1.2.2 Send Data

```
AT+QSPPSSEND=1,10           //Send 10 bytes via <connected_id>=1
>1234567890
OK
AT+QSPPSSEND=1,0            //Query the sent and unacknowledged data
+QSPPSSEND: 10,10,0

OK

.....
AT+QSPPSSEND=1,800          //Send 800 bytes via <connected_id>=1
>1234567890xxxxx...
+CME ERROR: 8020            //The buffer is full and the 800 bytes of data are not sent.
```

//When there is enough space in the buffer, the data can be sent again

```
AT+QSPSEND=1,200
>1234567890xxxxx...
OK
```

### 2.9.1.3. SPP Connection (Transparent Access Mode)

In transparent access mode, the corresponding UART port enters the exclusive mode. The data received from UART port will be sent to BT device directly, and the received data from BT device will be outputted to UART port directly as well. “+++” is used to exit from transparent access mode. When “+++” returns **OK**, the module will be switched to buffer access mode. **ATO** can be used to return back to transparent access mode. Please note that in transparent mode, the host cannot execute AT command. If BT connection is closed, the module will report **CLOSED** and exit from transparent access mode.

“+++” or DTR (**AT&D1** should be set) can be used to exit from transparent access mode. To prevent “+++” from being misinterpreted as data, the following sequence should be followed:

- 1) Do not input any character within T1 time (1 second) before inputting “+++”.
- 2) Input “+++”, time gap between each “+” cannot be longer than 20ms, and no other characters can be inputted during this time.
- 3) Do not input any character within T1 time (1 second) after “+++” has been inputted.
- 4) Exit from transparent access mode and **OK** is returned.

**ATO** will change the access mode of connection which exits from transparent access mode lately. If the connection enters the transparent access mode successfully, **CONNECT** will be returned. If there is no connection entering transparent access mode before, **ATO** will return NO CARRIER.

<b>AT+QBTCONN=1,0,2</b>	//Connect to paired device 1 in transparent access mode
<b>OK</b>	
<b>+QBTCONN: 1,1,"GT-e",F0728C0D50AE,"SPP"</b>	
<b>CONNECT</b>	//Enter transparent access mode
Aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	//Received data from remote BT
<b>+++</b>	//Send “+++” to exit from transparent access mode
<b>OK</b>	//Exit from transparent access mode without disconnecting
<b>ATO</b>	//Use <b>ATO</b> command to enter transparent mode
<b>CONNECT</b>	//Enter transparent mode again
Bbbbbbbbbbbbbbbbbbbbbb	//Received data
<b>CLOSED</b>	//Disconnected passively

```
+QBTIND: "disc",1,1,"GT-e",F0728C0D50AE,"SPP"

/***** accept connection *****/
+QBTIND: "conn","GT-e",F0728C0D50AE,"SPP"
AT+QBTACPT=1,2 //Accept connection in transparent access mode
OK

+QBTACPT: 1,2,"GT-e",F0728C0D50AE,"SPP" //URC notifying successful connection

CONNECT //Transparent access mode
+++ //Input "+++"
OK //Exit from transparent access mode
ATO //ATO command
CONNECT //Enter transparent access mode again
+++ //Input "+++"
OK //Exit from transparent access mode
AT+QBTDISCONN=2 //Disconnected proactively
OK

CLOSED

+QBTDISCONN: 2,1,"GT-e",F0728C0D50AE,"SPP"
```

## 2.9.2. HFP Connection

### 2.9.2.1. Module Acts as an AG Device

MC60, M66, M66 R2.0 and M66-DS support HFP profile, and the following connection sequence shows how the module acts as an AG device.

```
//Connect to BT headset first

AT+QBTSCAN
OK

+QBTSCAN: 1,"Motorola H270",00241C6743E9 //Found BT headset
AT+QBTSCANC //Stop scanning
+QBTSCAN: 0

OK
AT+QBTPAIR=1 //Pairing
OK
```

```
+QBTPAIRCNF: 1,1,0,"Motorola H270",00241C6743E9 //Paired successfully
AT+QBTCONN=1,6 //Connect to HFG profile
OK

+QBTCONN: 1,1,"A16S",00241C6743E9,"HFG_PROFILE" //Connected successfully

//If BT headset has been paired with the module before, it will connect to the module automatically when it
is turned on.

+QBTCONN: 1,1,"A16S",00241C6743E9,"HFG_PROFILE" //Connect to the module automatically

//When BT headset is connected to the module, the speech voice will output through the BT headset. The
speech voice output path can be set by AT+QBTHFGAUDIO.

AT+QBTHFGAUDIO=1 //Speech voice outputs from module side
OK

AT+QBTHFGAUDIO=0 //Speech voice outputs from HF device side
OK
```

#### 2.9.2.2. Module Acts as an HF Device

The following connection sequence shows how the module acts as an HF device.

```
AT+QBTSCAN
OK

+QBTSCAN: 1,"GT-e",F0728C0D50AE //Found the phone device
AT+QBTSCANC //Stop scanning
+QBTSCAN: 0

OK
AT+QBTPAIR=1 //Pairing
OK

+QBTPAIR: "GT-e",F0728C0D50AE,348531
AT+QBTPAIRCNF=1
OK

+QBTPAIRCNF: 1,1,1,"GT-e",F0728C0D50AE
AT+QBTCONN=1,5 //Connect to HF profile, and module acts as a HF device
OK

+QBTCONN: 1,1,"GT-e",F0728C0D50AE,"HF_PROFILE" //Connected successfully
```

//After connecting to AG device, the following command can be used to control the AG device (phone)

**AT+QBTATD=10086** //Call 10086

OK

+QBTIND: "ring",1,1,10086

+QBTIND: "ring",4,1,10086

**AT+QBTATH** //Hung up voice call

OK

+QBTIND: "ring",0,1,10086

//When a call is coming, the following URC will be output

+QBTIND: "ring",2,1,055165869386 //A call is coming

**AT+QBTATA** //Answer the call

OK

# 3 AT Commands for BT

## 3.1. AT Command Syntax

Test Command	AT+<x>=?	This command returns the list of parameters and value ranges set by corresponding Write Command or internal processes.
Read Command	AT+<x>?	This command returns currently set value of the parameter or parameters.
Write Command	AT+<x>=<...>	This command sets user-definable parameter values.
Execution Command	AT+<x>	This command reads non-variable parameters affected by internal processes in GSM engine.

## 3.2. Description of AT Commands

### 3.2.1. AT+QBTPWR Enable/Disable BT

AT+QBTPWR Enable/Disable BT	
Test Command AT+QBTPWR=?	Response +QBTPWR: (list of supported <on_off>s)  OK
Read Command AT+QBTPWR?	Response +QBTPWR: <power_status>  OK or +CME ERROR: 100
Write Command AT+QBTPWR=<on_off>	Response OK or +CME ERROR: 100

## Parameter

<on_off>	<u>0</u>	Disable
	1	Enable
<power_status>	<u>0</u>	BT function has been disabled
	1	BT function has been enabled

## Example

```
AT+QBTWPR=1 //Enable BT
OK
```

### 3.2.2. AT+QBTNAME Read/Write Local Name of BT Device

AT+QBTNAME Read/Write Local Name of BT Device	
Test Command <b>AT+QBTNAME=?</b>	Response <b>OK</b> or <b>ERROR</b>
Read Command <b>AT+QBTNAME?</b>	Response <b>+QBTNAME: &lt;device_name&gt;</b>  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>
Write Command <b>AT+QBTNAME=&lt;device_name&gt;</b>	Response <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

## Parameter

<device\_name> String type. Device name. Max length: 54 bytes.

## Example

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

OK

### 3.2.3. AT+QBTADDR Read Local Address of BT Device

#### AT+QBTADDR Read Local Address of BT Device

Test Command <b>AT+QBTADDR=?</b>	Response <b>OK</b>
Read Command <b>AT+QBTADDR?</b>	Response <b>+QBTADDR: &lt;addr&gt;</b>  <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

#### Parameter

<b>&lt;addr&gt;</b>	Device address. (e.g. A662616202C3: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
---------------------	---

#### Example

```
AT+QBTADDR?
+QBTADDR: A662616202C3
OK
```

### 3.2.4. AT+QBTVISB Set Visibility of BT

#### AT+QBTVISB Set Visibility of BT

Test Command <b>AT+QBTVISB=?</b>	Response <b>+QBTVISB: (list of supported &lt;n&gt;s),(list of supported &lt;time&gt;s)</b>  <b>OK</b>
Read Command <b>AT+QBTVISB?</b>	Response <b>+QBTVISB: &lt;n&gt;</b>  <b>OK</b>

	If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>
Write Command <b>AT+QBTVISB=&lt;n&gt;[,&lt;time&gt;]</b>	Response <b>OK</b>  If <n>=2: <b>OK</b>  <b>+QBTVISB: 0</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

### Parameter

<b>&lt;n&gt;</b>	0	Invisible
	<u>1</u>	Visible forever
	2	Visible temporarily; the device can be discovered by the other BT devices for a certain time
<b>&lt;time&gt;</b>	Integer type. The duration of BT being visible. Unit: second. Range: 1-255.	

### Example

```

AT+QBTVISB=0           //Invisible
OK
AT+QBTVISB=1
OK
AT+QBTVISB=2,60
OK
+QBTVISB: 0           //Become invisible after a timeout

```

### 3.2.5. AT+QBTSCAN Scan BT Devices

#### AT+QBTSCAN Scan BT Devices

Test Command <b>AT+QBTSCAN=?</b>	Response <b>+QBTSCAN:</b> (range of supported <timeout>s),(range of supported <device_Number>s),(range of supported <Cod>s)  <b>OK</b>
-------------------------------------	---

Read Command <b>AT+QBTSCAN?</b>	<p>Response</p> <p><b>+QBTSCAN: &lt;timeout&gt;,&lt;device_Number&gt;,&lt;Cod&gt;</b></p> <p><b>OK</b></p> <p>If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b></p>
Write Command <b>AT+QBTSCAN=&lt;timeout&gt;[,&lt;deviceNumber&gt;[,&lt;Cod&gt;]]</b>	<p>Response</p> <p><b>OK</b></p> <p><b>+QBTSCAN: &lt;searched_device_id&gt;,&lt;name&gt;,&lt;addr&gt;</b></p> <p>...</p> <p><b>+QBTSCAN: 0[,&lt;errorcode&gt;]</b></p> <p>If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b></p>
Execution Command <b>AT+QBTSCAN</b>	<p>Response</p> <p><b>OK</b></p> <p><b>+QBTSCAN: &lt;searched_device_id&gt;,&lt;name&gt;,&lt;addr&gt;</b></p> <p>...</p> <p><b>+QBTSCAN: 0[,&lt;errorcode&gt;]</b></p> <p>If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b></p>

## Parameter

<b>&lt;timeout&gt;</b>	1-255	Max time of searching BT devices. Unit: second. Default: 60.
<b>&lt;searched_device_id&gt;</b>	0	Searching is completed.
	1-20	Assign an ID for searched device.
<b>&lt;device_Number&gt;</b>	1- <u>20</u>	The max number of devices allowed to be searched.
<b>&lt;Cod&gt;</b>	<u>0</u> -4294967295	The Class of Device/Service (CoD). For more details, please refer to BT specification.
<b>&lt;name&gt;</b>	String type. Device name.	
<b>&lt;addr&gt;</b>	Device address.	

## Example

```
AT+QBTSCAN=60,10,0
OK
```

+QBTSCAN: 1,"BILLHE-PC",065C3A2F40F0

+QBTSCAN: 2,"bt\_dev1",016261E65467

+QBTSCAN: 3,"2233",326261926A88

AT+QBTSCANC

+QBTSCAN: 0

OK

AT+QBTSCAN?

+QBTSCAN: 60,10,0

OK

### 3.2.6. AT+QBTSCANC Stop Scanning

#### AT+QBTSCANC Stop Scanning

Execution Command

AT+QBTSCANC

Response

OK

If there is any error relating to ME functionality:

+CME ERROR: <errorcode>

### 3.2.7. AT+QBTPAIR Pair BT Device

#### AT+QBTPAIR Pair BT Device

Test Command

AT+QBTPAIR=?

Response

+QBTPAIR: (list of supported <searched\_device\_id>s)

OK

Write Command

AT+QBTPAIR=<searched\_device\_id>

Response

OK

+QBTPAIR: <name>,<addr>[,<numeric\_compare>]

If the searched device does not respond:

+QBTPAIRCNF: 0,<addr>

If there is any error relating to ME functionality:

+CME ERROR: <errorcode>

## Parameter

<b>&lt;searched_device_id&gt;</b>	Searched device ID. Range: 1-20.
<b>&lt;name&gt;</b>	String type. Device name.
<b>&lt;addr&gt;</b>	Device address.
<b>&lt;numeric_compare&gt;</b>	6-byte random password in Numeric Comparison Mode.

### 3.2.8. AT+QBTPAIRCNF Confirm Pairing

#### AT+QBTPAIRCNF Confirm Pairing

Test Command <b>AT+QBTPAIRCNF=?</b>	Response <b>+QBTPAIRCNF:</b> (list of supported <op>s),(list of supported <passkey>s)  <b>OK</b>
Write Command <b>AT+QBTPAIRCNF=&lt;op&gt;[,&lt;passkey&gt;]</b>	Response <b>OK</b>  <b>+QBTPAIRCNF:</b> <result>,<paired_device_id>,<is_first>,<name>,<addr>  If the pairing failed: <b>+QBTPAIRCNF: 0,&lt;addr&gt;</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

## Parameter

<b>&lt;op&gt;</b>	0	Reject pairing
	1	Accept pairing
<b>&lt;passkey&gt;</b>	String type. Passkey.	
<b>&lt;result&gt;</b>	0	Failed
	1	Successful
<b>&lt;paired_device_id&gt;</b>	1-10	Allocated paired device ID.
<b>&lt;is_first&gt;</b>	It indicates whether it is the first time to pair.	
	0	No
	1	Yes
<b>&lt;name&gt;</b>	String type. It indicates device name.	
<b>&lt;addr&gt;</b>	Device address.	

### 3.2.9. AT+QBTUNPAIR Delete Paired BT Device

#### AT+QBTUNPAIR Delete Paired BT Device

Test Command <b>AT+QBTUNPAIR=?</b>	Response <b>+QBTUNPAIR:</b> (range of supported <paired_device_id>s)  <b>OK</b>
Write Command <b>AT+QBTUNPAIR=&lt;paired_device_id&gt;</b>	Response <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

#### Parameter

<paired_device_id>	1-10	Paired device ID.
--------------------	------	-------------------

### 3.2.10. AT+QBTSTATE Query Device Status

#### AT+QBTSTATE Query Device Status

Test Command <b>AT+QBTSTATE=?</b>	Response <b>+QBTSTATE:</b> (list of supported <state>s)  <b>OK</b>
Execution Command <b>AT+QBTSTATE</b>	Response <b>+QBTSTATE: &lt;state&gt;</b> <b>+QBTSTATE: 0,&lt;paired_device_id&gt;,&lt;name&gt;,&lt;addr&gt;</b> ... <b>+QBTSTATE: 1,&lt;connected_id&gt;,&lt;name&gt;,&lt;addr&gt;,&lt;profile_name&gt;</b> ... <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

## Parameter

<state>	0	Initial on
	1	Deactivating
	2	Activating
	5	Idle
	6	Scanning
	7	Scan result indication
	8	Scan cancelling
	9	Initiate pairing
	12	Connecting
	13	Unpairing
	14	Deleting paired device
	15	Deleting all
	16	Disconnecting
	19	Numeric confirmation
	20	Numeric confirmation response
	25	Confirmation Indication of accepting connection
	26	Service refreshing
	29	Device name setting
	30	Authentication setting
	31	Release all connections
	36	Activating profiles
<paired_device_id>	1-10	Paired device ID.
<connected_id>	0-2	Connected device ID.
<name>	String type. Device name.	
<addr>	Device address.	
<profile_name>	String type. Profile name.	
	"SPP"	
	"OBEX_PBA_PROFILE_CLIENT"	
	"OBEX_PBA_PROFILE"	
	"OBEX_OBJECT_PUSH_SERVICE"	
	"OBEX_OBJECT_PUSH_CLIENT"	
	"HF_PROFILE"	
	"HFG_PROFILE"	

### 3.2.11. AT+QBTGPROF Check Supported Profiles

#### AT+QBTGPROF Check Supported Profiles

Test Command	Response
AT+QBTGPROF=?	OK
	ERROR

Write Command <b>AT+QBTGPROF=&lt;paired_device_id&gt;</b>	Response <b>+QBTGPROF: &lt;profile_id&gt;,&lt;profile_name&gt;</b> ... <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>
--	--

## Parameter

<b>&lt;paired_device_id&gt;</b>	1-10	Paired device ID.
<b>&lt;profile_id&gt;</b>	0	"SPP",0x1101
	1	"OBEX_PBA_PROFILE_CLIENT",0x112E
	2	"OBEX_PBA_PROFILE",0x112F
	3	"OBEX_OBJECT_PUSH_SERVICE",0x1105
	4	"OBEX_OBJECT_PUSH_CLIENT",0xfffd
	5	"HF_PROFILE",0x111E
	6	"HFG_PROFILE",0x111F
<b>&lt;profile_name&gt;</b>		String type. Profile name. "SPP" "OBEX_PBA_PROFILE_CLIENT" "OBEX_PBA_PROFILE" "OBEX_OBJECT_PUSH_SERVICE" "OBEX_OBJECT_PUSH_CLIENT" "HF_PROFILE" "HFG_PROFILE"

### NOTE

The command returns the profiles supported by both the local device and the device on other side.

## 3.2.12. AT+QBTCONN Connect with BT Device

### AT+QBTCONN Connect with BT Device

Test Command <b>AT+QBTCONN=?</b>	Response <b>+QBTCONN: (range of supported &lt;paired_device_id&gt;s),(list of supported &lt;profile_id&gt;s),(list of supported &lt;mode&gt;s)</b>  <b>OK</b>
Write Command <b>AT+QBTCONN=&lt;paired_device_id&gt;,&lt;</b>	Response <b>OK</b>

profile\_id>[,<mode>]

**+QBTCONN:** <result>,<connected\_id>,<name>,<addr>,<profile\_name>

If there is any error relating to ME functionality:

**+CME ERROR:** <errorcode>

## Parameter

<result>	0	Failed
	1	Successful
<paired_device_id>	1-10	Paired device ID
<profile_id>	0	"SPP",0x1101
	1	"OBEX_PBA_PROFILE_CLIENT",0x112E
	2	"OBEX_PBA_PROFILE",0x112F
	3	"OBEX_OBJECT_PUSH_SERVICE",0x1105
	4	"OBEX_OBJECT_PUSH_CLIENT",0xfffd
	5	"HF_PROFILE",0x111E
	6	"HFG_PROFILE",0x111F
<mode>	Modes used for SPP connection.	
	0	AT command mode.
	1	Buffer Access Mode.
	2	Transparent Access Mode.
<connected_id>	0-2	Connected device ID.
<name>	String type. Device name.	
<addr>	Device address.	
<profile_name>	String type. Profile name.	
	"SPP"	
	"OBEX_PBA_PROFILE_CLIENT"	
	"OBEX_PBA_PROFILE"	
	"OBEX_OBJECT_PUSH_SERVICE"	
	"OBEX_OBJECT_PUSH_CLIENT"	
	"HF_PROFILE"	
	"HFG_PROFILE"	

### NOTE

The access mode cannot be changed when SPP is successfully connected; and SPP should be disconnected first if the mode needs to be changed.

### 3.2.13. AT+QBTCONND Connect to SPP Directly Using BT Device Address

#### AT+QBTCONND Connecting to SPP Directly Using BT Device Address

Test Command <b>AT+QBTCONND=?</b>	Response <b>+QBTCONND:</b> (list of supported <addr>s),(list of supported <mode>s),(list of supported <password>s)  <b>OK</b>
Write Command <b>AT+QBTCONND=&lt;addr&gt;[,&lt;mode&gt;[,&lt;password&gt;]]</b>	Response <b>OK</b>  <b>+QBTCONN:</b> <result>,<connected_id>,<name>,<addr>,<profile_name>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

#### Parameter

<result>	0 Failed 1 Successful
<mode>	Modes used for SPP connection 0 AT command mode 1 Buffer Access Mode 2 Transparent Access Mode
<connected_id>	0-2 Connected device ID
<name>	String type. Device name.
<addr>	Device address (double quotes are not required) Sequence: Nap, UAP, LAP
<password>	String type. Password in PIN Code pairing (double quotes are required)
<profile_name>	String type. Profile name. "SPP" "OBEX_PBA_PROFILE_CLIENT" "OBEX_PBA_PROFILE" "OBEX_OBJECT_PUSH_SERVICE" "OBEX_OBJECT_PUSH_CLIENT" "HF_PROFILE" "HFG_PROFILE"

## NOTE

Direct SPP connection only needs to use BT address with the sequence of NAP-UAP-LAP. Either “PIN Code pairing (with password) as originating party” or “SSP pairing (Numeric Comparison Mode) as originating party” is internally applied in this connection, so users do not need to conduct scanning or concern the pairing process.

## Example

**AT+QBTADDR?**

**+QBTADDR: 02C362A66261**

OK

**AT+QBTCONND=4D7BDF076261,1,"0000"**

OK

**+QBTIND: "pair","12",4D7BDF076261,167831** //First time to connect

**+QBTPAIRCNF: 1,1,1,"12",4D7BDF076261**

**+QBTCONN: 1,1,"12",4D7BDF076261,"SPP"** //Connected successfully

**AT+QSPSEND=1,94** //Send data to print

>

!"#\$%&'()\*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^\_`abcdefghijklmnopqrstuvwxyz{|}~

OK

**AT+QBTDISCONN**

//In this condition, users can use this AT Command without a connected device ID.

OK

**+QBTDISCONN: 1,1,"12",4D7BDF076261,"SPP"**

## 3.2.14. AT+QBTACPT Accept Connection

### AT+QBTACPT Accept Connection

Test Command  
**AT+QBTACPT=?**

Response  
**+QBTACPT: (list of supported <op>s)**

OK

Write Command  
**AT+QBTACPT=<op>[,<mode>]**

Response  
OK

**+QBTA**CPT: <result>,<connected\_id>,<name>,<addr>,<profile\_name>

If there is any error relating to ME functionality:

**+CME ERROR**: <errorcode>

## Parameter

<op>	0	Reject connection
	1	Accept connection
<result>	0	Failed
	1	Successful
<mode>	Modes used for SPP connection.	
	0	AT command mode
	1	Buffer Access Mode
	2	Transparent Access Mode
<connected_id>	0-2	Connected device ID.
<name>	String type. Device name.	
<addr>	Device address.	
<profile_name>	String type. Profile name.	
	"SPP"	
	"OBEX_PBA_PROFILE_CLIENT"	
	"OBEX_PBA_PROFILE"	
	"OBEX_OBJECT_PUSH_SERVICE"	
	"OBEX_OBJECT_PUSH_CLIENT"	
	"HF_PROFILE"	
	"HFG_PROFILE"	

### 3.2.15. AT+QBTDISCONN Disconnection

#### AT+QBTDISCONN Disconnection

Test Command <b>AT+QBTDISCONN=?</b>	Response <b>+QBTDISCONN</b> : (list of supported <connected_id>s)
	<b>OK</b>
Write Command <b>AT+QBTDISCONN=&lt;connected_id&gt;</b>	Response <b>OK</b>
	<b>+QBTDISCONN</b> : <connected_id>,<paired_device_id>,<name>,<addr>,<profile_name>
If there is any error relating to ME functionality:	

**+CME ERROR: <errorcode>**

## Parameter

<b>&lt;connected_id&gt;</b>	Connected device ID. 0 Connection 0 1 Connection 1 2 Connection 2
<b>&lt;paired_device_id&gt;</b>	Paired device ID.
<b>&lt;name&gt;</b>	String type. Device name.
<b>&lt;addr&gt;</b>	Device address.
<b>&lt;profile_name&gt;</b>	String type. Profile name. "SPP" "OBEX_PBA_PROFILE_CLIENT" "OBEX_PBA_PROFILE" "OBEX_OBJECT_PUSH_SERVICE" "OBEX_OBJECT_PUSH_CLIENT" "HF_PROFILE" "HFG_PROFILE"

### 3.2.16. AT+QSPSEND Send Data

#### AT+QSPSEND Send Data

Test Command  
**AT+QSPSEND=?**

Response:  
**+QSPSEND:** (list of supported <connected\_id>s)[,(list of supported <send\_length>s)]

**OK**

Write Command  
Send data with changeable length  
**AT+QSPSEND=<connected\_id>**  
Response ">", then type data to send, tap "CTRL+Z" to send, tap "ESC" to cancel the operation

Response:  
  
If the connection has been established and the sending is successful:  
**OK**

If the connection has been established but the sending buffer does not have enough space:  
**+CME ERROR: 8020**

If the connection has not been established, abnormally closed, or the parameter is incorrect:  
**+CME ERROR: <errorcode>**

<p>Write Command</p> <p>Send data with fixed length</p> <p><b>AT+QSPSEND=&lt;connected_id&gt;,&lt;send_length&gt;</b></p> <p>Response "&gt;", type data until the data length is equal to &lt;send_length&gt;</p>	<p>Response:</p> <p>If the connection has been established and sending is successful:</p> <p><b>OK</b></p> <p>If the connection has been established but the sending buffer does not have enough space:</p> <p><b>+CME ERROR: 8020</b></p> <p>If the connection has not been established, abnormally closed, or the parameter is incorrect:</p> <p><b>+CME ERROR: &lt;errorcode&gt;</b></p>
<p>Write Command</p> <p>When &lt;send_length&gt; is 0, query the sent data</p> <p><b>AT+QSPSEND=&lt;connected_id&gt;,0</b></p>	<p>Response</p> <p>If the specified connection has existed:</p> <p><b>+QSPSEND:&lt;total_send_length&gt;,&lt;sent_bytes&gt;,&lt;left_bytes&gt;</b></p> <p><b>OK</b></p> <p>Or</p> <p><b>+CME ERROR: &lt;errorcode&gt;</b></p>

## Parameter

<connected_id>	Connected device ID.
<send_length>	Integer type. Length of data to be sent, which cannot exceed 1460.
<total_send_length>	Integer type. Total length of the sent data.
<sent_bytes>	Integer type. Length of the data already sent.
<left_bytes>	Integer type. Length of the remaining data.

### 3.2.17. AT+QSPREAD Read Data

<b>AT+QSPREAD Read Data</b>	
<p>Test Command</p> <p><b>AT+QSPREAD=?</b></p>	<p>Response:</p> <p><b>+QSPREAD: (list of supported &lt;connected id&gt;s),(range of supported &lt;read_length&gt;)</b></p> <p><b>OK</b></p>
<p>Write Command</p> <p><b>AT+QSPREAD=&lt;connected_id&gt;,&lt;read_length&gt;</b></p>	<p>Response</p> <p>If the specified channel has received data:</p> <p><b>+QSPREAD: &lt;read_actual_length&gt;&lt;CR&gt;&lt;LF&gt;&lt;data&gt;</b></p>

	OK If there is no data: <b>+QSPPREAD: 0</b>
	OK  If the connection does not exist: <b>+CME ERROR: &lt;errorcode&gt;</b>
Write Command When <b>&lt;read_length&gt;</b> is 0, query retrieved data via <b>AT+QSPPREAD=&lt;connected_id&gt;,0</b>	Response: If the specified connection has existed: <b>+QSPPREAD: &lt;total_receive_length&gt;,&lt;have_read_length&gt;,&lt;unread_length&gt;</b>  OK Or <b>+CME ERROR: &lt;errorcode&gt;</b>

## Parameter

<b>&lt;connected_id&gt;</b>	Connected device ID.
<b>&lt;read_length&gt;</b>	Max length of data to be retrieved with a range of 0-1500.
<b>&lt;total_receive_length&gt;</b>	Total length of the received data.
<b>&lt;have_read_length&gt;</b>	Length of the received data that is already read.
<b>&lt;unread_length&gt;</b>	Length of the received data that is unread.

### 3.2.18. AT+QBTATA Answer Incoming Call

#### AT+QBTATA Answer Incoming Call

Execution Command <b>AT+QBTATA</b>	Response <b>OK</b> or <b>+CME ERROR: &lt;errorcode&gt;</b>
---------------------------------------	---

### 3.2.19. AT+QBTATH Hung up Voice Call

#### AT+QBTATH Hung up Voice Call

Execution Command <b>AT+QBTATH</b>	Response <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>
---------------------------------------	--

### 3.2.20. AT+QBTATD Dial Voice Call

#### AT+QBTATD Dial Voice Call

Write Command  
**AT+QBTATD=<number>**

Response  
**OK**

If there is any error relating to ME functionality:  
**+CME ERROR: <errorcode>**

#### Parameter

**<number>** Phone number.  
Digits and symbols that can be dialed include: \*, #, 0~9

### 3.2.21. AT+QBTATDL Redial Last Phone Number

#### AT+QBTATDL Redial Last Phone Number

Execution Command  
**AT+QBTATDL**

Response  
**OK**

If there is any error relating to ME functionality:  
**+CME ERROR: <errorcode>**

### 3.2.22. AT+QBTHFGAUDIO Select Audio Output Path While Connecting with an HF Device

#### AT+QBTHFGAUDIO Select Audio Output Path While Connecting with an HF Device

Test Command  
**AT+QBTHFGAUDIO=?**

Response  
**+QBTHFGAUDIO: (list of supported <path>s)**  
  
**OK**

Read Command  
**AT+QBTHFGAUDIO?**

Response  
**+QBTHFGAUDIO: <path>**  
  
**OK**

Write Command  
**AT+QBTHFGAUDIO=<path>**

Response  
  
**OK**

If there is any error relating to ME functionality:  
**+CME ERROR: <errorcode>**

#### Parameter

<b>&lt;path&gt;</b>	Audio output path when the module is connected with an HF device.
0	Audio outputs from the HF device side
1	Audio outputs from the module side

### 3.2.23. AT+QBTOPPPUSH OPP Client Pushes Files

#### AT+QBTOPPPUSH OPP Client Pushes Files

Test Command <b>AT+QBTOPPPUSH=?</b>	Response <b>+QBTOPPPUSH:</b> (range of supported <paired_id>s),<file 1>[,<file2>[,...[,<file5>]]]  <b>OK</b>
Write Command <b>AT+QBTOPPPUSH=&lt;paired_id&gt;,&lt;file 1&gt;[,&lt;file2&gt;[,...[,&lt;file5&gt;]]]</b>	Response <b>OK</b>  If there is any error relating to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>

#### Parameter

<b>&lt;paired_id&gt;</b>	1-10 Paired device ID.
<b>&lt;file1&gt;</b>	First file to be sent. The complete file path should be provided (double quotes are required).
<b>&lt;file2&gt;...&lt;file5&gt;</b>	Files to be sent (double quotes are required). If several files are sent and some of the file paths are not valid, the module will try to send valid file paths. The valid file path will be reported in URC <b>+QBTOPPPUSH</b> . At most 5 files can be sent.

### 3.2.24. AT+QBTOPPACPT OPP Server Decides Whether to Accept Push Request from OPP Client

#### AT+QBTOPPACPT OPP Server Decides Whether to Accept Push Request from OPP Client

Test Command <b>AT+QBTOPPACPT=?</b>	Response <b>+QBTOPPACPT:</b> (list of supported <op>s),(list of supported <path>s)
--	---

	OK
Write Command <b>AT+QBTOPPACPT=&lt;op&gt;,&lt;path&gt;</b>	Response OK  <b>+QBTOPPACPT: &lt;result &gt;,&lt;file&gt;</b>

### Parameter

<b>&lt;op&gt;</b>	0 Reject 1 Accept
<b>&lt;path&gt;</b>	File path, which can be UFS, RAM and SD card. 1 RAM 2 UFS 3 SD card
<b>&lt;file&gt;</b>	File sent by the OPP client.

### 3.2.25. AT+QBTCFG Configure BT Pairing Mode

The pairing modes consist of Pin Code pairing mode and SSP mode (SSP mode by default). For Pin Code pairing mode, password entering is required. **AT+QBTCFG="pair"** is used to configure the BT pairing mode.

<b>AT+QBTCFG Configure BT Pairing Mode</b>	
Test Command <b>AT+QBTCFG=?</b>	Response <b>+QBTCFG: "&lt;config_type&gt;"[(list of supported &lt;pair_mode&gt;s)]</b>  OK
Read Command <b>AT+QBTCFG?</b>	Response <b>+QBTCFG: "pair",&lt;pair_mode&gt;</b>  OK
Write Command <b>AT+QBTCFG="pair",&lt;pair_mode&gt;</b>	Response OK  If there is any error related to ME functionality: <b>+CME ERROR: &lt;errorcode&gt;</b>
Maximum Response Time	300ms
Reference	

## Parameter

<config_type>	Configuration option. "pair" Used to configure pairing mode
<pair_mode>	BT pairing mode. 0 Pin Code pairing mode 1 SSP mode

## NOTES

1. This command should be executed before turning on BT with **AT+QBTPWR=1**.
2. It is necessary to reset the command after the module is restarted.

## Example

```

AT+QBTCFG=? //Test mode of AT+QBTCFG
+QBTCFG: "pair",(0,1)

OK

AT+QBTCFG? //Query the current values of BT configurations
+QBTCFG: "pair",1

OK

AT+QBTCFG="pair",0 //Set pairing mode (Pin Code pairing mode)
OK

AT+QBTCFG="pair" //Get the current value of BT pairing mode
+QBTCFG: "pair",0

OK

```

### 3.3. Description of URCs

#### 3.3.1. +QBTIND: "pair" Notify Pairing

##### +QBTIND: "pair" Notify Pairing

+QBTIND: "pair",<name>,<addr>  
[,<numeric\_compare>]

The URC will be reported when the pairing request is initiated by the other device.

##### Parameter

<name>	String type. Device name.
<addr>	Device address
<numeric_compare>	6-byte random password in Numeric Comparison Mode

##### NOTE

When receiving a pairing indication:

Use **AT+QBTPAIRCNF=1** without passkey if there is a <numeric\_compare> parameter.

Use **AT+QBTPAIRCNF=1,<passkey>** with passkey if there is no <numeric\_compare> parameter.

#### 3.3.2. +QBTIND: "conn" Notify Connection

##### +QBTIND: "conn" Notify Connection

+QBTIND: "conn",<name>,<addr>,<profile\_name>

The URC will be reported when a connection request is initiated by device on the other side.

##### Parameter

<name>	String type. Device name.
<addr>	Device address.
<profile_name>	String type. Profile name. "SPP" "OBEX_PBA_PROFILE_CLIENT" "OBEX_PBA_PROFILE" "OBEX_OBJECT_PUSH_SERVICE" "OBEX_OBJECT_PUSH_CLIENT" "HF_PROFILE" "HFG_PROFILE"

### 3.3.3. +QBTIND: "disc" Notify Disconnection

#### +QBTIND: "disc" Notify Disconnection

**+QBTIND: "disc",<connected\_id>,<paired\_device\_id>,<name>,<addr>,<profile\_name>**

This URC will be reported when disconnecting the module and the paired device while disconnecting command is not proactively executed, or when failing to connect while proactively initiating the connection.

#### Parameter

<b>&lt;connected_id&gt;</b>	Connected device ID
<b>&lt;paired_device_id&gt;</b>	Paired device ID
<b>&lt;name&gt;</b>	String type. Device name.
<b>&lt;addr&gt;</b>	Device address.
<b>&lt;profile_name&gt;</b>	String type. Profile name. "SPP" "OBEX_PBA_PROFILE_CLIENT" "OBEX_PBA_PROFILE" "OBEX_OBJECT_PUSH_SERVICE" "OBEX_OBJECT_PUSH_CLIENT" "HF_PROFILE" "HFG_PROFILE"

### 3.3.4. +QBTIND: "recv" Notify Incoming Data

In buffer access mode, after receiving data, URC **+QBTIND: "recv",<connected\_id>** will be reported to notify the host. Then host can retrieve data via **AT+QSPPREAD=<connected\_id>,<read\_length>**. Please note that if the buffer is not empty while the module receives data again, a new URC will not be reported until all the received data has been retrieved via **AT+QSPPREAD=<connected\_id>,<read\_length>** from the buffer.

#### +QBTIND: "recv" Notify Incoming Data

**+QBTIND: "recv",<connected\_id>**

This URC notifies the incoming data in the buffer access mode. And the data can be retrieved via **AT+QSPPREAD**.

#### Parameter

<b>&lt;connected_id&gt;</b>	Connected device ID.
-----------------------------	----------------------

### 3.3.5. +QBTIND: "ring" Notify Incoming Call

#### +QBTIND: "ring" Call Indication

**+QBTIND: ring",<call\_state>,<call\_id>,<number>** This URC notifies incoming call when the module acts as a HF device.

#### Parameter

<b>&lt; call_state &gt;</b>	The call state.
0	Idle state
1	Outgoing state
2	Incoming state
4	Active state
8	Hold state
<b>&lt;call_id&gt;</b>	The call ID.
<b>&lt;number&gt;</b>	The call number.

### 3.3.6. +QBTVGS Notify When Volume of AG Speaker is Changed

#### +QBTVGS Notify When Volume of AG Speaker is Changed

**+QBTVGS: <volume>** This URC notifies when the volume of AG speaker is changed, and when the volume of the HF speaker synchronizes with that of AG speaker.

#### Parameter

<b>&lt;volume&gt;</b>	Volume level. Range: 0-15
0	Mute
15	Max volume

### 3.3.7. +QHFCME Notify When Changing of Call State is Finished

#### +QHFCM Notify When Changing of Call State is Finished

**+QHFCME: <result>** This URC notifies when the changing of call state is finished, and usually is reported when the execution of HFP AT command is failed.

## Parameter

<result>	0-254	Failed
	255	Successful

### 3.3.8. +QBTOPPPUSH Notify When File Sending is Finished

#### +QBTOPPPUSH Notify When File Sending is Finished

**+QBTOPPPUSH: <result>,<file>** This URC notifies when file sending is finished.

## Parameter

<result>	1	Successful
	0	Failed
<file>	File that has been sent.	

## Example

```
AT+QBTOPPPUSH=1,"UFS/test.txt"
OK
+QBTOPPPUSH: 1,"UFS/test.txt"
```

### 3.3.9. +QBTIND: "OPPPrecv" Notify When OPP Server Receives Send Request from OPP Client

#### +QBTIND: "OPPPrecv" Notify When OPP Server Receives Send Request from OPP Client

**+QBTIND: "OPPPrecv",<addr>,<bt\_name>,<file>** This URC notifies when OPP server receives send request from OPP client

## Parameter

<addr>	BT address of the OPP client.
<bt_name>	BT name of the OPP client.
<file>	File sent by the OPP client.

### 3.3.10. +QBTOPPACPT Notify When OPP Server Finishes Receiving File

#### **+QBTOPPACPT Notify When OPP Server Finishes Receiving File**

**+QBTOPPACPT: <result>,<file>**

This URC notifies when OPP server finishes receiving file

#### Parameter

<result>	1	Successful
	0	Failed
<file>	File received from the OPP server.	

#### Example

```
+QBTIND: "OPPrecv", 4CC9DEA96261,"QUECTEL-BT","test.txt"  
AT+QBTOPPACPT=1,1  
OK  
  
+QBTOPPACPT: 1,"test.txt"
```

# 4 Appendix A Reference

**Table 1: Related Documents**

SN	Document Name	Remark
[1]	GSM 07.07	Digital cellular telecommunications (Phase 2+); AT command set for GSM Mobile Equipment (ME)
[2]	GSM 07.10	Support GSM 07.10 multiplexing protocol
[3]	Specification of the Bluetooth System	Bluetooth Specification from <a href="http://www.bluetooth.com">Http://www.bluetooth.com</a>

**Table 2: Terms and Abbreviations**

Abbreviation	Description
AG	Audio Gateway
BT	Bluetooth
HF	Hands-Free Unit
HFP	Hands-Free Profile
RAM	Random Access Memory
SPP	Serial Port Profile
SSP	Simple Safe Pair
UFS	Universal Flash Storage
URC	Unsolicited Result Code
OPP	Object Push Profile
SDAP	Service Discovery Application Profile
GOEP	Generic Object Exchange Profile
(U)SIM	(Universal) Subscriber Identity Module

# 5 Appendix B Summary of <errorcode>

**Table 3: Different Coding Schemes of +CME ERROR: <errorcode>**

Code of <errorcode>	Meaning
4	Operation not supported
8002	Device ID invalid
8003	Busy in scanning
8004	Parameter invalid
8005	Busy in pairing
8006	Already paired
8007	Scanning error
8008	Power request failed
8009	Setting request failed
8010	Request of reading address failed
8011	Request of setting visibility failed
8012	Scanning request failed
8013	Request of scan cancelling failed
8014	Pairing request failed
8015	Request of pairing confirming failed
8016	Unpairing request failed
8017	Request of checking profiles failed
8018	Connecting request failed
8019	Request of connection accepting failed

8020	BT buffer full
8021	BT not connected
8022	BT memory wrong
8023	The data exceeds the specified length
8024	BT disconnected
8101	Disconnecting request failed
8102	Request of checking device status failed
8107	Request of <b>AT+QBTHFATD</b> failed
8108	Request of <b>AT+QBTHFATA</b> failed
8109	Request of <b>AT+QBTHFATH</b> failed
8110	Request of <b>AT+QBTCLCC</b> failed
8111	Request of <b>AT+QBTHFRVC</b> failed
8201	BT already connected