

M203C BLE

AT Commands Manual

GSM/GPRS/GNSS Module Series

Rev. M203C_BLE_AT_Commands_Manual_V1.0

Contents

About the Document	3
Contents	4
Table Index	5
1 Introduction	6
2 BLE AT Commands Description	7
2.1. AT+QBT_PWR Power on/off BT	7
2.2. AT+QBTLEADDR Read Local Address of BLE Device	8
2.3. AT+QBTGATSREG Register/Deregister a GATT Server	8
2.4. AT+QBTGATSS Add/Remove a Service	9
2.5. AT+QBTGATSC Add a Characteristic to an Existing Service	11
2.6. AT+QBTGATSD Add a Descriptor to an Existing Service	12
2.7. AT+QBTGATSST Start/Stop a Service	13
2.8. AT+QBTGATSL Start/Stop Advertising	14
2.9. AT+QBTGATSIND Send an Indication or Notification to a Client	15
2.10. AT+QBTGATSRSP Send a Response to a Client's Read or Write Operation	16
2.11. AT+QBTFMPSREG* Register/Deregister an FMP Service	17
2.12. AT+QBTPXPSREG* Register/Deregister a PXP Service	17
2.13. AT+QBTGATADV Set Advertising Parameters	18
2.14. AT+QBTGATCPU Update Connection Parameters	19
2.15. AT+QBTGATSDISC Disconnect Server Actively	19
2.16. AT+QBTLETXPWR Set the Level of BLE Transmission Power	20
3 Description of URC	22
3.1. Notify the GATT Connection or Disconnection	22
3.2. Notify the GATT Client's Write Request	22
3.3. Notify the GATT Client's Read Request	23
3.4. Notify the PXP Connection*	24
3.5. Notify the PXP Link Loss Alert*	25
3.6. Notify the PXP Disconnection Alert*	25
3.7. Notify the FMP Connection or Disconnection*	26
3.8. Notify the FMP Client's Write Request*	26
4 Appendix A References	27

Table Index

TABLE 1: TERMS AND ABBREVIATIONS	27
TABLE 2: FORMAT MAP OF PROPERTIES AND PERMISSION	27
TABLE 3: DIFFERENT CODING SCHEMES OF +CME ERROR: <ERR>	28

1 Introduction

This document presents the BLE AT Commands Set for M203C module.

NOTE

The AT commands, parameter meanings and URCs marked with “*” in this document are still under development.

2 BLE AT Commands Description

2.1. AT+QBTPWR Power on/off BT

AT+QBTPWR Power on/off BT	
Test Command AT+QBTPWR=?	Response: +QBTPWR: (list of supported <on_off>s) OK
Read Command AT+QBTPWR?	Response: +QBTPWR: <power status> OK
Write Command AT+QBTPWR=<on_off>	Response: OK If there is any error relating to ME functionality, response: +CME ERROR: <err>

Parameter

<on_off>	0	Off
	1	On
<power status>	0	Power off
	1	Power on

Example

AT+QBTPWR=1	//Power on BT.
OK	

2.2. AT+QBTLEADDR Read Local Address of BLE Device

AT+QBTLEADDR Read Local Address of BLE Device

Test Command AT+QBTLEADDR=?	Response OK
Read Command AT+QBTLEADDR?	Response +QBTLEADDR: <ble_addr> OK If there is any error relating to ME functionality, response: +CME ERROR: <err>

Parameter

<ble_addr>	Local address of BLE device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
-------------------------	---

Example

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

2.3. AT+QBTGATSREG Register/Deregister a GATT Server

AT+ QBTGATSREG Register/Deregister a GATT Server

Test Command AT+QBTGATSREG=?	Response: +QBTGATSREG: (list of supported <op>s),<gserv_id> OK
Read Command AT+QBTGATSREG?	Response: +QBTGATSREG: <gserv_id>,<role>,<ctx_handle> ... OK
Write Command AT+QBTGATSREG=<op>,<gserv_id>	Response: +QBTGATSREG: <op>,<gserv_id>,<result>

OK

If there is any error relating to ME functionality, response:

+CME ERROR: <err>

Parameter

<op>	0	Deregister
	1	Register
<gserv_id>	User ID (or the name) of GATT server A Hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).	
<result>	0	SUCCESS
	1	FAIL
<role>	1	Server
	2	Client*
<ctx_handle>	Handle of registered GATT server	

Example

AT+QBTGATSREG=1,"ABC2" //Register a GATT server.

+QBTGATSREG: 1,"ABC2",0

OK

2.4. AT+QBTGATSS Add/Remove a Service

AT+ QBTGATSS Add/Remove a Service

Test Command
AT+QBTGATSS=?

Response:

+QBTGATSS:

(list of supported <op>s),<gserv_id>,<service_uuid>,
(list of supported <num_handles>s),(list of supported
<is_primary>s),(list of supported <inst>s)

OK

Read Command
AT+QBTGATSS?

Response:

+QBTGATSS:

<gserv_id>,<service_uuid>,<num_handles>,<service_ha
ndle>,<is_primary>,<inst>,<is_started>

...

	OK
Write Command Add a service AT+QBTGATSS=1,<gserv_id>,<service_uuid>,<num_handles>,<is_primary>,<inst>	Response: +QBTGATSS: 1,<gserv_id>,<service_uuid>,<is_primary>,<result>,<inst>,<service_handle> OK If there is any error relating to ME functionality, response: +CME ERROR: <err>
Write Command Remove a service AT+QBTGATSS=0,<gserv_id>,<service_handle>	Response: +QBTGATSS: 0,<gserv_id>,<result>,<service_handle> OK If there is any error relating to ME functionality, response: +CME ERROR: <err>

Parameter

<op>	0 Remove 1 Add
<gserv_id>	User ID (or the name) of GATT server A Hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end to be even).
<service_uuid>	UUID of this service. A string with HEX value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<num_handles>	Number of handles of this service. The range is 1-65535.
<is_primary>	0 Not primary service 1 Primary service
<inst>	Instance ID of this service. The range is 0-255.
<service_handle>	Handle of this service. The range is 0-65535.
<result>	0 SUCCESS 1 FAIL
<is_started>	0 Service is in stop state 1 Service is in start state

Example

```
AT+QBTGATSS=1,"ABC2","3418",5,1,254 //Add a service.
+QBTGATSS: 1,"ABC2","3418",1,0,254,256
```


OK

AT+QBTGATSS=0,"ABC2",256
+QBTGATSS: 0,"ABC2",0,256

//Remove a service.

OK

2.5. AT+QBTGATSC Add a Characteristic to an Existing Service

AT+QBTGATSC Add a Characteristic to an Existing Service

Test Command
AT+QBTGATSC=?

Response:

+QBTGATSC: 1,<gserv_id>,(list of supported
<service_handle>s),<char_uuid>,(list of supported
<inst>s),(list of supported <prop>s),(list of supported
<permission>s)

OK

Write Command
AT+QBTGATSC=1,<gserv_id>,<service_handle>,<char_uuid>,<inst>,<prop>,<permission>

Response:

+QBTGATSC:
1,<gserv_id>,<service_handle>,<char_uuid>,<inst>,<result>,<char_handle>

OK

If there is any error relating to ME functionality, response:
+CME ERROR: <err>

Parameter

<gserv_id>	User ID (or the name) of GATT server A Hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<service_handle>	Handle of this service. The range is 0-65535.
<char_uuid>	UUID of this characteristic. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<inst>	Instance ID of this characteristic. The range is 0-255.
<prop>	Properties of this characteristic. For more details, please refer to Table 2 .
<permission>	Permission of this characteristic. For more details, please refer to Table 2 .

<char_handle>	Handle of this characteristic.	
<result>	0	SUCCESS
	1	FAIL

Example

```
AT+QBTGATSC=1,"ABC2",256,"332a",2,58,17 //Add a characteristic.
+QBTGATSC: 1,"ABC2",256,"332A",2,0,260
OK
```

2.6. AT+QBTGATSD Add a Descriptor to an Existing Service

AT+ QBTGATSD Add a Descriptor to an Existing Service

Test Command AT+QBTGATSD=?	Response +QBTGATSD: 1,<gserv_id>,(list of supported <service_handle>s),<desc_uuid>,(list of supported <inst>s),(list of supported <permission>s) OK
Write Command AT+QBTGATSD=1,<gserv_id>,<service_handle>,<desc_uuid>,<inst>,<permission>	Response +QBTGATSD: 1, <gserv_id>,<service_handle>,<desc_uuid>,<inst>,<result>,<desc_handle> OK If there is any error relating to ME functionality, response: +CME ERROR: <err>

Parameter

<gserv_id>	User ID (or the name) of GATT server A Hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<desc_uuid>	UUID of this descriptor. A string with hex value (string should be included in quotation marks). The max length is 32, and the min length is 4 (if the length is odd, then fill '0' at the end of it to be even).
<inst>	Instance ID of this descriptor. The range is 0-255.
<permission>	Permission of this descriptor. For more details, please refer to Table 2 .
<desc_handle>	Handle of this descriptor. The range is 0-65535.
<service_handle>	Handle of this service. The range is 0-65535.

<result>	0	SUCCESS
	1	FAIL

Example

```
AT+QBTGATSD=1,"ABC2",256,"1329",1,17 //Add a descriptor.
+QBTGATSD: 1,"ABC2",256,"1329",1,0,262

OK
```

2.7. AT+QBTGATSST Start/Stop a Service

AT+ QBTGATSST Start/Stop a Service

Test Command AT+QBTGATSST=?	Response: +QBTGATSST: (list of supported <op>s),<gserv_id>, (list of supported <service_handle>s)[,(list of supported <transport>s)] OK
Write Command Start a service AT+QBTGATSST=1,<gserv_id>,<service_handle>,<transport>	Response: +QBTGATSS: 1,<gserv_id>,<result>,<service_handle> OK If there is any error relating to ME functionality, response: +CME ERROR: <err>
Write Command Stop a service AT+QBTGATSST=0,<gserv_id>,<service_handle>	Response: +QBTGATSST: 0,<gserv_id>,<result>,<service_handle> OK If there is any error relating to ME functionality, response: +CME ERROR: <err>

Parameter

<gserv_id>	User ID (or the name) of GATT server A Hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<service_handle>	Handle of this service. The range is 0-65535.

<transport>	Transport method to start this service.	
	0	LE
	1	BR/EDR*
	2	Dual*
<result>	0	SUCCESS
	1	FAIL

Example

```

AT+QBTGATSST=1,"ABC2",256,0           //Start a service.
+QBTGATSST: 1,"ABC2",0,256

OK

```

2.8. AT+QBTGATSL Start/Stop Advertising

AT+QBTGATSL Start/Stop Advertising

Test Command AT+QBTGATSL=?	Response: +QBTGATSL: <gserv_id>,(list of supported <op>s) OK
Write Command AT+QBTGATSL=<gserv_id>,<op>	Response: +QBTGATSL: <gserv_id>,<result> OK If there is any error relating to ME functionality, response: +CME ERROR: <err>

Parameter

<op>	0	Stop
	1	Start
<gserv_id>	User ID (or the name) of GATT server A Hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).	
<result>	0	SUCCESS
	1	FAIL

Example

```
AT+QBTGATSL="ABC2",1
+QBTGATSL: "ABC2",0

OK
```

2.9. AT+QBTGATSIND Send an Indication or Notification to a Client

AT+QBTGATSIND Send an Indication or Notification to a Client

Test Command
AT+QBTGATSIND=?

Response:
+QBTGATSIND: <gserv_id>,(list of supported <conn_id>s),(list of supported <attr_handle>s), (list of supported <need_confirm>s) ,<ind_value>

OK

Write Command
AT+QBTGATSIND=<gserv_id>,<conn_id>,<attr_handle>,<need_confirm>,<ind_value>

Response:
+QBTGATSIND:
<result>,<gserv_id>,<conn_id>,<value_handle>

OK

If there is any error relating to ME functionality, response:
+CME ERROR: <err>

Parameter

<gserv_id>	User ID (or the name) of GATT server A Hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill the '0' at the end of it to even).
<conn_id>	ID of current connection. The range is 0-255.
<attr_handle>	Handle of attribute. The range is 0-65535.
<value_handle>	Handle of this characteristic value. The range is 0-65535.
<ind_value>	The value that needs to be notified (if the length is odd, then fill '0' at the end of it to be even).
<need_confirm>	Need client confirmation when sending indication (not notification). 0 NO 1 YES
<result>	0 SUCCESS 1 FAIL

Example

```
AT+QBTGATSIND="ABC2",1,258,1,"74ab"           //Send an indication to a client.
+QBTGATSIND: 0,"ABC2",1,2

OK
```

2.10. AT+QBTGATSRSP Send a Response to a Client's Read or Write Operation

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

Test Command AT+QBTGATSRSP=?	Response: +QBTGATSRSP: <gserv_id>,(list of supported <response>s),(list of supported <conn_id>s),(list of supported <trans_id>s),(list of supported <attr_handle>s),<rsp_value> OK
Write Command AT+QBTGATSRSP=<gserv_id>,<response>,<conn_id>,<trans_id>,<attr_handle>,<rsp_value>	Response: +QBTGATSRSP: <result>,<gserv_id>,<conn_id>,<attr_handle> OK If there is any error relating to ME functionality, response: +CME ERROR: <err>

Parameter

<gserv_id>	User ID (or the name) of GATT server A Hex value string (string should be included in quotation marks). Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<response>	Response result for client request. The range is 0-255.
<conn_id>	ID of current connection. The range is 0-255.
<trans_id>	ID of current transaction. The range is 0-65535.
<attr_handle>	Handle of attribute. The range is 0-65535.
<rsp_value>	The character value that needs to be responded (if the value length is odd, then fill '0' at the end of it to be even).
<result>	0 SUCCESS 1 FAIL

Example

```
+QBTGATRREQ: "ABC2",1,452,107602B38034,258,0,0 //Client's read operation.  
AT+QBTGATSRSP="ABC2",0,1,452,258,"adb2" //Send response to a client.  
+QBTGATSRSP: 0,"ABC2",1,258  
OK
```

2.11. AT+QBTFMPSREG* Register/Deregister an FMP Service

AT+ QBTFMPSREG* Register/Deregister an FMP Service

Test Command AT+QBTFMPSREG=?	Response: +QBTFMPSREG: (list of supported <op>s) OK
Write Command AT+QBTFMPSREG=<op>	Response: OK If there is any error relating to ME functionality, response: +CME ERROR: <err>

Parameter

<op>	0	Deregister
	1	Register

Example

```
AT+QBTFMPSREG=1  
OK
```

2.12. AT+QBTPXPSREG* Register/Deregister a PXP Service

AT+ QBTPXPSREG* Register/Deregister a PXP Service

Test Command AT+QBTPXPSREG=?	Response: +QBTPXPSREG: (list of supported <op>s) OK
Write Command AT+QBTPXPSREG=<op>	Response: OK

If there is any error relating to ME functionality, response:
+CME ERROR: <err>

Parameter

<op>	0	Deregister
	1	Register

Example

```
AT+QBTXPSREG=1
OK
```

2.13. AT+QBTGATADV Set Advertising Parameters

AT+QBTGATADV Set Advertising Parameters

Test Command AT+QBTGATADV=?	Response +QBTGATADV: (list of supported <min_interval>s),(list of supported <max_interval>s) OK
Write Command AT+QBTGATADV=<min_interval>,<max_interval>	Response OK If there is any error relating to ME functionality, response: +CME ERROR: <err>

Parameter

<min_interval>	Minimum advertising interval for undirected and low duty cycle directed advertising. The range is 32-16384.
<max_interval>	Maximum advertising interval for undirected and low duty cycle directed advertising. The range is 32-16384.

Example

```
AT+QBTGATADV=800,2000
OK
```


2.14. AT+QBTGATCPU Update Connection Parameters

AT+ QBTGATCPU Update Connection Parameters

Test Command AT+QBTGATCPU=?	Response +QBTGATCPU: <bt_addr>,(list of supported <min_interval>s),(list of supported <max_interval>s),(list of supported <timeout>s),(list of supported <latency>s) OK
Write Command AT+QBTGATCPU=<bt_addr>,<min_interval>,<max_interval>,<timeout>,<latency>	Response OK If there is any error relating to ME functionality, response: +CME ERROR: <err>

Parameter

<bt_addr>	Address of the peer device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<min_interval>	Minimum value of the connection interval. The range is 6-3200.
<max_interval>	Maximum value of the connection interval. The range is 6-3200.
<timeout>	Supervision timeout for the connection. The range is 10-3200.
<latency>	Maximum slave latency allowed for the connection specified as the number of connection events. The range is 0-499.

Example

```
AT+QBTGATCPU=E62CA017A503,288,304,600,4
OK
```

2.15. AT+QBTGATSDISC Disconnect Server Actively

AT+ QBTGATSDISC Disconnect Server Actively

Test Command AT+QBTGATSDISC=?	Response +QBTGATSDISC : (list of supported <conn_id>s) OK
Write Command AT+QBTGATSDISC=<conn_id>	Response OK +QBTGATSCON:

	0,<gserv_id>,<result>,<bt_addr>,<conn_id>
--	---

	If there is any error relating to ME functionality, response: +CME ERROR: <err>
--	---

Parameter

<conn_id>	ID of current connection. The range is 0-255.
<gserv_id>	User ID (or the name) of GATT server
<result>	0 SUCCESS 1 FAIL
<bt_addr>	Address of the peer device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)

Example

```
AT+QBTGATSDISC=1
OK
```

2.16. AT+QBTLETXPWR Set the Level of BLE Transmission Power

AT+ QBTLETXPWR Set the Level of BLE Transmission Power

Test Command AT+QBTLETXPWR=?	Response +QBTLETXPWR: (list of supported <tx_level>s) OK
Read Command AT+QBTLETXPWR?	Response +QBTLETXPWR: <tx_level> OK
Write Command AT+QBTLETXPWR=<tx_level>	Response OK If there is any error relating to ME functionality, response: +CME ERROR: <err>

Parameter

<tx_level>	The level of BLE transmission power. The range is 0-7.
------------	--

NOTE

This command takes effect after restarting BT.

Example

```
AT+QBTLEXPWR=2
```

```
OK
```

```
AT+QBTLEXPWR?
```

```
+QBTLEXPWR: 2
```

```
OK
```

3 Description of URC

3.1. Notify the GATT Connection or Disconnection

Notify the GATT Connection or Disconnection

+QBTGATSCON:

<op>,<gserv_id>,<result>,<bt_addr>,
<conn_id>

Parameter

<op>	0	Disconnect
	1	Connect
<gserv_id>	User ID (or the name) of GATT server A Hex value string (string should be included in quotation marks), Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).	
<bt_addr>	Address of the peer device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)	
<result>	0	SUCCESS
	1	FAIL
<conn_id>	ID of current connection. The range is 0-255.	

Example

+QBTGATSCON: 1,"ABC2",0,CB2CD7923F46,1

3.2. Notify the GATT Client's Write Request

Notify the GATT Client's Write Request

+QBTGATWREQ:

<gserv_id>,<conn_id>,<trans_id>,<
bt_addr>,<attr_handle>,<value>,<ne
ed_rsp>,<is_prepare>,<offset>

Parameter

<gserv_id>	User ID (or the name) of GATT server A Hex value string (string should be included in quotation marks), Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
<conn_id>	ID of current connection. The range is 0-255.
<trans_id>	ID of current transaction. The range is 0-255.
<bt_addr>	Address of the peer device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<attr_handle>	Handle of attribute. The range is: 0-65535.
<value>	The value that needs to be written. Hex format.
<need_rsp>	Whether the client needs server's response. 1 YES 0 NO
<is_prepare>	Whether the server executes request immediately. 0 NO 1 YES
<offset>	Offset of the request. The range is 0-65535.

Example

```
+QBTGATWREQ: "ABC2",1,19,CB2CD7923F46,259,37383336,1,0,0
AT+QBTGATSRSP="ABC2",0,1,19,259,"adb2"
+QBTGATSRSP:0,"ABC2",1,259

OK
```

3.3. Notify the GATT Client's Read Request

Notify the GATT Client's Read Request

```
+QBTGATRREQ:
<gserv_id>,<conn_id>,<trans_id>,<bt
_addr>,<attr_handle>,<is_long>,<offs
et>
```

Parameter

<gserv_id>	User ID (or the name) of GATT server A Hex value string (string should be included in quotation marks), Each character of it should be in set {'0'~'9','a'~'f','A'~'F'}. The max length is 32 bytes (if the length is odd, then fill '0' at the end of it to be even).
------------	---

<conn_id>	ID of current connection. The range is 0-255.
<trans_id>	ID of current transaction. The range is 0-255.
<bt_addr>	Address of the peer device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<attr_handle>	Handle of attribute. The range is 0-65535.
<is_long>	Notify server the order number of the request (0 means the first request)
<offset>	Offset of the request. The range is 0-65535.

Example

```
+QBTGATRREQ: "ABC2",1,18,CB2CD7923F46,259,0,0
AT+QBTGATSRSP="ABC2",0,1,18,259,"adb2"
+QBTGATSRSP: 0,"ABC2",1,259
OK
```

3.4. Notify the PXP Connection*

Notify the PXP Connection*

```
+QBTPXPSCON: <bt_addr>
```

Parameter

<bt_addr>	Address of the peer device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
------------------------	--

Example

```
+QBTPXPSCON: CB2CD7923F46
```

3.5. Notify the PXP Link Loss Alert*

Notify the PXP Link Loss Alert*

+QBTPXPLLAT:
<bt_addr>,<alert_level>

Parameter

<bt_addr>	Address of the peer device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<alert_level>	Value of alert level characteristic. The range is 0-255.

Example

+QBTPXPLLAT: CB2CD7923F46,0

3.6. Notify the PXP Disconnection Alert*

Notify the PXP Disconnection Alert*

+QBTPXPDISAT:
<bt_addr>,<alert_level>

Parameter

<bt_addr>	Address of the peer device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<alert_level>	Value of alert level characteristic. The range is 0-255.

Example

+QBTPXPDISAT: CB2CD7923F46,0

3.7. Notify the FMP Connection or Disconnection*

Notify the FMP Connection or Disconnection*

+QBTFMPSCON: <op>,<bt_addr>

Parameter

<op>	0	Disconnect
	1	Connect
<bt_addr>	Address of the peer device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)	

Example

+QBTFMPSCON: 1,CB2CD7923F46

3.8. Notify the FMP Client's Write Request*

Notify the FMP Client's Write Request*

+QBTFMPSWREQ:
<bt_addr>,<alert_level>

Parameter

<bt_addr>	Address of the peer device (e.g.: A662616202C3. Meaning: LAP: 0xA66261, UAP: 0x62, NAP: 0x02c3)
<alert_level>	Value of alert level characteristic. The range is 0-255.

Example

+QBTFMPSWREQ: CB2CD7923F46,0

4 Appendix A References

Table 1: Terms and Abbreviations

Abbreviation	Description
BLE	Bluetooth Low Energy
BR	Basic Rate
BT	Bluetooth
DTE	Data Terminal Equipment
EDR	Enhanced Data Rate
FMP	Find Me Profile
GATT	Generic Attribute Profile
GSM	Global System for Mobile Communication
ID	Identification
ME	Mobile Equipment
PXP	Proximity Profile
UE	User Equipment
UUID	Universally Unique Identifier

Table 2: Format Map of Properties and Permission

Properties	Format Map
Default	0
Broadcast	1

Read	2
Write without response	4
Write	8
Notify	16
Indicate	32
Signed write	64
Extended properties	128

Permission	Format Map
Read	1
Read with encrypted protection	2
Read with MITM protection	4
Write	16
Write with encrypted protection	32
Write with MITM protection	64
Signed write	128
Signed write with MITM protection	256

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

Code of <err>	Meaning
4	Operation not supported
8300	GATT server register failed
8301	GATT server deregister failed
8302	GATT add service failed
8303	GATT remove service failed
8304	GATT add include service failed

8305	GATT add characteristic failed
8306	GATT add descriptor failed
8307	GATT service start failed
8308	GATT service stop failed
8309	GATT indication to client failed
8310	GATT send response to client failed
8311	GATT start listening failed
8312	GATT stop listening failed
8313	GATT start FMP failed
8314	GATT stop FMP failed
8315	GATT start PXP failed
8316	GATT stop PXP failed
