GigaDevice Semiconductor Inc.

GD32VW553 AT Command User Guide

Application Notes

AN151

Revision 1.4

(Nov.2025)

Table of Contents

List of Tables

1. AT command formats

1.1. Command types

Table 1-1. Command types

Type	Format	Description
Help command	AT+<x>=?	View command parameters and value ranges
Query command	AT+<x>?	Query the current parameter value of the specified target
Execution command	AT+<x> or AT+<x>=<…>	Run command Set the specified target parameter value

1.2. Command formats

Table 1-2. Command formats

Field	Description
AT	Command prefix
<CMD>	Command string
[ ]	Optional part
<>	Mandatory part. For specific commands, some parameters are mandatory to be entered
[p1],[p2],[p3],…	Parameters, which support both strings and numbers. Enter the IP address in the string format "x.x.x.x" String: Must be enclosed in double quotation marks Number: Both decimal and hexadecimal numbers are supported
<CR LF>	Line break

Note: AT [+<CMD>] [=] [p1],[p2],[p3],…<CR LF>

When the AT MQTT command set is supported, the total length of each AT command must not exceed 256 bytes.

1.3. Response formats

Table 1-3. Response formats

Output type	Description
[+<CMD>:<MSG>]	Output result or error prompt
<RSP>	OK: success ERROR: failure

Note: The Chinese characters in the response format are only explanations of the command response and are not actually displayed.

2. List of AT commands

Table 2-1. AT commands

Command	Description
AT	Test AT command mode
AT+HELP	Query all AT commands
AT+RST	Module reset
AT+GMR	Query version information
AT+TASK	Query all tasks of the current operating system
AT+HEAP	Query the free HEAP of the current operating system
AT+SYSRAM	Query the current free SRAM space
AT+UART	Set the LOG UART parameter or read the current parameter
AT+FS	Operate the file system
AT+FSMOUNT	Mount or unmount the file system
AT+CWMODE_CUR	Query or set the current WiFi operating mode: SoftAP/STA/ MONITOR
AT+CWJAP_CUR	Connect to AP
AT+CWLAP	Scan and display the AP list
AT+CWSTATUS	Query the current WiFi operating mode and status
AT+CWQAP	Disconnect from AP
AT+CWSAP_CUR	Start the SoftAP mode
AT+CWLIF	Query information about all STAs connected to SoftAP
AT+CWAUTOCONN	Set whether to automatically connect to the AP after power-on
AT+PING	Ping function
AT+CIPSTA	Query or set the IP address of the local STA
AT+CIPSTART	Create TCP connection or UDP transfer
AT+CIPSEND	Send data
AT+CIPSERVER	Start the TCP/UDP server
AT+CIPCLOSE	Close TCP connection or UDP transfer
AT+CIPSTATUS	Query Wi-Fi connection information
AT+CIPMUX	Query or enable/disable multi-connection mode
AT+CIPSTATE	Query TCP/UDP connection information
AT+CIFSR	Query local IP address information
AT+CIPMODE	Query or set the Transmission mode
AT+CIPDOMAIN	Domain name resolution
AT+TRANSINTVL	Query or set the Data Transmission Interval in Passthrough Mode
AT+CIPSNTPCFG	Query or set the time zone and SNTP server
AT+CIPSNTPTIME	Query SNTP time
AT+CIPSNTPINTV	Query or set the SNTP time synchronization interval
AT+WSCFG	Set WebSocket parameters
AT+WSHEAD	Query or set WebSocket request headers

Command	Description
AT+WSOPEN	Query or open a WebSocket connection
AT+WSSEND	Send data to the WebSocket connection
AT+WSCLOSE	Close the WebSocket connection
AT+MQTTUSERCFG	Set MQTT user properties
AT+MQTTLONGCLIENTID	Set MQTT client ID
AT+MQTTLONGUSERNAME	Set MQTT username
AT+MQTTLONGPASSWORD	Set MQTT password
AT+MQTTCONNCFG	Set MQTT connection properties
AT+MQTTCONN	Connect MQTT broker or query MQTT state
AT+MQTTPUB	Publish MQTT message(string)
AT+MQTTPUBRAW	Publish long MQTT message
AT+MQTTSUB	Subscribe to MQTT topic or query subscribed topic
AT+MQTTUNSUB	Unsubscribe to MQTT topic
AT+MQTTCLEAN	Close MQTT connection and release resources
AT+HTTPCLIENT	Send HTTP client request
AT+HTTPGETSIZE	Get the size of the HTTP resource
AT+HTTPCGET	Get HTTP resource
AT+HTTPCPOST	Post HTTP data of the specified length
AT+HTTPCPUT	Put HTTP data of the specified length
AT+HTTPURLCFG	Query or set long HTTP URL
AT+HTTPCHEAD	Query or set HTTP request header
AT+BLEENABLE	Enable BLE
AT+BLEDISABLE	Disable BLE
AT+BLENAME	Set the name
AT+BLEADVSTART	Start BLE advertising
AT+BLEADVSTOP	Stop BLE advertising
AT+BLEADVDATA	Set the advertising data
AT+BLEADVDATAEX	Set the advertising data
AT+BLESCANRSPDATA	Set the scan response data
AT+BLEPASSTH	Enable passthrough mode
AT+BLEPASSTHAUTO	Enable passthrough mode automatically
AT+BLESCANPARAM	Set scan parameters
AT+BLESCAN	Start scan
AT+BLESYNC	Start or cancel BLE synchronization
AT+BLESYNCSTOP	Stop BLE synchronization
AT+BLECONN	Initiate a BLE connection
AT+BLESCONNPARAM	Set or query the connection parameters
AT+BLEDISCONN	Disconnect the established BLE connection
AT+BLEMTU	Update or query the mtu
AT+BLEPHY	Update or query phy
AT+BLEDATALEN	Data length extension

Command	Description
AT+BLEADDR	Query or set the BLE bd address
AT+BLESETAUTH	Set the authentication
AT+BLEPAIR	Start pairing
AT+BLEENCRYPT	Start encrypting
AT+BLEPASSKEY	Enter the passkey
AT+BLECOMPARE	Enter the numeric comparison result
AT+BLELISTENCDEV	List the bond devices
AT+BLECLEARENCDEV	Clear the bond devices
AT+BLEGATTSSVC	List the devices registered locally
AT+BLEGATTSCHAR	List the characteristic of the service
AT+BLEGATTSDESC	List the descriptor of the characteristic
AT+BLEGATTSLISTALL	List the information of all local services
AT+BLEGATTSNTF	Send notification
AT+BLEGATTSIND	Send indication
AT+BLEGATTSSETATTRVAL	Set the value of the characteristic
AT+BLEGATTCDISCSVC	Discover the service
AT+BLEGATTCDISCCHAR	Discover the characteristic
AT+BLEGATTCDISCDESC	Discover the descriptor
AT+BLEGATTCRD	Read attribute value
AT+BLEGATTCWR	Write attribute value

3. AT basic command set

3.1. AT

Table 3-1. Test AT command mode

Command	Parameters	Response
Execution command AT		Execution result
Example: AT Correct response: OK		

3.2. AT+HELP

Table 3-2. Querying all AT commands

Command	Parameters	Response
Execution command AT+HELP		Display the list of all AT commands
Example: AT+HELP Correct response: AT COMMAND LIST: ================================== AT AT+HELP …… OK		

3.3. AT+RST

Table 3-3. Module reset command

Command	Parameters	Response
Execution command AT+RST		Restart message
Example: AT+RST Correct response: OK ==================================		

```
SDK Version: v1.0.3-908337de9f3c0871
Build date: 2025/07/28 16:15:20
OK
READY
```

3.4. AT+GMR

Table 3-4. Querying version information

Command	Parameters	Response (similar format information)
Execution command AT+GMR		Related version information

```
Example:
AT+GMR
Correct response:
==================================
SDK Version: v1.0.3-908337de9f3c0871
Build date: 2025/07/28 16:15:20
OK
```

3.5. AT+TASK

Table 3-5. Querying all tasks of the current operating system

Command	Parameters	Response (similar format information)
Execution command AT+TASK		Current task information list

```
Example:
AT+TASK
Correct response:
ATCMD           X    20    402  2      0x2001a780
...
RX              B    18    416  6      0x200203c8
OK
```

3.6. AT+HEAP

Table 3-6. Querying the free HEAP of the current operating system

Command	Parameters	Response (similar format information)
Execution command		HEAP usage

AT+HEAP		
Example: AT+HEAP Correct response: ================================== Total free heap size = 113784 Total min free heap size = 109480 OK		

3.7. AT+SYSRAM

Table 3-7. Querying the current free SRAM space

Command	Parameters	Response (similar format information)
Execution command AT+SYSRAM		Remaining SRAM space
Example: AT+SYSRAM Correct response: ================================== Free SRAM size = 108472 OK		

3.8. AT+UART

Table 3-8. Querying or setting serial port parameters

Command	Parameters	Response
Help command AT+UART=?		+UART=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>
Query command AT+UART?		Current serial port parameter
Execution command AT+UART=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>	<baudrate>: UART baud rate <databits>: Data bit 8: 8 bit <stopbits>: Stop bit 1: 1 bit 2: 1.5 bit 3: 2 bit <parity>: parity bit 0: None 1: Odd 2: Even	Execution result

Command	Parameters	Response
	<flow control>: Flow control 0: Disable flow control 1: Enable RTS 2: Enable CTS 3: Enable both RTS and CTS	
Example: AT+UART=115200,8,1,0,0 Correct response: OK		

3.9. AT+TRANSINTVL

Table 3-9. Querying or setting the Data Transmission Interval in Passthrough Mode

Command	Parameters	Response
Help command AT+TRANSINTVL=?		+TRANSINTVL=<interval>
Query command AT+TRANSINTVL?		Current Transmission Interval +TRANSINTVL:<interval>
Execution command AT+TRANSINTVL =<interval>	<interval>：Data transmission interval. Unit: milliseconds. Range: 0~1000. Default value: 20.	Execution result
Example: AT+TRANSINTVL=800 Correct Resonse: OK		

3.10. AT+FS

Table 3-10. Operating the file system

Command	Parameters	Response
Help command AT+FS=?		+FS=<type>,<operation>,<filename>,<offset>,<length>

Execution command
AT+FS=<type>,<operation>,<filename>,<offset>,<length>
	<type>: type of file system, only FatFS is supported, integer
	0: FatFS
	<operation>: operations on the file system
	0: delete a file or delete a folder and all files within the folder
	1: write the file
		Execution result
		if <operation> is equal to 1:
		> <input from keyboard>
		OK
		if <operation> is not equal to 1:
		OK

<table>
<tr><th>Command</th><th>Parameters</th><th>Response</th></tr>
<tr><td></td><td>2: read the file
3: query size of the file
4: query files in the specified path
<filename>: name of the file, string parameter
<offset>: offset address, only valid for read and write operations, integer
<length>: length, only valid for read and write operations, integer, unit: bytes</td><td></td></tr>
<tr><td colspan="3">Example 1:
AT+FS=0,1,"a/e.txt",0,10
Correct Resonse:
>
OK

Example 2:
AT+FS=0,2,"a/e.txt",0,5
Correct Resonse:
FATFS: read 5Bytes, content:
12345
OK

Example 3:
AT+FS=0,3,"a/e.txt"
Correct Resonse:
10Bytes
OK

Example 4, query all files in the root directory:
AT+FS=0,4,"."
Correct Resonse:
DIR: //a
FILE: //a/e.txt 10Bytes
OK

Example 5:
AT+FS=0,0,"a"
Correct Resonse:
OK</td></tr>
</table>

Command	Parameters	Response
Note: This command will automatically mount the file system. After completing file system operations with this command, it is strongly recommended to use the AT+FSMOUNT=0 command to unmount the file system to release memory. When the length of data to be read is greater than the actual file size, only the data of actual length will be returned.		

3.11. AT+FSMOUNT

Table 3-11. Mounting or unmounting the file system

Command	Parameters	Response
Help command AT+FSMOUNT =?		+FSMOUNT=<mount>
Execution command AT+FSMOUNT=<mount>	<mount>: mount or unmount file system, integer 0: unmount file system 1: mount file system	Execution result
Example 1: AT+FSMOUNT=1 Correct Resonse: OK Example 2: AT+FSMOUNT=0 Correct Resonse: OK Note: After completing file system operations with the AT+FS command, it is strongly recommended to use the AT+FSMOUNT=0 command to unmount the file system to release memory.		

4. AT WiFi command set

4.1. AT+CWMODE_CUR

Table 4-1. Querying or setting the current WiFi operating mode

Command	Parameters	Response
Help command AT+CWMODE_CUR=?		+CWMODE_CUR=<mode:0-3>
Query command AT+CWMODE_CUR?		Current operating mode +CWMODE_CUR:<mode>
Execution command AT+CWMODE_CUR=<mode>	<mode>: 0: MONITOR mode 1: STA mode 2: SoftAP mode 3: Wi-Fi concurrent mode	Execution result
Example: AT+CWMODE_CUR=2 Correct response: OK		

4.2. AT+CWJAP_CUR

Table 4-2. Querying the information of connected AP or connecting to AP

Command	Parameters	Response
Help command AT+CWJAP_CUR=?		+CWJAP_CUR=<ssid>,<pwd>
Query command AT+CWJAP_CUR?		+CWJAP_CUR: <ssid>,<mac>,<channel>,<rssi>
Execution command AT+CWJAP_CUR=<ssid>,<pwd>	<ssid>: String parameter <pwd>: String parameter	Execution result
Example 1: AT+CWJAP_CUR="totolink","12345678" Correct response 1: WIFI CONNECTED OK Example 2: AT+CWJAP_CUR="tplink","" Correct response 2: WIFI CONNECTED		

Command	Parameters	Response
OK		

4.3. AT+CWLAP

Table 4-3. Scanning and listing surrounding AP information

Command	Parameters	Response
Help command AT+CWLAP=?		+CWLAP=[ssid]
Execution command AT+CWLAP[=<ssid>]	<ssid>: String parameter	Scan results +CWLAP: <ssid>,<rssi>,<mac>,<channel>,<encrypt>
Example 1: AT+CWLAP Correct response 1: +CWLAP: iQOO Neo5, -44, d6:4f:86:cb:c8:d0, 1, WPA2 CCMP; +CWLAP: GD-guest, -43, 08:3a:38:cc:2f:d1, 1, OPEN ; +CWLAP: OpenWrt, -33, c4:70:ab:d9:bd:11, 1, OPEN ; +CWLAP: GD-internet, -44, 08:3a:38:cc:2f:d0, 1, OPEN ; +CWLAP: Redmi K40, -56, ba:fa:07:50:63:f6, 1, WPA2 CCMP; +CWLAP: D-Link_DIR-822, -30, 1c:5f:2b:fd:be:60, 1, WPA2 CCMP; +CWLAP: iPhone 24 Pro Max Ultr, -48, fa:da:47:72:f0:b3, 2, WPA2 CCMP; +CWLAP: TP-LINK_8659, -20, 68:77:24:bd:86:59, 4, WPA2/WPA3 CCMP; OK Example 2, if the parameter ssid is provided, only the corresponding AP information is displayed: AT+CWLAP= “xiaomi_4a” Correct response 2: +CWLAP: xiaomi_4a,-55, 88:c3:97:0d:c3:70,1, OPEN OK		

4.4. AT+CWSTATUS

Table 4-4. Querying WiFi status, STA, SoftAP, or MONITOR

Command	Parameters	Response
Execution command AT+CWSTATUS		+CWSTATUS: STA, connected, <ssid>,<channel>,<mac> Or +CWSTATUS: MONITOR, <channel>, <mac>

Command	Parameters	Response
		Or +CWSTATUS: STA, disconnected Or +CWSTATUS: SoftAP, <ssid>, <password>, <channel>
Example: AT+CWSTATUS Correct response: +CWSTATUS: STA, connected, xiaomi_4a, 1, 76:ba:ed:20:22:a2 OK		

4.5. AT+CWQAP

Table 4-5. Disconnecting from AP

Command	Parameters	Response
Execution command AT+CWQAP		Disconnection message
Example: AT+CWQAP Correct response: OK		

4.6. AT+CWSAP_CUR

Table 4-6. Starting SoftAP

Command	Parameters	Response
Help command AT+CWSAP_CUR=?		+CWSAP_CUR=<ssid>,<pwd>,<chl:1-13>,<hidden:0-1>
Execution command AT+CWSAP_CUR=<ssid>,<pwd>,<chl>,<hidden>	<ssid>: String parameter <pwd>: String parameter <chl>: 1~13 <hidden>: 0: SSID Broadcast 1: Hidden SSID	Execution result
Example: AT+CWSAP_CUR="test_ap","12345678",6,0 Correct response: OK		

4.7. AT+CWLIF

Table 4-7. Viewing clients connected to SoftAP

Command	Parameters	Response
Execution command AT+CWLIF		+CWLIF: [0] <mac1> +CWLIF: [1] <mac2> ……
Example: AT+CWLIF Correct response: +CWLIF: [0] e0:2b:e9:8a:46:ac OK		

4.8. AT+CWAUTOCONN

Table 4-8. Setting whether to automatically connect to the AP after power-on

Command	Parameters	Response
Help command AT+CWAUTOCONN=?		+CWAUTOCONN=<enable>
Query command AT+CWAUTOCONN?		+CWAUTOCONN:<enable>
Execution command AT+CWAUTOCONN=<enable>	<enable>: 0-1 0: disable auto connect 1: enable auto connect	Execution result
Example: AT+CWAUTOCONN=1 Correct response: OK		
Additional description: After +CWAUTOCONN is set to 1, if the AP is successfully connected, the AP information will be saved in FLASH. After restarting, the AP will be automatically connected according to the AP information stored in FLASH.		

5. AT TCPIP command set

5.1. AT+PING

Table 5-1. Ping function

<table>
<tr><th>Command</th><th>Parameters</th><th>Response</th></tr>
<tr><td>Help command
AT+PING=?</td><td></td><td>+PING=<ip or domain name></td></tr>
<tr><td>Execution command
AT+PING=<ip or domain></td><td><ip>: string, IP address or domain name</td><td>+<delay_time>
+<delay_time>
……</td></tr>
<tr><td colspan="3">Example 1:
AT+PING="192.168.0.1"
Correct response 1:
+80
+47
+49
+55
+53
OK

Example 2 when pinging a public domain name, the Internet must be connected; otherwise, it will fail:
AT+PING="www.baidu.com"
Correct response 2:
+149
+47
+51
+47
+112
OK</td></tr>
</table>

5.2. AT+CIPSTA

Table 5-2. Querying or setting the IP address of the local STA

Command	Parameters	Response
Help command AT+CIPSTA=?		+CIPSTA=<ip>,<netmask>,<gw>
Query command AT+CIPSTA?		+CIPSTA:<ip> +CIPSTA:<netmask> +CIPSTA:<gw>
Execution command AT+CIPSTA=<ip>,<netmask>,<gw>	<ip>: String parameter <netmask>: String parameter <gw>: String parameter	Execution result

Command	Parameters	Response
Example 1: AT+CIPSTA? Correct response 1: +CIPSTA: 192.168.185.43 +CIPSTA: 255.255.255.0 +CIPSTA: 192.168.185.1 OK Example 2: AT+CIPSTA="192.168.185.45","255.255.255.0","192.168.185.1" Correct response 2: OK		

5.3. AT+CIPSTART

Table 5-3. Creating TCP connection or UDP transfer

Command	Parameters	Response
Help command AT+CIPSTART=?		+CIPSTART=[<con_id>,]<type:TCP or UDP>,<remote ip>,<remote port>[,udp local port][,tcp keep alive]
Execution command AT+CIPSTART=[<con_id>,]<type>,<remote ip>,<remote port>, [udp local port] , [tcp keep alive]	<con_id>: connection ID, used for multi-connection mode <type>: "TCP" or "UDP", string parameter <remote ip>: Server IP, string parameter <remote port>: Server Port, integer [udp local port]: The UDP local port number, integer [tcp keep alive]: Keepalive time, integer, unit: seconds.	Execution result <con_id>,OK or ERROR
Example 1, establish a TCP connection in single-connection mode: AT+CIPSTART="TCP","192.168.0.2",2001,1 Correct response: 0,OK Example 2, establish a TCP connection in multi-connection mode: AT+CIPSTART=2,"TCP","192.168.0.2",2001,60 Correct response: 2,OK		

Example 3, establish a UDP transfer in single-connection mode:

AT+CIPSTART="UDP", "192.168.0.2",5001

Correct response:

0,OK

Example 4, establish a UDP transfer in multi-connection mode with local port number 8888 specified:

AT+CIPSTART=3,"UDP", "192.168.0.2",5001,8888

Correct response:

3,OK

Note: In this test, the tester needs to run the sokit or other network tool on the test machine.

5.4. AT+CIPSEND

Table 5-4. Sending data

Command	Parameters	Response
Help command AT+CIPSEND=?		Usage: Normal Mode Usage: +CIPSEND=[con_id,]<len>[,<remote ip>,<remote port>] PassThrough Mode Usage: +CIPSEND
Execution command in Normal transmission mode AT+CIPSEND=[con_id,]<len>,[<remote ip>, <remote port>]	[con_id]: connection ID, integer <len>: < =2048, length of sent data, integer [remote ip]: Remote IP address, string parameter [remote port]: Remote port, integer	><input from keyboard> SEND OK
Execution command in WiFi passthrough transmission mode AT+CIPSEND		OK ><input from keyboard>

Example 1, in single-connection mode, con_id can be omitted:

AT+CIPSEND=10

Correct response 1:

>

SEND OK

OK

Example 2, in multi-connection mode, con_id needs to be specified:

```
AT+CIPSEND=1,10
```

Correct response 2:

```
>
SEND OK
OK
```

Example 3, UDP transmission can specify the peer IP and port.:

```
AT+CIPSEND=1,20,"192.168.0.2",5001
```

Correct response 3:

```
>
SEND OK
OK
```

Example 4: UART WiFi passthrough transmission when the GD32VW553 works as a TCP client in single connection

Connect to the router.

```
AT+CWJAP=”test_ap”,”1234567890”
```

Query the device’s IP address, take 192.168.1.27 for example.

```
AT+CIPSTA?
```

Connect the PC to the same router which GD32VW553 is connected to. Use a network tool on the PC to create a TCP Server. For Example, the TCP Server on PC is 192.168.1.2, and the port is 5678. Connect the GD32VW553 to the TCP server as a TCP client over TCP.

```
AT+CIPSTART="TCP","192.168.1.2",5678,0
```

Enable the UART WiFi Passthrough Receiving Mode.

```
AT+CIPMODE=1
```

Enter the UART WiFi Passthrough mode and send data.

```
AT+CIPSEND
OK
>
```

Stop Sending data. When receiving a packet that contains only +++, the UART WiFi PassThrough transmission process will be stopped. Then please wait at least 1 second before sending the next AT command.

```
+++
```

Exit the UART WiFi PassThrough Receiving Mode.

```
AT+CIPMODE=0
```

Close TCP connection.

```
AT+CIPCLOSE
```

Note:

Enter the WiFi Passthrough Mode, the GD32VW553 can receive 8129 bytes and send 2920 bytes at most each time. If the data received by GD32VW553 reaches or exceeds 2920 bytes, the data will be immediately sent in chunks of 2910 bytes. Otherwise, it will wait for 20 milliseconds (You can configure this interval using AT+TRANSINTVL command) before being sent. When a single packet

containing +++ is received, the GD32VW553 will exit the data sending mode under the WiFi Passthrough Mode. Please wait at least on second before sending the next AT command. WiFi Passthrough Mode can only be used for single connection in the WiFi Passthrough Mode. For UDP WiFi passthrough, the UDP's remote server, remote port and local port must be specified. In this test, the tester needs to run the sokit or other network tool on the test machine. The Passthrough mode only supports single TCP connection and fixed UDP communication peer.

5.5. AT+CIPSERVER

Table 5-5. Starting the TCP server

Command	Parameters	Response
Help command AT+CIPSERVER=?		+CIPSERVER=<mode:0-1>[,<type>,<port>]
Execution command AT+CIPSERVER=<mode>,[,<type>,<port>]	<mode>: 0: Close the server 1: Create a server <type>: "TCP" or "UDP", string parameter [port]: Optional parameters, integer	Execution result
Example 1, when setting up the server, <type> and <port> must be specified: AT+CIPSERVER=1,"TCP",5004 Correct response: OK Example 2, close the server will also close all connections: AT+CIPSERVER=0 Correct response: OK Note: 1. A server can only be established in multi-connection mode (AT+CIPMUX=1). 2. Only one server is allowed to be established.		

5.6. AT+CIPCLOSE

Table 5-6. Closing TCP connection or UDP transfer

Command	Parameters	Response
Help command AT+CIPCLOSE=?		+CIPCLOSE=[con_id]
Execution command AT+CIPCLOSE=[con_id]	[con_id]: connection ID, integer	Execution result
Example 1: AT+CIPCLOSE=2		

Correct response: CLOSED 2 OK Example 2, in single-connection mode, con_id can be omitted: AT+CIPCLOSE Correct response: CLOSED OK Example 3, when con_id is set to MAX_CLIENT_NUM (default is 9), all connections will be closed: Correct response: CLOSED OK

5.7. AT+CIPSTATUS

Table 5-7. Querying Wi-Fi connection information

Command	Parameters	Response
Execution command AT+CIPSTATUS		STATUS: 5
Example: AT+CIPSTATUS Correct response: STATUS: 2 OK		
Additional description: STATUS 2: STA has been connected to the AP and obtained an IP address. 3: STA has been connected to the AP and created TCP connection or UDP transfer clients. 4: The dhcp process is ongoing. 5: Other connection status.		

5.8. AT+CIFSR

Table 5-8. Querying local IP address information

Command	Parameters	Response
Execution command AT+CIFSR		+CIFSR:APIP,<ip> +CIFSR:APMAC,<mac> Or +CIFSR:STAIP,<ip> +CIFSR:STAMAC,<mac>
Example: AT+CIFSR Correct response:		

Command	Parameters	Response
+CIFSR:STAIP,192.168.2.3 +CIFSR:STAMAC,76:ba:ed:20:22:a2 OK		

5.9. AT+CIPMODE

Table 5-9. Querying or Setting the Transmission Mode

Command	Parameters	Response
Help command AT+CIPMODE=?		+CIPMODE=<mode:0-1>
Query command AT+CIPMODE?		Current Transmission Mode +CIPMODE:<mode>
Execution command AT+CIPMODE=<mode>	<mode>:Transmission Mode 0: Normal Transmission Mode 1: WiFi Passthrough Receiving Mode	Execution result
Example: AT+CIPMODE=1 Correct response: OK Note: WiFi Passthrough Receiving Mode can only be enabled in TCP single connection mode, UDP mode when the remote host, remote port and local port are specified. The maximum receive length is 2920 Bytes each time in WiFi Passthrough Receiving Mode.		

5.10. AT+CIPMUX

Table 5-10. Querying or enable/disable multi-connection mode

Command	Parameters	Response
Help command AT+CIPMUX=?		+CIPMUX=<mode:0-1>
Query command AT+CIPMUX?		Current connection mode +CIPMUX:<mode>
Execution command AT+CIPMUX=<mode>	<mode>：connection mode 0: single connection mode 1: multi-connection mode	Execution result
Example: AT+CIPMUX=1 Correct response: OK		

Command	Parameters	Response
Note: The connection mode can only be changed when all connections are closed. If a TCP/UDP server is established, the server must be closed before the connection mode can be changed.		

5.11. AT+CIPSTATE

Table 5-11. Querying TCP/UDP connection information

Command	Parameters	Response
Execution command AT+CIPSTATE		Return connection information when a connection exists: +CIPSTATE: <con_id>,<type>,<remote ip>,<remote port>,<local port>,<fd>,<role> Return when there is no connection: OK
Example: AT+CIPSTATE Correct response: +CIPSTATE:3,TCP,192.168.19.115,5000,51955,3,0 OK		
Additional description: type: “TCP” or “UDP”。 role: 0, module as a client; 1, module as a server.		

5.12. AT+CIPDOMAIN

Table 5-12. Domain name resolution

Command	Parameters	Response
Help command AT+CIPDOMAIN=?		+CIPDOMAIN=<"domain name">[,<ip network>]
Execution command AT+CIPDOMAIN =<"domain name">[,<ip network>]	<"domain name">: domain name, string parameter [,<ip network>]: Preferred IP type, integer, default value is 1. 1: Prefer resolving to IPv4 address 2: Resolve only to IPv4 address 3: Resolve only to IPv6 address	Execution result +CIPDOMAIN:<IP>
Example: AT+CIPDOMAIN="www.baidu.com",1		

Correct response: +CIPDOMAIN:<"36.152.44.132"> OK Note: If you need to resolve public domain names, must first connect to the Internet.

5.13. AT+CIPSNTPCFG

Table 5-13. Querying or setting the time zone and SNTP server

Command	Parameters	Response
Help command AT+CIPSNTPCFG=?		+CIPSNTPCFG:<enable>,<timezone>,[<SNTP server1>,<SNTP server2>,<SNTP server3>]
Query command AT+CIPSNTPCFG?		+CIPSNTPCFG: <enable>,<timezone>,[<SNTP server1>,<SNTP server2>,<SNTP server3>]
Execution command AT+CIPSNTPCFG =<enable>,<timezone>,[<SNTP server1>,<SNTP server2>,<SNTP server3>]	<enable>: 0: disable SNTP service 1: enable SNTP service <timezone>: timezone Format: hour|minute, hour:-12~+14, minute:00~59 [<SNTP server1>]: SNTP server, string parameter	Execution result
Example: AT+CIPSNTPCFG=1,800,"cn.pool.ntp.org" Correct response: OK Example: AT+CIPSNTPCFG=1,-1200 Correct response: OK Example: AT+CIPSNTPCFG=0,800 Correct response: OK Note: Enabling the SNTP service requires an internet connection.		

Command	Parameters	Response
After enabling the SNTP service, the message "+TIME_UPDATED" will be printed once the time synchronization is complete.		

5.14. AT+CIPSNTPTIME

Table 5-14. Querying SNTP time

Command	Parameters	Response
Execution command AT+CIPSNTPTIME		synchronized time: SNTP time: <SNTP time> unsynchronized time: Please start the SNTP or wait for the SNTP time update
Example: AT+CIPSTATE Correct response: SNTP time: 2025-09-02 Tuesday 15:38:14 OK		
Additional description: The SNTP service needs to be enabled. Wait for the "+TIME_UPDATED" message to be printed before retrieving the synchronized time.		

5.15. AT+CIPSNTPINTV

Table 5-15. Querying or setting the SNTP time synchronization interval

Command	Parameters	Response
Help command AT+CIPSNTPINTV=?		+CIPSNTPINTV=<interval second>
Query command AT+CIPSNTPINTV?		+CIPSNTPINTV:<interval second>
Execution command AT+CIPSNTPINTV=<interval second>	<interval second>： SNTP time synchronization interval, integer, unit: seconds, range: [15, 4294967]	Execution result
Example: AT+CIPSNTPINTV? Correct response: +CIPSNTPINTV:20 OK Example: AT+CIPSNTPINTV=20 Correct response:		

Command	Parameters	Response
OK		

5.16. AT+WSCFG

Table 5-16. Setting WebSocket parameters

Command	Parameters	Response
Help command AT+WSCFG=?		+WSCFG=<link_id>,<ping_intv_sec>,<ping_timeout_sec>[,<buffer_size>]
Execution command AT+WSCFG=<link_id>,<ping_intv_sec>,<ping_timeout_sec>[,<buffer_size>]	<link_id>: WebSocket connection ID, integer, range: 0~2, up to three WebSocket connections are supported <ping_intv_sec>: WebSocket Ping interval, integer, unit: seconds, range: 1~7200, default value is 10, meaning a WebSocket Ping packet is sent every 10 seconds <ping_timeout_sec>: WebSocket Ping timeout, integer, unit: seconds, range: 1~7200, default value is 120, meaning if no WebSocket Pong packet is received within 120 seconds, the connection will be closed <buffer_size>: WebSocket buffer size, integer, unit: bytes, range: 1~8192, default value is 1024	Execution result
Example: AT+WSCFG=0,15,120,1024 Correct response: OK Note: Please set this command before the AT+WSOPEN command, otherwise this command will not take effect.		

5.17. AT+WSHEAD

Table 5-17. Querying or setting WebSocket request headers

<table>
<tr><th>Command</th><th>Parameters</th><th>Response</th></tr>
<tr><td>Help command
AT+WSHEAD=?</td><td></td><td>+WSHEAD=<req_header_len></td></tr>
<tr><td>Query command
AT+WSHEAD?</td><td><index>: Index value of the WebSocket request header, integer
<"req_header">: WebSocket request header, string parameter</td><td>+WSHEAD:<index>,<"req_header">
...</td></tr>
<tr><td>Execution command
AT+WSHEAD=<req_header_len></td><td><req_header_len>: length of WebSocket request header, integer, unit: bytes
0: clear all configured WebSocket request headers
other values: set a new WebSocket request header</td><td>Execution result
OK
> <input from keyboad></td></tr>
<tr><td colspan="3">Example:
AT+WSHEAD=31
Correct response:
OK
>
OK

Example:
AT+WSHEAD?
Correct response:
+WSHEAD:0,"Sec-WebSocket-Protocol: chat-v2"
+WSHEAD:1,"Cookie: session_id=abc123"
OK

Note:
This command can only set one WebSocket request header at a time, but it can be called multiple times to support up to 5 different WebSocket request headers.
The format of the WebSocket request header is: key: value.
The WebSocket request headers set by this command are global; once configured, all WebSocket commands will carry these headers.</td></tr>
</table>

5.18. AT+WSOPEN

Table 5-18. Querying or opening a WebSocket connection

Command	Parameters	Response
Help command AT+WSOPEN=?		+WSOPEN=<link_id>,<"uri">[,<"subprotocol">][,<timeout_ms>][,<"auth">]
Query command AT+WSOPEN?	<state>: state of the WebSocket connection, integer 0: WebSocket connection is closed 1: WebSocket is reconnecting 2: WebSocket connection is established 3: WebSocket Pong reception timeout or connection data read error, waiting to reconnect 4: received a WebSocket close frame from the server, sending a close frame to the server	+WSOPEN:<link_id>,<state>,<"uri"> ...
Execution command AT+WSOPEN=<link_id>,<"uri">[,<"subprotocol">][,<timeout_ms>][,<"auth">]	<link_id>: WebSocket connection ID, integer, range: 0~2, up to three WebSocket connections are supported <"uri">: the Uniform Resource Identifier (URI) of the WebSocket server, string parameter <"subprotocol">: subprotocol of WebSocket, string parameter <timeout_ms>: timeout for establishing a WebSocket connection, integer, unit: milliseconds, range: 0~180000, default value is 15000 <"auth">: authentication of WebSocket, string parameter	Execution result +WS_CONNECTED:<link_id>
Example: AT+WSOPEN=0,"wss://echo.websocket.events" Correct response: +WS_CONNECTED:0 OK		

Command	Parameters	Response
Example: AT+WSOPEN? Correct response: +WSOPEN:0,2,"wss://echo.websocket.events" OK Note: For more information about the <"subprotocol"> parameter, please refer to Section 1.9 of RFC6455. For more information about the <"auth"> parameter, please refer to Section 4.1.12 of RFC6455.		

5.19. AT+WSSEND

Table 5-19. Sending data to the WebSocket connection

Command	Parameters	Response
Help command AT+WSSEND=?		+WSSEND=<link_id>,<length>[,<opcode>][,<timeout_ms>]

Execution command
AT+WSSEND=<link_id>,<length>[,<opcode>][,<timeout_ms>]
	<link_id>: WebSocket connection ID, integer, range: 0~2, up to three WebSocket connections are supported
	<length>:length of the data to be sent, integer, unit: bytes. The maximum length that can be sent is the smaller value between <buffer_size> in AT+WSCFG minus 10 and the available system heap size
	<opcode>: opcode in the sent WebSocket frame, integer, range: 0~0xF, default value is 1 (text frame)
	0x0: continuation frame
	0x1: text frame
	0x2: binary frame
	0x3-0x7: reserved for other non-control frames
	0x8: close frame
	0x9: ping frame
	0xA: pong frame
	0xB-0xF: reserved for other control frames
	<timeout_ms>: send timeout,
		Execution result
		OK
		> <input from keyboad>
		if the data is transmitted successfully, AT returns:
		SEND OK
		if the connection is not established or the connection is closed during data transmission, AT returns:
		ERROR

Command	Parameters	Response
	integer, unit: milliseconds, range: 0~60000, default value is 10000	
Example: AT+WSSEND=0,5 Correct response: OK > SEND OK Note: For more information about the <opcode> parameter, please refer to Section 5.2 of RFC6455.		

5.20. AT+WSCLOSE

Table 5-20. Closing the WebSocket connection

Command	Parameters	Response
Help command AT+WSCLOSE=?		+WSCLOSE=<link_id>
Execution command AT+WSCLOSE=<link_id>	<link_id>: WebSocket connection ID, integer, range: 0~2, up to three WebSocket connections are supported	Execution result
Example: AT+WSCLOSE=0 Correct response: OK		

6. AT MQTT command set

6.1. AT+MQTTUSERCFG

Table 6-1. Setting MQTT user properties

Command	Parameters	Response
Help command AT+MQTTUSERCFG=?		+MQTTUSERCFG=<LinkID>,<scheme>,<"client_id">,<"username">,<"password">,<cert_key_ID>,<CA_ID>
Execution command AT+MQTTUSERCFG=<LinkID>,<scheme>,<"client_id">,<"username">,<"password">,<cert_key_ID>,<CA_ID>	<LinkID>: only supports link ID 0, integer <scheme>: connection scheme 1: MQTT over TCP 2: MQTT over TLS(not validate certificates) 3: MQTT over TLS(validate server certificate) 4: MQTT over TLS(provide client certificate) 5: MQTT over TLS (validate server certificate and provide client certificate) <"client_id">: client ID, string parameter, maximum length is 256 bytes <"username">: username for logging into the MQTT broker, string parameter, maximum length is 64 bytes <"password">: password for logging into the MQTT broker, string parameter, maximum length is 64 bytes <cert_key_ID>: certificate ID, only supports one set of cert certificates, parameter is 0, integer <CA_ID>: CA ID, only supports one set of CA certificates, parameter is 0, integer	Execution result
Example: AT+MQTTUSERCFG=0,3,"Gigadevice","user","123456",0,0		

Correct response: OK

6.2. AT+MQTTLONGCLIENTID

Table 6-2. Setting MQTT client ID

Command	Parameters	Response
Help command AT+MQTTLONGCLIENTID=?		+MQTTLONGCLIENTID=<LinkID>,<length>
Execution command AT+MQTTLONGCLIENTID=<LinkID>,<length>	<LinkID>: only supports link ID 0, integer <length>: length of MQTT client ID, integer, range: 1~1024	Execution result OK > <input from keyboad>
Example: AT+MQTTLONGCLIENTID=0,10 Correct response: OK > OK Note: Both the AT+MQTTUSERCFG command and the AT+MQTTLONGCLIENTID command can be used to set the MQTT client ID. The differences between them are as follows: The AT+MQTTUSERCFG command is limited by the total length of the AT command, so it can set a relatively shorter client ID, whereas the AT+MQTTLONGCLIENTID command can be used to set a relatively longer client ID. You must configure AT+MQTTUSERCFG before using AT+MQTTLONGCLIENTID.		

6.3. AT+MQTTLONGUSERNAME

Table 6-3. Setting MQTT username

Command	Parameters	Response
Help command AT+MQTTLONGUSERNAME=?		+MQTTLONGUSERNAME=<LinkID>,<length>
Execution command AT+MQTTLONGUSERNAME=<LinkID>,<length>	<LinkID>: only supports link ID 0, integer <length>: length of MQTT username, integer, range: 1~1024	Execution result OK > <input from keyboad>
Example:		

AT+MQTTLONGUSERNAME=0,8 Correct response: OK > OK Note: Both the AT+MQTTUSERCFG command and the AT+MQTTLONGUSERNAME command can be used to set the MQTT username. The differences between them are as follows: The AT+MQTTUSERCFG command is limited by the total length of the AT command, so it can set a relatively shorter username, whereas the AT+MQTTLONGUSERNAME command can be used to set a relatively longer username. You must configure AT+MQTTUSERCFG before using AT+MQTTLONGUSERNAME.

6.4. AT+MQTTLONGPASSWORD

Table 6-4. Setting MQTT password

Command	Parameters	Response
Help command AT+MQTTLONGPASSWORD=?		+MQTTLONGPASSWORD=<LinkID>,<length>
Execution command AT+MQTTLONGPASSWORD=<LinkID>,<length>	<LinkID>: only supports link ID 0, integer <length>: length of MQTT password, integer, range: 1~1024	Execution result OK > <input from keyboad>
Example: AT+MQTTLONGPASSWORD=0,12 Correct response: OK > OK Note: Both the AT+MQTTUSERCFG command and the AT+MQTTLONGPASSWORD command can be used to set the MQTT password. The differences between them are as follows: The AT+MQTTUSERCFG command is limited by the total length of the AT command, so it can set a relatively shorter password, whereas the AT+MQTTLONGPASSWORD command can be used to set a relatively longer password. You must configure AT+MQTTUSERCFG before using AT+MQTTLONGPASSWORD.		

6.5. AT+MQTTCONNCFG

Table 6-5. Setting MQTT connection properties

Command	Parameters	Response
Help command AT+MQTTCONNCFG=?		+MQTTCONNCFG=<LinkID>,<keepalive>,<disable_clean_session>,<"lwt_topic">,<"lwt_msg">,<lwt_qos>,<lwt_retain>
Execution command AT+MQTTCONNCFG=<LinkID>,<keepalive>,<disable_clean_session>,<"lwt_topic">,<"lwt_msg">,<lwt_qos>,<lwt_retain>	<LinkID>: only supports link ID 0, integer <keepalive>: MQTT ping timeout, integer, unit: seconds, range: 0~7200, default value is 0 and it is forcibly changed to 120 seconds <disable_clean_session>: flag of setting MQTT clean session 0: enable clean session 1: disable clean session <"lwt_topic">: will topic, string parameter, maximum length is 128 bytes <"lwt_msg">: will message, string parameter, maximum length is 128 bytes <lwt_qos>: will QoS, integer, range: 0~2, default value is 0 <lwt_retain>: will retain, integer, range: 0~1, default value is 0	Execution result
Example: AT+MQTTCONNCFG=0,0,0,"will_topic","will_message",0,0 Correct response: OK Note: For more information about the parameter <disable_clean_session>, please refer to the Clean Session section in the MQTT 3.1.1 protocol.		

6.6. AT+MQTTCONN

Table 6-6. Connecting MQTT broker or querying MQTT state

Command	Parameters	Response
Help command AT+MQTTCONN=?		+MQTTCONN=<LinkID>,<"host">,<port>,<reconnect>
Query command AT+MQTTCONN?	<LinkID>: only supports link ID 0, integer <state>: MQTT state 0: MQTT not configured 1: AT+MQTTUSERCFG is setted 2: AT+MQTTCONNCFG is setted 3: disconnected 4: connected 5: connected but not subscribe to topics 6: connected and subscribed to topics <scheme>: connection scheme 1: MQTT over TCP 2: MQTT over TLS(not validate certificates) 3: MQTT over TLS(validate server certificate) 4: MQTT over TLS(provide client certificate) 5: MQTT over TLS (validate server certificate and provide client certificate) <"host">: domain name of MQTT broker, string parameter, maximum length is 128 bytes <port>: port of MQTT broker, integer, maximum port is 65535 <reconnect>: reconnect 0: MQTT not automatically reconnect 1: MQTT automatically reconnect and will consume an amount of memory resources	+MQTTCONN:<LinkID>,<state>,<scheme>,<"host">,<port>,<reconnect>

Command	Parameters	Response
Execution command AT+MQTTCONN=<LinkID>,<"host">,<port>,<reconnect>		Execution result +MQTTCONNECTED:<LinkID>,<scheme>,<"host">,<port>,<reconnect>
Example 1, when MQTT is not configured: AT+MQTTCONN? Correct response: +MQTTCONN:0,0 OK Example 2, when AT+MQTTUSERCFG is setted: AT+MQTTCONN? Correct response: +MQTTCONN:0,1,3 OK Example 3, when MQTT client is connected but does not subscribe to topics: AT+MQTTCONN? Correct response: +MQTTCONN:0,5,3,"192.168.43.50",8883,0 OK Example 4: AT+MQTTCONN=0,"192.168.43.50",8883,0 Correct response: +MQTTCONNECTED:0,3,"192.168.43.50",8883,0 OK Note: When MQTT client connects to MQTT broker successfully, a message of +MQTTCONNECTED:<LinkID>,<scheme>,<"host">,<port>,<reconnect> will be prompted. If MQTT client does not automatically reconnect and the connection is dropped after establishment, or if automatic reconnection is enabled but the connection drops after establishment and the reconnection attempts reach the maximum limit, this command cannot be used again to re-establish the connection. You must first use the AT+MQTTCLEAN=0 command to clear cached information, reconfigure the parameters, and then establish a new connection. Use Mosquitto as the MQTT broker.		

6.7. AT+MQTTPUB

Table 6-7. Publishing MQTT message(string)

Command	Parameters	Response

Help command AT+MQTTPUB=?		+MQTTPUB=<LinkID>,<"topic">,<"data">,<qos>,<retain>
Execution command AT+MQTTPUB=<LinkID>,<"topic">,<"data">,<qos>,<retain>	<LinkID>: only supports link ID 0, integer <"topic">: MQTT topic, string parameter, maximum length is 128 bytes <"data">: MQTT message, string parameter <qos>: QoS of published message, integer, range: 0~2, default value is 0 <retain>: retain of published message, integer, range: 0~1, default value is 0	Execution result
Example: AT+MQTTPUB=0,"topic_test","helloworld",0,0 Correct response: OK Note: This command cannot send \0. If you need to send this data, please use the AT+MQTTPUBRAW command.		

6.8. AT+MQTTPUBRAW

Table 6-8. Publishing long MQTT message

Command	Parameters	Response
Help command AT+MQTTPUBRAW=?		+MQTTPUBRAW=<LinkID>,<"topic">,<length>,<qos>,<retain>
Execution command AT+MQTTPUBRAW=<LinkID>,<"topic">,<length>,<qos>,<retain>	<LinkID>: only supports link ID 0, integer <"topic">: MQTT topic, string parameter, maximum length is 128 bytes <length>: length of MQTT message, integer <qos>: QoS of published message, integer, range: 0~2, default value is 0 <retain>: retain of published message, integer, range: 0~1, default value is 0	Execution result OK > <input from keyboard> serial data transmit successfully, AT returns: +MQTTPUB:OK serial data transmit unsuccessfully, AT returns: +MQTTPUB:FAIL

Example: AT+MQTTPUBRAW=0,"topic_test",9,0,0 Correct response: OK > +MQTTPUB:OK OK

6.9. AT+MQTTSUB

Table 6-9. Subscribing to MQTT topic or querying subscribed topic

Command	Parameters	Response
Help command AT+MQTTSUB=?		AT+MQTTSUB=<LinkID>,<"topic">,<qos>
Query command AT+MQTTSUB?	<LinkID>: only supports link ID 0, integer <state>: MQTT state 0: MQTT not configured 1: AT+MQTTUSERCFG is setted 2: AT+MQTTCONNCFG is setted 3: disconnected 4: connected 5: connected but not subscribe to topics 6: connected and subscribed to topics <"topic">: subscribed topic, string parameter <qos>: QoS of subscribed message, integer, range: 0~2	+MQTTSUB:<LinkID>,<state>,<"topic1">,<qos> +MQTTSUB:<LinkID>,<state>,<"topic2">,<qos> +MQTTSUB:<LinkID>,<state>,<"topic3">,<qos> ...

Execution command
AT+MQTTSUB=<LinkID>,<"topic">,<qos>
		Execution result
		if topic not subscribed previously and the subscription is successful, AT returns:
		OK
		if topic is subscribed preciously, AT returns:
		ALREADY SUBSCRIBE

Example 1: AT+MQTTSUB? Correct response:		

Command	Parameters	Response
+MQTTSUB:0,6,"topic_test1",0 +MQTTSUB:0,6,"topic_test",0 OK Example 2, when MQTT client does not subscribe to topic_test previously: AT+MQTTSUB=0,"topic_test",0 Correct response: OK Example 3, when MQTT client subscribed to topic_test previously: AT+MQTTSUB=0,"topic_test",0 Correct response: ALREADY SUBSCRIBE OK Note: This command can be called multiple times to subscribe to different topics. When AT receives an MQTT message from a subscribed topic, it returns: +MQTTSUBRECV:<LinkID>,<"topic">,<data_length>,data		

6.10. AT+MQTTUNSUB

Table 6-10. Unsubscribing to MQTT topic

Command	Parameters	Response
Help command AT+MQTTUNSUB=?		+MQTTUNSUB=<LinkID>,<"topic">

Execution command
AT+MQTTUNSUB=<LinkID>,<"topic">
	<LinkID>: only supports link ID 0, integer
	<"topic">: MQTT topic, string parameter, maximum length is 128 bytes
		Execution result
		if topic is subscribed previously and the unsubscription is successful, AT returns:
		OK
		if topic is unsubscribed previously, AT returns:
		NO UNSUBSCRIBE

Example 1: when MQTT client subscribed to topic_test previously: AT+MQTTUNSUB=0,"topic_test" Correct response: OK Example 2: when MQTT client does not subscribe to topic_test previously: AT+MQTTUNSUB=0,"topic_test" Correct response:		

NO UNSUBSCRIBE OK Note: This command can be called multiple times to unsubscribe to different topics.

6.11. AT+MQTTCLEAN

Table 6-11. Closing MQTT connection and releasing resources

Command	Parameters	Response
Help command AT+MQTTCLEAN=?		+MQTTCLEAN=<LinkID>
Execution command AT+MQTTCLEAN=<LinkID>	<LinkID>: only supports link ID 0, integer	Execution result
Example: AT+MQTTCLEAN=0 Correct response: OK Note: When the MQTT client is passively disconnected, such as due to network disconnection or MQTT broker shutdown, a message of +MQTTDISCONNECTED:<LinkID> will be prompted.		

6.12. AT MQTT ERROR CODE

Table 6-12. AT MQTT error code

Type of error	Error code
AT_MQTT_NO_CONFIGURED	0x6001
AT_MQTT_NOT_IN_CONFIGURED_STATE	0x6002
AT_MQTT_UNINITIATED_OR_ALREADY_CLEAN	0x6003
AT_MQTT_ALREADY_CONNECTED	0x6004
AT_MQTT_MALLOC_FAILED	0x6005
AT_MQTT_NULL_LINK	0x6006
AT_MQTT_NULL_PARAMTER	0x6007
AT_MQTT_PARAMETER_COUNTS_IS_WRONG	0x6008
AT_MQTT_TLS_CONFIG_ERROR	0x6009
AT_MQTT_PARAM_PREPARE_ERROR	0x600A
AT_MQTT_CLIENT_START_FAILED	0x600B
AT_MQTT_CLIENT_PUBLISH_FAILED	0x600C
AT_MQTT_CLIENT_SUBSCRIBE_FAILED	0x600D

AT_MQTT_CLIENT_UNSUBSCRIBE_FAILED	0x600E
AT_MQTT_CLIENT_DISCONNECT_FAILED	0x600F
AT_MQTT_LINK_ID_READ_FAILED	0x6010
AT_MQTT_LINK_ID_VALUE_IS_WRONG	0x6011
AT_MQTT_SCHEME_READ_FAILED	0x6012
AT_MQTT_SCHEME_VALUE_IS_WRONG	0x6013
AT_MQTT_CLIENT_ID_READ_FAILED	0x6014
AT_MQTT_CLIENT_ID_IS_NULL	0x6015
AT_MQTT_CLIENT_ID_IS_OVERLENGTH	0x6016
AT_MQTT_USERNAME_READ_FAILED	0x6017
AT_MQTT_USERNAME_IS_NULL	0x6018
AT_MQTT_USERNAME_IS_OVERLENGTH	0x6019
AT_MQTT_PASSWORD_READ_FAILED	0x601A
AT_MQTT_PASSWORD_IS_NULL	0x601B
AT_MQTT_PASSWORD_IS_OVERLENGTH	0x601C
AT_MQTT_CERT_KEY_ID_READ_FAILED	0x601D
AT_MQTT_CERT_KEY_ID_VALUE_IS_WRONG	0x601E
AT_MQTT_CA_ID_READ_FAILED	0x601F
AT_MQTT_CA_ID_VALUE_IS_WRONG	0x6020
AT_MQTT_CA_LENGTH_ERROR	0x6021
AT_MQTT_CA_READ_FAILED	0x6022
AT_MQTT_CERT_LENGTH_ERROR	0x6023
AT_MQTT_CERT_READ_FAILED	0x6024
AT_MQTT_KEY_LENGTH_ERROR	0x6025
AT_MQTT_KEY_READ_FAILED	0x6026
AT_MQTT_PATH_READ_FAILED	0x6027
AT_MQTT_PATH_IS_NULL	0x6028
AT_MQTT_PATH_IS_OVERLENGTH	0x6029
AT_MQTT_VERSION_READ_FAILED	0x602A
AT_MQTT_KEEPALIVE_READ_FAILED	0x602B
AT_MQTT_KEEPALIVE_IS_NULL	0x602C
AT_MQTT_KEEPALIVE_VALUE_IS_WRONG	0x602D
AT_MQTT_DISABLE_CLEAN_SESSION_READ_FAILED	0x602E
AT_MQTT_DISABLE_CLEAN_SESSION_VALUE_IS_WRONG	0x602F
AT_MQTT_LWT_TOPIC_READ_FAILED	0x6030
AT_MQTT_LWT_TOPIC_IS_NULL	0x6031
AT_MQTT_LWT_TOPIC_IS_OVERLENGTH	0x6032
AT_MQTT_LWT_MSG_READ_FAILED	0x6033
AT_MQTT_LWT_MSG_IS_NULL	0x6034
AT_MQTT_LWT_MSG_IS_OVERLENGTH	0x6035
AT_MQTT_LWT_QOS_READ_FAILED	0x6036
AT_MQTT_LWT_QOS_VALUE_IS_WRONG	0x6037

AT_MQTT_LWT_RETAIN_READ_FAILED	0x6038
AT_MQTT_LWT_RETAIN_VALUE_IS_WRONG	0x6039
AT_MQTT_HOST_READ_FAILED	0x603A
AT_MQTT_HOST_IS_NULL	0x603B
AT_MQTT_HOST_IS_OVERLENGTH	0x603C
AT_MQTT_PORT_READ_FAILED	0x603D
AT_MQTT_PORT_VALUE_IS_WRONG	0x603E
AT_MQTT_RECONNECT_READ_FAILED	0x603F
AT_MQTT_RECONNECT_VALUE_IS_WRONG	0x6040
AT_MQTT_TOPIC_READ_FAILED	0x6041
AT_MQTT_TOPIC_IS_NULL	0x6042
AT_MQTT_TOPIC_IS_OVERLENGTH	0x6043
AT_MQTT_DATA_READ_FAILED	0x6044
AT_MQTT_DATA_IS_NULL	0x6045
AT_MQTT_DATA_IS_OVERLENGTH	0x6046
AT_MQTT_QOS_READ_FAILED	0x6047
AT_MQTT_QOS_VALUE_IS_WRONG	0x6048
AT_MQTT_RETAIN_READ_FAILED	0x6049
AT_MQTT_RETAIN_VALUE_IS_WRONG	0x604A
AT_MQTT_PUBLISH_LENGTH_READ_FAILED	0x604B
AT_MQTT_PUBLISH_LENGTH_VALUE_IS_WRONG	0x604C
AT_MQTT_RECV_LENGTH_IS_WRONG	0x604D
AT_MQTT_CREATE_SEMA_FAILED	0x604E
AT_MQTT_CREATE_EVENT_GROUP_FAILED	0x604F
AT_MQTT_URI_PARSE_FAILED	0x6050
AT_MQTT_IN_DISCONNECTED_STATE	0x6051
AT_MQTT_HOSTNAME_VERIFY_FAILED	0x6052
Note: MQTT error code is printed in the format: ERR CODE:0x<%08x>.	

7. AT HTTP command set

7.1. AT+HTTPCLIENT

Table 7-1. Sending HTTP client request

Command	Parameters	Response
Help command AT+HTTPCLIENT=?		+HTTPCLIENT=<method>,<content-type>,<"url">,[<"host">],[<"path">],<transport_type>[,<"data">][,<"http_req_header">][,<"http_req_header">][...]
Execution command AT+HTTPCLIENT=<method>,<content-type>,<"url">,[<"host">],[<"path">],<transport_type>[,<"data">][,<"http_req_header">][,<"http_req_header">][...]	<method>: method of HTTP client request, integer 1: HEAD 2: GET 3: POST <content-type>: data type of HTTP client request, integer 0: application/x-www-form-urlencoded 1: application/json 2: multipart/form-data 3: text/xml <"url">: HTTP URL, string parameter, when the <host> and <path> parameters are empty, this parameter will automatically override them [<"host">]: domain name or IP address, string parameter [<"path">]: HTTP path, string parameter <transport_type>: transport type of HTTP client request, integer 1: HTTP OVER TCP 2: HTTP OVER SSL [<"data">]: when <method> is a POST request, this parameter is the data sent to the HTTP server, otherwise, this parameter does not exist. integer [<"http_req_header">]: request	Execution result

<table>
<tr><td></td><td>headers sent to the server, up to 5 headers can be included, string parameter</td><td></td></tr>
<tr><td colspan="3">Example 1: HEAD request
AT+HTTPCLIENT=1,3,"https://www.baidu.com/","","",2
Correct response:
+HTTPCLIENT:277
OK

Example 2: GET request
AT+HTTPCLIENT=2,3,"https://www.baidu.com/","","",2
Correct response:
+HTTPCLIENT:29506,<!DOCTYPE html>
…(the specific content of the HTTP resource)
OK

Example 3: POST request
AT+HTTPCLIENT=3,3,"https://192.168.237.1/portal.html","","",2,"ssid=test&password=12345678"
Correct response:
SEND OK

Note:
If the total length of the command containing the URL exceeds 256 bytes, please use the AT+HTTPURLCFG command to pre-configure the URL, and set the <"url"> parameter in this command to "".
If the url parameter is not empty, the HTTP client will use it and ignore the host and path parameters.
If the url parameter is omitted or empty, the HTTP client will use the host and path parameters.
If the total length of the command containing the <"data"> parameter exceeds 256 bytes, please use the AT+HTTPCPOST command.
To set more HTTP request headers, please use the AT+HTTPCHEAD command.</td></tr>
</table>

7.2. AT+HTTPGETSIZE

Table 7-2. Getting the size of the HTTP resource

Command	Parameters	Response
Help command AT+HTTPGETSIZE=?		+HTTPGETSIZE=<"url">[,<tx size>][,<rx size>][,<timeout>]
Execution command AT+HTTPGETSIZE=<"url">[,<tx size>][,<rx size>][,<timeout>]	<"url">: HTTP URL, string parameter [<tx size>]: HTTP send buffer size, integer, unit: bytes, range: 0~10240, default value is 2048 [<rx size>]: HTTP receive buffer	Execution result +HTTPGETSIZE:<size>

	size, integer, unit: bytes, range: 0~10240, default value is 2048 [<timeout>]: timeout, integer, unit: milliseconds, range: 0~180000, default value is 5000 <size>: size of HTTP resource, integer	
Example: AT+HTTPGETSIZE="https://www.baidu.com",,2048 Correct response: +HTTPGETSIZE:277 OK Note: If the total length of the command containing the URL exceeds 256 bytes, please use the AT+HTTPURLCFG command to pre-configure the URL, and set the <"url"> parameter in this command to "". If you need to set HTTP request headers, please use the AT+HTTPCHEAD command.		

7.3. AT+HTTPCGET

Table 7-3. Getting HTTP resource

Command	Parameters	Response
Help command AT+HTTPCGET=?		+HTTPCGET=<"url">[,<tx size>][,<rx size>][,<timeout>]
Execution command AT+HTTPCGET=<"url">[,<tx size>][,<rx size>][,<timeout>]	<"url">: HTTP URL, string parameter [<tx size>]: HTTP send buffer size, integer, unit: bytes, range: 0~10240, default value is 2048 [<rx size>]: HTTP receive buffer size, integer, unit: bytes, range: 0~10240, default value is 2048 [<timeout>]: timeout, integer, unit: milliseconds, range: 0~180000, default value is 5000	Execution result +HTTPCGET:<size>,<data>
Example: AT+HTTPCGET="https://www.baidu.com",4096,4096 Correct response: +HTTPCGET:29506,<!DOCTYPE html> …(the specific content of the HTTP resource) OK		

Note:
If the total length of the command containing the URL exceeds 256 bytes, please use the AT+HTTPURLCFG command to pre-configure the URL, and set the <"url"> parameter in this command to "".
If you need to set HTTP request headers, please use the AT+HTTPCHEAD command.

7.4. AT+HTTPCPOST

Table 7-4. Posting HTTP data of the specified length

Command	Parameters	Response
Help command AT+HTTPCPOST=?		+HTTPCPOST=<"url">,<length>[,<http_req_header_cnt>][,<http_req_header>..<http_req_header>]

Execution command
AT+HTTPCPOST=<"url">,<length>[,<http_req_header_cnt>][,<http_req_header>..<http_req_header>]
	<"url">: HTTP URL, string parameter
	<length>: length of HTTP data for POST, integer
	[<http_req_header_cnt>]: count of [<http_req_header>] parameter, integer, the max count is 5
	[<http_req_header>]: request headers sent to the server, up to 5 headers can be included, string parameter
		Execution result
		OK
		> <input from keyboad>
		if POST successfully，AT returns:
		SEND OK
		if POST unsuccessfully，AT returns:
		SEND FAIL

Example:
AT+HTTPCPOST="https://192.168.237.1/portal.html",31
Correct response:
OK
>
SEND OK

Note:
If the total length of the command containing the URL exceeds 256 bytes, please use the AT+HTTPURLCFG command to pre-configure the URL, and set the <"url"> parameter in this command to "".
Data type of the client request of this command defaults to application/x-www-form-urlencoded.
If you need to set HTTP request headers, please use the AT+HTTPCHEAD command.

7.5. AT+HTTPCPUT

Table 7-5. Putting HTTP data of the specified length

Command	Parameters	Response
Help command AT+HTTPCPUT=?		+HTTPCPUT=<"url">,<length>[,<http_req_header_cnt>][,<http_req_header>..<http_req_header>]

Execution command
AT+HTTPCPUT=<"url">,<length>[,<http_req_header_cnt>][,<http_req_header>..<http_req_header>]
	<"url">: HTTP URL, string parameter
	<length>: length of HTTP data for PUT, integer
	[<http_req_header_cnt>]: count of [<http_req_header>] parameter, integer, the max count is 5
	[<http_req_header>]: request headers sent to the server, up to 5 headers can be included, string parameter
		Execution result
		OK
		> <input from keyboad>
		if PUT successfully，AT returns:
		SEND OK
		if PUT unsuccessfully，AT returns:
		SEND FAIL

Example: AT+HTTPCPUT="https://192.168.237.1/portal.html",31 Correct response: OK > SEND OK Note: If the total length of the command containing the URL exceeds 256 bytes, please use the AT+HTTPURLCFG command to pre-configure the URL, and set the <"url"> parameter in this command to "". If you need to set HTTP request headers, please use the AT+HTTPCHEAD command.		

7.6. AT+HTTPURLCFG

Table 7-6. Querying or setting long HTTP URL

Command	Parameters	Response
Help command AT+HTTPURLCFG=?		+HTTPURLCFG=<url length>
Query command AT+HTTPURLCFG?		[+HTTPURLCFG:<url length>,<data>]
Execution command AT+HTTPURLCFG=<url	<url length>: length of HTTP URL, integer, unit: bytes	Execution result OK

Command	Parameters	Response
length>	0: clear HTTP URL configuration 8~8192: length of the configured HTTP URL <data>: HTTP URL data, string type	> <input from keyboad> SET OK
Example 1: AT+HTTPURLCFG=21 Correct response: OK > SET OK Example 2: AT+HTTPURLCFG? Correct response: +HTTPURLCFG:21,https://www.baidu.com OK		

7.7. AT+HTTPCHEAD

Table 7-7. Querying or setting HTTP request header

Command	Parameters	Response
Help command AT+HTTPCHEAD=?		+HTTPCHEAD=<req_header_len>
Query command AT+HTTPCHEAD?		+HTTPCHEAD:<index>,<"req_header">
Execution command AT+HTTPCHEAD=<req_header_len>	<index>: index value of the HTTP request header, integer <"req_header">: HTTP request header, string type <req_header_len>: length of HTTP request header, integer, unit: bytes 0: clear all configured HTTP request headers other values: length of the newly configured HTTP request header	Execution result OK > <input from keyboad>
Example 1: AT+HTTPCHEAD=18 Correct response:		

Command	Parameters	Response
OK > OK Example 2: AT+HTTPCHEAD? Correct response: +HTTPCHEAD:0,"Range: bytes=0-255" OK Note: The format of the HTTP request header is key: value. This command can only set one HTTP request header at a time, but it can be called multiple times to set different HTTP request headers. The HTTP request headers set by this command are global; once set, all HTTP commands will carry these headers. If the key in the HTTP request header set by this command is the same as the key in the request header of other HTTP commands, the header set by this command will be used.		

8. AT BLE command set

8.1. AT+BLEENABLE

Table 8-1. Enabling BLE

Command	Parameters	Response
Execution command AT+BLEENABLE		Execution result
Example 1: AT+BLEENABLE Correct response 1: OK		

8.2. AT+BLEDISABLE

Table 8-2. Disabling BLE

Command	Parameters	Response
Execution command AT+BLEDISABLE		Execution result
Example 1: AT+BLEDISABLE Correct response 1: OK		

8.3. AT+BLENAME

Table 8-3. Setting the name

Command	Parameters	Response
Help command AT+BLENAME=?		+BLENAME=<name>
Query command AT+BLENAME?		+BLENAME:<name>
Execution command AT+BLENAME=<name>	<name>: device name	Execution result
Example 1: AT+BLENAME? Correct response 1: +BLENAME:GD-BLE-01:23:45:67:89:ab OK		

Command	Parameters	Response
Example 2: AT+BLENAME=test Correct response 2: OK Note: 1.The name will also change synchronously in the advertising after setting.		

8.4. AT+BLEADVSTART

Table 8-4. Starting BLE advertising

Command	Parameters	Response
Help command AT+BLEADVSTART=?		+BLEADVSTART=<type>,[intv],[ch_map],[prop],[pri_phy],[sec_phy],[wl_enable],[own_addr_type],[disc_mode],[addr_type],[addr]
Execution command AT+BLEADVSTART=<type>,[intv],[ch_map],[prop],[pri_phy],[sec_phy],[wl_enable],[own_addr_type],[disc_mode],[addr_type],[addr]	<type>: advertising type [intv]: advertising interval [ch_map]: channel map [property]: property configuration [pri_phy]: primary channel phy [sec_phy]: secondary channel phy [wl_enable]: whether to enable white list [own_addr_type]: own address type [disc_mode]: discovery mode [addr_type]: target device address type [addr]: target device address	Execution result
Example 1: AT+BLEADVSTART=0 Correct response 1: OK		

8.5. AT+BLEADVSTOP

Table 8-5. Stopping BLE advertising

Command	Parameters	Response
Help command		+BLEADVSTOP=<adv_idx>

Command	Parameters	Response
AT+BLEADVSTOP=?		
Execution command AT+BLEADVSTOP=< adv_idx >	<adv_idx>: advertising index	Execution result
Example 1: AT+BLEADVSTOP=0 Correct response 1: OK		

8.6. AT+BLEADVDATA

Table 8-6. Setting the advertising data

Command	Parameters	Response
Help command AT+BLEADVDATA=?		+BLEADVDATA=<data>
Execution command AT+BLEADVDATA=<data>	<data>: advertising data, Hex string. For example, AT+BLEADVDATA="020106020941" represents setting the advertising data to "0x02 0x01 0x06 0x02 0x09 0x41".	Execution result
Example 1: AT+BLEADVDATA="020106020941" Correct response 1: OK		

8.7. AT+BLEADVDATAEX

Table 8-7. Setting the advertising data

Command	Parameters	Response
Help command AT+BLEADVDATAEX=?		+BLEADVDATAEX =<dev_name>,<uuid>,<manufacturer_data>,<include_power>
Execution command AT+BLEADVDATAEX =<dev_name>,<uuid>,<manufacturer_data>,<include_power>	<dev_name>: device name <uuid>: service uuid <manufacturer_data>: manufacturer data <include_power>: whether to include power	Execution result
Example 1: AT+BLEADVDATAEX="test","a002","2b0c112233",1		

Command	Parameters	Response
Correct response 1: OK		

8.8. AT+BLESCANRSPDATA

Table 8-8. Setting the scan response data

Command	Parameters	Response
Help command AT+BLESCANRSPDATA=?		+BLESCANRSPDATA=<data>
Execution command AT+BLESCANRSPDATA=<data>	<data>: advertising data, Hex string. For example, AT+BLESCANRSPDATA =” 020941”represents setting the advertising data to "0x02 0x09 0x41".	Execution result
Example 1: AT+BLESCANRSPDATA="020941" Correct response 1: OK		

8.9. AT+BLEPASSTH

Table 8-9. Enabling passthrough mode

Command	Parameters	Response
Execution command AT+BLEPASSTH		Execution result
Example 1: Start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Enable passthrough mode. AT+BLEPASSTH Stop Sending data. +++ No response.		

8.10. AT+BLEPASSTHAUTO

Table 8-10. Enabling passthrough mode automatically

<table>
<tr><th>Command</th><th>Parameters</th><th>Response</th></tr>
<tr><td>Help command
AT+BLEPASSTHAUTO=?</td><td></td><td>+BLEPASSTHAUTO=<enable&gt;</td></tr>
<tr><td>Execution command
AT+BLEPASSTHAUTO=<enable&gt;</td><td>&lt;enable>: whether to enter passthrough mode automatically</td><td>Execution result</td></tr>
<tr><td colspan="3">NOTE:
Master and slave automatically enter pass-through mode via the same command.

Example 1:
Enable passthrough mode automatically.
AT+BLEPASSTHAUTO=1

Start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device.
Passthrough mode is enabled automatically.

Stop Sending data.
+++
Correct response 1:
OK</td></tr>
</table>

8.11. AT+BLEPASSTHCLI

Table 8-11. Enabling passthrough mode

<table>
<tr><th>Command</th><th>Parameters</th><th>Response</th></tr>
<tr><td>Execution command
AT+BLEPASSTHCLI</td><td></td><td>Execution result</td></tr>
<tr><td colspan="3">Example 1:
Start BLE advertising, AT+BLEADVSTART=0 or Initiate a BLE connection, AT+BLECONN=0,<addr> then the target device connects to local device.

Enable passthrough mode.
AT+BLEPASSTHCLI

Stop Sending data.
+++
No response.</td></tr>
</table>

8.12. AT+BLESCANPARAM

Table 8-12. Setting scan parameters

Command	Parameters	Response
Help command AT+BLESCANPARAM=?		+BLESCANPARAM=<type>,<own_addr_type>,<dup_filt_pol>,<scan_intv_1m>,<scan_win_1m>
Query command AT+BLESCANPARAM?		+BLESCANPARAM:<type>,<own_addr_type>,<dup_filt_pol>,<scan_intv_1m>,<scan_win_1m>
Execution command AT+BLESCANPARAM:<type>,<own_addr_type>,<dup_filt_pol>,<scan_intv_1m>,<scan_win_1m>	<type>: scan type <own_addr_type>: local address type <dup_filt_pol>: duplicate packet filtering policy <scan_intv_1m >: scan interval in unit of 625us for 1M PHY <scan_win_1m>: scan window in unit of 625us for 1M PHY	Execution result
Example 1: AT+BLESCANPARAM? Correct response 1: +BLESCANPARAM:0,0,1,160,32 OK Example 2: AT+BLESCANPARAM=0,0,1,160,48 Correct response 2: OK		

8.13. AT+BLESCAN

Table 8-13. Starting scan

Command	Parameters	Response
Help command AT+BLESCAN=?		+BLESCAN=<enable>
Execution command AT+BLESCAN=<enable>	<enable> ： whether to start scan	Execution result
Example 1:		

Command	Parameters	Response
AT+BLESCAN=1 Correct response 1: OK		

8.14. AT+BLESYNC

Table 8-14. Starting or cancelling BLE synchronization

Command	Parameters	Response
Help command AT+BLESYNC=?		+BLESYNC=<enable>,<addr_type>, <addr>
Execution command AT+BLESYNC=<enable>,<addr_type>,<addr>	<enable>: whether to start synchronization <addr_type>: address type <addr>: target device address	Execution result
Example 1: Start BLE synchronization. AT+BLESYNC=1,0,AB:89:67:45:23:01 Correct response 1: OK Example 2: Cancel BLE synchronization. AT+BLESYNC=0 Correct response 2: OK		

8.15. AT+BLESYNCSTOP

Table 8-15. Stopping BLE synchronization

Command	Parameters	Response
Execution command AT+BLESYNCSTOP		Execution result
Example 1: AT+BLESYNCSTOP Correct response 1: OK		

8.16. AT+BLECONN

Table 8-16. Initiating a BLE connection

Command	Parameters	Response
Help command AT+BLECONN=?		+BLECONN=<addr_type>,<addr>
Execution command AT+BLECONN=<type>,<addr>	<addr_type>: address type <addr>: target device address	Execution result
Example 1: Peer device start advertising AT+BLECONN=0, AB:89:67:45:23:01(peer addr) Correct response 1: OK		

8.17. AT+BLESCONNPARAM

Table 8-17. Setting or querying the connection parameters

Command	Parameters	Response
Help command AT+BLECONNPARAM=?		+BLECONNPARM=<conn_idx>,<interval>,<latancy>,<supv_to>
Query command AT+BLECONNPARAM?		+BLECONNPARAM:<conn_idx>,<interval>,<latancy>,<supv_to>
Execution command AT+BLECONNPARAM=<conn_idx>,<interval>,<latancy>,<supv_to>	<conn_idx>: connection index < interval >: connection interval < latancy >: slave latency < supv_to >: supervision timeout	Execution result
Example 1: Query the connection parameters. Initiate a BLE connection, AT+BLECONN=0,<addr>, or start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Query the connection parameters. AT+BLECONNPARAM? Correct response 1: +BLECONNPARAM:0,40,0,500 OK		

Command	Parameters	Response
Example 2: Set the connection parameters. Initiate a BLE connection, AT+BLECONN=0,<addr>, or start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Set the connection parameters. AT+BLECONNPARAM=0,50,20,500 Correct response 2: OK		

8.18. AT+BLEDISCONN

Table 8-18. Disconnecting the established BLE connection

Command	Parameters	Response
Help command AT+BLEDISCONN=?		+BLEDISCONN=<conn_idx>
Execution command AT+BLEDISCONN=<conn_idx>	<conn_idx>: connection index	Execution result
Example 1: Initiate a BLE connection, AT+BLECONN=0,<addr>, or start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Disconnect the established BLE connection. AT+BLEDISCONN=0 Correct response 1: OK		

8.19. AT+BLEMTU

Table 8-19. Updating or querying the mtu

Command	Parameters	Response
Help command AT+BLEMTU=?		+BLEMTU=<conn_idx>,<pref_mtu>
Query command AT+BLEMTU?		+BLEMTU:<conn_idx>,<mtu_size>
Execution command	<conn_idx>: connection index	Execution result

Command	Parameters	Response
AT+BLEMTU=<conn_idx>, <pref_mtu>	<pref_mtu>:prefer mtu	
Example 1: Query the mtu. Initiate a BLE connection, AT+BLECONN=0,<addr>, or start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Query the mtu. AT+BLEMTU? Correct response 1: +BLEMTU:0,23 OK Example 2: Update the mtu. Initiate a BLE connection. AT+BLECONN=0,<addr> Update the mtu. AT+BLEMTU=0,1000 Correct response 2: OK		

8.20. AT+BLEPHY

Table 8-20. Updating or querying phy

Command	Parameters	Response
Help command AT+BLEPHY=?		+BLEPHY=<conn_idx>,<tx_phy>,<rx_phy>,<phy_opt>
Query command AT+BLEPHY？		+BLEPHY:<conn_idx>,<tx_phy>,<rx_phy>
Execution command AT+BLEPHY=<conn_idx>,<tx_phy>,<rx_phy>,<phy_opt>	<conn_idx>: connection index <tx_phy>: tx phy <rx_phy>: rx phy <phy_opt>: coded phy option	Execution result
Example 1: Query phy. Initiate a BLE connection, AT+BLECONN=0,<addr>, or start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Query phy. AT+BLEPHY?		

Command	Parameters	Response
Correct response 1: +BLEPHY:0,0,0 OK Example 2: Update phy. Initiate a BLE connection, AT+BLECONN=0,<addr>, or start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Update phy. AT+BLEPHY=0,1,1,0 Correct response 2: OK		

8.21. AT+BLEDATALEN

Table 8-21. Data length extension

Command	Parameters	Response
Help command AT+BLEDATALEN=?		+BLEDATALEN=<conn_idx>,<tx_oct>
Execution command AT+BLEDATALEN=<conn_idx>,<tx_oct>	<conn_idx>: connection index <tx_oct>: tx payload octets	Execution result
Example 1: Initiate a BLE connection, AT+BLECONN=0,<addr>, or start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Data length extension. AT+BLEDATALEN=0,200 Correct response 1: OK		

8.22. AT+BLEADDR

Table 8-22. Querying or setting the BLE bd address

Command	Parameters	Response
Help command AT+BLEADDR=?		+BLEADDR=<bd_addr>

Command	Parameters	Response
Query command AT+BLEADDR?		+BLEADDR:<bd_addr>
Execution command AT+BLEADDR=<bd_addr>	< bd_addr >: ble bd address	Execution result
Example 1: Query the ble bd address AT+BLEADDR? Correct response 1: +BLEBDADDR:77:66:55:44:33:22 OK Example 2: Set the ble bd address AT+BLEADDR=22:33:44:55:66:77 Correct response 2: OK		

8.23. AT+BLESETAUTH

Table 8-23. Setting the authentication

Command	Parameters	Response
Help command AT+BLESETAUTH=?		+BLESETAUTH=<bond>,<mitm>,<sc>,<iocap>,<oob>,<key_size>

Command	Parameters	Response
Execution command AT+BLESETAUTH=<bond>,<mitm>,<sc>,<iocap>,<oob>,<key_size>	< bond > : bonding flag 0x00: no bonding 0x01: bonding <mitm>: mitm flag 0x00: mitm protection not required 0x01: mitm protection required < sc >: secure connections flag 0x00: secure connections pairing is not supported 0x01: secure connections pairing is supported < iocap>: io capability to set 0x00: display only 0x01: display yes no 0x02: keyboard only 0x03: no input no output 0x04: keyboard display <oob>: oob flag for authention [key size]: encryption key size requirement, default is 16 if not set	Execution result
Example 1: AT+BLESETAUTH=1,0,0,3,0,16 Correct response 1: OK		

8.24. AT+BLEPAIR

Table 8-24. Starting pairing

Command	Parameters	Response
Help command AT+BLEPAIR=?		+BLEPAIR=<conidx>
Execution command AT+BLEPAIR=<conidx>	<conidx>: connection index	Execution result
Example 1: Initiate a BLE connection, AT+BLECONN=0,<addr>, or start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Start pairing. AT+BLEPAIR=0		

Command	Parameters	Response
Correct response 1: OK		

8.25. AT+BLEENCRYPT

Table 8-25. Starting encrypting

Command	Parameters	Response
Help command AT+BLEENCRYPT=?		+BLEENCRYPT =<conidx >
Execution command AT+BLEENCRYPT=<conidx >	< conidx >: connection index	Execution result
Example 1: Start encrypting(devices have been paired before). Initiate a BLE connection, AT+BLECONN=0,<addr>, or start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Start encrypting. AT+BLEENCRYPT=0 Correct response 1: OK		

8.26. AT+BLEPASSKEY

Table 8-26. Entering the passkey

Command	Parameters	Response
Help command AT+BLEPASSKEY=?		+BLEPASSKEY=<conidx>,<passkey>
Execution command AT+BLEPASSKEY=<conidx>,<passkey>	< conidx >: connection index <passkey>: passkey	Execution result
Example 1: Set the authentication. AT+BLESETAUTH=1,1,0,2,0,16		

Command	Parameters	Response
Initiate a BLE connection, AT+BLECONN=0,<addr>, or start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Enter the passkey. AT+BLEPASSKEY=0,123456(enter the value displayed) Correct response 1: OK		

8.27. AT+BLECOMPARE

Table 8-27. Entering the numeric comparison result

Command	Parameters	Response
Help command AT+BLECOMPARE=?		+BLECOMPARE=<conidx>,<value>
Execution command AT+BLECOMPARE=<conidx>,<value>	<conidx >: connection index <value>: numeric value	Execution result
Example 1: Set the authentication. AT+BLESETAUTH=1,1,1,4,0,16 Initiate a BLE connection, AT+BLECONN=0,<addr>, or start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Enter the numeric comparison result. AT+BLECOMPARE=0,1 Correct response 1: OK		

8.28. AT+BLELISTENCDEV

Table 8-28. Listing the bond devices

Command	Parameters	Response
Query command AT+BLELISTENCDEV?		+BLEADDR:<dev_idx><addr>

Command	Parameters	Response
Example 1: Devices have been paired before. AT+BLELISTENCDEV? Correct response 1: +BLELISTENCDEV=0,AB:89:67:45:23:01 +BLELISTENCDEV=1,D0:20:DD:EE:5C:3C OK		

8.29. AT+BLECLEARENCDEV

Table 8-29. Clearing the bond devices

Command	Parameters	Response
Help command AT+BLECLEARENCDEV=?		+BLECLEARENCDEV=<dev_idx>
Execution command AT+BLECLEARENCDEV=<dev_idx>	<dev_idx>: device index	Execution result
Example 1: Devices have been paired before. AT+BLECLEARENCDEV=0 Correct response 1: OK		

8.30. AT+BLEGATTSSVC

Table 8-30. Listing the devices registered locally

Command	Parameters	Response
Query command AT+BLEGATTSSVC?		+BLEGATTSSVC:<svc_id><uuid>
Example 1: AT+BLEGATTSSVC? Correct response 1: +BLEGATTSSVC:0,0000000000000000000000000000180A,1 +BLEGATTSSVC:1,00001111000000000123456789ABCDEF,1 +BLEGATTSSVC:2,00000000000000000000000000000101,1 OK		

8.31. AT+BLEGATTSCHAR

Table 8-31. Listing the characteristic of the service

Command	Parameters	Response
Help command AT+BLEGATTSCHAR=?		+BLEGATTSCHAR=<svc_idx>
Execution command AT+BLEGATTSCHAR=<svc_idx>	< svc_idx >: service index	Execution result +BLEGATTSCHAR:<uuid ><value_index>
Example 1: AT+BLEGATTSCHAR=1 Correct response 1: +BLEGATTSCHAR:00002222000000000123456789ABCDEF,2 +BLEGATTSCHAR:00003333000000000123456789ABCDEF,4 +BLEGATTSCHAR:00004444000000000123456789ABCDEF,6 OK		

8.32. AT+BLEGATTSDESC

Table 8-32. Listing the descriptor of the characteristic

Command	Parameters	Response
Help command AT+BLEGATTSDESC=?		+BLEGATTSDESC=<svc_idx>,<char_idx>
Execution command AT+BLEGATTSDESC=<svc_idx>,<char_idx>	< svc_idx >: service index <char_idx>: characteristic index	Execution result +BLEGATTSDESC:<uuid ><desc_idx>
Example 1: AT+BLEGATTSDESC=1,6 Correct response 1: +BLEGATTSDESC:00000000000000000000000000002902,7 OK		

8.33. AT+BLEGATTSLISTALL

Table 8-33. Listing the information of all local services

Command	Parameters	Response
Query command AT+BLEGATTSLISTALL?		+BLEGATTSSVC:<svc_id><uuid> +BLEGATTSCHAR:<uuid ><value_index> +BLEGATTSDESC:<uuid ><desc_idx>
Example 1: AT+BLEGATTSLISTALL? Correct response 1: +BLEGATTSCHAR:00000000000000000000000000002A29,2 +BLEGATTSCHAR:00000000000000000000000000002A24,4 +BLEGATTSCHAR:00000000000000000000000000002A25,6 +BLEGATTSCHAR:00000000000000000000000000002A27,8 +BLEGATTSCHAR:00000000000000000000000000002A26,10 +BLEGATTSCHAR:00000000000000000000000000002A28,12 +BLEGATTSCHAR:00000000000000000000000000002A23,14 +BLEGATTSCHAR:00000000000000000000000000002A2A,16 +BLEGATTSCHAR:00000000000000000000000000002A50,18 +BLEGATTSSVC:1,00001111000000000123456789ABCDEF,1 +BLEGATTSCHAR:00002222000000000123456789ABCDEF,2 +BLEGATTSCHAR:00003333000000000123456789ABCDEF,4 +BLEGATTSCHAR:00004444000000000123456789ABCDEF,6 +BLEGATTSDESC:00000000000000000000000000002902,7 +BLEGATTSSVC:2,00000000000000000000000000000101,1 +BLEGATTSCHAR:00000000000000000000000000000102,2 +BLEGATTSCHAR:00000000000000000000000000000103,4 +BLEGATTSDESC:00000000000000000000000000002902,5 OK		

8.34. AT+BLEGATTSNTF

Table 8-34. Sending notification

Command	Parameters	Response
Help command AT+BLEGATTSNTF=?		+BLEGATTSNTF=<conn_idx>,<svc_id>,<char_idx>,<tx_len>
Execution command AT+BLEGATTSNTF=<conn_idx>,<svc_id>,<char_idx>,<tx_len>	<conn_idx>: connection index <svc_id>: service id <char_idx>: characteristic index <tx_len>: data length	Execution result

Command	Parameters	Response
Example 1: Start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Send notification. AT+BLEGATTSNTF=0,1,6,5 > Enter AAAAA(the target device will receive the data). Correct response 1: OK		

8.35. AT+BLEGATTSIND

Table 8-35. Sending indication

Command	Parameters	Response
Help command AT+BLEGATTSIND=?		+BLEGATTSIND=<conn_idx>,<svc_id>,<char_idx>,<tx_len>
Execution command AT+BLEGATTSIND=<conn_idx>,<svc_id>,<char_idx>,<tx_len>	<conn_idx>: connection index <svc_id>: service id <char_idx>: characteristic index <tx_len>: data length	Execution result
Example 1: Start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Send indication. AT+BLEGATTSIND=0,1,6,5 > Enter AAAAA(the target device will receive the data). Correct response 1: OK		

8.36. AT+BLEGATTSSETATTRVAL

Table 8-36. Setting the value of the characteristic

Command	Parameters	Response
Help command AT+BLEGATTSSETATTRVAL=?		+BLEGATTSSETATTRVAL=<conn_idx>,<svc_id>,<char_idx>,<tx_len>
Execution command AT+BLEGATTSSETATTRVAL=<conn_idx>,<svc_id>,<char_idx>,<tx_len>	<conn_idx>: connection index <svc_id>: service id <char_idx>: characteristic index <tx_len>: data length	Execution result
Example 1: Start BLE advertising, AT+BLEADVSTART=0, then the target device connects to local device. Set the value of the characteristic AT+BLEGATTSSETATTRVAL=0,1,4,5 > Enter AAAAA(local data changed). Correct response 1: OK		

8.37. AT+BLEGATTCDISCSVC

Table 8-37. Discovering the service

Command	Parameters	Response
Help command AT+BLEGATTCDISCSVC=?		+BLEGATTCDISCSVC=<conn_idx>,<start_hdl>,<end_hdl>
Execution command AT+BLEGATTCDISCSVC=<conn_idx>,<start_hdl>,<end_hdl>	<conn_idx>: connection index < start_hdl >: start attribute handle < end_hdl >: end attribute handle	Execution result +BLEGATTCDISCSVC:<start_hdl >,<end_hdl >,<uuid>
Example 1: Initiate a BLE connection. AT+BLECONN=0,<addr> Discover the service. AT+BLEGATTCDISCSVC=0,1,ffff		

Command	Parameters	Response
Correct response 1: +BLEGATTCDISCSVC:1,8,00001111000000000123456789ABCDEF +BLEGATTCDISCSVC:9,14,00000000000000000000000000000101 +BLEGATTCDISCSVC:16,25,00000000000000000000000000001801 +BLEGATTCDISCSVC:32,40,00000000000000000000000000001800 +BLEGATTCDISCSVC:43,61,0000000000000000000000000000180A OK		

8.38. AT+BLEGATTCDISCCHAR

Table 8-38. Discovering the characteristic

Command	Parameters	Response
Help command AT+BLEGATTCDISCCHAR=?		+BLEGATTCDISCCHAR=<conn_idx>,<start_hdl>,<end_hdl>
Execution command AT+BLEGATTCDISCCHAR=<conn_idx>,<start_hdl>,<end_hdl>	<conn_idx>: connection index < start_hdl >: start attribute handle < end_hdl >: end attribute handle	Execution result +BLEGATTCDISCCHAR:<char_hdl >,<val_hdl >,<prop >,<uuid>
Example 1: Initiate a BLE connection. AT+BLECONN=0,<addr> Discover the characteristic. AT+BLEGATTCDISCCHAR=0,1,ffff Correct response 1: +BLEGATTCDISCCHAR:2,3,2,00002222000000000123456789ABCDEF +BLEGATTCDISCCHAR:4,5,12,00003333000000000123456789ABCDEF +BLEGATTCDISCCHAR:6,7,16,00004444000000000123456789ABCDEF +BLEGATTCDISCCHAR:10,11,12,00000000000000000000000000000102 +BLEGATTCDISCCHAR:12,13,16,00000000000000000000000000000103 +BLEGATTCDISCCHAR:17,18,32,00000000000000000000000000002A05 +BLEGATTCDISCCHAR:20,21,10,00000000000000000000000000002B29 +BLEGATTCDISCCHAR:22,23,2,00000000000000000000000000002B2A +BLEGATTCDISCCHAR:24,25,2,00000000000000000000000000002B3A +BLEGATTCDISCCHAR:33,34,10,00000000000000000000000000002A00 +BLEGATTCDISCCHAR:35,36,10,00000000000000000000000000002A01 +BLEGATTCDISCCHAR:37,38,2,00000000000000000000000000002A04 +BLEGATTCDISCCHAR:39,40,2,00000000000000000000000000002AA6 +BLEGATTCDISCCHAR:44,45,2,00000000000000000000000000002A29 +BLEGATTCDISCCHAR:46,47,2,00000000000000000000000000002A24 OK		

8.39. AT+BLEGATTCDISCDESC

Table 8-39. Discovering the descriptor

Command	Parameters	Response
Help command AT+BLEGATTCDISCDESC=?		+BLEGATTCDISCDESC=<conn_idx>,<start_hdl>,<end_hdl>
Execution command AT+BLEGATTCDISCDESC=<conn_idx>,<start_hdl>,<end_hdl>	<conn_idx>: connection index < start_hdl >: start attribute handle < end_hdl >: end attribute handle	Execution result +BLEGATTCDISCDESC:<desc_hdl >,<uuid>
Example 1: Initiate a BLE connection. AT+BLECONN=0,<addr> Discover the descriptor. AT+BLEGATTCDISCDESC=0,1,ffff Correct response 1: +BLEGATTCDISCDESC:8,00000000000000000000000000002902 +BLEGATTCDISCDESC:14,00000000000000000000000000002902 +BLEGATTCDISCDESC:19,00000000000000000000000000002902 OK		

8.40. AT+BLEGATTCRD

Table 8-40. Reading attribute value

Command	Parameters	Response
Help command AT+BLEGATTCRD=?		+BLEGATTCRD=<conn_idx>,<handle>,<max_len>
Execution command AT+BLEGATTCRD=<conn_idx>,<handle>,<max_len>	<conn_idx>: connection index < handle >: attribute handle < max_len >: max length	Execution result +BLEGATTCRD:<conn_idx >,<length>,<data>
Example 1: Initiate a BLE connection. AT+BLECONN=0,<addr> Read attribute value AT+BLEGATTCRD=0,3,100 Correct response 1:		

Command	Parameters	Response
+BLEGATTCRD:0,2,2222 OK		

8.41. AT+BLEGATTCWR

Table 8-41. Writing attribute value

Command	Parameters	Response
Help command AT+BLEGATTCWR=?		+BLEGATTCWR=<conn_idx>,<handle>,<write_type>,<len>
Execution command AT+BLEGATTCWR=<conn_idx>,<handle>,<write_type>,<len>	<conn_idx>: connection index < handle >: attribute handle < write_type >: write type <len>: write length	Execution result
Example 1: Initiate a BLE connection. AT+BLECONN=0,<addr> Write attribute value. AT+BLEGATTCWR=0,5,0,5 > Enter AAAAA(the target device will receive the data). Correct response 1: OK		

9. Revision history

Table 9-1. Revision history

Revision No.	Description	Date
1.0	Initial release.	Nov.24.2023
1.1	Add new Command: AT+TRANSINTVL and AT+CIPMODE, and extend AT+CIPSEND command to support passthrough transmission mode.	Jul.16.2024
1.2	Add new AT Command about BLE.	Oct.8.2024
1.3	Add new AT Command about BLE.	Mar.19.2025
1.4	Add new AT command about MQTT and HTTP, and extend AT command about basic command and TCPIP.	Nov.19.2025

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company according to the laws of the People’s Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.