L503C-Y/L511C-Y Custom AT Command User Guide

LTE Module Series

Version： V1.4

Date： 2022-12-09

Shanghai Mobiletek Communication Ltd

Notice

Some features of the product and its accessories described herein rely on the software installed, capacities and settings of local network, and therefore may not be activated or may be limited by local network operators or network service providers.
Thus, the descriptions herein may not exactly match the product or its accessories which you purchase. Shanghai Mobiletek Communication Ltd reserves the right to change or modify any information or specifications contained in this manual without prior notice and without any liability.

Copyright

DISCLAIMER

ALL CONTENTS OF THIS MANUAL ARE PROVIDED “AS IS”. EXCEPT AS REQUIRED BY APPLICABLE LAWS, NO WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDINGBUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE MADE IN RELATION TO THE ACCURACY, RELIABILITY OR CONTENTS OF THIS MANUAL.
TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL SHANGHAI MOBILETEKCOMMUNICATION LTD BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES, OR LOSS OF PROFITS, BUSINESS, REVENUE, DATA, GOODWILL SAVINGS OR ANTICIPATED SAVINGS REGARDLESS OF WHETHER SUCH LOSSES ARE FORSEEABLE OR NOT.

Revision History

Date	Version	Description of chage	Author
2022-09-14	V1.0	Initial	ym.shu
2022-10-14	V1.1	Add AT+QGSN, AT+QLTS, AT+QCELLEX=1 and AT+QSSLRECV command	ym.shu
2022-10-27	V1.2	Remove some commands	LI
2022-11-03	V1.3	Add AT+CBAUD command	ym.shu
2022-12-09	V1.4	Modify AT+CTZU and AT+QSSLCFG command	LI

1. Contents

1. Introduction

1.1 Overview

This document introduces the supported AT command set of L503C-Y/L511C-Y project.

We don‘t suggest using proprietary command in a multiple command. There might be abnormal situation occurs.

1.2 References

[1] 3GPP TS 27.007 V3.13.0 (2003-03)

[2] ETSI TS 27.005 V3.1.0 (2000-01)

[3] ITU-T V.25 ter (07/1997)

2. General Commands

2.1. AT+GSN Request TA Serial Number Identification

This command is used to request TA Serial Number Identification (serialNumber)

Execution Command **AT+GSN**	Response **<IMEI>** **OK** or **ERROR**

2.2. AT+QCFG Preferred Mode Selection

This command is used to preferred mode selection

Test Command **AT+QCFG=?**	Response **+QCFG: " ",(0,1)** **OK**
Read Command **AT+QCFG?**	Response **+QCFG: <path>,<value>** **OK**
Set Command **AT+QCFG=<path>[,<value>]**	Response **OK** **or** **ERROR**

Parameters are defined below:

Parameter note:	Description
<path>	The default path is “urc/cache”
<value>	integer type 0 URC reporting is prohibited 1 Enable URC reporting

2.3. AT+CTZU Automatic time zone update

This command is set the module to automatically time zone update

Test Command **AT+CTZU=?**	Response +CTZU: (0,1,3) **OK**
Set Command **AT+CTZU=<value>**	Response **OK** or **ERROR**
Read Command **AT+CTZU?**	Response **+CTZU: <value>** **OK**

Parameters are defined below:

Parameter note:	Description
<value>	integer type 0 Off 1 Enable 3 retutn RTC time

2.4. AT+GMM Request TA model identification

This command requests TA model identification (may equal to +CGMM)

Test Command **AT+GMM=?**	Response **+GMM:<list of supported technologies>,<module>** **OK**
Execution Command **AT+GMM**	Response **<module identification>** **OK**

2.5. AT+GMR Request revision identification

This command request TA revision identification (may equal to +CGMR)

Test Command **AT+GMR=?**	Response **OK**
Execution Command **AT+GMR**	Response **<Revision:<version>** **OK**

2.6. AT+QENG Get Nearby Cell Information

This command is used to get nearby cell information

Set Command **AT+QENG="servingcell/neighbourcell"**	Response servingcell: Nearby cell information for LTE +QENG:"servingcell",<state>,"LTE",<istdd>,<mcc>,<mnc>,<cellid>,<pcid>,<earfcn>,<freq_bandind_ind>,<ul_bandwidth>,<dl_bandwidth>,<tac>,<rsrp>,<rsrq>,<rssi>,<sinr>,<srxlev> **OK** or **ERROR** neighbourcell: +QENG: "neighbourcell inter","LTE",<earfcn>,<PCI>,<RSRQ>,<RSRP>,-,-,<srxlev>,<threshX_low>,<threshX_high>,<cell_resel_priority>[…] **OK** or **ERROR**
Test Command **AT+QENG=?**	Response +QENG: ("servingcell","neighbourcell") **OK**

2.7. AT+QCELLINFO Get Serving Cell and Neighbor Cell information

This command is used to get serving cell and neighbor cell information.

TEST Command **AT+QCELLINFO=?**	Response **OK**

Read Command **AT+QCELLINFO?**	Response serving cell and neighbor cell for LTE +QCELLINFO: "servingcell","LTE",<MCC>,<MNC>,<TAC>,<cellID>,<PCI>,<RX_lev> ,<RSSI>,<earfcn> [+QCELLINFO: "neighbourcell intra","LTE",<MCC>,<MNC>,<TAC>,<cellID>,<PCI>,<RX_lev>,<RSSI >,<earfcn> […]] [+QCELLINFO: "neighbourcell inter","LTE",<MCC>,<MNC>,<TAC>,<cellID>,<PCI>,<RX_lev>,<RSSI >,<earfcn> **OK**

2.8. AT+QGSN Request product serial number identification

Returns the IMEI number of the phone

Execution Command **AT+QGSN**	Response **+QGSN:<IMEI>** **OK**

2.9. AT+QCELLEX Get Serving Cell and Neighbor Cell information

This command is used to get serving cell and neighbor cell information.

Set Command **AT+QCELLEX=1**	Response serving cell and neighbor cell for LTE +QCELL: "servingcell","LTE",<MCC>,<MNC>,<TAC>,<cellID>,<PCI>,<RX_lev> ,<RSSI>,<earfcn> [+QCELL: "neighbourcell intra","LTE",<MCC>,<MNC>,<TAC>,<cellID>,<PCI>,<RX_lev>,<RSSI >,<earfcn> […]] [+QCELL: "neighbourcell inter","LTE",<MCC>,<MNC>,<TAC>,<cellID>,<PCI>,<RX_lev>,<RSSI >,<earfcn> **OK**

TEST Command **AT+QCELLEX=?**	Response **OK**

2.10. AT+QLTS

Gets the update time when being synchronized over the network.

Execution Command **AT+QLTS**	Response **+QLTS: <time>,<dst>** **OK** or **ERROR**
TEST Command **AT+QLTS=?**	Response **OK**

Parameters are defined below:

Parameter note:	Description
<time>	String type.Format is "yyyy / MM / dd, hh: mm: ss + zz", used to represent the year (yyyy), month (MM), day (dd), time (hh), minutes (mm), seconds (ss), time zone.
<dst>	daylight saving time

2.11. AT+CBAUD UART BAUD rate setting

Specifies the data rate, in addition to 921600 bits/s or 9600 bits/s, at which the DCE will accept commands. May be used to select operation at rates at which the DCE is not capable of automatically detecting the data rate being used by the DTE.

Test Command **AT+CBAUD=?**	Response **+CBAUD:(list of supported<rate>s)** **OK**
Execution Command **AT+CBAUD=<rate>**	Response **OK**
Read Command **AT+CBAUD?**	Response **+BAUDRATE:<rate>** **OK**

Parameters are defined below:

Parameters	Description
<rate>	The rate, in bits per second, at which the DTE-DCE interface should operate. Currently, the following rates are supported: 600,1200,2400,4800,9600,19200,38400,57600,115200,230400,460800.If unspecified, Default rate is 115200 bps.

2.12. ATO - Set Data mode

Set the data transfer mode

Command	Response
ATO[<value>]	**CONNECT**

Parameters are defined below:

Parameters	Description
<value>	**0** Switch from command mode to data mode.

3. Hardware AT Commands

3.1 AT+CGNETLED Network LED Control

This command is used to set the Network LED state to enable or disable.

Write Command **AT+CGNETLED=<mode>**	Response **OK** or **ERROR**
Test Command **AT+CGNETLED=?**	Response **+CGNETLED:(0-1)** **OK**
Read Command **AT+CGNETLED?**	Response **+CGNETLED: <mode>** **OK**

Parameters are defined below:

Parameters	Description
<mode>	0 – disable 1 – enable

3.2 AT+QSCLK Configure Slow Clock

This Command is used to Configure Slow Clock.

Write Command **AT+QSCLK=<n>**	Response **OK** or **ERROR**
Read Command **AT+QSCLK?**	Response **+QSCLK:<n>** **OK**

Test Command **AT+QSCLK=?**	Response **+QSCLK: (0-2)** **OK**

Parameters are defined below:

Parameters	Description
<n>	0 Disable slow clock, module will not enter sleep mode. 1 Enable slow clock, it is controlled by DTR. When DTR is high, module can enter sleep mode. When DTR changes to low level, module can quit sleep mode. 2 The serial port goes to sleep without data within 2 seconds

3.3 AT+RESET Reboot the Module

This Command is used to reboot the module

Write Command **AT+RESET**	Response **OK** or **ERROR**

3.4 AT+POWEROFF Poweroff the Module

This Command is used to poweroff the module

Write Command **AT+POWEROFF**	Response **OK**

4. Proprietary AT Commands for PS

4.1 AT+QIREGAPP Set APN, user name and password

This command is used to start the task and set the access point APN, user name and password.

Write Command **AT+QIREGAPP=<apn>,<username>,<password>**	Response **OK** or **ERROR**
Read Command **AT+QIREGAPP?**	Response **+QIREGAPP: <apn>,<username>,<password>** **OK** or **ERROR**
Test Command **AT+QIREGAPP=?**	Response **+QIREGAPP:"APN","USER","PWD"** **OK**

Parameters are defined below:

Parameters	Description
<apn>	string parameter; indicates the name of GPRS / CSD access point
<username>	string parameter; indicates the user name of GPRS / CSD access point
<password>	string parameter; indicates the user password of GPRS / CSD access point

4.2 AT+QPDPTIMER Setting PDP Connection parameters

This command is used to set PDP connection parameters.

Write Command **AT+QPDPTIMER**	Response **OK**
Read Command **AT+QPDPTIMER?**	Response **OK**
Test Command **AT+QPDPTIMER=?**	Response **OK**

4.3 AT+QPPPTIMER PPP Activation time setting

This command is used to set PPP activation time.

Write Command **AT+QPPPTIMER**	Response **OK**
Read Command **AT+QPPPTIMER?**	Response **OK**
Test Command **AT+QPPPTIMER=?**	Response **OK**

4.4 AT+CCID Read CCID of SIM Card

This command is used to read SIM card ICCID if SIM inserted. If SIM not inserted, return +CME ERROR: 10

Execution Command **AT+CCID**	Response **+CCID: <iccid>** **OK** or **ERROR**

Parameters are defined below:

Parameters	Description
<iccid>	String type

5. TCP IP AT Commands

5.1 AT+QICSGP Configure APN

This command is used to configure apn, username, and password

Write Command **AT+QICSGP=<cid>,<context_type>,<APN>,<username>,<password>,<authentication>**	Response **OK** or **ERROR**
Test Command **AT+QICSGP=?**	Response **+QICSGP: (1-15),(1-3),<APN>,<username>,<password>,(0-3)** **OK**
Read Command **AT+QICSGP?**	Response **OK**

Parameters are defined below:

Parameters	Description
<cid>	1-15
<context_type>	Connect type 1: IPV4 2: IPV4V6 3: IPV6
<APN>	string
<username>	string
<password>	string
<authentication>	Int type 0 none 1 PAP 2 CHAP 3 PAP and CHAP

5.2 AT+QIACT Active PDP Context and Open packet network

This command is used to open packet network.

Read Command **AT+QIACT?**	Response **+QIACT: 1,<context_state>,<context_type>,<IP_address>** **OK** or **ERROR**
Write Command **AT+QIACT=<contextID>**	Response **OK** or **ERROR**
Test Command **AT+QIACT=?**	Response **+QIACT: (1,15)** **OK**

Parameters are defined below:

Parameters	Description
<contextID>	Int type <1-15> Error Code:902 pdp already active
<context_state>	Indicate the current network state 0: network close (deactivated) 1: network open(activated)
<context_type>	Connect type 1: IPV4 2: IPV4V6 3: IPV6
<IP_address>	String type

5.3 AT+QIDEACT Deactive PDP Context

This command is used to deactive PDP context

Test Command **AT+QIDEACT=?**	Response **+QIDEACT:(1-15)** **OK**
Execution Command **AT+QIDEACT=<contextID>**	Response **OK** or **ERROR**

Parameters are defined below:

Parameters	Description
<contextID>	contextID , <1-15>

5.4 AT+QIOPEN Open socket service

This command is used to establish a connection with TCP server and UDP server.

Test Command **AT+QIOPEN=?**	Response **+QIOPEN:(1-15),(0-11),"TCP/UDP/TCPLISTEN/UDPSERVICE","IP_address/domain_name",(0-65535),(0-65535),(0-2)** **OK**
Write Command **AT+QIOPEN=<contextID>,<connectID>,<service_type>,"<IP_address>/<domain_name>",<remote_port>[,<local_port>[,<access_mode>]]**	Response **OK** **+QIOPEN: <connectID>,<err>**
ReadCommand **AT+QIOPEN?**	Response **ERROR**

Parameters are defined below:

Parameters	Description
<contextID>	Int Type <1-15>
<connectID>	Int Type <0-11>

<SERVICE_TYPE>	Int Type TCP: as client startup tcp connect UDP:as client startup udp connect TCP LISTENER:startup TCP server as listen connect UDP SERVICE:startup udp service
<IP_address>	Remote server ip address
<domain_name>	Remote server domain address
<remote_port>	Remote server's port only useful as SERVICE_TYPE is "TCP" or "UDP" Scoal is 1-65535
<LOCAL_PORT>	Local port scoal is 1-65535
<access_mode>	Socket service mode 0:buffer mode 1:normal mode 2;transparent mode

5.5 AT+QICLOSE Close socket service

This command is used to close socket service

Test Command **AT+QICLOSE=?**	Response **+QICLOSE:(0-11),(0-65535)** **OK**
Write Command **AT+QICLOSE=<connect ID> [,<timeout>]**	Response **OK** or **ERROR**
ReadCommand **AT+QICLOSE?**	Response **OK** or **ERROR**

Parameters are defined below:

Parameters	Description
<connectID>	Int type:<0-11>
<timeout>	Int type output response time:<0-65535>,default is 10

5.6 AT+QISEND Send data through TCP or UDP connection

This command is used to send data

Test Command **AT+QISEND=?**	Response **+QISEND:(0-11),(0-1460)** **OK**
Write Command **AT+QISEND=<connectID>** **AT+QISEND=<connectID>,<send_length>**	Response Response ">", then type data for send, tap CTRL+Z to send If sending is successful: **SEND OK** If sending fails: **SEND FAIL**
AT+QISEND=<connectID>,0	**+QISEND: <totallen>,<ackbyteslen>,<unackbyteslen>** **OK**

Parameters are defined below:

Parameters	Description
<connectID>	0-11 Socket ID A numeric parameter which indicates the connection number
<send_length>	0-1460 0 indicates the length of the data that the query has sent,1-1460, indicating the length of data that can be set once sent
<totallen>	The total length of data that has been sent
<ackbyteslen>	The confirmed byte length
<unackbyteslen>	Unconfirmed byte length

5.7 AT+QIRD – get buffer data

This command is used to get data

Test Command **AT+QIRD=?**	Response **+QIRD: (0-11),(0-1500)** **OK**

Write Command **AT+QIRD= <connectID>,<req_len>**	Response **+QIRD: <got_len> <data>** **OK** or **ERROR**
ReadCommand **AT+QIRD?**	**OK**

Parameters are defined below:

Parameters	Description
<connectID>	Integer type 0-11 Socket file description returned by +QIOPEN
<req_len>	Integer type The length of request data.
<got_len>	Integer type The got data length.
<data>	String type

5.8 AT+QIDNSCFG Config DNS service address

This command is used to config dns service address

Test Command **AT+QIDNSCFG=?**	Response **+QIDNSCFG:+QIDNSCFG: (1-15),("IP ADDR"),("IP ADDR")** **OK**
Write Command **AT+QIDNSCFG=<pridns addr>[,<secdnsaddr>]**	Response **OK** or **ERROR**
Read Command **AT+QIDNSCFG?**	Response **+QIDNSCFG:<list of ipv4 dns server>,<list of ipv6 dns server>** **OK** or **ERROR**

Parameters are defined below:

Parameters	Description
<pridnsaddr>	Primary dns service ip address
<secdnsaddr>	Second dns service ip address

5.9 AT+QIDNSGIP Use Domain get IP address

This command is used to use domain get ip address

Write Command **AT+QIDNSGIP=<contextID>,<hostname>**	Response **OK** or **ERROR** **URC format return result:** **+QIURC: "dnsgip",<err>,<IP_count>,<DNS_ttl>** **.....** **+QIURC: "dnsgip",<hostIPaddr>**
Test Command **AT+QIDNSGIP=?**	Response **+QIDNSGIP:(1-15),("host name")** **OK**
Read Command **AT+QIDNSGIP?**	Response **OK**

Parameters are defined below:

Parameters	Description
<contextID>	PDP context ID: 1-15
<hostname>	String type host name
<IP_count>	<hostname>’s ip address count
<DNS_ttl>	The value of ttl dns
<hostIPaddr>	Get ip address

5.10 AT+QISDE Control echo send data

This command is used to control echo send data

Test Command **AT+QISDE=?**	Response **+QISDE:(0-1)** **OK**
Write Command **AT+QISDE=<echo>**	Response **OK** or **ERROR**
Read Command **AT+QISDE?**	Response **+QISDE: <echo>** **OK**

Parameters are defined below:

Parameters	Description
<echo>	0: echo off 1:echo on

5.11 AT+QIMUX Control multi connection

Configure parameters of TCP/IP

Test Command **AT+QIMUX=?**	Response **+QIMUX:(0-1)** **OK**
Write Command **AT+QIMUX=<mode>**	Response **OK** or **ERROR**
Read Command **AT+QIMUX?**	Response **+QIMUX:<mode>** **OK**

Parameters are defined below:

Parameters	Description
<mode>	0: not use mutil connection 1:use mutil connection

5.12 AT+QIHEAD Add IP Head

This command is used to add an IP Head at the Beginning of a Package Received.

Write Command **AT+QIHEAD=<mode>**	Response **OK** or **ERROR**
Read Command **AT+QIHEAD?**	Response **+QIHEAD: <mode>** **OK**
Test Command **AT+QIHEAD=?**	Response **+QIHEAD:(0-1)** **OK**

Parameters are defined below:

Parameters	Description
<mode>	0 Normal mode, Not add IP header 1 add IP header

5.13 AT+QIPROMPT Config whether display “>” and ”send ok”

This command is used to config send data whether display “>” and “send ok”

Execution Command **AT+QIPROMPT=<send prompt>**	Response **OK** or **ERROR**
Read Command **AT+QIPROMPT?**	Response **+QIPROMPT:<send prompt>** **OK**

Test Command **AT+QIPROMPT=?**	Response **+QIPROMPT:(0-1)** **OK**

Parameters are defined below:

Parameters	Description
<send prompt>	0 send data not display “>”，return “SEND OK” 1 send data display “>”，return “SEND OK”

5.14 TCP URC and UDP URC

+QIURC: "closed",<connectID> when TCP Socket service disconnect URC <connectID> Int type, Socket ID ：0~11
+QIURC: "recv",<connectID>,<currentrecvlength><CR><LF> **<data>** normal mode，reveive data and URC <connectID> Int type, Socket ID：0~11 <currentrecvlength> Int type, The length of receive data <data> receive data

6. SSL Commands

6.1 AT+QSSLCFG Configure the parameters of SSL context

The command is used to select a directory.

Write Command **AT+QSSLCFG="sslversion",<sslctxID>[,<sslversion>]**	Response **OK** or **ERROR**
AT+QSSLCFG="ignorelocaltime",<on/off>	**OK** or **ERROR**
AT+QSSLCFG="session_cache",<sslctxID>[,<on/off>]	**OK** or **ERROR**
AT+QSSLCFG="seclevel",<level>	**OK** or **ERROR**
AT+QSSLCFG=cacert,<sslctxID>,<cacertpath>	**OK** or **ERROR**
AT+QSSLCFG=clientcert,<sslctxID>,<name>	**OK** or **ERROR**
AT+QSSLCFG=clientkey,<sslctxID>,<name>	**OK** or **ERROR**
AT+QSSLCFG=ciphersuite,<sslctxID>,<ciphersuite>	**OK** or **ERROR**

Test Command **AT+QSSLCFG=?**	Response **+QSSLCFG: "sslversion":(0-5),(0-4)** **+QSSLCFG: "ciphersuite":(0-5),<cipher_suites>** **+QSSLCFG: "cacert":(0-5),<cacertpath>** **+QSSLCFG: "clientcert":(0-5),<client_cert_path>** **+QSSLCFG: "clientkey":(0-5),<client_key_path>** **+QSSLCFG: "seclevel":(0-5),(0-2)** **+QSSLCFG: "session_cache":(0-5),(0,1)** **+QSSLCFG: "sni":(0-5),(0,1)** **+QSSLCFG: "ignorelocaltime":(0-5),(0,1)** **OK**
Read Command **AT+QSSLCFG?**	**sscxtId: caCertPath, clientCertPath,** **clientKeyPath,"seclevel":0,"cache":1,"ignore":1** **OK**

Parameters are defined below:

Parameters	Description
<sslctxID>	Numeric type, SSL context ID, range is 0~5.
<sslversion>	Numeric type, SSL Version 0 SSL3.0 1 TLS1.0 2 TLS1.1 3 TLS1.2 4 All
<ignorelocaltime>	Numeric format, indicates how to deal with expired certificate 0 Care about time check for certification 1 Ignore time check for certification
<session_cache>	On or off SSL session cache 0 off 1 on
<cacertpath>	String format, the path of the trusted CA certificate.
<client_cert_path>	String format, the path of the client certificate.
<client_key_path>	String format, the path of the client private key.
<seclevel>	Numeric format, the authentication mode 0 No authentication 1 Manage server authentication 2 If server need, manage server authentication

<ciphersuite>	0X0035 TLS_RSA_WITH_AES_256_CBC_SHA 0X002F TLS_RSA_WITH_AES_128_CBC_SHA 0X0005 TLS_RSA_WITH_RC4_128_SHA 0X0004 TLS_RSA_WITH_RC4_128_MD5 0X000A TLS_RSA_WITH_3DES_EDE_CBC_SHA 0X003D TLS_RSA_WITH_AES_256_CBC_SHA256 0XC002 TLS_ECDH_ECDSA_WITH_RC4_128_SHA 0XC003 TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA 0XC004 TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA 0XC005 TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA 0XC007 TLS_ECDHE_ECDSA_WITH_RC4_128_SHA 0XC008 TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA 0XC009 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA 0XC00A TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA 0XC011 TLS_ECDHE_RSA_WITH_RC4_128_SHA 0XC012 TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA 0XC013 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA 0XC014 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA 0xC00C TLS_ECDH_RSA_WITH_RC4_128_SHA 0XC00D TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA 0XC00E TLS_ECDH_RSA_WITH_AES_128_CBC_SHA 0XC00F TLS_ECDH_RSA_WITH_AES_256_CBC_SHA 0XC023 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 0xC024 TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 0xC025 TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 0xC026 TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384 0XC027 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 0XC028 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 0xC029 TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256 0XC02A TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384 0XC02F TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 0XFFFF All cipher suites
<sni>	Integer, server name indicater 0 Off 1 On

Example:

AT Commands	Response
AT+QSSLCFG="ignorelocaltime",1	**OK**

AT+QSSLCFG="session_cache",0,1	**OK**
AT+QSSLCFG="sslversion",0,3	**OK**
AT+QSSLCFG=seclevel,0	**+QSSLCFG:"seclevel",0,0** **OK**
AT+QSSLCFG=cacert,0,"RAM:cacert.pem"	**OK**
AT+QSSLCFG=clientcert,0,"RAM:clientcert.pem"	**OK**
AT+QSSLCFG=clientkey,0,"RAM:clientkey.pem"	**OK**

6.2 AT+QSSLOPEN Open a SSL socket to connect remote server

Write Command **AT+QSSLOPEN=<pdpctxID>,<sslctxID>,<clientID>,<serveraddr>,<server_port>[,<access_mode>]**	Response If the <access_mode> is transparent access mode and SSL connection is successfully set up, response: **CONNECT** Else, response: **ERROR** If the <access_mode> is buffer access mode or direct push mode, response: **OK** **+QSSLOPEN: <clientID>,<err>** <err> is 0 when service is started successfully, else <err> is not 0. or **ERROR**
Test Command **AT+QSSLOPEN=?**	Response **+QSSLOPEN: (1-15),(0-5),(0-3),<serveraddr>,<serverport>[,<access_mode>]** **OK**

Parameters are defined below:

Parameters	Description

<pdpctxID>	Numeric type, PDP context ID, range is 1-15.
<sslctxID>	Numeric type, SSL context ID, range is 0-5.
<clientID>	Numeric type, socket index, range is 0-3.
<serveraddr>	String type, the address of remote server.
<server_port>	Numeric type, the listening port of remote server.
<access_mode>	Numeric type, the access mode of SSL connection 0 Buffer access mode 1 Direct push mode 2 Transparent mode

Example:

AT Commands	Response
AT+QSSLOPEN=1,0,0,"10.1.1.0",30073,1	**OK** **+QSSLOPEN: 0,0**

6.3 AT+QSSLSEND After the connection is established, the module can send data through the SSL connection.

Write Command **AT+QSSLSEND=<clientID>**	Response AT+QSSLSEND=<clientID> Response ">", then input data to send, tap CTRL+Z to send, tap ESC to cancel the operation > <input data> <CTRL-Z> If connection has been established and sending is successful, response: **SEND OK** If connection has been established but sending buffer is full, response: **SEND FAIL** If connection has not been established, abnormally closed, or parameter is incorrect, response: **ERROR**
Write Command **AT+QSSLSEND=<clientID>,<sendlen>**	Response ">", input data until the data length is equal to <sendlength> > <input data with specified length> If connection has been established and sending is successful, response: **SEND OK** If connection has been established but sending buffer is full, response: **SEND FAIL** If connection has not been established, abnormally closed, or parameter is incorrect, response: **ERROR**
Test Command **AT+QSSLSEND=?**	Response +QSSLSEND: (0-3),(1-1460) **OK**

Parameters are defined below:

Parameters	Description
<clientID>	Numeric type, socket index, range is 0-3.
<sendlen>	Numeric type, the length of sending data, range is 1-1460.

Example:

AT Commands	Response
AT+QSSLOPEN=1,0,0,"10.1.1.0",30073,1	**OK** **+QSSLOPEN: 0,0**
AT+QSSLSEND=0,4	**>1234** **SEND OK**

6.4 AT+QSSLCLOSE

This command is used to close a SSL connection. If all the SSL connections based on one SSL context have been closed, the module will release the SSL context.

Write Command **AT+QSSLCLOSE=<clientID>**	Response **OK** or **ERROR**
Test Command **AT+QSSLCLOSE=?**	Response **OK**

Parameters are defined below:

Parameters	Description
<clientID>	Numeric type, socket index, range is 0-3.

Example:

AT Commands	Response
AT+QSSLOPEN=1,0,0,"10.1.1.0",30073,1	**OK** **+QSSLOPEN: 0,0**
AT+QSSLCLOSE=0	**OK**

6.5 AT+QSSLSTATE

This command is used to query the Socket connection status, and can only query the SSL connection status.

Write Command **AT+QSSLSTATE=<clientID>**	Response **+QSSLSTATE: <clientID>,"SSLClient",<IP_address>,<remote_port>,<local_port>,<socket_state>,<PDP_ctxID>,<serverID>,<access_mode>,<AT_port>,<SSL_ctxID>** **OK**
Action Command **AT+QSSLSTATE**	Response **+QSSLSTATE: <clientID>,"SSLClient",<IP_address>,<remote_port>,<local_port>,<socket_state>,<PDP_ctxID>,<serverID>,<access_mode>,<AT_port>,<SSL_ctxID>** **OK**
Test Command **AT+QSSLSTATE=?**	Response **OK**

Parameters are defined below:

Parameters	Description
<clientID>	Numeric type, socket index, range is 0-3.
<IP_address>	String type. remote server address.
<remote_port>	Integer type. Remote server port. Range: 0~65535
<local_port>	Integer type. local port. Range: 0~65535.

<socket_state>	Integer type. SSL connection status 0 -- "Initial",1 -- "Opening",2 -- "Connected",4 -- "Closing"
<PDP_ctxID>	Integer type. PDP context identifier. Range: 1~15
<serverID>	Integer type. This parameter is reserved
<access_mode>	Integer type. Indicates the data access mode for the SSL connection. 0 -- cache mode,1--Straight spit mode,2 -- Transparent mode
<AT_port>	Integer type. COM port.
<SSL_ctxID>	Integer type. SSL context identifier. Range: 0~5.

6.6 AT+QSSLRECV

This command is used to get the network data manually.

Write Command **AT+QSSLRECV=<clientID>,<readlen>**	Response **+QSSLRECV:<have_readlen><data>** **OK** or **ERROR**
Test Command **AT+QSSLRECV=?**	Response **+QSSLRECV: (0-3),(1-1500)** **OK**

Parameters are defined below:

Parameters	Description
<clientID>	Numeric type, socket index, range is 0-3.
<readlen>	Integer type.The data length to be read.range is 1-1500.
<have_readlen>	Integer type.The length of the data that have read.
<data>	The read data.

6.7 URC +QSSLURC:"recv"

+QSSLURC:"recv",<clientID>,<currentrecvlength><CR><LF><data>	The URC of SSL data incoming in direct push mode.

Parameters are defined below:

Parameters	Description
<clientID>	Numeric type, socket index, range is 0-3.
<currentrecvlength>	Integer type, the length of actual received data.
<data>	The received data.

6.8 URC +QSSLURC:"closed"

<clientID> SSL connection is closed.

+QSSLURC:"closed",<clientID>	Lots of reasons can cause this phenomenon, such as the Internet closes the connection or the state of GPRS PDP is deactivated. The<socket_state> of <clientID> will be “closing”.

Parameters are defined below:

Parameters	Description
<clientID>	Numeric type, socket index, range is 0-3.

6.9 Error Code

<err>	**Meaning**
0	Operate successfully
1	parameter error

2	ssl config error
3	mbedtls error
4	connect error
5	already has a connect
6	has no open connect
7	internal error, for example can't restore from deep sleep
8	memory not enough
557	socket listen failed
558	socket write failed
559	socket read failed
560	socket accept failed
561	open PDP context failed
562	close PDP context failed
563	socket Identity has been used
564	DNS busy
565	DNS parse failed
566	socket connect failed
567	socket has been closed
568	operation busy
569	operation timeout
570	PDP context break down
571	cancel send

572	operation not allowed
573	APN not configured
574	port busy

7. FOTA Commands

7.1 AT+QFOTADL Update Firmware

This command is used to update firmware.

Write Command **AT+QFOTADL=<URL>,<upgrade_mode>,<download_URC_max>**	Response **OK** or **ERROR**
Read Command **AT+QFOTADL?**	Response **+QFOTADL: <percent>** **OK**
Test Command **AT+QFOTADL=?**	Response **+HTTPFOTADL: <URL>,(0-100)** **OK**

Parameters are defined below:

Parameters	Description
<URL>	String type. Maximum length: 255 bytes. The URL should start with "http: / /" or "HTTPS: / /".
<upgrade_mode>	Integer. Upgrade mode after successfully downloading the target firmware package. 0. After downloading the target firmware package successfully, restart the module first and then upgrade the firmware 1. Upgrade the firmware immediately after the target firmware package is downloaded successfully

<download_URC_max>	Integer. Download progress URC. The last URC indicates that the download is complete. For example, if it is set to 50, Then 50 download progress URCs will be reported, among which the 25th URC means half of the download is completed and the 50th URC means half of the download is completed The URC indicates that the download is complete. 0 prohibit reporting download progress URC 5 – 100 maximum number of reports for download progress URC
<percent>	Integer. Download or upgrade progress.
<err>	Integer. 0 indicates the upgrade was successful, and other values indicate errors

8. Error Code

Description	Error Code
CME_PHONE_FAILURE	0 // phone failure
CME_NO_CONNECTION	1 // no connection to phone
CME_PHONE_ADP_LINK_RSVD	2 // phone adaptor link reserved
CME_OPERATION_NOT_ALLOWED	3 // operation not allowed
CME_OPERATION_NOT_SUPPORTED	4 // operation not supported
CME_PH_SIM_PIN_REQUIRED	5 // PH SIM PIN required
CME_PH_FSIM_PIN_REQUIRED	6 // PH-FSIM PIN required
CME_PH_FSIM_PUK_REQUIRED	7 // PH-FSIM PUK required
CME_NO_SIM	10 // SIM not inserted
CME_SIM_PIN_REQUIRED	11 // SIM PIN required
CME_SIM_PUK_REQUIRED	12 // SIM PUK required
CME_SIM_FAILURE	13 // SIM failure
CME_SIM_BUSY	14 // SIM busy
CME_SIM_WRONG	15 // SIM wrong
CME_INCORRECT_PASSWD	16 // incorrect password
CME_SIM_PIN2_REQUIRED	17 //SIM PIN2 required
CME_SIM_PUK2_REQUIRED	18 //SIM PUK2 required
CME_MEMORY_FULL	20 //memory full

CME_INVALID_INDEX	21 //invalid index
CME_NOT_FOUND	22 //not found
CME_MEMORY_FAILURE	23 //memory failure
CME_TEXT_STRING_TOO_LONG	24 //text string too long
CME_INVALID_CHAR_IN_STRING	25 //invalid characters in text string
CME_DAIL_STRING_TOO_LONG	26 //dial string too long
CME_INVALID_CHAR_IN_DIAL_STRING	27 //invalid characters in dial string
CME_NO_NW_SERVICE	30 //no network service
CME_NW_TIMEOUT	31 //network timeout
CME_NW_NOT_ALLOWED	32 //network not allowedemergency calls only
CME_NW_PIN_REQUIRED	40 //networkpersonalization PIN required
CME_NW_PUK_REQUIRED	41 //network personalization PUK required
CME_NW_SUB_PIN_REQUIRED	42 //network subset personalization PIN required
CME_NW_SUB_PUK_REQUIRED	43 //network subset personalization PUK required
CME_SP_PIN_REQUIRED	44 //service provider personalization PIN required
CME_SP_PUK_REQUIRED	45 //service provider personalization PUK required
CME_CP_PIN_REQUIRED	46 //corporate personalization PIN required
CME_CP_PUK_REQUIRED	47 //corporate personalization PUK required
CME_HD_KEY_REQUIRED	48 //hidden key required
CME_INVALID_PARAM	50 //Invalid Param
CME_UNKNOWN	100 //unknown

CME_ILLEGAL_MS	103 //Illegal MS (#3)
CME_ILLEGAL_ME	106 //Illegal ME (#6)
CME_GPRS_NOT_ALLOWED	107 //GPRS services not allowed (#7)
CME_PLMN_NOT_ALLOWED	111 //PLMN not allowed (#11)
CME_LA_NOT_ALLOWED	112 //Location area not allowed (#12)
CME_ROAMING_NOT_ALLOWED	113 //Roaming not allowed in this location area (#13)
CME_SERVICE_OP_NOT_SUPPORTED	132 //service option not supported (#32)
CME_SERVICE_OP_NOT_SUBSCRIBED	133 //requested service option not subscribed (#33)
CME_SERVICE_OP_OUT_OF_ORDER	134 //service option temporarily out of order (#34)
CME_UNSPECIFIED_GPRS_ERR	148 //unspecified GPRS error
CME_PDP_AUTH_FAILURE	149 //PDP authentication failure
CME_INVALID_MOBILE_CLASS	150 //invalid mobile class
CME_COMMMAND_TIMEOUT_ERR	151 //AT command timeout
CME_IMS_SRV_FAILURE	170 //IMSSRV failure
CMS_ME_FAILURE	300 //ME failure
CMS_SMS_SERVICE_RESV	301 //SMS service of ME reserved
CMS_OPERATION_NOT_ALLOWED	302 //operation not allowed
CMS_OPERATION_NOT_SUPPORTED	303 //operation not supported
CMS_INVALID_PDU_MODE_PARA	304 //invalid PDU mode parameter
CMS_INVALID_TEXT_MODE_PARA	305 //invalid text mode parameter
CMS_NO_SIM	310 //(U)SIM not inserted
CMS_SIM_PIN_REQUIRED	311 //(U)SIM PIN required

CMS_PH_SIM_PIN_REQUIRED	312 //PH-(U)SIM PIN required
CMS_SIM_FAILURE	313 //(U)SIM failure
CMS_SIM_BUSY	314 //(U)SIM busy
CMS_SIM_WRONG	315 //(U)SIM wrong
CMS_SIM_PUK_REQUIRED	316 //(U)SIM PUK required
CMS_SIM_PIN2_REQUIRED	317 //(U)SIM PIN2 required
CMS_SIM_PUK2_REQUIRED	318 //(U)SIM PUK2 required
CMS_MEMORY_FAILURE	320 //memory failure
CMS_INVALID_MEMORY_INDEX	321 //invalid memory index
CMS_MEMORY_FULL	322 //memory full
CMS_SMSC_ADDR_UNKNOWN	330 //SMSC address unknown
CMS_NO_NW_SERVICE	331 //no network service
CMS_NW_TIMEOUT	332 //network timeout
CMS_NO_CNMA_ACK_EXPECTED	340 //no +CNMA acknowledgement expected
CMS_UNKNOWN_ERROR	500 //unknown error
MIRC_PDP_ALREADY_ACTIVE	902 //pdp already active