WCDMA UGxx SSL

AT Commands Manual

UMTS/HSPA Module Series

Rev. WCDMA_UGxx_SSL_AT_Commands_Manual_V1.2

Date: 2015-04-01

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

Quectel Wireless Solutions Co., Ltd.
Office 501, Building 13, No.99, Tianzhou Road, Shanghai, China, 200233
Tel: +86 21 5108 6236
Mail: info@quectel.com

Or our local office, for more information, please visit:
http://www.quectel.com/support/salesupport.aspx

For technical support, to report documentation errors, please visit:
http://www.quectel.com/support/techsupport.aspx
Or Email: Support@quectel.com

GENERAL NOTES

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

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2014-12-13	Chris PENG	Initial
1.1	2015-03-03	Jessica GENG	1. Changed the document name from “UG95” to “UGxx”. 2. Modified the timeout value of command AT+QSSLCLOSE.
1.2	2015-04-01	Jessica GENG	Updated applicable modules.

Contents

Table Index

1 Introduction

This document describes how to use the SSL functionality of Quectel standard module. In some cases, in order to ensure communication privacy, the communication between the server and the client should be in an encrypted way. So that it can prevent data from eavesdropping, tampering, or forging during the communication process. The SSL function meets these demands.

This document is applicable to UGxx modules.

1.1. SSL Version and CipherSuite

So far, four SSL versions have been released. They are SSL3.0, TLS1.0, TLS1.1, and TLS1.2. The following versions are supported by Quectel modules.

Table 1: SSL Version

SSL Version
SSL3.0
TLS1.0
TLS1.1
TLS1.2

The following table shows the names of the CipherSuites that Quectel module supports. Please refer to RFC 2246 - The TLS Protocol Version 1.0 on the Ciphersuites definitions for details.

Table 2: SSL CipherSuite

CipherSuite Name	
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
0XFFFF	Support above all ciphersuite

1.2. Procedures of Using SSL Function

Step 1: Execute command “AT+QICSGP” to configure the APN, Username, Password of the context and so on.
Step 2: Execute command “AT+QIACT” to activate the specified PDP context. After the PDP context is activated, query the local IP address by command “AT+QILOCIP”.
Step 3: Execute command “AT+QSSLCFG” to configure the SSL version, ciphersuite, the path of trusted CA Cert and security level for the specified SSL context.
Step 4: Execute command “AT+QSSLOPEN” to open SSL client connection, <sslctxID> is used to specify SSL context, <access_mode> is used to specify data access mode.
Step 5: After the SSL connection has been established, you can send or receive data via this SSL connection. About how to send and receive data under each access mode, please refer to Chapter 1.3.
Step 6: Execute command “AT+QSSLCLOSE” to close SSL connection.
Step 7: Execute command “AT+QIDEACT” to deactivate PDP context.

1.3. Description of Data Access Mode

The module SSL Connection has three data access modes: buffer access mode, direct push mode, transparent access mode. When you use the command “AT+QSSLOPEN” to open SSL connection, you can specify the access mode via <access_mode>. After SSL connection has been established, you can switch the access mode via command “AT+QISWTMD”.

1. In buffer access mode, you can send data via command “AT+QSSLSEND”, and if the modem has received data from Internet, it will report URC: +QSSLURC: “recv”,<clientID>, and you can retrieve data via command “AT+QSSLRECV”.

2. In direct push mode, you can send data via command “AT+QSSLSEND”, and if the module modem has received data from Internet, it will output directly via COM port with following format: +QSSLURC: “recv”,<clientID>,<currectrecvlength><CR><LF><data>.

3. In transparent access mode, corresponding port will enter into exclusive mode, and the data received from COM port will be sent to Internet directly, and data received from Internet will be outputted via COM port directly. You can use "+++" or DTR (AT&D1 should be set) to switch to buffer access mode. In transparent access mode, if SSL connection encounters abnormal disconnection, the module modem will report URC: NO CARRIER.

4. Exit from transparent access data mode by "+++" or DTR (AT&D1 should be set). To prevent the "+++" from being misinterpreted as data, it should comply with the following sequence:

   1) Do not input any character within 1s before inputting "+++".
   2) Input "+++" during 1s, and no other characters can be inputted during this time.
   3) Do not input any character within 1s after "+++" has been inputted.
   4) Exit from transparent access mode, return OK.

5. There are two methods to return to transparent access mode:

   1) By AT+QISWTMD. Specify the <access_mode> as 2 by this command. If entering transparent access mode successfully, CONNECT will be returned.
   2) By ATO. ATO will change the access mode of connection which lately exits from transparent access mode. If entering transparent access mode successfully, CONNECT will be returned. If there is no connection enters transparent access mode before, ATO will return NO CARRIER.

1.4. Time Check for Certificate

To check whether a certificate is in the period of validity, you must parse the certificate, and compare the local time with the "Not before" and "Not after" of the certificate. If the local time is earlier than the time of "Not before" or later than the time of "Not after", the certificate will be considered expired.

When <ignoreltime> is 0, in order to avoid failure of certificate time check, you must use command "AT+CCLK" to configure the modem time to a validity period of the certificate.

1.5. Open SSL Connection Fails

When you fail to open SSL connection, please check the following aspects:

1. Query the status of the specified PDP context by command "AT+QIACT?" to check whether the specified PDP context is activated.

2. If the address of server is a domain name, please use command "AT+QIDNSCFG=<contextID>" to check whether the address of DNS server is valid. Because an invalid DNS server address cannot

convert domain name to IP address.

3. Please check the SSL configuration by command “AT+QSSLCFG”, especially SSL version and ciphersuite, make sure they are supported on server side. If you have configured <seclevel> as 1 or 2, you must upload trusted CA certificate to modem by FILE AT command. If server side has configured “SSLVerifyClient required”, you must upload the client cert and client private key to modem by FILE AT command. For details about certificate time check, please refer to Chapter 1.4.

2 Description of AT Command

2.1. AT Command Syntax

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

2.2. Description of AT Command

2.2.1. AT+QSSLCFG Configure the Parameters of SSL Context

This command is used to configure the SSL version, Cipher suites, secure level, CA certificate, client certificate and client key. These parameters will be used in the handshake procedure.

<sslctxID> is the index of the SSL context. The modules support 6 SSL contexts at most. On the basis of a SSL context, several SSL connections can be established. The settings such as the SSL versions and the Cipher suites are stored in the SSL context, and they will be applied to the new SSL connections associated with the SSL context.

AT+QSSLCFG Configure the Parameters of SSL Context	
Test Command **AT+QSSLCFG=?**	Response **+QSSLCFG: "sslversion",(0-5),(0-4)** **+QSSLCFG: "ciphersuite",(0-5),("0X0035","0X002F","0X0005","0X0004","0X000A","0X003D","0XFFFF")** **+QSSLCFG: "cacert",(0-5),<cacert_path>** **+QSSLCFG: "clientcert",(0-5),<client_cert_path>** **+QSSLCFG: "clientkey",(0-5),<client_key_path>** **+QSSLCFG: "seclevel",(0-5),(0-2)** **+QSSLCFG: "ignorelocaltime",(0-5),(0,1)**

	+QSSLCFG: "negotiatetime",(0-5),(10-300) **OK**
Configure the version for the <sslctxID> **AT+QSSLCFG="sslversion",<sslctxID>[,<ssl_version>]**	Response If <ssl_version> is omitted, query the value of "version" with specified <sslctxID>, and response: **+QSSLCFG: "sslversion",<sslctxID>,<ssl_version>** **OK** Else, set the value of "version" with specified <sslctxID>, and response: **OK** or **ERROR**
Configure the ciphersuites for the <sslctxID> **AT+QSSLCFG="ciphersuite",<sslctxID>[,<cipher_suites>]**	Response If <cipher_suites> is omitted, query the value of "ciphersuite" with specified <sslctxID>, and response: **+QSSLCFG: "ciphersuite",<sslctxID>,<cipher_suites>** **OK** Else, set the value of "ciphersuite" with specified sslctxID, and response: **OK** or **ERROR**
Configure the path of CA Cert for the <sslctxID> **AT+QSSLCFG="cacert",<sslctxID>[,<cacert_path>]**	Response If <cacert_path> is omitted, query the value of "cacert" with specified <sslctxID>, and response: **+QSSLCFG: "cacert",<sslctxID>,<cacert_path>** **OK** Else, set the value of "cacert" with specified <sslctxID>, and response: **OK** or **ERROR**
Configure the path of client Cert for the <sslctxID> **AT+QSSLCFG="clientcert",<sslctxID>[,<client_cert_path>]**	Response If <client_cert_path> is omitted, query the value of "client_cert" with specified <sslctxID>, and response: **+QSSLCFG: "clientcert",<sslctxID>,<client_cert_path>** **OK**

	Else, set the value of “client_cert” with specified <sslctxID>, and response: **OK** or **ERROR**
Configure the path of client Key for the <sslctxid> **AT+QSSLCFG="clientkey",<sslctxID>[,<client_key_path>]**	Response If <client_key_path> is omitted, query the value of “client_key” with specified <sslctxID>, and response: **+QSSLCFG: "clientkey",<sslctxID>,<client_key_path>** **OK** Else, set the value of “clientkey” with specified <sslctxID>, and response: **OK** or **ERROR**
Configure the security level for the <sslctxid> **AT+QSSLCFG="seclevel",<sslctxID>[,<seclevel>]**	Response If <seclevel> is omitted, query the value of “seclevel” with specified <sslctxID>, and response: **+QSSLCFG: "seclevel",<sslctxID>,<seclevel>** **OK** Else, set the value of “seclevel” with specified <sslctxID>, and response: **OK** or **ERROR**
Configure the ignore time check for certification for the <sslctxID> **AT+QSSLCFG="ignorelocaltime",<sslctxID>[,<ignoreltime>]**	Response If <ignoreltime> is omitted, query the value of “ignorelocaltime” with specified <sslctxID>, and response: **+QSSLCFG: "ignorelocaltime",<sslctxID>,<ignoreltime>** **OK** Else, set the value of “ignorelocaltime” with specified <sslctxid>, and response: **OK** or **ERROR**
Configure the negotiate timeout for the <sslctxID> **AT+QSSLCFG="negotiatetime",<sslct**	Response If <negotiatetime> is omitted, query the value of “negotiatetimeout” with specified <sslctxID>, and response:

xID>[,<negotiatetime>]	**+QSSLCFG: "negotiatetime",<sslctxID>,<negotiatetime>** **OK** Else, set the value of “negotiatetimeout” with specified <sslctxID>, and response: **OK** or **ERROR**

Parameter

<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 version
<cipher_suites>	String type, SSL 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 “0XFFFF” All support
<ignoreltime>	Numeric format, indicates how to deal with expired certificate 0 Care time check for certification 1 Ignore time check for certification
<cacert_path>	String format, the path of the trust 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 Manage server and client authentication if requested by the remote server
<negotiatetime>	Numeric format, indicates max timeout used to SSL negotiate stage, value range is 10-300, unit: seconds, default value is 300

2.2.2. AT+QSSLOPEN Open a SSL Socket to Connect Remote Server

AT+QSSLOPEN is used to set up a SSL connection. During the negotiation between the module and the Internet, parameters configured by QSSLCFG will be used in the handshake procedure. After shaking hands with the Internet successfully, the module can send or receive data via this SSL connection. Also the module can set up several SSL connection based on one SSL context.

According to description of Chapter 1.2, before executing QSSLOPEN command, you must executing QIACT command to active PDP context.

It is suggested to wait 150 seconds for the URC response as "+QSSLOPEN: <clientID>,<errorcode>".
If the URC response has not been received in 150 seconds, you could use AT+QSSLCLOSE to close the SSL connection.

AT+QSSLOPEN Open a SSL Socket to Connect Remote Server	
Test Command **AT+QSSLOPEN=?**	Response **+QSSLOPEN: (1-20),(0-5),(0-11),<serveraddr>,<server_port>[,(0-2)]** **OK**
Write Command **AT+QSSLOPEN=<pdpctxID>,<sslctxID>,<clientID>,<serveraddr>,<server_port>[,<access_mode>]**	Response If the <access_mode> is transparent access mode and it is successful to create SSL connection, response: **CONNECT** Else, response: **ERROR** Error description can be got via **AT+QIGETERROR**. If the <access_mode> is buffer access mode or direct push mode, response: **OK** **+QSSLOPEN: <clientID>,<errorcode>** <errorcode> is 0 when service is started successfully, else <errorcode> is not 0. Or **ERROR** Error description can be got via "AT+QIGETERROR"
Maximum Response Time	150 seconds, determined by network and <negotiatetimout>

Parameter

<pdpctxID>	Numeric type, PDP context ID, range is 1-20
<sslctxID>	Numeric type, SSL context ID, range is 0-5
<clientID>	Numeric type, socket index, range is 0-11
<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
<errorcode>	Refer to *Quectel_WCDMA_UGxx_TCPIP_AT_Commands_Manual*

2.2.3. AT+QSSLSEND Send Data via SSL Connection

After the connection is established, the module can send data through the SSL connection.

AT+QSSLSEND Send Data via SSL Connection	
Test Command **AT+QSSLSEND=?**	Response **+QSSLSEND: (0-11)[,(1-1460)]** **OK**
Write Command **AT+QSSLSEND=<clientID>** Response ">", then type data to send, tap CTRL+Z to send, tap ESC to cancel the operation	Response **>** **<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 ">", type data until the data length is equal to <sendlength>	Response **>** **<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**

Parameter

<clientID>	Numeric type, socket index, range is 0-11
<sendlen>	Numeric type, the length of send data, range is 1-1460

2.2.4. AT+QSSLRECV Receive Data via SSL Connection

When you open SSL connection, and specify <access_mode>=0, if the module receives data from the Internet, it will report URC: +QSSLURC: "recv",<clientID>, and you can read data from buffer by AT+QSSLRECV command.

AT+QSSLRECV Receive Data via SSL Connection	
Test Command **AT+QSSLRECV=?**	Response **+QSSLRECV: (0-11),(1-1500)** **OK**
Write Command **AT+QSSLRECV=<clientID>,<readlen>**	Response If the specified connection has received data, response: **+QSSLRECV: <havereadlen><CR><LF><data>** **OK** If the buffer is empty, directly response: **+QSSLRECV: 0** **OK** If parameters is not correct or connection is not established, response: **ERROR**

Parameter

<clientID>	Numeric type, socket index, range is 0-11
<readlen>	Numeric type, the length of data will be retrieved, range is 1-1500
<havereadlen>	Numeric type, the actual data length which is obtained by QSSLRECV
<data>	The retrieved data

2.2.5. AT+QSSLCLOSE Close SSL Connection

Close a SSL connection. If all the SSL connections based on one SSL context have been closed, the module will release the SSL context.

AT+QSSLCLOSE Close SSL Connection	
Test Command **AT+QSSLCLOSE=?**	Response **+QSSLCLOSE: (0-11)[,(0-65535)]** **OK**
Write Command **AT+QSSLCLOSE=<clientID>[,<close_timeout>]**	Response If closes successfully, response: **OK** If failed to close, response: **ERROR**

Parameter

<clientID>	Numeric type, socket index, range is 0-11
<close_timeout>	Numeric type, the timeout value of QSSLCLOSE, range is 0-65535. Unit: s Default value is 10s. If <close_timeout>=0, means close immediately

2.2.6. AT+QSSLSTATE Query the State of SSL Connection

Query the socket connection status. This command can only query the status of SSL connection.

AT+QSSLSTATE Query the State of SSL Connection	
Test Command **AT+QSSLSTATE=?**	Response **OK**
Write Command **AT+QSSLSTATE=<clientID>**	Response **[+QSSLSTATE:<clientID>,"SSL CLIENT",<IP_address>,<remote_port>,<local_port>,<socket_state>,<pdpctxID>,<serverID>,<access_mode>,<at_p**

	ort>,<sslctxID>] **OK**
Execute Command **AT+QSSLSTATE**	Response **[List of (+QSSLSTATE: <clientID>,"SSL CLIENT",<IP_address>,<remote_port>,<local_port>,<socket_state>,<pdpctxID>,<serverID>,<access_mode>,<at_port>,<sslctxID>)]** **OK**

Parameter

<clientID>	Numeric type, socket index, range is 0-11		
<IP_address>	String type, the address of remote server		
<remote_port>	Numeric type, the port of remote server		
<local_port>	Numeric type, the local port		
<socket_state>	Numeric type, the state of SSL connection		
	0	"Initial"	Connection not established
	1	"Opening"	Client is connecting or server is trying to listen
	2	"SSL handshake"	SSL handshake
	3	"Connected"	Client/incoming connection has been established
	4	"Listening"	Server is listening
	5	"Closing"	Connection is closing
<pdpctxID>	Numeric type, PDP context ID, range is 1-20		
<serverID>	Numeric type, reserved		
<access_mode>	Numeric type, the access mode of SSL connection		
	0	Buffer access mode	
	1	Direct push mode	
	2	Transparent access mode	
<at_port>	String type, name of COM port		
<sslctxID>	Numeric type, SSL Context index, range is 0-5		

2.3. URC Description

2.3.1. Notify Received Data

Notify received data which comes from peer.

Notify Received Data	
+QSSLURC: "recv",<clientID>	The URC of SSL data incoming in buffer access mode. You can receive SSL data by AT+QSSLRECV.

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

Parameter

<clientID>	Integer type, socket index, range is 0-11
<currentrecvlength>	Integer type, the length of actual received data
<data>	The received data

2.3.2. Notify Abnormal Close

Notify that the connection has been disconnected. 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”. Host must execute AT+QSSLCLOSE=<clientID> to change the <socket_state> to “initial”.

Notify Abnormal Close	
+QSSLURC: "closed",<clientID>	<clientID> SSL connection is closed.

Parameter

<clientid>	Integer type, socket index, range is 0-11

2.3.3. Notify SSL Security Error

Notify host encounter security error while transferring data by SSL Connection.

Notify SSL Security Error	
+QSSLURC: "security",<clientID>,<errorcode>	<clientID> SSL connection encounter security error.

Parameter

<clientID>	Socket index, range is 0-11	
<errorcode>	Security error code	
	1	Encrypt error
	2	Decrypt error
	3	Data verify error

3 Example

3.1. Configure and Activate the PDP Context

3.1.1. Configure Context

```
AT+QICSGP=1,1,“UNINET”,””,””,1      //Configure context 1, APN is “UNINET” for China Unicom.
OK
```

3.1.2. Activate Context

```
AT+QIACT=1                          //Activate context 1
OK                                  //Activate successfully
AT+QIACT?                           //Query the state of context
+QIACT: 1,1,1,"10.7.157.1"

OK
```

3.1.3. Deactivate Context

```
AT+QIDEACT=1                        //Deactivate context 1
OK                                  //Deactivate successfully
```

3.2. Configure SSL Context

```
AT+QSSLCFG=“sslversion”,1,1
OK
```

```
AT+QSSLCFG=“ciphersuite”,1,“0X0035”
OK
```

```
AT+QSSLCFG=“seclevel”,1,1
OK
```

```
AT+QSSLCFG=“cacert”,1,“RAM:ca.pem”
OK
```

3.3. SSL Client Works in Buffer Access Mode

3.3.1. Set up SSL Connection and Enter into Buffer Access Mode

```
AT+QSSLOPEN=2,1,1,"220.180.239.201",8712,0
OK

+QSSLOPEN: 0,0                    //Set up SSL connection successfully
AT+QSSLSTATE                      //Query status of all SSL connections
+QSSLSTATE: 1,"SSL CLIENT","220.180.239.201",8712,4100,3,1,2,0,"usbmodem",1

OK
```

3.3.2. Send Data in Buffer Access Mode

```
AT+QSSLSEND=1                     //Send changeable length data.
> GET /4M.txt HTTP/1.1
> HOST: 220.180.239.201:8011
>
><CTRL-Z>
SEND OK
AT+QISEND=1,52                    //Send fixed length data and the data length is 52
> GET /4M.txt HTTP/1.1
> HOST: 220.180.239.201:8011
>
SEND OK
```

3.3.3. Receive Data in Buffer Access Mode

```
+QSSLURC: “recv”,1                          //The <clientID> 1 received data

AT+QSSLRECV=1,1500                          //Read data, the length is 1500
+QSSLRECV: 291                              //The actual received data length is 291
HTTP/1.1 200 OK
Date: Tue, 03 Sep 2013 11:04:13 GMT
Server: Apache/2.2.22 (Win32) PHP/5.2.4 mod_ssl/2.2.22 OpenSSL/0.9.8t
Last-Modified: Mon, 02 Sep 2013 03:30:38 GMT
```

```
ETag: "f700000001e36d-406538-4e55e322ae3bc"
Accept-Ranges: bytes
Content-Length: 4220216
Content-Type: text/plain



OK
AT+QSSLRECV=1,1500
+QSSLRECV: 0                    //No Data in buffer

OK
```

3.3.4. Close SSL Connection

```
AT+QSSLCLOSE=1          //Close a connection whose <clientID> is 1. Depending on the Network, the
                          maximum response time is 10s.
OK
```

3.4. SSL Client Works in Direct Push Mode

3.4.1. Set up SSL Connection and Enter into Direct Push Mode

```
AT+QSSLOPEN= 2,1,1,"220.180.239.201",8712,1
OK

+QSSLOPEN: 1,0                          //Set up SSL connection successfully
AT+QSSLSTATE                            //Query status of all SSL connections
+QSSLSTATE: 1,"SSL CLIENT","220.180.239.201",8712,4100,3,1,2,1,"usbmodem",1

OK
```

3.4.2. Send Data in Direct Push Mode

```
AT+QSSLSEND=1                           //Send changeable length data
> GET /4M.txt HTTP/1.1
> HOST: 220.180.239.201:8011
>
><CTRL-Z>
SEND OK
```

```
AT+QISEND=1,52                          //Send fixed length data and the data length is 52
> GET /4M.txt HTTP/1.1
> HOST: 220.180.239.201:8011
>
SEND OK
```

3.4.3. Receive Data in Direct Push Mode

```
+QSSLURC: "recv",1,291
HTTP/1.1 200 OK
Date: Tue, 03 Sep 2013 12:07:19 GMT
Server: Apache/2.2.22 (Win32) PHP/5.2.4 mod_ssl/2.2.22 OpenSSL/0.9.8t
Last-Modified: Mon, 02 Sep 2013 03:30:38 GMT
ETag: "f700000001e36d-406538-4e55e322ae3bc"
Accept-Ranges: bytes
Content-Length: 4220216
Content-Type: text/plain
```

3.4.4. Close SSL Connection

```
AT+QSSLCLOSE=1          //Close a connection whose <clientID> is 1. Depending on the Network, the
                          maximum response time is 10s.
OK
```

3.5. SSL Client Works in Transparent Access Mode

3.5.1. Set up SSL Connection and Send Data in Transparent Access Mode

```
AT+QSSLOPEN= 2,1,1,"220.180.239.201",8712,2
CONNECT
```

3.5.2. Send Data in Transparent Access Mode

```
GET /4M.txt HTTP/1.1<CR><LF>
HOST: 220.180.239.201:8011<CR><LF>
<CR><LF>
```

3.5.3. Receive Data in Transparent Access Mode

```
HTTP/1.1 200 OK
Date: Tue, 03 Sep 2013 12:07:19 GMT
Server: Apache/2.2.22 (Win32) PHP/5.2.4 mod_ssl/2.2.22 OpenSSL/0.9.8t
Last-Modified: Mon, 02 Sep 2013 03:30:38 GMT
ETag: "f700000001e36d-406538-4e55e322ae3bc"
Accept-Ranges: bytes
Content-Length: 4220216
Content-Type: text/plain
```

3.5.4. Close SSL Connection

```
AT+QSSLCLOSE=1          //Close a connection whose <clientID> is 1. Depending on the Network, the
                          maximum response time is 10s.
OK
```

4 Appendix A Reference

Table 3: Related Documents

SN	Document Name	Remark
[1]	GSM 07.07	Digital cellular telecommunications (Phase 2+); AT command set for GSM Mobile Equipment (ME)
[2]	GSM 07.10	Support GSM 07.10 multiplexing protocol
[3]	Quectel_WCDMA_UGxx_TCPIP_AT_Commands_Manual	TCPIP AT commands manual

Table 4: Terms and Abbreviations

Abbreviation	Description
SSL	Security Socket Layer
DTR	Data Terminal Ready
DNS	Domain Name Server
PDP	Packet Data Protocol