

EG800Q&EG91xQ Series

SSL Application Note

LTE Standard Module Series

Version: 1.3

Date: 2024-11-25

Status: Released



At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:

Quectel Wireless Solutions Co., Ltd.

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

Tel: +86 21 5108 6236

Email: info@quectel.com

Or our local offices. For more information, please visit:

<http://www.quectel.com/support/sales.htm>.

For technical support, or to report documentation errors, please visit:

<http://www.quectel.com/support/technical.htm>.

Or email us at: support@quectel.com.

Legal Notices

We offer information as a service to you. The provided information is based on your requirements and we make every effort to ensure its quality. You agree that you are responsible for using independent analysis and evaluation in designing intended products, and we provide reference designs for illustrative purposes only. Before using any hardware, software or service guided by this document, please read this notice carefully. Even though we employ commercially reasonable efforts to provide the best possible experience, you hereby acknowledge and agree that this document and related services hereunder are provided to you on an “as available” basis. We may revise or restate this document from time to time at our sole discretion without any prior notice to you.

Use and Disclosure Restrictions

License Agreements

Documents and information provided by us shall be kept confidential, unless specific permission is granted. They shall not be accessed or used for any purpose except as expressly provided herein.

Copyright

Our and third-party products hereunder may contain copyrighted material. Such copyrighted material shall not be copied, reproduced, distributed, merged, published, translated, or modified without prior written consent. We and the third party have exclusive rights over copyrighted material. No license shall be granted or conveyed under any patents, copyrights, trademarks, or service mark rights. To avoid ambiguities, purchasing in any form cannot be deemed as granting a license other than the normal non-exclusive, royalty-free license to use the material. We reserve the right to take legal action for noncompliance with abovementioned requirements, unauthorized use, or other illegal or malicious use of the material.

Trademarks

Except as otherwise set forth herein, nothing in this document shall be construed as conferring any rights to use any trademark, trade name or name, abbreviation, or counterfeit product thereof owned by Quectel or any third party in advertising, publicity, or other aspects.

Third-Party Rights

This document may refer to hardware, software and/or documentation owned by one or more third parties ("third-party materials"). Use of such third-party materials shall be governed by all restrictions and obligations applicable thereto.

We make no warranty or representation, either express or implied, regarding the third-party materials, including but not limited to any implied or statutory, warranties of merchantability or fitness for a particular purpose, quiet enjoyment, system integration, information accuracy, and non-infringement of any third-party intellectual property rights with regard to the licensed technology or use thereof. Nothing herein constitutes a representation or warranty by us to either develop, enhance, modify, distribute, market, sell, offer for sale, or otherwise maintain production of any our products or any other hardware, software, device, tool, information, or product. We moreover disclaim any and all warranties arising from the course of dealing or usage of trade.

Privacy Policy

To implement module functionality, certain device data are uploaded to Quectel's or third-party's servers, including carriers, chipset suppliers or customer-designated servers. Quectel, strictly abiding by the relevant laws and regulations, shall retain, use, disclose or otherwise process relevant data for the purpose of performing the service only or as permitted by applicable laws. Before data interaction with third parties, please be informed of their privacy and data security policy.

Disclaimer

- a) We acknowledge no liability for any injury or damage arising from the reliance upon the information.
- b) We shall bear no liability resulting from any inaccuracies or omissions, or from the use of the information contained herein.
- c) While we have made every effort to ensure that the functions and features under development are free from errors, it is possible that they could contain errors, inaccuracies, and omissions. Unless otherwise provided by valid agreement, we make no warranties of any kind, either implied or express, and exclude all liability for any loss or damage suffered in connection with the use of features and functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage may have been foreseeable.
- d) We are not responsible for the accessibility, safety, accuracy, availability, legality, or completeness of information, advertising, commercial offers, products, services, and materials on third-party websites and third-party resources.

Copyright © Quectel Wireless Solutions Co., Ltd. 2024. All rights reserved.

About the Document

Revision History

Version	Date	Author	Description
-	2023-04-11	Greyson DONG	Creation of the document
1.0	2023-06-13	Greyson DONG	First official release
1.1	2023-09-15	Greyson DONG	<ol style="list-style-type: none"> Updated the applicable modules: <ul style="list-style-type: none"> Added EG916Q-GL. Updated EG800Q-EU to EG800Q series. Added the command: AT+QSSLCFG="dtlsversion",<SSL_ctxID>[,<DTLS_version>] (Chapter 2.2.1).
1.2	2024-05-22	Greyson DONG	Updated EG915Q-NA to EG915Q series.
1.3	2024-11-25	Fawei ZHOU	<ol style="list-style-type: none"> Updated the declaration of AT command examples (Chapter 2.1.3). Updated the server address in command examples (Chapter 3).

Contents

About the Document.....	3
Contents	4
Table Index.....	6
1 Introduction	7
1.1. SSL Version and Cipher Suite	7
1.2. Using SSL Function	9
1.3. Description of Data Access Modes	10
1.4. Certificate Validity Check	11
1.5. Server Name Indication	11
2 Description of SSL AT Commands	12
2.1. AT Command Description	12
2.1.1. Definitions.....	12
2.1.2. AT Command Syntax	12
2.1.3. Declaration of AT Command Examples	13
2.2. Description of AT Commands	13
2.2.1. AT+QSSLCFG Configure Parameters of an SSL Context.....	13
2.2.2. AT+QSSLOPEN Open an SSL Socket to Connect a Remote Server	23
2.2.3. AT+QSSLSEND Send Data over SSL Connection	24
2.2.4. AT+QSSLRECV Receive Data over SSL Connection	26
2.2.5. AT+QSSLCLOSE Close an SSL Connection.....	27
2.2.6. AT+QSSLSTATE Query the State of SSL Connection.....	27
2.3. Description of URCs	29
2.3.1. +QSSLURC: "recv" Notify Received Data	29
2.3.2. +QSSLURC: "closed" Notify Abnormal Disconnection	29
3 Examples	30
3.1. Configure and Activate a PDP Context.....	30
3.1.1. Configure a PDP Context.....	30
3.1.2. Activate a PDP Context.....	30
3.1.3. Deactivate a PDP Context	30
3.2. Configure an SSL Context	30
3.3. SSL Client in Buffer Access Mode	31
3.3.1. Set up an SSL Connection and Enter Buffer Access Mode.....	31
3.3.2. Send Data in Buffer Access Mode	31
3.3.3. Receive Data in Buffer Access Mode	31
3.3.4. Close an SSL Connection	32
3.4. SSL Client in Direct Push Mode.....	32
3.4.1. Set up an SSL Connection and Enter Direct Push Mode	32
3.4.2. Send Data in Direct Push Mode.....	32
3.4.3. Receive Data in Direct Push Mode	33
3.4.4. Close an SSL Connection	33

3.5.	SSL Client in Transparent Transmission Mode	33
3.5.1.	Set up an SSL Connection and Send Data in Transparent Transmission Mode.....	33
3.5.2.	Set up an SSL Connection and Receive Data in Transparent Transmission Mode.....	33
3.5.3.	Close an SSL Connection	33
4	Check for Failure in SSL Connection	34
5	Result Codes	35
6	Appendix References	37

Table Index

Table 1: SSL Versions	7
Table 2: Supported SSL Cipher Suites	8
Table 3: Type of AT Commands	12
Table 4: Result Codes	35
Table 5: Related Documents	37
Table 6: Terms and Abbreviations	37

1 Introduction

Quectel EG800Q series and EG91xQ family (EG915Q series and EG916Q-GL) modules support SSL function. The SSL function is to ensure the privacy of communication. In some cases, the communication between the server and the client should be encrypted to prevent data from being eavesdropped, tampered with or forged during the communication process.

This document introduces how to use the SSL function on Quectel EG800Q series and EG91xQ family modules through AT commands.

1.1. SSL Version and Cipher Suite

The following SSL versions are supported.

Table 1: SSL Versions

SSL Versions
SSL 3.0
TLS 1.2
TLS 1.1
TLS 1.0

The following table shows SSL cipher suites supported by Quectel EG800Q series and EG91xQ family modules, and all the SSL cipher suites are supported by default. For detailed description of cipher suites, see *RFC 2246-The TLS Protocol Version 1.0*.

Table 2: Supported SSL Cipher Suites

Codes of Cipher Suites	Names of Cipher Suites
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	Support all the cipher suites listed above

1.2. Using SSL Function

Step 1: Configure <APN>, <username>, <password> and other parameters of a PDP context by **AT+QICSGP**. See **document [1]** for details.

Step 2: Activate the PDP context by **AT+QIACT**, then query the assigned IP address by **AT+QIACT?**. See **document [1]** for details.

Step 3: Configure the SSL version, cipher suite, path of trusted CA certificate, authentication mode, the path of the client certificate and private key, etc. for a specified SSL context by **AT+QSSLCFG**.

Step 4: Open an SSL socket to connect a remote server by **AT+QSSLOPEN**.

Step 5: After the SSL connection has been established, data will be sent or received over the connection. For details about how to send and receive data in each access mode, see **Chapter 1.3**.

Step 6: Close SSL connection by **AT+QSSLCLOSE**.

Step 7: Deactivate the PDP context by **AT+QIDEACT**. See **document [1]** for details.

1.3. Description of Data Access Modes

The SSL connection supports the following three data access modes:

- Buffer access mode
- Direct push mode
- Transparent transmission mode

When opening an SSL connection over **AT+QSSLOPEN**, the data access mode can be specified by the **<access_mode>**. After the SSL connection has been established, **AT+QISWTMD** can be used to switch the data access mode. For details of **AT+QISWTMD**, see *document [1]* for details.

1. In buffer access mode, data can be sent via **AT+QSSLSEND**, and the module buffers data upon receiving them and reports a URC in the format of **+QSSLURC: "recv",<clientID>** to notify the host of the incoming data. In this case, the host can retrieve the buffered data with **AT+QSSLRECV**.
2. In direct push mode, data can be sent via **AT+QSSLSEND**, and the module outputs the received data directly over UART/USB modem/USB AT port in the format of **+QSSLURC: "recv",<clientID>,<currentrecvlength><CR><LF><data>**.
3. In transparent transmission mode, the corresponding COM port is exclusively used for sending/receiving data directly to/from the Internet. It cannot be used for other purposes such as running AT commands, etc.

● Exit transparent transmission mode

To make the module exit transparent transmission mode either:

- 1) Execute **+++**. To prevent the **+++** from being misinterpreted as data, follow the requirements below:
 - a) Do not input any other character at least 1 second before and after inputting **+++**.
 - b) Input **+++** within 1 second, and wait until **OK** is returned. After **OK** is returned, the module switches to buffer access mode.

OR

- 2) Change DTR from LOW to HIGH to make the module enter command mode (the COM port can now be used for running AT commands, as well as for sending/retrieving data). In this case, set **AT&D1** (see *document [2]*) before the module enters transparent transmission mode.

● Return to transparent transmission mode

To return to transparent transmission mode either:

- 1) Execute **AT+QISWTMD**. Before execution specify the **<access_mode>** as 2. Once transparent transmission mode is entered successfully, **CONNECT** is returned.

OR

- 2) Execute **ATO**. After a connection exits transparent transmission mode, executing **ATO** switches the data access mode back to transparent transmission mode. Once transparent transmission mode is entered successfully, **CONNECT** is returned. If no connection has entered transparent transmission mode, **ATO** returns **NO CARRIER**. See *document [2]* for detailed information about **ATO**.

1.4. Certificate Validity Check

To check certificate validity, the certificate must be parsed, and the local time compared with the “Not before” and “Not after” of the certificate. If the local time is earlier than the “Not before” time or later than the “Not after” time the certificate has expired.

When validity check of certificate is required (set **<ignore_ltime>** as 0 when executing **AT+QSSLCFG**), to avoid failure of certificate validity check, execute **AT+CCLK** to configure the module time within the validity time period of the certificate. For details of **AT+CCLK**, see *document [2]*.

1.5. Server Name Indication

SNI (Server Name Indication) allows the server to safely host multiple TLS Certificates since it provides Server Host Name information as an extension in the client hello message. It thus enhances connection security with multiple virtual servers based on a single IP address. This feature is only applicable to TLS protocol.

2 Description of SSL AT Commands

2.1. AT Command Description

2.1.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on command line.
- **[...]** Optional parameter of a command or an optional part of TA information response. Square brackets do not appear on the command line. When an optional parameter is not given in a command, the new value equals to its previous value or the default setting, unless otherwise specified.
- **Underline** Default setting of a parameter.

2.1.2. AT Command Syntax

All command lines must start with **AT** or **at** and end with **<CR>**. Information responses and result codes always start and end with a carriage return character and a line feed character: **<CR><LF><response><CR><LF>**. In tables presenting commands and responses throughout this document, only the commands and responses are presented, and **<CR>** and **<LF>** are deliberately omitted.

Table 3: Type of AT Commands

Command Type	Syntax	Description
Test Command	AT+<cmd>=?	Test the existence of the corresponding command and return information about the type, value, or range of its parameter.
Read Command	AT+<cmd>?	Check the current parameter value of the corresponding command.
Write Command	AT+<cmd>=<p1>[,<p2>[,<p3>[...]]]	Set user-definable parameter value.
Execution Command	AT+<cmd>	Return a specific information parameter or perform a specific action.

2.1.3. Declaration of AT Command Examples

The AT command examples in this document are provided to help you familiarize with AT commands and learn how to use them. The examples, however, should not be taken as Quectel's recommendation or suggestions about how you should design a program flow or what status you should set the module into. Sometimes multiple examples may be provided for one AT command. However, this does not mean that there exists a correlation among these examples and that they should be executed in a given sequence. The URLs, domain names, IP addresses, usernames/accounts, and passwords (if any) in the AT command examples are provided for illustrative and explanatory purposes only, and they should be modified to reflect your actual usage and specific needs.

2.2. Description of AT Commands

2.2.1. AT+QSSLCFG Configure Parameters of an SSL Context

This command configures the SSL version, cipher suite, path of trusted CA certificate, authentication mode, the path of the client certificate and private key, etc. for a specified SSL context. These parameters will be used in the handshake procedure.

<SSL_ctxID> is the index of the SSL context. The module supports 6 SSL contexts at most. On the basis of one SSL context, several SSL connections can be established. The settings such as the SSL version and the cipher suite are stored in the SSL context, and they will be applied to the new SSL connections associated with the SSL context.

AT+QSSLCFG Configure Parameters of an SSL Context	
Test Command AT+QSSLCFG=?	Response +QSSLCFG: "sslversion", (list of supported <SSL_ctxID>s),(list of supported <SSL_version>s) +QSSLCFG: "seclevel", (list of supported <SSL_ctxID>s),(list of supported <seclevel>s) +QSSLCFG: "ciphersuite", (list of supported <SSL_ctxID>s), <cs_id> +QSSLCFG: "negotiatetime", (list of supported <SSL_ctxID>s),(list of supported <negotiate_time>s) +QSSLCFG: "sni", (list of supported <SSL_ctxID>s),(list of supported <SNI>s) +QSSLCFG: "cacert", (list of supported <SSL_ctxID>s), <cacertpath> +QSSLCFG: "cacertex", (list of supported <ca_id>s), <cacertexpath> +QSSLCFG: "clientcert", (list of supported <SSL_ctxID>s), <client_cert_path>

	<p>+QSSLCFG: "clientkey",(list of supported <SSL_ctxID>s),<client_key_path>,<key_pwd></p> <p>+QSSLCFG: "dtls",(list of supported <SSL_ctxID>s),(list of supported <DTLS_enable>s)</p> <p>+QSSLCFG: "dtlsversion",(list of supported <SSL_ctxID>s),(list of supported <DTLS_version>s)</p> <p>+QSSLCFG: "psk",(list of supported <SSL_ctxID>s),<identity>,<key></p> <p>+QSSLCFG: "ignoremulticertchainverify",(list of supported <SSL_ctxID>s),(list of supported <ignore_multicertchain_verify>s)</p> <p>+QSSLCFG: "ignoreinvalidcertsign",(list of supported <SSL_ctxID>s),(list of supported <ignore_invalid_certsign>s)</p> <p>+QSSLCFG: "ignorecertitem",(list of supported <SSL_ctxID>s),(list of supported <ignore_check_item>s)</p> <p>+QSSLCFG: "ignorelocaltime",(list of supported <SSL_ctxID>s),(list of supported <ignore_ltime>s)</p> <p>+QSSLCFG: "session_cache",(list of supported <SSL_ctxID>s),(list of supported <session_cache_enable>s)+QSSLCFG: "closetimemode",(list of supported <SSL_ctxID>s),(list of supported <close_time_mode>s)</p> <p>+QSSLCFG: "renegotiation",(list of supported <SSL_ctxID>s),(list of supported <renegotiation_enable>s)</p> <p>+QSSLCFG: "alpn",(list of supported <SSL_ctxID>s),<ALPN_name></p> <p>OK</p>
<p>Write Command</p> <p>Configure the SSL version for a specified SSL context:</p> <p>AT+QSSLCFG="sslversion",<SSL_ctxID>[,<SSL_version>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query the SSL version for a specified SSL context:</p> <p>+QSSLCFG: "sslversion",<SSL_ctxID>,<SSL_version></p> <p>OK</p> <p>If the optional parameter is specified, set the SSL version for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure the authentication mode for a specified SSL context:</p> <p>AT+QSSLCFG="seclevel",<SSL_ctxID></p>	<p>Response</p> <p>If the optional parameter is omitted, query the authentication mode for a specified SSL context:</p> <p>+QSSLCFG: "seclevel",<SSL_ctxID>,<seclevel></p>

<p>>[,<seclvl>]</p>	<p>OK</p> <p>If the optional parameter is specified, set the authentication mode for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure the SSL cipher suites for a specified SSL context:</p> <p>AT+QSSLCFG="ciphersuite",<SSL_ctxID>[,<cs_id>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query the SSL cipher suites for a specified SSL context:</p> <p>+QSSLCFG: "ciphersuite",<SSL_ctxID>,<cs_id></p> <p>OK</p> <p>If the optional parameter is specified, set the SSL cipher suite for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure the maximum timeout in SSL negotiation stage for a specified SSL context:</p> <p>AT+QSSLCFG="negotiatetime",<SSL_ctxID>[,<negotiate_time>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query the maximum timeout in SSL negotiation stage for a specified SSL context:</p> <p>+QSSLCFG: "negotiatetime",<SSL_ctxID>,<negotiate_time></p> <p>OK</p> <p>If the optional parameter is specified, set the maximum timeout in SSL negotiation stage for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure Server Name Indication feature for a specified SSL context:</p> <p>AT+QSSLCFG="sni",<SSL_ctxID>[,<SNI>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query whether the Server Name Indication feature is enabled for a specified SSL context:</p> <p>+QSSLCFG: "sni",<SSL_ctxID>,<SNI></p> <p>OK</p> <p>If the optional parameter is specified, disable/enable Server Name Indication feature for a specified SSL context:</p> <p>OK</p>

	<p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure the path of trusted CA certificate for a specified SSL context:</p> <p>AT+QSSLCFG="cacert",<SSL_ctxID>[,<cacertpath>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query the path of trusted CA certificate for a specified SSL context:</p> <p>+QSSLCFG: "cacert",<SSL_ctxID>,<cacertpath></p> <p>OK</p> <p>If the optional parameter is specified, set the path of trusted CA certificate for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure the path of trusted CA certificate for a specified <ca_id>:</p> <p>AT+QSSLCFG="cacertex"[,<ca_id>[,<cacertexpath>]]</p>	<p>Response</p> <p>If all optional parameters are omitted, query the path of trusted CA certificate chain for all SSL contexts:</p> <p>+QSSLCFG: "cacertex",0,<cacertexpath></p> <p>...</p> <p>+QSSLCFG: "cacertex",5,<cacertexpath></p> <p>OK</p> <p>If only <cacertpath> is omitted, query the path of trusted CA certificate for a specified <ca_id>:</p> <p>+QSSLCFG: "cacertex",<ca_id>,<cacertexpath></p> <p>OK</p> <p>If all optional parameters are specified, set the path of trusted CA certificate for a specified <ca_id>:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure the path of client certificate for a specified SSL context:</p> <p>AT+QSSLCFG="clientcert",<SSL_ctxID>[,<client_cert_path>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query the path of client certificate for a specified SSL context:</p> <p>+QSSLCFG: "clientcert",<SSL_ctxID>,<client_cert_path></p> <p>OK</p> <p>If the optional parameter is specified, set the path of client certificate for a specified SSL context:</p>

	<p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure the path of client private key for a specified SSL context:</p> <p>AT+QSSLCFG="clientkey",<SSL_ctxID>[,<client_key_path>,<key_pwd>]</p>	<p>Response</p> <p>If the optional parameters are omitted, query the path of client private key for a specified SSL context:</p> <p>+QSSLCFG:</p> <p>"clientkey",<SSL_ctxID>,<client_key_path>,<key_pwd></p> <p>OK</p> <p>If the optional parameters are specified, set the path of client private key for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure the DTLS function for a specified SSL context:</p> <p>AT+QSSLCFG="dtls",<SSL_ctxID>[,<DTLS_enable>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query whether the DTLS function is enabled for a specified SSL context:</p> <p>+QSSLCFG: "dtls",<SSL_ctxID>,<DTLS_enable></p> <p>OK</p> <p>If the optional parameter is specified, enable/disable the DTLS function for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure the DTLS version for a specified SSL context:</p> <p>AT+QSSLCFG="dtlsversion",<SSL_ctxID>[,<DTLS_version>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query the DTLS version supported by a specified SSL context:</p> <p>+QSSLCFG: "dtlsversion",<SSL_ctxID>,<DTLS_version></p> <p>OK</p> <p>If the optional parameter is specified, configure the DTLS version for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure the PSK used in handshake for a specified SSL context:</p>	<p>Response</p> <p>If the optional parameters are omitted, query the current configuration for a specified SSL context:</p>

<p>AT+QSSLCFG="psk",<SSL_ctxID>[,<identity>,<key>]</p>	<p>+QSSLCFG: "psk",<SSL_ctxID>,<identity>,<key></p> <p>OK</p> <p>If the optional parameters are specified, set the PSK used in handshake for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure whether to ignore multiple level certificate chain verification for a specified SSL context:</p> <p>AT+QSSLCFG="ignoremulticertchainverify",<SSL_ctxID>[,<ignore_multicertchain_verify>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query whether the multiple level certificate chain verification is ignored for a specified SSL context:</p> <p>+QSSLCFG: "ignoremulticertchainverify",<SSL_ctxID>,<ignore_multicertchain_verify></p> <p>OK</p> <p>If the optional parameter is specified, set whether or not to ignore multiple level certificate chain verification for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure whether to ignore the invalid certificate signature for a specified SSL context:</p> <p>AT+QSSLCFG="ignoreinvalidcertsign",<SSL_ctxID>[,<ignore_invalid_certsign>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query whether the invalid certificate signature is ignored for a specified SSL context:</p> <p>+QSSLCFG: "ignoreinvalidcertsign",<SSL_ctxID>,<ignore_invalid_certsign></p> <p>OK</p> <p>If the optional parameter is specified, set whether or not to ignore the invalid certificate signature for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure whether to ignore one or more checks specified in the certificate sent by the server for a specified SSL context:</p> <p>AT+QSSLCFG="ignorecertitem",<SSL</p>	<p>Response</p> <p>If the optional parameter is omitted, query whether one or more checks specified in the certificate sent by the server is ignored for a specified SSL context:</p> <p>+QSSLCFG: "ignorecertitem",<SSL_ctxID>,<ignore_ch</p>

<p>_ctxID>[,<ignore_check_item>]</p>	<p>eck_item></p> <p>OK</p> <p>If the optional parameter is specified, set whether or not to ignore one or more checks specified in the certificate sent by the server for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure whether to ignore certificate validity check for a specified SSL context:</p> <p>AT+QSSLCFG="ignorelocaltime",<SSL_ctxID>[,<ignore_ltime>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query whether the certificate validity check is ignored for a specified SSL context:</p> <p>+QSSLCFG: "ignorelocaltime",<SSL_ctxID>,<ignore_ltime></p> <p>OK</p> <p>If the optional parameter is specified, set whether or not to ignore certificate validity check for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Enable/Disable SSL session resumption function for a specified SSL context:</p> <p>AT+QSSLCFG="session_cache",<SSL_ctxID>[,<session_cache_enable>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query whether the SSL session resumption function is enabled for a specified SSL context:</p> <p>+QSSLCFG: "session_cache",<SSL_ctxID>,<session_cache_enable></p> <p>OK</p> <p>If the optional parameter is specified, enable/disable the SSL session resumption function:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Enable/disable the delay in closing the SSL connection for a specified SSL context:</p> <p>AT+QSSLCFG="closetimemode",<SSL_ctxID>[,<close_time_mode>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query whether the delay in closing the SSL connection is enabled for a specified SSL context:</p> <p>+QSSLCFG: "closetimemode",<SSL_ctxID>,<close_time_mode></p>

	<p>OK</p> <p>If the optional parameter is specified, enable/disable the delay in closing the SSL connection for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Enable/disable TLS renegotiation function for a specified SSL context:</p> <p>AT+QSSLCFG="renegotiation",<SSL_ctxID>[,<renegotiation_enable>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query whether the TLS renegotiation function is enabled for a specified SSL context:</p> <p>+QSSLCFG: "renegotiation",<SSL_ctxID>,<renegotiation_enable></p> <p>OK</p> <p>If the optional parameter is specified, enable/disable the TLS renegotiation function for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
<p>Write Command</p> <p>Configure the ALPN information for a specified SSL context:</p> <p>AT+QSSLCFG="alpn",<SSL_ctxID>[,<ALPN_name>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query the ALPN information for a specified SSL context:</p> <p>+QSSLCFG: "alpn",<SSL_ctxID>,<ALPN_name></p> <p>OK</p> <p>If the optional parameter is specified, configure the ALPN information for a specified SSL context:</p> <p>OK</p> <p>Or</p> <p>ERROR</p>
Maximum Response Time	300 ms
Characteristics	<p>The command takes effect immediately.</p> <p>The configurations are not saved.</p>

Parameter

<SSL_ctxID>	Integer type. SSL context ID. Range: 0–5.
<SSL_version>	Integer type. SSL version.
0	SSL 3.0

	1	TLS 1.0
	2	TLS 1.1
	3	TLS 1.2
	<u>4</u>	All
<seclvl>	Integer type. The authentication mode.	
	<u>0</u>	No authentication
	1	Perform server authentication
	2	Perform server and client authentication if requested by the remote server
<cs_id>	Numeric type in HEX format. SSL cipher suites.	
	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
	<u>0xFFFF</u>	Support all cipher suites
<negotiate_time>	Integer type. Indicates maximum timeout used in SSL negotiation stage. Range: 10–300. Default value: 300. Unit: second.	

<SNI>	Integer type. Disables/enables Server Name Indication feature. 0 Disable 1 Enable
<cacertpath>	String type. The path of the trusted CA certificate.
<ca_id>	Integer type. The certificate index of the CA certificate chain.
<cacertxpath>	String type. The path of the trusted CA certificate.
<client_cert_path>	String type. The path of the client certificate.
<client_key_path>	String type. The path of the client private key.
<key_pwd>	String type. The password of the client private key.
<DTLS_enable>	Integer type. Enables/disables the DTLS function. 0 Disable 1 Enable
<DTLS_version>	Integer type. Configures DTLS version. 0 DTLS1.0 1 DTLS1.2 2 ALL
<identity>	String type. Identity of PSK. The length is 0–255.
<key>	String type. Key of PSK. The length is 0–255.
<ignore_multicertchain_verify>	Integer type. Indicates whether or not to ignore the multiple level certificate chains verification. 0 Not to ignore 1 Ignore
<ignore_invalid_certsig>	Integer type. Indicates whether or not to ignore the invalid certificate signature. 0 Not to ignore 1 Ignore
<ignore_check_item>	Integer type. Indicates whether the client ignores one or more checks specified in the certificate sent by the server. The parameter applies an accumulative value if the client ignores more checks. 0 Not ignore any check item in the certificate 1 Ignore that the certificate validity has expired 4 Ignore certificate common name does not match expected CN 8 Ignore that the certificate is not correctly signed by the trusted CA 256 Ignore other reasons (The reason used to verify the callback) 2048 Ignore Usage does not match keyUsage extension 4096 Ignore Usage does not match extendedKeyUsage extension 8192 Ignore Usage does not match nsCertType extension 32768 Ignore that the certificate signed with unacceptable public key algorithm (such as RSA, ECDSA) 65536 Ignore that the certificate signed with an unacceptable key 1048575 Ignore all check items, that is, not to check the certificate
<ignore_ltime>	Integer type. Indicates whether or not to ignore certificate validity check. 0 Not to ignore 1 Ignore

<session_cache_enable>	Integer type. Enables/disables the SSL session resumption function. 0 Disable 1 Enable
<close_time_mode>	Integer type. Enables/disables the delay in closing the SSL connection. 0 Disable, and the unit of SSL close linger time is second 1 Enable, and the unit of SSL close linger time is millisecond
<renegotiation_enable>	Integer type. Enable/disable the TLS renegotiation function. 0 Disable 1 Enable
<ALPN_name>	String type. Configures ALPN protocol name. When the content of this parameter is null (only double quotes are specified), TLS does not contain ALPN extension content.

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

This command sets up an SSL connection, that is, opens an SSL socket to connect a remote server. During the negotiation between the module and the Internet, parameters configured by **AT+QSSLCFG** will be used in the handshake procedure. After shaking hands with the Server successfully, the module can send or receive data over this SSL connection. Also, the module can set up several SSL connections based on one SSL context.

According to steps mentioned in **Chapter 1.2**, before executing **AT+QSSLOPEN**, execute **AT+QIACT** first to activate the PDP context.

It is suggested to wait for a specific period of time (refer to the Maximum Response Time below) for **+QSSLOPEN: <clientID>,<err>** URC to be outputted. If the URC response cannot be received during the time, **AT+QSSLCLOSE** can be used to close the SSL connection.

AT+QSSLOPEN Open an SSL Socket to Connect a Remote Server

Test Command
AT+QSSLOPEN=?

Response

+QSSLOPEN: (list of supported **<PDP_ctxID>s**),(list of supported **<SSL_ctxID>s**),(list of supported **<clientID>s**),**<serveraddr>**,**<server_port>**[(list of supported **<access_mode>s**)]

OK

Write Command
AT+QSSLOPEN=<PDP_ctxID>,<SSL_ctxID>,<clientID>,<serveraddr>,<server_port>[,<access_mode>]

Response

If the **<access_mode>=2** and the SSL connection is successfully set up:

CONNECT

If there is any error:

ERROR

	<p>Error description can be got over AT+QIGETERROR.</p> <p>If the <access_mode>=0/1: OK</p> <p>+QSSLOPEN: <clientID>,<err> <err> is 0 when SSL socket is opened successfully, and <err> is not 0 when opening SSL socket fails.</p> <p>If there is any error: ERROR Error description can be got over AT+QIGETERROR.</p>
Maximum Response Time	Maximum network response time of 150 s, plus configured time of <negotiate_time> .
Characteristics	<p>The command takes effect immediately.</p> <p>The configurations are not saved.</p>

Parameter

<PDP_ctxID>	Integer type. PDP context ID. Range: 1–15.
<SSL_ctxID>	Integer type. SSL context ID. Range: 0–5.
<clientID>	Integer type. Socket index. Range: 0–11.
<serveraddr>	String type. Remote server address.
<server_port>	Integer type. Listening port of remote server.
<access_mode>	Integer type. The access mode of SSL connection. 0 Buffer access mode 1 Direct push mode 2 Transparent transmission mode
<err>	Integer type. Result code. See Chapter 5 for details.
<negotiate_time>	Integer type. Indicates maximum timeout used in SSL negotiation stage. Range: 10–300. Default value: 300. Unit: second.

2.2.3. AT+QSSSEND Send Data over SSL Connection

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

AT+QSSSEND Send Data over SSL Connection	
Test Command AT+QSSSEND=?	Response +QSSSEND: (list of supported <clientID> s)[,(list of supported <sendlen> s)] OK

Write Command Send variable-length data AT+QSSLSSEND=<clientID>	Response > After the above response, input the data to be sent. Tap CTRL+Z to send, and tap ESC to cancel the operation. If the connection has been established and sending is successful: SEND OK If connection has been established but sending buffer is full: SEND FAIL If the connection has not been established, abnormally closed, or any parameter is incorrect: ERROR
Write Command Send fixed-length data AT+QSSLSSEND=<clientID>,<sendlen>	Response > After the above response, input the data until the data length equals <sendlen> . If connection has been established and sending is successful: SEND OK If connection has been established but sending buffer is full, response: SEND FAIL If the connection has not been established, abnormally closed, or any parameter is incorrect: ERROR
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<clientID>	Integer type. Socket index. Range: 0–11.
<sendlen>	Integer type. The length of sending data. Range: 1–1460. Unit: byte.

NOTE

The maximum length of fixed-length data or variable-length data is 1460 bytes.

2.2.4. AT+QSSLRECV Receive Data over SSL Connection

When an SSL connection is opened with **<access_mode>** specified as 0, the module will report URC as **+QSSLURC: "recv",<clientID>** when it receives data from the Internet. You can read the data from buffer by **AT+QSSLRECV**.

AT+QSSLRECV Receive Data over SSL Connection	
Test Command AT+QSSLRECV=?	Response +QSSLRECV: (list of supported <clientID>s),(list of supported <readlen>s) OK
Write Command AT+QSSLRECV=<clientID>,<readlen>	Response If the specified socket connection has received data: +QSSLRECV: <have_readlen><CR><LF><data> OK If the buffer is empty: +QSSLRECV: 0 OK If the connection has not been established, abnormally closed, or any parameter is incorrect: ERROR
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<clientID>	Integer type. Socket index. Range: 0–11.
<readlen>	Integer type. The length of data to be retrieved. Range: 1–1500. Unit: byte.
<have_readlen>	Integer type. The actual data length obtained by AT+QSSLRECV . Unit: byte.
<data>	The retrieved data.

2.2.5. AT+QSSLCLOSE Close an SSL Connection

This command closes an SSL connection. If all SSL connections based on the same SSL context are closed, the module releases the SSL context.

AT+QSSLCLOSE Close an SSL Connection	
Test Command AT+QSSLCLOSE=?	Response +QSSLCLOSE: (list of supported <clientID>s),(list of supported <close_timeout>s) OK
Write Command AT+QSSLCLOSE=<clientID>[,<close_timeout>]	Response If the SSL connection is successfully closed: OK If it is failed to close the connection: ERROR
Maximum Response Time	Determined by <close_timeout>
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<clientID>	Integer type. Socket index. Range: 0–11.
<close_timeout>	Integer type. The timeout of executing AT+QSSLCLOSE . Range: 0–65535. Default value: 10. 0 means closing immediately. The unit of <close_timeout> depends on the configuration of AT+QSSLCFG="closetimemode" : if <close_time_mode>=0, the unit is second. If <close_time_mode>=1, the unit is microsecond.

2.2.6. AT+QSSLSTATE Query the State of SSL Connection

This command queries the socket connection status and can only query the SSL connection status.

AT+QSSLSTATE Query the State of SSL Connection	
Test Command AT+QSSLSTATE=?	Response OK
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

Execution Command AT+QSSLSTATE	Response List of (+QSSLSTATE: <clientID>,"SSLClient",<IP_address>,<remote_port>,<local_port>,<socket_state>,<PDP_ctxID>,<serverID>,<access_mode>,<AT_port>,<SSL_ctxID>)s OK
Maximum Response Time	300 ms
Characteristics	-

Parameter

<clientID>	Integer type. Socket index. Range: 0–11.
<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 state. 0 "Initial" Connection has not been established 1 "Opening" Client is connecting 2 "Connected" Client connection has been established 4 "Closing" Connection is closing
<PDP_ctxID>	Integer type. PDP context ID. Range: 1–15.
<serverID>	Integer type. Reserved.
<access_mode>	Integer type. The access mode of SSL connection. 0 Buffer access mode 1 Direct push mode 2 Transparent transmission mode
<AT_port>	String type. The COM port. "usbat" USB AT port "uart1" Main UART "usbmodem" USB Modem port "uart2" Auxiliary UART
<SSL_ctxID>	Integer type. SSL context ID. Range: 0–5.

2.3. Description of URCs

2.3.1. +QSSLURC: "recv" Notify Received Data

The URC notifies the data received from peer in buffer access mode and direct push mode.

+QSSLURC: "recv" Notify Received Data	
+QSSLURC: "recv",<clientID>	The URC of SSL data incoming in buffer access mode. SSL data can be received by AT+QSSLRECV .
+QSSLURC: "recv",<clientID>,<current_recvlength><CR><LF><data>	The URC of SSL data incoming in direct push mode.

Parameter

<clientID>	Integer type. Socket index. Range: 0–11.
<current_recvlength>	Integer type. The length of actual received data. Unit: byte.
<data>	The received data.

2.3.2. +QSSLURC: "closed" Notify Abnormal Disconnection

The URC notifies that the SSL connection has been disconnected. Disconnection can be caused by many reasons. For example, the Internet closes the connection or the state of PDP is deactivated. The SSL connection state based on a specified socket will be "closing". In such case, **AT+QSSLCLOSE=<clientID>** must be executed to change the SSL connection state to "initial".

+QSSLURC: "closed" Notify Abnormal Disconnection	
+QSSLURC: "closed",<clientID>	The SSL connection based on the specified socket is closed.

Parameter

<clientID>	Integer type. Socket index. Range: 0–11.
------------	--

3 Examples

3.1. Configure and Activate a PDP Context

3.1.1. Configure a PDP Context

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

3.1.2. Activate a PDP Context

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

3.1.3. Deactivate a PDP Context

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

3.2. Configure an SSL Context

```
AT+QSSLCFG="sslversion",1,1 //Set SSL context ID and SSL version as 1.
OK
AT+QSSLCFG="ciphersuite",1,0X0035 //Set SSL context ID as 1 and SSL cipher suites as
TLS_RSA_WITH_AES_256_CBC_SHA.
OK
AT+QSSLCFG="secllevel",1,1 //Set SSL context ID as 1 and authentication mode as
perform server authentication.
OK
```

```
AT+QSSLCFG="cacert",1,"RAM:cacert.pem" //Set SSL context ID as 1 and the path of the trusted CA
                                         certificate as RAM:cacert.pem.
```

OK

3.3. SSL Client in Buffer Access Mode

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

```
AT+QSSLOPEN=1,1,4,"192.0.2.2",8010,0
```

OK

```
+QSSLOPEN: 4,0 //Set up an SSL connection successfully.
```

```
AT+QSSLSTATE //Query the status of all SSL connections.
```

```
+QSSLSTATE: 4,"SSLClient","192.0.2.2",8010,65344,2,1,4,0,"usbmodem",1
```

OK

3.3.2. Send Data in Buffer Access Mode

3.3.2.1. Send Variable-Length Data

```
AT+QSSLSEND=4 //Send variable-length data.
```

>

```
Test data from SSL
```

```
<CTRL+Z>
```

SEND OK

3.3.2.2. Send Fixed-Length Data

```
AT+QSSLSEND=4,18 //Send fixed-length data with the data length of 18 bytes.
```

>

```
Test data from SSL
```

SEND OK

3.3.3. Receive Data in Buffer Access Mode

```
+QSSLURC: "recv",4 //The socket 4 (<clientID>=4) has received data.
```

```
AT+QSSLRCV=4,1500 //Read data. The length of data to be retrieved is 1500 bytes.
```



```
+QSSLRCV: 18 //The actual received data length is 18 bytes.
Test data from SSL

OK
AT+QSSLRCV=4,1500
+QSSLRCV: 0 //No data in buffer.

OK
```

3.3.4. Close an SSL Connection

```
AT+QSSLCLOSE=4 //Close an SSL connection (<clientID>=4). Depending on the
network, the maximum response time is 10 s.

OK
```

3.4. SSL Client in Direct Push Mode

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

```
AT+QSSLOPEN=1,1,4,"192.0.2.2",8011,1
OK

+QSSLOPEN: 4,0 //Set up SSL connection successfully.
AT+QSSLSTATE //Query the status of all SSL connections.
+QSSLSTATE: 4,"SSLClient","192.0.2.2",8011,65047,2,1,4,1,"usbmodem",1

OK
```

3.4.2. Send Data in Direct Push Mode

```
AT+QSSLSEND=4 //Send variable-length data.
>
Test data from SSL
<CTRL+Z>
SEND OK
AT+QSSLSEND=4,18 //Send fixed-length data and the data length is 18 bytes.
>
Test data from SSL
SEND OK
```

3.4.3. Receive Data in Direct Push Mode

```
+QSSLURC: "recv",4,18
Test data from SSL
```

3.4.4. Close an SSL Connection

```
AT+QSSLCLOSE=4 //Close a connection (<clientID>=4). Depending on the
network, the maximum response time is 10 s.
OK
```

3.5. SSL Client in Transparent Transmission Mode

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

```
AT+QSSLOPEN=1,1,4,"192.0.2.2",8011,2 //Set up an SSL connection.
CONNECT //Enter transparent transmission mode.
//Client is sending data from COM port to the Internet directly. (The data
is not visible in the example.)
OK //Use +++ or DTR (set AT&D1 first) to exit transparent transmission
mode. The NO CARRIER result code indicates that the server has
stopped the SSL connection.
```

3.5.2. Set up an SSL Connection and Receive Data in Transparent Transmission Mode

```
AT+QSSLOPEN=1,1,4,"192.0.2.2",8011,2 //Set up an SSL connection.
CONNECT
<Received data> //Client is reading the data.
OK //Use +++ or DTR (set AT&D1 first) to exit transparent transmission
mode. The NO CARRIER result code indicates that the server has
stopped the SSL connection.
```

3.5.3. Close an SSL Connection

```
AT+QSSLCLOSE=4 //Close a connection (<clientID>=4). Depending on the network, the
maximum response time is 10 s.
OK
```

4 Check for Failure in SSL Connection

To identify reasons for the failure to open an SSL connection:

1. Query the status of the specified PDP context by **AT+QIACT?** to check whether the specified PDP context has been activated.
2. Since an invalid DNS server address cannot convert domain name to IP address, if the address of server is a domain name, check whether the DNS server address is valid by **AT+QIDNSCFG=<contextID>**. See *document [1]* for detailed information about **AT+QIDNSCFG**.
3. Check the SSL configuration by **AT+QSSLCFG**, especially the SSL version and cipher suite to ensure that they are supported on server side. If **<seclevel>** has been configured as 1 or 2, then the trusted CA certificate has to be uploaded to the module with **AT+QFUPL**. If the server side has configured “SSLVerifyClient required”, then the client certificate and client private key have to be uploaded to the module with **AT+QFUPL**. For details about certificate validity check, see *Chapter 1.4*. see *document [3]* for more details of **AT+QFUPL**.

5 Result Codes

If an **ERROR** or URC error code is returned after executing SSL AT commands, the details of errors can be queried by **AT+QIGETERROR**. Please note that **AT+QIGETERROR** just returns result code of the last SSL AT command. See [document \[1\]](#) for details of **AT+QIGETERROR**.

Table 4: Result Codes

<err>	Description
0	Operation successful
550	Unknown error
551	Operation blocked
552	Invalid parameter
553	Memory not enough
554	Create socket failed
555	Operation not supported
556	Socket bind failed
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 connection 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

6 Appendix References

Table 5: Related Documents

Document Name
[1] Quectel_EG800Q&EG91xQ_Series_TCP(IP)_Application_Note
[2] Quectel_EG800Q&EG91xQ_Series_AT_Commands_Manual
[3] Quectel_EG800Q&EG91xQ_Series_FILE_Application_Note

Table 6: Terms and Abbreviations

Abbreviation	Description
ALPN	Application Layer Protocol Negotiation
APN	Access Point Name
CA	Certificate Authority
DNS	Domain Name Server
DTR	Data Terminal Ready
DTLS	Datagram Transport Layer Security
PDP	Packet Data Protocol
SNI	Server Name Indication
SSL	Security Socket Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
UART	Universal Asynchronous Receiver/Transmitter
URC	Unsolicited Result Code

USB

Universal Serial Bus
