BG96 SSL Application Note

LPWA Module Series

Rev. BG96_SSL_Application_Note_V1.1

Date: 2020-03-14

Status: Released

www.quectel.com

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company 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 office. 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 to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE 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

Revision History

Version	Date	Author	Description
1.0	2017-11-07	Sherlock ZHAO/ Parker ZHOU	Initial
1.1	2020-03-14	Terrence YANG	Added the following AT commands ● AT+QSSLCFG="dtls" ● AT+QSSLCFG="dtlsversion" ● AT+QSSLCFG="sni" ● AT+QSSLCFG="checkhost"

Contents

Table Index

1 Introduction

This document describes how to use the SSL functionality of Quectel BG96 module. In some cases, in order to ensure communication privacy, the communication between the server and the client should be in an encrypted way to prevent data from eavesdropping, tampering, or forging during the communication process. The SSL function meets these demands.

1.1. SSL Versions and Cipher Suites

The following are the SSL versions supported by BG96.

Table 1: SSL Versions

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

The following are the SSL cipher suites supported by BG96. For detailed description of cipher suites, please refer to *RFC 2246-The TLS Protocol Version 1.0*.

Table 2: Supported SSL Cipher Suites

Code of Cipher Suites	Name 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
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
0XC027	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
0XC028	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
0XC02F	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
0XFFFF	Support all cipher suites above

1.2. The Process of Using SSL Function

Step 1: Configure **<APN>**, **<username>**, **<password>** and other parameters of a PDP context by **AT+QICSGP**. Please refer to ***document [3]*** for details of the command.

Step 2: Activate the PDP context by **AT+QIACT**, then the assigned IP address can be queried by **AT+QIACT?**. Please refer to ***document [3]*** for details of the command.

Step 3: Configure the SSL version, cipher suite, path of trusted CA certificate and the security level for a specified SSL context by **AT+QSSLCFG**.

Step 4: Open SSL client connection by **AT+QSSLOPEN**. **<SSL_ctxID>** is used to specify the SSL context, and **<access_mode>** is used to specify the data access mode.

Step 5: After the SSL connection has been established, data will be sent or received via the connection. For details about how to send and receive data under different data access modes, please refer to ***Chapter 1.3***.

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

Step 7: Deactivate the PDP context by **AT+QIDEACT** command.

1.3. Description of Data Access Modes

The SSL connection supports the following three kinds of data access modes:

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

When opening an SSL connection via **AT+QSSLOPEN** command, the data access mode can be specified by the parameter **<access_mode>**. After the SSL connection has been established, **AT+QISWTMD** command can be used to switch the data access mode.

1. In buffer access mode, data are sent via **AT+QSSLSEND** command, and if the module has received data from the Internet, it will report a URC as **+QSSLURC: "recv",<clientID>**. In such a case, the data can be retrieved via **AT+QSSLRECV** command.

2. In direct push mode, data are sent via **AT+QSSLSEND** command, and if the module has received data from the Internet, the data will be outputted directly via UART1/USB modem/USB AT port in the following format: **+QSSLURC: "recv",<clientID>,<current_recvlength><CR><LF><data>**.

3. In transparent access mode, the corresponding port enters exclusive mode. The data received from COM port will be sent to the Internet directly, and the received data from the Internet will be outputted to COM port directly. **+++** or DTR (**AT&D1** should be set first) can be used to switch the data access mode to buffer access mode. In transparent access mode, if any abnormal SSL disconnection happens, the module will report **NO CARRIER**.

4. To exit from transparent access mode, **+++** or DTR (**AT&D1** should be set first) can be used. To prevent the **+++** from being misinterpreted as data, the following sequence should be followed:

   1) Do not input any character within 1s (at least or longer) before inputting **+++**.
   2) Input **+++** within 1s, and no other characters can be inputted during the time.
   3) Do not input any character within 1s after **+++** has been inputted.
   4) Use **+++** or DTR (**AT&D1** should be set first) to make the module exit from transparent access mode, and wait until **OK** is returned.

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

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

1.4. Certificate Validity Check

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

When certificate validity check is required (**<ignore_ltime>** is set as 0 when executing **AT+QSSLCFG**), in order to avoid a failure, **AT+CCLK** command should be used to configure the module time within the validity time period of the certificate.

2 Description of SSL AT Commands

2.1. Description of AT Commands

2.1.1. AT+QSSLCFG Configure Parameters of an SSL Context

This command can be used to configure the SSL version, cipher suites, security level, CA certificate, client certificate and client key. 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",(**range of supported **<SSL_ctxID>**s**),(**range of supported **<SSL_version>**s**)** **+QSSLCFG: "ciphersuite",(**range of supported **<SSL_ctxID>**s**),(**list of supported **<cipher_suites>**s**)** **+QSSLCFG: "dtls",(**range of supported **<SSL_ctxID>**s**),(**list of supported **<DTLS_enable>**s**)** **+QSSLCFG: "dtlsversion",(**range of supported **<SSL_ctxID>**s**),(**list of supported **<DTLS_version>**s**)** **+QSSLCFG: "cacert",(**range of supported **<SSL_ctxID>**s**),<CA_cert_path>** **+QSSLCFG: "clientcert",(**range of supported **<SSL_ctxID>**s**),<client_cert_path>** **+QSSLCFG: "clientkey",(**range of supported **<SSL_ctxID>**s**),<client_key_path>** **+QSSLCFG: "seclevel",(**range of supported **<SSL_ctxID>**s**),(**range of supported **<seclevel>**s**)** **+QSSLCFG: "sni",(**range of supported **<SSL_ctxID>**s**),(**list of supported **<SNI>**s**)** **+QSSLCFG: "checkhost",(**range of supported **<SSL_ctxID>**s**),(**list of supported **<check_host>**s**)**

	+QSSLCFG: "ignorelocaltime",(range of supported **<SSL_ctxID>**s**),(**range of supported **<ignore_ltime>**s**)** **+QSSLCFG: "negotiatetime",(**range of supported **<SSL_ctxID>**s**),(**range of supported **<negotiate_time>**s**)** **OK**
Write Command **AT+QSSLCFG="sslversion",<SSL_ctxID>[,<SSL_version>]**	Response If **<SSL_version>** is omitted, query the SSL version of the specified SSL context: **+QSSLCFG: "sslversion",<SSL_ctxID>,<SSL_version>** **OK** If **<SSL_version>** is specified, configure the SSL version of the specified SSL context: **OK** Or **ERROR**
Write Command **AT+QSSLCFG="ciphersuite",<SSL_ctxID>[,<cipher_suites>]**	Response If **<cipher_suites>** is omitted, query the SSL cipher suite of the specified SSL context: **+QSSLCFG: "ciphersuite",<SSL_ctxID>,<cipher_suites>** **OK** If **<cipher_suites>** is specified, configure the SSL cipher suite of the specified SSL context: **OK** Or **ERROR**
Write Command **AT+QSSLCFG="dtls",<SSL_ctxID>[,<DTLS_enable>]**	Response If **<DTLS_enable>** is omitted, query whether DTLS feature is enabled for the specified SSL context: **+QSSLCFG: "dtls",<SSL_ctxID>,<DTLS_enable>** **OK** If **<DTLS_enable>** is specified, configure whether to enable DTLS feature for the specified SSL context: **OK** Or **ERROR**

Write Command **AT+QSSLCFG="dtlsversion",<SSL_ctxID>[,<DTLS_version>]**	Response If **<DTLS_version>** is omitted, query the DTLS version of the specified SSL context: **+QSSLCFG: "dtlsversion",<SSL_ctxID>,<DTLS_version>** **OK** If **<DTLS_version>** is specified, configure the DTLS version of the specified SSL context: **OK** Or **ERROR**
Write Command **AT+QSSLCFG="cacert",<SSL_ctxID>[,<cacertpath>]**	Response If **<cacertpath>** is omitted, query the path of trusted CA certificate for the specified SSL context: **+QSSLCFG: "cacert",<SSL_ctxID>,<cacertpath>** **OK** If **<cacertpath>** is specified, configure the path of trusted CA certificate for the specified SSL context: **OK** Or **ERROR**
Write Command **AT+QSSLCFG="clientcert",<SSL_ctxID>[,<client_cert_path>]**	Response If **<client_cert_path>** is omitted, query the path of client certificate for the specified SSL context: **+QSSLCFG: "clientcert",<SSL_ctxID>,<client_cert_path>** **OK** If **<client_cert_path>** is specified, configure the path of client certificate for the specified SSL context: **OK** Or **ERROR**
Write Command **AT+QSSLCFG="clientkey",<SSL_ctxID>[,<client_key_path>]**	Response If **<client_key_path>** is omitted, query the path of client certificate for the specified SSL context: **+QSSLCFG: "clientkey",<SSL_ctxID>,<client_key_path>** **OK**

	If **<client_key_path>** is specified, configure the path of client certificate for the specified SSL context: **OK** Or **ERROR**
Write Command **AT+QSSLCFG="sni",<SSL_ctxID>[,<SNI>]**	Response If **<SNI>** is omitted, query whether server name indication feature is enabled for the specified SSL context: **+QSSLCFG: "sni",<SSL_ctxID>,<SNI>** **OK** If **<SNI>** is specified, configure whether to enable server name indication feature for the specified SSL context: **OK** Or **ERROR**
Write Command **AT+QSSLCFG="seclevel",<SSL_ctxID>[,<seclevel>]**	Response If **<seclevel>** is omitted, query the authentication mode of the specified SSL context: **+QSSLCFG: "seclevel",<SSL_ctxID>,<seclevel>** **OK** If **<seclevel>** is specified, configure the authentication mode of the specified SSL context: **OK** Or **ERROR**
Write Command **AT+QSSLCFG="checkhost",<SSL_ctxID>[,<check_host>]**	Response If **<check_host>** is omitted, query whether the hostname validation feature is enabled for the specified SSL context: **+QSSLCFG: "checkhost",<SSL_ctxID>,<check_host>** **OK** If **<check_host>** is specified, configure whether to enable hostname validation feature for the specified SSL context: **OK** Or **ERROR**
Write Command **AT+QSSLCFG="ignorelocaltime",<SSL_ctxID>[,<ignore_ltime>]**	Response If **<ignore_ltime>** is omitted, query whether the validity check for certification is ignored for the specified SSL

	context: **+QSSLCFG: "ignorelocaltime",<SSL_ctxID>,<ignore_ltime>** **OK** If **<ignore_ltime>** is specified, configure whether to ignore certification validity check for the specified SSL context: **OK** Or **ERROR**
Write Command **AT+QSSLCFG="negotiatetime",<SSL_ctxID>[,<negotiate_time>]**	Response If **<negotiate_time>** is omitted, query the maximum timeout of SSL negotiation for the specified SSL context: **+QSSLCFG: "negotiatetime",<SSL_ctxID>,<negotiate_time>** **OK** If **<negotiate_time>** is specified, configure the maximum timeout of SSL negotiation for the specified SSL context: **OK** Or **ERROR**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations will not be saved.

Parameter

<SSL_ctxID>	Integer type. SSL context ID. The range is 0-5.	
<SSL_version>	Integer type. SSL Version.	
	0	SSL3.0
	1	TLS1.0
	2	TLS1.1
	3	TLS1.2
	4	All
<cipher_suites>	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 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 0XC027 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 0XC028 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 0XC02F TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 0XFFFF Support all
<DTLS_enable>	Integer type. Enable or disable DTLS feature. 0 Disable DTLS feature 1 Enable DTLS feature
<DTLS_version>	Integer type. DTLS version. 0 DTLS1.0 1 DTLS1.2 2 All
<cacertpath>	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.
<seclevel>	Integer type. The authentication mode. 0 No authentication 1 Manage server authentication 2 Manage server and client authentication if requested by the remote server
<SNI>	Integer type. Whether to enable server name indication feature, currently, the only server names supported are DNS hostnames. 0 Disable server name indication 1 Enable server name indication
<check_host>	Integer type. Whether to enable hostname validation feature (Subject Common Name (CN) matches the specified hostname). 0 Disable hostname validation 1 Enable hostname validation
<ignore_ltime>	Integer type. How to deal with expired certificate. 0 Concern validity check for certification 1 Ignore validity check for certification
<negotiate_time>	Integer type. The maximum timeout of SSL negotiation. The value range is 10-300, and the default value is 300. Unit: second.

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

This command is used to set up an SSL connection. 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 Internet successfully, the module can send or receive data via this SSL connection. Also, the module can set up several SSL connections based on one SSL context.

As mentioned in ***Chapter 1.2***, before executing **AT+QSSLOPEN**, **AT+QIACT** command should be executed 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 URC **+QSSLOPEN: <connectID>,<err>** to be outputted. If the URC cannot be received during the maximum response time, **AT+QSSLCLOSE** command 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: (**range of supported **<PDP_ctxID>**s**),(**range of supported **<SSL_ctxID>**s**),(**range of supported **<clientID>**s**),<serveraddr>,<server_port>[,(**range 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>** is transparent access mode and the SSL connection is successfully set up: **CONNECT** If the **<access_mode>** is buffer access mode or direct push mode: **OK** **+QSSLOPEN: <clientID>,<err>** **<err>** is 0 when SSL socket is opened successfully, otherwise **<err>** is not 0. If there is any error: **ERROR** Error description can be got via **AT+QIGETERROR**.
Maximum Response Time	Maximum network response time of 150 s, plus configured time of **<negotiate_time>**.
Characteristics	The command takes effect immediately. The configurations will not be saved.

Parameter

<PDP_ctxID>	Integer type. PDP context ID. The range is 1-16.
<SSL_ctxID>	Integer type. SSL context ID. The range is 0-5.
<clientID>	Integer type. Socket index. The range is 0-11.
<serveraddr>	String type. The address of remote server.
<server_port>	Integer type. The listening port of remote server. The range is 0-65535.

<access_mode>	Integer type. The access mode of SSL connection. The default value will be used if this parameter is omitted in Write Command. 0 Buffer access mode 1 Direct push mode 2 Transparent mode
<err>	Integer type. The error code of the operation. Please refer to ***Chapter 5***.
<negotiate_time>	Please refer to **AT+QSSLCFG** command for details.

2.1.3. 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: (**range of supported **<clientID>**s**)[,(**range of supported **<sendlen>**s**)]** **OK**
Write Command Send variable-length data **AT+QSSLSEND=<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 the connection has been established but sending buffer is full: **SEND FAIL** If the connection has not been established, abnormally closed, or the parameter is incorrect: **ERROR**
Write Command Send fixed-length data **AT+QSSLSEND=<clientID>,<sendlen>**	Response **>** After the above response, input the data until the data length equals **<sendlen>**. If the connection has been established and sending is successful: **SEND OK** If the connection has been established but sending buffer is

	full: **SEND FAIL** If the connection has not been established, abnormally closed, or the parameter is incorrect: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

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

2.1.4. AT+QSSLRECV Retrieve Data via SSL Connection

When an SSL connection is opened with **<access_mode>** specified as 0 (buffer access mode), the module will report URC as **+QSSLURC: "recv",<clientID>** when it receives data from the Internet. The received data can be read from buffer with **AT+QSSLRECV** command.

AT+QSSLRECV Retrieve Data via SSL Connection	
Test Command **AT+QSSLRECV=?**	Response **+QSSLRECV: (**range of supported **<clientID>**s**),(**range of supported **<readlen>**s**)** **OK**
Write Command **AT+QSSLRECV=<clientID>,<readlen>**	Response If the specified connection has received data: **+QSSLRECV: <have_readlen><CR><LF><data>** **OK** If the buffer is empty: **+QSSLRECV: 0** **OK** If any parameter is incorrect or the connection cannot be established: **ERROR**
Maximum Response Time	300 ms

Characteristics	/

Parameter

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

2.1.5. AT+QSSLCLOSE Close an SSL Connection

This command is used to close an SSL connection. If all SSL connections based on the same SSL context are closed, the module will release the SSL context.

AT+QSSLCLOSE Close an SSL Connection	
Test Command **AT+QSSLCLOSE=?**	Response **+QSSLCLOSE: (**range of supported **<clientID>**s**),(**range 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 parameter **<close_timeout>**
Characteristics	The command takes effect immediately. The configuration will not be saved.

Parameter

<clientID>	Integer type. Socket index. The range is 0-11.
<close_timeout>	Integer type. The timeout value of **AT+QSSLCLOSE**. The range is 0-65535, and the default value is 10. Unit: second. 0 means close immediately. The default value will be used if this parameter is omitted in Write Command.

2.1.6. AT+QSSLSTATE Query the State of SSL Connection

This command is used to query the state 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>,"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. The range is 0-11.
<IP_address>	String type. The address of remote server.
<remote_port>	Integer type. The port number of remote server. The range is 0-65535.
<local_port>	Integer type. The local port. The range is 0-65535.
<socket_state>	Integer type. The state of SSL connection. 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. The range is 1-16.
<serverID>	Integer type. Reserved. The value is usually the same as **<clientID>**.
<access_mode>	Integer type. The access mode of SSL connection. 0 Buffer access mode 1 Direct push mode 2 Transparent access mode
<AT_port>	String type. COM port.
<SSL_ctxID>	Integer type. SSL context ID. The range is 0-5.

2.2. Description of URC

2.2.1. +QSSLURC: "recv" Notify Incoming Data

The URC is used to notify incoming data in buffer access mode and direct push mode.

+QSSLURC: "recv" Notify Incoming Data	
+QSSLURC: "recv",<clientID>	Notify the incoming of SSL data in buffer access mode. SSL data can be retrieved by **AT+QSSLRECV**.
+QSSLURC: "recv",<clientID>,<current_recvlength><CR><LF><data>	Notify the incoming of SSL data in direct push mode.

Parameter

<clientID>	Integer type. Socket index. The range is 0-11.
<current_recvlength>	Integer type. The length of actual received data.
<data>	The received data.

2.2.2. +QSSLURC: "closed" Notify Abnormal Close

The URC is used to 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 SSL connection state based on the specified socket will be "closing". In such case, **AT+QSSLCLOSE=<connectID>** must be executed to change the SSL connection state to "initial".

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

Parameter

<clientID>	Integer type. Socket index. The range is 0-11.

3 Examples

3.1. Configure and Activate a PDP Context

3.1.1. Configure a PDP Context

```
AT+QICSGP=1,1,"CMCIOT","","",1         //Configure context 1. APN is "CMCIOT".
OK
```

3.1.2. Activate a PDP Context

```
AT+QIACT=1                           //Activate context 1.
OK                                   //Activated the context 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 the context successfully.
```

3.2. Configure an SSL Context

```
AT+QSSLCFG="sslversion",1,1
OK
AT+QSSLCFG="ciphersuite",1,0X0035
OK
AT+QSSLCFG="seclevel",1,1
OK
AT+QSSLCFG="cacert",1,"cacert.pem"
OK
```

3.3. SSL Client Works in Buffer Access Mode

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

```
AT+QSSLOPEN=1,1,4,"220.180.239.201",8010,0
OK

+QSSLOPEN: 4,0                  //Set up the SSL connection successfully.
AT+QSSLSTATE                    //Query the state of all SSL connections.
+QSSLSTATE: 4,"SSLClient","220.180.239.201",8010,65344,2,1,4,0,"usbmodem",1

OK
```

3.3.2. Send Data in Buffer Access 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
```

3.3.3. Retrieve Data in Buffer Access Mode

```
+QSSLURC: "recv",4              //The socket 4 (<clientID>=) has received data.
AT+QSSLRECV=4,1500              //Retrieve the data. The length of data to be retrieved is 1500
                                bytes.
+QSSLRECV: 18                   //The retrieved data length is 18 bytes.
Test data from SSL

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

OK
```

3.3.4. Close an SSL Connection

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

3.4. SSL Client Works in Direct Push Mode

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

```
AT+QSSLOPEN= 1,1,4,"220.180.239.201",8011,1
OK

+QSSLOPEN: 4,0                    //Set up the SSL connection successfully.
AT+QSSLSTATE                      //Query the status of all SSL connections.
+QSSLSTATE: 4,"SSLClient","220.180.239.201",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. Retrieve Data in Direct Push Mode

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

3.4.4. Close an SSL Connection

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

3.5. SSL Client Works in Transparent Access Mode

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

```
AT+QSSLOPEN= 1,1,4,"220.180.239.201",8011,2   //Set up an SSL connection.
CONNECT                                       //Enter transparent access mode.
                              //The client is sending data from COM port to the Internet directly.
                              (The data is not visible in the example.)
OK                            //Use +++ or DTR (AT&D1 should be set first) to exit from transparent
                              access mode. The result code NO CARRIER indicates that the server
                              has stopped the SSL connection.
```

3.5.2. Set up an SSL Connection and Retrieve Data in Transparent Access Mode

```
AT+QSSLOPEN= 1,1,4,"220.180.239.201",8011,2    //Set up an SSL connection.
CONNECT
<Received data>               //The client is reading the data.
OK                            //Use +++ or DTR (AT&D1 should be set first) to exit from transparent
                              access mode. The result code NO CARRIER indicates that the server
                              has stopped the SSL connection.
```

3.5.3. Close an SSL Connection

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

3.6. DTLS Test Process based on PSK Encryption

3.6.1. Set up an DTLS Connection

```
AT+QFUPL="0_server.psk"        //Upload PSK file first. The PSK file should be named as
                               SSL_ctxID_server.psk (0_server.psk for instance) and its format
                               should be "idxxxxxxxxx&keyxxxxxxx" (eg, DTLS_Client&1a2b3c4d).
CONNECT
<Input file bin data>
+QFUPL: 24,553
OK
AT+QSSLCFG="dtls",0,1                 //Enable DTLS feature for SSL context 0
OK
AT+QSSLCFG="dtlsversion",0,0          //Configure DTLS version to DTLS1.0 for SSL context 0
OK
AT+QSSLCFG="ciphersuite",0,0XFFFF     //Configure cipher suite for SSL context 0
OK
AT+QSSLOPEN=1,0,0,"220.180.239.201",8010,0
OK

+QSSLOPEN: 0,0                        //Set up the SSL connection successfully.
AT+QSSLSTATE                          //Query the status of all SSL connections.
+QSSLSTATE: 0,"SSLClient","220.180.239.201",8010,65344,2,1,4,0,"usbmodem",1

OK
```

4 Error Handling

4.1. Failed to Open SSL Connection

If it is failed to open SSL connection, please check the following aspects:

1. Query the status of the specified PDP context with **AT+QIACT?** to check whether the specified PDP context has been activated.

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

3. Check the SSL configuration with **AT+QSSLCFG**, especially the SSL version and cipher suite, so as to make sure 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, please refer to ***Chapter 1.4***. And for more details about related FILE AT commands, please refer to ***document [4]***.

5 Summary of Error Codes

If **ERROR** is returned after executing SSL AT commands, the details of error can be queried with **AT+QIGETERROR**. Please note that **AT+QIGETERROR** command just returns error code of the last SSL AT command.

Table 3: Summary of Error Codes

<err>	Meaning
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 A References

Table 4: Related Documents

SN	Document Name	Remark
[1]	GSM 07.07	Digital cellular telecommunications (Phase 2+); AT command set for GSM Mobile Equipment
[2]	GSM 07.10	Support GSM 07.10 multiplexing protocol
[3]	Quectel_BG96_TCP(IP)_Application_Note	Introduction about BG96 TCP/IP AT commands
[4]	Quectel_BG96_FILE_AT_Commands_Manual	Introduction about BG96 FILE AT commands

Table 5: Terms and Abbreviations

Abbreviation	Description
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
UART	Universal Asynchronous Receiver/Transmitter
URC	Unsolicited Result Code
USB	Universal Serial Bus