BC660K-GL SSL Application Note

NB-IoT Module Series

Version: 1.0

Date: 2021-12-29

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.

About the Document

Revision History

Version	Date	Author	Description
-	2021-09-09	Randy LI	Creation of the document
1.0	2021-12-29	Randy LI	First official release

Contents

Table Index

1 Introduction

To keep the communication secure and prevent sensitive data from being eavesdropped, tampered, or forged during the communication process, SSL is used to safeguard the communication between the server and the client by using encryption algorithms.

This document describes how to apply the SSL function of Quectel BC660K-GL module.

1.1. SSL Versions

The following SSL versions are supported by BC660K-GL.

Table 1: Supported SSL Versions

SSL Version
TLS1.0
TLS1.1
TLS1.2

1.2. SSL Cipher Suites

The following table shows SSL cipher suites supported by Quectel BC660K-GL module. Please refer to https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml for details of the cipher suites.

Table 2: Supported SSL Cipher Suites (Official IANA Names)

Cipher Suite Code	Cipher Suite Name
0X002F	TLS_RSA_WITH_AES_128_CBC_SHA

0X0035	TLS_RSA_WITH_AES_256_CBC_SHA
0X000A	TLS_RSA_WITH_3DES_EDE_CBC_SHA
0X00FF	TLS_EMPTY_RENEGOTIATION_INFO_SCSV

1.3. DTLS Versions

The following DTLS versions are supported by BC660K-GL.

Table 3: **Supported DTLS Versions**

DTLS Version
DTLS1.0
DTLS1.2

2 SSL Related AT Commands

This chapter gives details of the AT Command set supported by the Quectel NB-IoT module BC660K-GL.

2.1. AT Command Introduction

2.1.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on the 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 settings, 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 4: Types of AT Commands

Command Type	Syntax	Description
Test Command	**AT+<cmd>=?**	Test the existence of corresponding Write Command and return information about the type, value, or range of its parameter.
Read Command	**AT+<cmd>?**	Check the current parameter value of a corresponding Write 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.

2.2. Declaration of AT Command Examples

The AT command examples in this document are provided to help you learn about how to use the AT commands introduced herein. 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.

2.3. Description of AT Commands

2.3.1. AT+QSSLCFG Configure Parameters of an SSL Context

The command configures optional parameters for SSL function.

AT+QSSLCFG Configure Parameters of an SSL Context	
Test Command **AT+QSSLCFG=?**	Response **+QSSLCFG: (**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),"seclevel",(**range of supported **<seclevel>**s**)** **+QSSLCFG: (**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),"sslversion",(**range of supported **<SSL_version>**s**)** **+QSSLCFG: (**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),"dataformat",(**list of supported **<send_data_format>**s**),(**list of supported **<recv_data_format>**s**)** **+QSSLCFG: (**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),"timeout",(**range of supported **<timeout>**s**)** **+QSSLCFG: (**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),"debug",(**range of supported **<debug_level>**s**)** **+QSSLCFG: (**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),"cacert"**

	+QSSLCFG: (range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),"clientcert"** **+QSSLCFG: (**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),"clientkey"** **+QSSLCFG: (**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),"dtls",(**list of supported **<DTLS_enable>**s**)** **+QSSLCFG: (**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),"dtlsversion",(**range of supported **<DTLS_version>**s**)** **OK**
Write Command Query all current settings of the specified context **AT+QSSLCFG=<contextID>,<connectID>**	Response **+QSSLCFG: <contextID>,<connectID>,"seclevel",<seclevel>** **+QSSLCFG: <contextID>,<connectID>,"sslversion",<SSL_version>** **+QSSLCFG: <contextID>,<connectID>,"dataformat",<send_data_format>,<recv_data_format>** **+QSSLCFG: <contextID>,<connectID>,"timeout",<timeout>** **+QSSLCFG: <contextID>,<connectID>,"debug",<debug_level>** **+QSSLCFG: <contextID>,<connectID>,"cacert",<checksum>** **+QSSLCFG: <contextID>,<connectID>,"clientcert",<checksum>** **+QSSLCFG: <contextID>,<connectID>,"clientkey",<checksum>** **+QSSLCFG: <contextID>,<connectID>,"dtls",<DTLS_enable>** **+QSSLCFG: <contextID>,<connectID>,"dtlsversion",<DTLS_version>** **OK** Or **ERROR**
Write Command Configure the authentication mode for the specified SSL context **AT+QSSLCFG=<contextID>,<connectID>,"seclevel"[,<seclevel>]**	Response If the optional parameter is omitted, query the current setting: **+QSSLCFG: <contextID>,<connectID>,"seclevel",<seclevel>** **OK**

	If the optional parameter is specified, set the authentication mode for the specified SSL context: **OK** If there is any error: **ERROR**
Write Command Configure the SSL version for the specified SSL context **AT+QSSLCFG=<contextID>,<connectID>,"sslversion"[,<SSL_version>]**	Response If the optional parameter is omitted, query the current setting: **+QSSLCFG: <contextID>,<connectID>,"sslversion",<SSL_version>** **OK** If optional parameter is specified, set the SSL version for the specified SSL context: **OK** If there is any error: **ERROR**
Write Command Configure the format of data to be sent/received **AT+QSSLCFG=<contextID>,<connectID>,"dataformat"[,<send_data_format>,<recv_data_format>]**	Response If the optional parameters are omitted, query the current setting: **+QSSLCFG: <contextID>,<connectID>,"dataformat",<send_data_format>,<recv_data_format>** **OK** If the optional parameters are specified, configure the format of data to be sent/received: **OK** If there is any error: **ERROR**
Write Command Configure the timeout of connection and message delivery for the specified SSL context **AT+QSSLCFG=<contextID>,<connectID>,"timeout"[,<timeout>]**	Response If the optional parameter is omitted, query the current setting: **+QSSLCFG: <contextID>,<connectID>,"timeout",<timeout>** **OK** If the optional parameter is specified, configure the timeout

	of connection and message delivery for the specified SSL context: **OK** If there is any error: **ERROR**
Write Command Configure the printable debug log level for the specified SSL context **AT+QSSLCFG=<contextID>,<connectID>,"debug"[,<debug_level>]**	Response If the optional parameter is omitted, query the current setting: **+QSSLCFG: <contextID>,<connectID>,"debug",<debug_level>** **OK** If optional parameter is specified, configure the printable debug log level for the specified SSL context: **OK** If there is any error: **ERROR**
Write Command Configure the content of trusted CA certificate in PEM format for the specified SSL context: **AT+QSSLCFG=<contextID>,<connectID>,"cacert"**	Response **>** After the above response, input the data to be sent. Then tap "**CTRL**" + "**Z**" to send the data or tap "**Esc**" to cancel the operation." **+QSSLCFG: <contextID>,<connectID>,"cacert",<checksum>** **OK** If there is any error: **ERROR**
Write Command Configure the content of client certificate in PEM format for the specified SSL context: **AT+QSSLCFG=<contextID>,<connectID>,"clientcert"**	Response **>** After the above response, input the data to be sent. Then tap "**CTRL**" + "**Z**" to send the data or tap "**Esc**" to cancel the operation. **+QSSLCFG: <contextID>,<connectID>,"clientcert",<checksum>** **OK** If there is any error: **ERROR**

Write Command Configure the content of client private key in PEM format for the specified SSL context: **AT+QSSLCFG=<contextID>,<connectID>,"clientkey"**	Response **>** After the above response, input the data to be sent. Then tap "**CTRL**" **+** "**Z**" to send the data or tap "**Esc**" to cancel the operation. **+QSSLCFG: <contextID>,<connectID>,"clientkey",<checksum>** **OK** If there is any error: **ERROR**
Write Command Enable/disable the DTLS function for the specified SSL context **AT+QSSLCFG=<contextID>,<connectID>,"dtls"[,<DTLS_enable>]**	Response If the optional parameter is omitted, query the current setting: **+QSSLCFG: <contextID>,<connectID>,"dtls",<DTLS_enable>** **OK** If the optional parameter is specified, enable/disable the DTLS function for the specified SSL context: **OK** If there is any error: **ERROR**
Write Command Configure the DTLS version of the specified SSL context **AT+QSSLCFG=<contextID>,<connectID>,"dtlsversion"[,<DTLS_version>]**	Response If the optional parameter is omitted, query the current setting: **+QSSLCFG: <contextID>,<connectID>,"dtlsversion",<DTLS_version>** **OK** If the optional parameter is specified, configure the DTLS version of the specified SSL context: **OK** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. Except for the certificate configuration, other configurations

	will not be saved.

Parameter

<contextID>	Integer type. SSL context ID. Range: 0–10 (currently only 0 is supported).
<connectID>	Integer type. SSL connect ID. Range: 0–4 (currently only 0 is supported).
<seclevel>	Integer type. The authentication mode. 0 No authentication 1 Perform server authentication 2 Perform server and client authentication if requested by the remote server
<SSL_version>	Integer type. SSL version. 1 TLS 1.0 2 TLS 1.1 3 TLS 1.2 4 All protocols are supported, the specific protocol version used needs to be negotiated with the server.
<send_data_format>	Integer type. The format of the sent data. 0 Text format 1 Hex format
<recv_data_format>	Integer type. The format of the received data. 0 Text format 1 Hex format
<timeout>	Integer type. Timeout value of connection or message delivery. Range: 10–300. Default value: 90. Unit: second.
<debug_level>	Integer type. The printable debug log level. 0 No debug log 1 Error debug log 2 State debug log 3 Info debug log 4 Detail debug log
<checksum>	Integer type. The length of certificate. Unit: byte.
<DTLS_enable>	Integer type. Enable or disable DTLS feature. 0 Disable DTLS feature. 1 Enable DTLS feature .
<DTLS_version>	Integer type. DTLS version. 0 DTLS 1.0 1 DTLS 1.2 2 All protocols are supported, the specific protocol version used needs to be negotiated with the server.

NOTE

1. **<debug_level>** is used during debugging only. And the bigger the value is, the more log will be generated.
2. If **<seclevel>** is set to 0, no certificates need to be configured. If **<seclevel>** is set to 1, server CA certificate needs to be configured. If **<seclevel>** is set to 2, server CA certificate, client certificate and client private key need to be configured.
3. The configuration of **<timeout>** is only valid for TLS connection and invalid for DTLS connection.

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

This command opens an SSL socket to connect a remote server.

AT+QSSLOPEN Open an SSL Socket to Connect a Remote Server	
Test Command **AT+QSSLOPEN=?**	Response **+QSSLOPEN: (**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),<host_name>,(**range of supported **<port>**s**),(**list of supported **<connect_mode>**s**)** **OK**
Read Command **AT+QSSLOPEN?**	Response **OK**
Write Command **AT+QSSLOPEN=<contextID>,<connectID>,<host_name>,<port>,<connect_mode>**	Response **OK** **+QSSLOPEN: <contextID>,<connectID>,<err>** If there is any error: **ERROR**
Maximum Response Time	It is determined by **<timeout>** of **AT+QSSLCFG** (default value: 90 s) when a TLS server is connected. It is 123 s when a DTLS server is connected.
Characteristics	The command takes effect immediately. The configurations will not be saved.

Parameter

<contextID>	Integer type. SSL context ID. Range: 0–10 (currently only 0 is supported).
<connectID>	Integer type. SSL connect ID. Range: 0–4 (currently only 0 is supported).
<host_name>	String type. IP address or domain name of SSL server. Maximum length: 150 bytes.
<port>	Integer type. Port number of the remote server. Range: 1–65535.

<connect_mode>	Integer type. Data transmission mode of the SSL connection. 0 Non-transparent mode 1 Transparent mode (not supported currently)
<err>	Result code. 0 indicates a successful operation and any other value indicates an error. Please refer to ***Chapter 4*** for more details.

2.3.3. AT+QSSLSEND Send Data Through SSL Connection

This command sends data through the SSL connection after the connection is established.

AT+QSSLSEND Send Data Through SSL Connection	
Test Command **AT+QSSLSEND=?**	Response **+QSSLSEND: (**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),(**range of supported **<send_length>**s**)** **OK**
Read Command **AT+QSSLSEND?**	Response **OK**
Write Command Send variable-length data **AT+QSSLSEND=<contextID>,<connectID>**	Response **>** After the above response, the module enters data mode and the data to be sent can be inputted directly. Tap "**CTRL**" **+** "**Z**" to send the data or tap "**Esc**" to cancel the operation. If the SSL connection is established and the data is sent successfully: **OK** **+QSSLSEND: <contextID>,<connectID>,<err>** If the SSL connection is not established, disconnected, or some other errors occur: **ERROR**
Write Command Send fixed-length data **AT+QSSLSEND=<contextID>,<connectID>,<send_length>**	Response **>** After the above response, the module enters data mode. After that, type the data to be sent until the data length reaches **<send_length>**. If the SSL connection is established and the data is sent successfully: **OK**

	+QSSLSEND: <contextID>,<connectID>,<err> If the SSL connection is not established, disconnected, or some other errors occur: **ERROR**
Maximum Response Time	**<timeout>** of **AT+QSSLCFG** (default value: 90). The actual response time is determined by network.
Characteristics	The command takes effect immediately. The configurations will not be saved.

Parameter

<contextID>	Integer type. SSL context ID. Range: 0–10 (currently only 0 is supported).
<connectID>	Integer type. SSL connect ID. Range: 0–4 (currently only 0 is supported).
<send_length>	Integer type. The length of the data to be sent. Range: 1–1024. Unit: byte.
<err>	Integer type. The result of connection. 0 indicates a successful operation and any other value indicates an error. Please refer to ***Chapter 4*** for more details.

NOTE

When **<send_data_format>** in **AT+QSSLCFG** is set to 1 (hex format), the character length of the data to be inputted after executing **AT+QSSLSEND=<contextID>,<connectID>,<send_length>** must be twice of **<send_length>** value.

2.3.4. AT+QSSLCLOSE Close an SSL Connection

This command closes an SSL connection. If all SSL connections of an 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 **<contextID>**s**),(**range of supported **<connectID>**s**)** **OK**
Read Command **AT+QSSLCLOSE?**	Response **OK**
Write Command **AT+QSSLCLOSE=<contextID>,<connectID>**	Response **OK** **+QSSLCLOSE: <contextID>,<connectID>,<err>**

	If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations will not be saved.

Parameter

<contextID>	Integer type. SSL context ID. Range: 0–10 (currently only 0 is supported).
<connectID>	Integer type. SSL connect ID. The range is 0–4 (currently only 0 is supported).
<err>	Result code. 0 indicates a successful operation and any other value indicates an error. Please refer to ***Chapter 4*** for more details.

2.4. Description of URCs

SSL URCs begin with **+QSSLURC:** and they are mainly used to notify the host of incoming data or disconnected SSL connection.

2.4.1. +QSSLURC: "recv" Notify Incoming Data

The URC notifies the host of incoming data.

+QSSLURC: "recv" Notify Incoming Data	
+QSSLURC: "recv",<contextID>,<connectID>,<length>,<data>	The URC notifies the host of incoming SSL data.

Parameter

<contextID>	Integer type. SSL context ID. Range: 0–10 (currently only 0 is supported).
<connectID>	Integer type. SSL connect ID. Range: 0–4 (currently only 0 is supported).
<length>	Integer type. The length of data. Range: 1–1400. Unit: byte.
<data>	String type. The incoming data.

2.4.2. +QSSLURC: "closed" Notify SSL Connection Disconnected

The URC notifies the host that the SSL connection has been disconnected. When this URC is reported, the module closes the SSL connection automatically, and the host does not need to execute **AT+QSSLCLOSE** to close the SSL connection.

+QSSLURC: "closed" Notify SSL Connection Disconnected	
+QSSLURC: "closed",<contextID>,<connectID>	The SSL connection based on the specified socket is closed.

Parameter

<contextID>	Integer type. SSL context ID. Range: 0–10 (currently only 0 is supported).
<connectID>	Integer type. SSL connect ID. Range: 0–4 (currently only 0 is supported).

3 Example

3.1. SSL Function of Two-Way Authentication

```
AT+QSCLK=0                                  //Disable sleep mode
OK
//Configure certificates and keys
AT+QSSLCFG=0,0,"seclevel",2                 //Perform server and client authentication
OK
AT+QSSLCFG=0,0,"cacert"                     //Configure CA certificate
>
//After the response >, input content of the trusted CA certificate in PEM format, tap "CTRL" + "Z" to send
+QSSLCFG: 0,0,"cacert",1216

OK
AT+QSSLCFG=0,0,"clientcert"                 //Configure client certificate
>
//After the response >, input content of the client certificate in PEM format, tap "CTRL" + "Z" to send.
+QSSLCFG: 0,0,"clientcert",1224

OK
AT+QSSLCFG=0,0,"clientkey"                  //Configure client private  key
>
//After the response >, input content of the client private key in PEM format, tap "CTRL" + "Z" to send.
+QSSLCFG: 0,0,"clientkey",1679

OK
AT+QSSLOPEN=0,0,"hf.quectel.com",8164,0     //Open an SSL server connection
OK

+QSSLOPEN: 0,0,0
AT+QSSLSEND=0,0                             //Send data to SSL server
>
//After the response >, input the data to be sent and tap "CTRL" + "Z" to send.
OK

+QSSLSEND: 0,0,0
```

```
+QSSLURC: "recv",0,0,10,"1234567890"          //Received data from SSL server
AT+QSSLCLOSE=0,0                              //Close the SSL connection
OK

+QSSLCLOSE: 0,0,0
AT+QSCLK=1                                    //Enable light sleep and deep sleep and wake up by
                                                PSM_EINT (falling edge)
OK
```

3.2. DTLS Function of Two-Way Authentication

```
AT+QSCLK=0                                    //Disable sleep mode
OK
//Configure certificates and keys
AT+QSSLCFG=0,0,"dtls",1                       //Enable DTLS feature
OK
//Configure certificates and keys
AT+QSSLCFG=0,0,"seclevel",2                   //Perform server and client authentication
OK
AT+QSSLCFG=0,0,"cacert"                       //Configure CA certificate
>
//After the response >, input content of the trusted CA certificate in PEM format, tap "CTRL" + "Z" to send
+QSSLCFG: 0,0,"cacert",1216

OK
AT+QSSLCFG=0,0,"clientcert"                   //Configure client certificate
>
//After the response >, input content of the client certificate in PEM format, tap "CTRL" + "Z" to send.
+QSSLCFG: 0,0,"clientcert",1224

OK
AT+QSSLCFG=0,0,"clientkey"                    //Configure client private key
>
//After the response >, input content of the client private key in PEM format, tap "CTRL" + "Z" to send.
+QSSLCFG: 0,0,"clientkey",1679

OK
AT+QSSLOPEN=0,0,"hf.quectel.com",8164,0       //Open an SSL server connection
OK

+QSSLOPEN: 0,0,0
```

```
AT+QSSLSEND=0,0                          //Send data to SSL server
>
//After the response >, input the data to be sent and tap "CTRL" + "Z" to send.
OK

+QSSLSEND: 0,0,0

+QSSLURC: "recv",0,0,10,"1234567890"     //Received data from SSL server
AT+QSSLCLOSE=0,0                         //Close the SSL connection
OK

+QSSLCLOSE: 0,0,0
AT+QSCLK=1                               //Enable light sleep and deep sleep and wake up by
                                         PSM_EINT (falling edge)
OK
```

NOTE

For more details about **AT+QSCLK**, please refer to ***document [1]***.

4 Result Codes

Table 5: Summary of Result Codes

<err>	Description
0	Successful operation
-1	Exception error
-2	Connection error
-3	Certificate error
-4	Key error
-5	Cipher error
-6	State error
-7	Time out
-9	Other errors

5 Appendix References

Table 6: Related Document

Document Name
[1] Quectel_BC660K-GL_AT_Commands_Manual

Table 7: Terms and Abbreviations

Abbreviation	Description
CA	Certificate Authority
DTLS	Datagram Transport Layer Security
EINT	External Interrupt
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol
NB-IoT	Narrowband Internet of Things
PEM	Privacy Enhanced Mail
PSK	Pre-Shared Key
PSM	Power Saving Mode
SSL	Security Socket Layer
TA	Terminal Adapter
TLS	Transport Layer Security
URC	Unsolicited Result Code
URL	Uniform Resource Locator