A7600 Series_SSL_AT Command Manual_V1.00

LWG Module

Shanghai SIMCom Wireless Solutions Ltd.
Building A, SIM Technology Building, No.633, Jinzhong Road
Changning District200335
Tel:86-21-31575100/31575200
support@simcom.com
www.simcom.com

Document Title:	A7600 Series_SSL_AT Command Manual
Version:	1.00
Date:	2019-05-28
Status:	Release
Document ID:	A7600 Series_SSL_AT Command Manual_V1.00

General Notes

SIMCom offers this information as a service to its customers, to support application and engineering efforts that use the products designed by SIMCom. The information provided is based upon requirements specifically provided to SIMCom by the customers. SIMCom has not undertaken any independent search for additional relevant information, including any information that may be in the customer's possession. Furthermore, system validation of this product designed by SIMCom within a larger electronic system remains the responsibility of the customer or the customer's system integrator. All specifications supplied herein are subject to change.

Copyright

Version History

Version	Date	Chapter	What is new
V1.00	2019-05-28		New version

Contents

This document is a reference guide to all the AT commands defined for SSL. Through these SSL AT commands, you can communicate with a TCP or SSL server.

1 Introduction

1.1 AT Commands for the SSL Context Management

Step 1: Configure SSL version by AT+CSSLCFG="sslversion",<ssl_ctx_index>,<sslversion>.

Step 2: Configure SSL authentication mode by AT+CSSLCFG="authmode",<ssl_ctx_index>, <authmode>.

Step 3: Configure the flag of ignore local time by

AT+CSSLCFG="ignorlocaltime",<ssl_ctx_index>,<ignoreltime>.

Step 4: Configure the max time in SSL negotiation stage by

AT+CSSLCFG="negotiatetime",<ssl_ctx_index>,<negotiatetime>.

Step 5: Configure the server root CA by AT+CSSLCFG="cacert",<ssl_ctx_index>,<ca_file>.

Step 6: Configure the client certificate by AT+CSSLCFG="clientcert",<ssl_ctx_index>,<clientcert_file>.

Step 7: Configure the client key by AT+CSSLCFG="clientkey",<ssl_ctx_index>,<clientkey_file>.

Step 8: Download the certificate into the module by AT+CCERTDOWN.

Step 9: Delete the certificate from the module by AT+CCERTDELE.

Step 10: List the certificates by AT+CCERTLIST.

1.2 The process of Using SSL AT Commands

Step 1: Ensure GPRS network is available before performing SSL related operations.

Step 2: Configure the parameter of PDP context by AT+CGDCONT.

Step 3: Activate the PDP context to start SSL service by AT+CCHSTART.

Step 4: Configure SSL context by AT+CSSLCFG (if connect to SSL/TLS server).

Step 5: Set the SSL context used in SSL connection by AT+CCHSSLCFG (if connect to SSL/TLS server).

Step 6: Connect to the server by AT+CCHOPEN.

Step 7: Send data to the server by AT+CCHSEND.

Step 8: Receive data from server by AT+CCHRECV in manual receive mode.

Step 9: Disconnect from the server by AT+CCHCLOSE.

Step 10: Deactivate the PDP context to stop SSL service by AT+CCHSTOP.

Note:

2 Description of AT Command

2.1 SSL Context Management AT

2.1.1 AT+CSSLCFG Configure the SSL Context

AT+CSSLCFG Configure the SSL Context	
Test Command **AT+CSSLCFG=?**	Response **+CSSLCFG: "sslversion",(0-9),(0-4)** **+CSSLCFG: "authmode",(0-9),(0-3)** **+CSSLCFG: "ignorelocaltime",(0-9),(0,1)** **+CSSLCFG: "negotiatetime",(0-9),(10-300)** **+CSSLCFG: "cacert",(0-9),(5-108)** **+CSSLCFG: "clientcert",(0-9),(5-108)** **+CSSLCFG: "clientkey",(0-9),(5-108)** **+CSSLCFG: "enableSNI",(0-9),(0,1)** **OK**
Read Command **AT+CSSLCFG?**	Response **+CSSLCFG: 0,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enableSNI>** **+CSSLCFG: 1,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enableSNI>** **+CSSLCFG: 2,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enableSNI>** **+CSSLCFG: 3,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enableSNI>** **+CSSLCFG: 4,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enableSNI>** **+CSSLCFG: 5,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enableSNI>**

	+CSSLCFG: 6,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enableSNI> **+CSSLCFG: 7,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enableSNI>** **+CSSLCFG: 8,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enableSNI>** **+CSSLCFG: 9,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enableSNI>** **OK**
Write Command /*Query the configuration of the specified SSL context*/ **AT+CSSLCFG=<ssl_ctx_index>**	Response **+CSSLCFG: <ssl_ctxindex>,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enableSNI>** **OK**
Write Command /*Configure the version of the specified SSL context*/ **AT+CSSLCFG="sslversion",<ssl_ctx_index>,<sslversion>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the authentication mode of the specified SSL context*/ **AT+CSSLCFG="authmode",<ssl_ctx_index>,<authmode>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the ignore local time flag of the specified SSL context*/ **AT+CSSLCFG="ignorelocaltime",<ssl_ctx_index>,<ignoreltime>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the negotiate timeout value of the specified SSL context*/	Response a)If successfully: **OK**

AT+CSSLCFG="negotiatetime",<ssl_ctx_index>,<negotiatetime>	b)If failed: **ERROR**
Write Command /*Configure the server root CA of the specified SSL context*/ **AT+CSSLCFG="cacert",<ssl_ctx_index>,<ca_file>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the client certificate of the specified SSL context*/ **AT+CSSLCFG="clientcert",<ssl_ctx_index>,<clientcert_file>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the client key of the specified SSL context*/ **AT+CSSLCFG="clientkey",<ssl_ctx_index>,<clientkey_file>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the enableSNI flag of the specified SSL context */ **AT+CSSLCFG="enableSNI",<ssl_ctx_index>,<enableSNI_flag>**	Response a)If successfully: **OK** b)If failed: **ERROR**

Defined Values

<ssl_ctx_index>	The SSL context ID. The range is 0-9.
<sslversion>	The SSL version, the default value is 4. 0 – SSL3.0 1 – TLS1.0 2 – TLS1.1 3 – TLS1.2 4 – All The configured version should be support by server. So you should use the default value if you are not sure that the version which the server supported.
<authmode>	The authentication mode, the default value is 0. 0 – no authentication. 1–server authentication. It needs the root CA of the server. 2–server and client authentication. It needs the root CA of the server, the cert and key of the client. 3–client authentication and no server authentication. It needs the cert and key of the client.
<ignoreltime>	The flag to indicate how to deal with expired certificate, the

	default value is 1. 0 – care about time check for certification. 1 – ignore time check for certification When set the value to 0, it need to set the right current date and time by AT+CCLK when need SSL certification.
<negotiatetime>	The timeout value used in SSL negotiate stage. The range is 10-300 seconds. The default value is 300.
<ca_file>	The root CA file name of SSL context. The file name must have type like ".pem" or ".der". The length of filename is from 5 to 108 bytes. If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename's UTF8 code). There are two ways to download certificate files to module: 1. By AT+CCERTDOWN. 2. By FTPS or HTTPS commands. Please refer to: SIM7500_SIM7600_SIM7800 Series_FTPS_AT Command Manual and SIM7500_SIM7600_SIM7800 Series_HTTP_AT Command Manual
<clientcert_file>	The client cert file name of SSL context. The file name must have type like ".pem" or ".der". The length of filename is from 5 to 108 bytes. If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename's UTF8 code). There are two ways to download certificate files to module: 1. By AT+CCERTDOWN. 2. By FTPS or HTTPS commands. Please refer to: SIM7500_SIM7600_SIM7800 Series_FTPS_AT Command Manual and SIM7500_SIM7600_SIM7800 Series_HTTP_AT Command Manual
<clientkey_file>	The client key file name of SSL context. The file name must have type like ".pem" or ".der". The length of filename is from 5 to 108 bytes. If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename's UTF8 code).

	There are two ways to download certificate files to module: 1. By AT+CCERTDOWN. 2. By FTPS or HTTPS commands. Please refer to: SIM7500_SIM7600_SIM7800 Series_FTPS_AT Command Manual and SIM7500_SIM7600_SIM7800 Series_HTTP_AT Command Manual
<enalbeSNI_flag>	The flag to indicate that enable the SNI flag or not, the default value is 0. 0 – not enable SNI. 1 – enable SNI.

2.1.2 AT+CCERTDOWN Download certificate into the module

AT+CCERTDOWN Download certificate into the module	
Test Command **AT+CCERTDOWN=?**	Response **+CCERTDOWN: (5-108),(1-10240)** **OK**
Write Command **AT+CCERTDOWN=<filename>,<len>**	Response a)If it can be download: > **<input data here>** **OK** b)If failed: **ERROR**

Defined Values

<filename>	The name of the certificate/key file. The file name must have type like ".pem" or ".der". The length of filename is from 5 to 108 bytes. If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename's UTF8 code). For example: If you want to download a file with name "中华.pem", you should convert the "中华.pem" to UTF8 coding (中华.pem), then input the hexadecimal (262378344532443B262378353334453B2E70656D) of UTF8 coding.

<len>	The length of the file data to send. The range is from 1 to 10240 bytes. User should note than every packet data should be no larger than 3072 bytes.

2.1.3 AT+CCERTLIST List certificates

AT+CCERTLIST List certificates	
Execute Command **AT+CCERTLIST**	Response **[+CCERTLIST:<file_name>** **[+CCERTLIST:<file_name>]** **…** **<CR><LF>]** **OK**

Defined Values

<filename>	The certificate/key files which has been downloaded to the module. If the filename contains non-ASCII characters, it will show the non-ASCII characters as UTF8 code.

2.1.4 AT+CCERTDELE Delete certificates

AT+CCERTDELE Delete certificate from the module	
Write Command **AT+CCERTDELE=<filename>**	Response a)If delete successfully: **OK** b)If failed: **ERROR**

Defined Values

<filename>	The name of the certificate/key file. The file name must have type like ".pem" or ".der". The length of filename is from 5 to 108 bytes. If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename's UTF8 code). For example: If you want to download a file with name "中华.pem", you should convert the "中华.pem" to UTF8 coding (中华.pem), then input the hexadecimal

	(262378344532443B262378353334453B2E70656D) of UTF8 coding.

2.2 SSL Services AT

2.2.1 AT+CCHSET Configure the report mode of sending and receiving data

AT+CCHSET is used to configure the mode of sending and receiving data. It must be called before AT+CCHSTART.

AT+CCHSET Configure the report mode of sending and receiving	
Test Command **AT+CCHSET=?**	Response **+CCHSET: (0,1),(0,1)** **OK**
Read Command **AT+CCHSET?**	Response **+CCHSET: <report_send_result>,<recv_mode>** **OK**
Write Command **AT+CCHSET=<report_send_result>[,<recv_mode>]**	Response a)If successfully: **OK** b)If failed: **ERROR**

Defined Values

<report_send_result>	Whether to report result of CCHSEND, the default value is 0: 0– No. 1–Yes. Module will report +CCHSEND: <session_id>,<err> to MCU when complete sending data.
<recv_mode>	The receiving mode: 0– Output the data to MCU whenever received data. 1 – Module caches the received data and notifies MCU with +CCHEVENT: <session_id>, RECV EVENT. MCU can use AT+CCHRECV to receive the cached data (only in manual receiving mode).

2.2.2 AT+CCHMODE Configure the mode of sending and receiving data

AT+CCHMODE is used to select transparent mode (data mode) or non-transparent mode (command mode). The default mode is non-transparent mode. This AT command must be called before calling AT+CCHSTART.
NOTE: There is only one session in the transparent mode, it's the first session.

AT+CCHMODE Configure the mode of sending and receiving	
Test Command **AT+CCHMODE=?**	Response **+CCHMODE: (0,1)** **OK**
Read Command **AT+CCHMODE?**	Response **+CCHMODE: <mode>** **OK**
Write Command **AT+CCHMODE=<mode>**	Response a)If successfully: **OK** b)If failed: **ERROR**

Defined Values

<mode>	The mode value: 0 – Normal. 1 – Transparent mode The default value is 0.

2.2.3 AT+CCHSTART Start SSL service

AT+CCHSTART is used to start SSL service by activating PDP context. You must execute AT+CCHSTART before any other SSL related operations.

AT+CCHSTART Start SSL service	
Execute Command **AT+CCHSTART**	Response a)If start SSL service successfully: **OK** **+CCHSTART: 0** b)If failed: **ERROR** c)If failed:

	ERROR **+CCHSTART: <err>**
Maximum Response Time	120000ms

Defined Values

<err>	The result code, please refer to chapter 2.3.1

2.2.4 AT+CCHSTOP Stop SSL service

AT+CCHSTOP is used to stop SSL service.

AT+CCHSTOP STOP SSL service	
Execute Command **AT+CCHSTOP**	Response a)If stop SSL service successfully: **OK** **+CCHSTOP: 0** b)If failed: **ERROR**

Defined Values

<err>	The result code, please refer to chapter 2.3.1

2.2.5 AT+CCHADDR Get the IPv4 address

AT+CCHADDR is used to get the IPv4 address after calling AT+CCHSTART.

AT+CCHSADDR Get the IPv4 address	
Execute Command **AT+CCHADDR**	Response **a) if successfully, response** **+CCHADDR: < ip_address>** **OK** **b) if pdp has not been activated, response** **ERROR**

Defined Values

<ip_address>	A string parameter that identifies the IPv4 address after PDP

	activated.

2.2.6 AT+CCHSSLCFG Set the SSL context

AT+CCHSSLCFG is used to set the SSL context which to be used in the SSL connection. It must be called before AT+CCHOPEN and after AT+CCHSTART. The setting will be cleared after AT+CCHOPEN failed or AT+CCHCLOSE.

NOTE: If you don't set the SSL context by this command before connecting to SSL/TLS server by AT+CCHOPEN, the CCHOPEN operation will use the SSL context as same as index <session_id> (the 1st parameter of AT+CCHOPEN) when connecting to the server.

AT+CCHSSLCFG Set the SSL context	
Test Command **AT+CCHSSLCFG=?**	Response **+CCHSSLCFG: (0,1),(0-9)** **OK**
Read Command **AT+CCHSSLCFG?**	Response **+CCHSSLCFG: <session_id>,[<ssl_ctx_index >]** **+CCHSSLCFG: <session_id>,[<ssl_ctx_index >]** **OK**
Write Command **AT+CCHSSLCFG=<session_id>,<ssl_ctx_index>**	Response a)If successfully: **OK** b)If failed: **ERROR**

Defined Values

<session_id>	The session_id to operate. It's from 0 to 1.
<ssl_ctx_index>	The SSL context ID which will be used in the SSL connection. Refer to the <ssl_ctx_index> of AT+CSSLCFG.

2.2.7 AT+CCHCFG Configure the Client Context

AT+CCHCFG is used to set the client session context. It must be called before AT+CCHOPEN and after AT+CCHSTART. The setting will be cleared after AT+CCHOPEN failed or AT+CCHCLOSE.

AT+CCHCFG Configure the Client Context	
Test Command	Response

AT+CCHCFG=?	**+CSSLCFG: "sendtimeout",(0-1),(60-150)** **+CSSLCFG: "sslctx",(0-1),(0-9)** **OK**
Read Command **AT+CCHCFG?**	Response **+CCHCFG: 0,<sendtimeout_val>,<sslctx_index>** **+CCHCFG: 1, <sendtimeout_val>,<sslctx_index>** **OK**
Write Command /*Configure the timeout value of the specified client when sending data*/ **AT+CCHCFG="sendtimeout",<session_id>,<sendtimeout_val>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the SSL context index, it's as same as AT+CCHSSLCFG*/ **AT+CCHCFG="sslctx",<session_id>,<sslctx_index>**	Response a)If successfully: **OK** b)If failed: **ERROR**

Defined Values

<session_id>	The session_id to operate. It's from 0 to 1.
<sendtimeout_val>	The timeout value used in sending data stage. The range is 60-150 seconds. The default value is 150.
<sslctx_index>	The SSL context ID which will be used in the SSL connection. Refer to the <ssl_ctx_index> of AT+CSSLCFG.

2.2.8 AT+CCHOPEN Connect to server

AT+CCHOPEN is used to connect the server.

NOTE: If you don't set the SSL context by AT+CCHSSLCFG before connecting a SSL/TLS server by

AT+CCHOPEN, it will use the <session_id>(the 1'st parameter of AT+CCCHOPEN) SSL context when connecting to the server.

AT+CCHOPEN Connect to server	
Test Command **AT+CCHOPEN=?**	Response **+CCHOPEN: (0,1),"ADDRESS",(1-65535)[,(1-2)[,(1-65535)]]** **OK**
Read Command **AT+CCHOPEN?**	Response If connect to a server, it will show the connected information. Otherwise, the connected information is empty. **+CCHOPEN: 0,"<host>",<port>,<client_type>,<bind_port>** **+CCHOPEN: 1,"<host>",<port>,<client_type>,<bind_port>** **OK**
Write Command **AT+CCHOPEN=<session_id>, "<host>",<port>[<client_type>,[<bind_port>]]**	Response a)If connect successfully: **OK** **+CCHOPEN: <session_id>,0** b)If connect successfully in transparent mode: **CONNECT [<text>]** c)If failed: **OK** **+CCHOPEN: <session_id>,<err>** e)If failed: **ERROR** f)If failed in transparent mode: **CONNECT FAIL**

Defined Values

<session_id>	The session index to operate. It's from 0 to 1.
<host>	The server address, maximum length is 256 bytes.
<port>	The server port which to be connected, the range is from 1 to 65535.
<client_type>	The type of client, default value is 2: 1 – TCP client. 2 – SSL/TLS client.
<bind_port>	The local port for channel, the range is from 1 to 65535.
<text>	CONNECT result code string; the string formats please refer ATX/AT\V/AT&E command.

<err>	The result code: 0 is success. Other values are failure. Please refer to chapter 2.3.1

2.2.9 AT+CCHCLOSE Disconnect from server

AT+CCHCLOSE is used to disconnect from the server.

AT+CCHCLOSE Disconnect from the Server	
Write Command **AT+CCHCLOSE=<session_id>**	Response a)If successfully: **OK** **+CCHCLOSE: <session_id>,0** b)If successfully in transparent mode: **OK** **CLOSED** c)If failed: **ERROR**

Defined Values

<session_id>	The session index to operate. It's from 0 to 1.
<err>	The result code: 0 is success. Other values are failure. Please refer to chapter 2.3.1.

2.2.10 AT+CCHSEND Send data to server

You can use AT+CCHSEND to send data to server.

AT+CCHSEND Send Data	
Test Command **AT+CCHSEND=?**	Response: **+CCHSEND: (0,1),(1-2048)** **OK**
Read Command **AT+CCHSEND?**	Response: **+CCHSEND: 0,<unsent_len_0>,1,<unsent_len_1>** **OK**

Write Command **AT+CCHSEND=<session_id>,<len>**	Response a)if parameter is right: > **<input data here>** When the total size of the inputted data reaches **<len>,** TA will report the following code. Otherwise, the serial port will be blocked. **OK** b)If parameter is wrong or other errors occur: **ERROR**

Defined Values

<session_id>	The session_id to operate. It's from 0 to 1.
<len>	The length of data to send. Its range is from 1 to 2048 bytes.
<unsent_len_0>	The data of connection 0 cached in sending buffer which is waiting to be sent.
<unsent_len_1>	The data of connection 1 cached in sending buffer which is waiting to be sent.

2.2.11 AT+CCHRECV Read the cached data that received from the server

You can use AT+CCHRECV to read the cached data which received from the server.
Note: If connection is closed by server, the cached data will not be cleaned.

AT+CCHRECV Read the cached data that received from server	
Read Command **AT+CCHRECV?**	Response **+CCHRECV: LEN,<cache_len_0>,<cache_len_1>** **OK**
Write Command **AT+CCHRECV=<session_id>[,<max_recv_len>]**	Response a)if parameter is right and there are cached data: **OK** **[+CCHRECV: DATA, <session_id>,<len>** **…** **+CCHRECV: DATA, <session_id>,<len>** **…]**

	+CCHRECV: <session_id>,<err> b) if parameter is not right or any other error occurs: **+CCHRECV: <session_id>,<err>** **ERROR** c) others: **ERROR**

Defined Values

<session_id>	The session id to operate. It's from 0 to 1.
<max_recv_len>	Maximum bytes of data to receive in the current AT+CCHRECV calling. It will read all the received data when the value is greater than the length of RX data cached for session <session_id>. 0 means the maximum bytes to receive is 2048 bytes. (But, when 2048 is greater than the length of RX data cached for session <session_id>, 0 means the length of RX data cached for session <session_id>). The default value is the length of RX data cached for session <session_id>. It will be not allowed when there is no data in the cache.
<cache_len_0>	The length of RX data cached for connection 0.
<cache_len_1>	The length of RX data cached for connection 1.
<len>	The length of data followed.
<err>	The result code: 0 is success. Other values are failure. Please refer to chapter 2.3.1.

2.3 Command result codes and unsolicited codes

2.3.1 Command result <err> codes

Result codes	
0	Operation succeeded
1	Alerting state(reserved)
2	Unknown error
3	Busy
4	Peer closed
5	Operation timeout
6	Transfer failed

7	Memory error
8	Invalid parameter
9	Network error
10	Open session error
11	State error
12	Create socket error
13	Get DNS error
14	Connect socket error
15	Handshake error
16	Close socket error
17	Nonet
18	Send data timeout
19	Not set certificates

2.3.2 Unsolicited result codes

Unsolicited codes	
+CCHEVENT: <session_id>,RECV EVENT	In manual receiving mode, when new data of a connection arriving to the module, this unsolicited result code will be reported to MCU.
+CCH_RECV_CLOSED: <session_id>,<err>	When receive data occurred any error, this unsolicited result code will be reported to MCU.
+CCH_PEER_CLOSED: <session_id>	The connection is closed by the server.

3 Example

Before all SSL related operations, we should ensure the following:
Ensure GPRS network is available:

```
AT+CSQ

+CSQ: 23,0


OK

AT+CREG?

+CREG: 0,1
```

OK

AT+CGREG?

+CGREG: 0,1

OK

3.1 Access to TCP server

Following commands shows how to communicate with a TCP server.

//Enable reporting +CHSEND result

AT+CCHSET=1

OK

//start SSL service, activate PDP context

AT+CCHSTART

OK

+CCHSTART: 0

//connect to TCP server
AT+CCHOPEN=0,"www.baidu.com"**,80,1**
OK

+CCHOPEN: 0,0
//send data to server
AT+CCHSEND=0,121
>GET / HTTP/1.1
Host: www.baidu.com
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:2.0) Gecko/20100101 Firefox/4.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-cn,zh;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: GB2312,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115

Connection: keep-alive
Cookie: BAIDUID=D6F6D0D297CCAE39BD45C683996696C7:FG=1;
Hm_lvt_9f14aaa038bbba8b12ec2a4a3e51d254=1321597443439;
USERID=e194072f4759c0f7c2b6e5d3b09298984fd1

OK

+CCHSEND: 0,0
//report the received data from server
+CCHRECV: DATA,0,757
HTTP/1.1 302 Found
Connection: Keep-Alive
Content-Length: 225
Content-Type: text/html
Date: Wed, 05 Sep 2018 08:59:38 GMT
Location: https://www.baidu.com/
Server: BWS/1.1
Set-Cookie: BIDUPSID=D6F6D0D297CCAE39BD45C683996696C7; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: PSTM=1536137978; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: BD_LAST_QID=11878059346481009304; path=/; Max-Age=1
X-Ua-Compatible: IE=Edge,chrome=1

<html>
<head><title>302 Found</title></head>
<body bgcolor="white">
** <h1>302 Found</h1> **
<hr> 7a367f7b87705e16b985e34ca59b8ae8b1d28d47
**Time : Tue Aug 21 10:55:16 CST 2018 **
</body>
</html>
//Disconnect from the Service
AT+CCHCLOSE=0
OK

+CCHCLOSE: 0,0
//stop SSL Service
AT+CCHSTOP
OK

+CCHSTOP: 0

3.2 Access to SSL/TLS server (not verify server and client)

Following commands shows how to access to a SSL/TLS server without verifying the server. It needs to configure the authentication mode to 0, and then it will connect to the server successfully.

```
//Set the SSL version of the first SSL context
AT+CSSLCFG=”sslversion”,0,4
OK
//Set the authentication mode(not verify server) of the first SSL context
AT+CSSLCFG=”authmode”,0,0
OK
//Enable reporting +CHSEND result
AT+CCHSET=1
OK
// start SSL service, activate PDP context
AT+CCHSTART
OK

+CCHSTART: 0
// Set the first SSL context to be used in the SSL connection
AT+CCHSSLCFG=0,0
OK
//connect to SSL/TLS server
AT+CCHOPEN=0,“www.baidu.com”, 443,2
OK

+CCHOPEN: 0,0
//send data to server
AT+CCHSEND=0,121
>GET / HTTP/1.1
Host: www.baidu.com
User-Agent: MAUI htp User Agent
Proxy-Connection: keep-alive
Content-Length: 0


OK
```

+CCHSEND: 0,0
//report the received data from server

+CCHRECV: DATA,0,917
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: no-cache
Connection: Keep-Alive
Content-Length: 227
Content-Type: text/html
Date: Tue, 04 Sep 2018 06:21:35 GMT
Etag: "5b7b7f40-e3"
Last-Modified: Tue, 21 Aug 2018 02:56:00 GMT
P3p: CP=" OTI DSP COR IVA OUR IND COM "
Pragma: no-cache
Server: BWS/1.1
Set-Cookie: BD_NOT_HTTPS=1; path=/; Max-Age=300
Set-Cookie: BIDUPSID=D95046B2B3D5455BF01A622DB8DED9EA; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: PSTM=1536042095; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com
Strict-Transport-Security: max-age=0
X-Ua-Compatible: IE=Edge,chrome=1

```
<html>
<head>
    <script>
        location.replace(location.href.replace("https://","http://"));
    </script>
</head>
<body>
    <noscript><meta http-equiv="refresh" content="0;url=http://www.baidu.com/"></noscript>
</body>
</html>
```

//Disconnect from the Service
AT+CCHCLOSE=0
OK

+CCHCLOSE: 0,0
//stop SSL Service
AT+CCHSTOP
OK

+CCHSTOP: 0

3.3 Access to SSL/TLS server (only verify the server)

Following commands shows how to access to a SSL/TLS server with verifying the server. It needs to configure the authentication mode to 1 and the right server root CA, and then it will connect to the server successfully.

//Set the SSL version of the first SSL context

AT+CSSLCFG=”sslversion”,0,4

OK

//Set the authentication mode(verify server) of the first SSL context

AT+CSSLCFG=”authmode”,0,1

OK

//Set the server root CA of the first SSL context

AT+CSSLCFG=”cacert”,0,”ca_cert.pem”

OK

//Enable reporting +CHSEND result
AT+CCHSET=1
OK
// start SSL service, activate PDP context
AT+CCHSTART
OK

+CCHSTART: 0
// Set the first SSL context to be used in the SSL connection
AT+CCHSSLCFG=0,0
OK
//connect to SSL/TLS server
AT+CCHOPEN=0,”www.baidu.com”,443,2
OK

+CCHOPEN: 0,0
//send data to server
AT+CCHSEND=0,121
>GET / HTTP/1.1
Host: www.baidu.com

User-Agent: MAUI htp User Agent
Proxy-Connection: keep-alive
Content-Length: 0

OK

+CCHSEND: 0,0
//report the received data from server

+CCHRECV: DATA,0,917
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: no-cache
Connection: Keep-Alive
Content-Length: 227
Content-Type: text/html
Date: Tue, 04 Sep 2018 06:21:35 GMT
Etag: "5b7b7f40-e3"
Last-Modified: Tue, 21 Aug 2018 02:56:00 GMT
P3p: CP=" OTI DSP COR IVA OUR IND COM "
Pragma: no-cache
Server: BWS/1.1
Set-Cookie: BD_NOT_HTTPS=1; path=/; Max-Age=300
Set-Cookie: BIDUPSID=D95046B2B3D5455BF01A622DB8DED9EA; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: PSTM=1536042095; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com
Strict-Transport-Security: max-age=0
X-Ua-Compatible: IE=Edge,chrome=1

```
<html>
<head>
    <script>
        location.replace(location.href.replace("https://","http://"));
    </script>
</head>
<body>
    <noscript><meta http-equiv="refresh" content="0;url=http://www.baidu.com/"></noscript>
</body>
</html>
```

//Disconnect from the Service
AT+CCHCLOSE=0
OK

+CCHCLOSE: 0,0
//stop SSL Service
AT+CCHSTOP
OK

+CCHSTOP: 0

3.4 Access to SSL/TLS server (verify server and client)

Following commands shows how to access to a SSL/TLS server with verifying the server and client. It needs to configure the authentication mode to 2, the right server root CA, the right client certificate and key, and then it will connect to the server successfully.

//Set the SSL version of the first SSL context

AT+CSSLCFG=”sslversion”,0,4

OK

//Set the authentication mode(verify server and client) of the first SSL context

AT+CSSLCFG=”authmode”,0,2

OK

//Set the server root CA of the first SSL context

AT+CSSLCFG=”cacert”,0,”ca_cert.pem”

OK

//Set the client certificate of the first SSL context

AT+CSSLCFG=”clientcert”,0,”cert.pem”

OK

//Set the client key of the first SSL context

AT+CSSLCFG=”clientkey”,0,”key_cert.pem”

OK

//Enable reporting +CHSEND result
AT+CCHSET=1
OK
// start SSL service, activate PDP context

AT+CCHSTART
OK

+CCHSTART: 0
// Set the first SSL context to be used in the SSL connection
AT+CCHSSLCFG=0,0
OK
//connect to SSL/TLS server
AT+CCHOPEN=0, ”www.baidu.com”,443,2
OK

+CCHOPEN: 0,0
//send data to server
AT+CCHSEND=0,121
>GET / HTTP/1.1
Host: www.baidu.com
User-Agent: MAUI htp User Agent
Proxy-Connection: keep-alive
Content-Length: 0

OK

+CCHSEND: 0,0
//report the received data from server

+CCHRECV: DATA,0,917
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: no-cache
Connection: Keep-Alive
Content-Length: 227
Content-Type: text/html
Date: Tue, 04 Sep 2018 06:21:35 GMT
Etag: "5b7b7f40-e3"
Last-Modified: Tue, 21 Aug 2018 02:56:00 GMT
P3p: CP=" OTI DSP COR IVA OUR IND COM "
Pragma: no-cache
Server: BWS/1.1
Set-Cookie: BD_NOT_HTTPS=1; path=/; Max-Age=300
Set-Cookie: BIDUPSID=D95046B2B3D5455BF01A622DB8DED9EA; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: PSTM=1536042095; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com

```
Strict-Transport-Security: max-age=0
X-Ua-Compatible: IE=Edge,chrome=1

<html>
<head>
    <script>
        location.replace(location.href.replace("https://","http://"));
    </script>
</head>
<body>
    <noscript><meta http-equiv="refresh" content="0;url=http://www.baidu.com/"></noscript>
</body>
</html>
```

//Disconnect from the Service

AT+CCHCLOSE=0

OK

+CCHCLOSE: 0,0

//stop SSL Service

AT+CCHSTOP

OK

+CCHSTOP: 0

3.5 Access to SSL/TLS server (only verify the client)

Following commands shows how to access to a SSL/TLS server with verifying the client. It needs to configure the authentication mode to 3, the right client certificate and key, and then it will connect to the server successfully.

//Set the SSL version of the first SSL context

AT+CSSLCFG=”sslversion”,0,4

OK

//Set the authentication mode(only verify client) of the first SSL context

AT+CSSLCFG=”authmode”,0,3

OK

//Set the client certificate of the first SSL context

AT+CSSLCFG=”clientcert”,0,”cert.pem”

OK

//Set the client key of the first SSL context

AT+CSSLCFG=”clientkey”,0,”key_cert.pem”

OK

//Enable reporting +CHSEND result
AT+CCHSET=1
OK
// start SSL service, activate PDP context
AT+CCHSTART
OK

+CCHSTART: 0
// Set the first SSL context to be used in the SSL connection
AT+CCHSSLCFG=0,0
OK
//connect to SSL/TLS server
AT+CCHOPEN=0, ”www.baidu.com”, 443,2
OK

+CCHOPEN: 0,0
//send data to server
AT+CCHSEND=0,121
>GET / HTTP/1.1
Host: www.baidu.com
User-Agent: MAUI htp User Agent
Proxy-Connection: keep-alive
Content-Length: 0

OK

+CCHSEND: 0,0
//report the received data from server

+CCHRECV: DATA,0,917
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: no-cache
Connection: Keep-Alive

```
Content-Length: 227
Content-Type: text/html
Date: Tue, 04 Sep 2018 06:21:35 GMT
Etag: "5b7b7f40-e3"
Last-Modified: Tue, 21 Aug 2018 02:56:00 GMT
P3p: CP=" OTI DSP COR IVA OUR IND COM "
Pragma: no-cache
Server: BWS/1.1
Set-Cookie: BD_NOT_HTTPS=1; path=/; Max-Age=300
Set-Cookie: BIDUPSID=D95046B2B3D5455BF01A622DB8DED9EA; expires=Thu, 31-Dec-37 23:55:55
GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: PSTM=1536042095; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/;
domain=.baidu.com
Strict-Transport-Security: max-age=0
X-Ua-Compatible: IE=Edge,chrome=1

<html>
<head>
    <script>
        location.replace(location.href.replace("https://","http://"));
    </script>
</head>
<body>
    <noscript><meta http-equiv="refresh" content="0;url=http://www.baidu.com/"></noscript>
</body>
</html>
```

//Disconnect from the Service

AT+CCHCLOSE=0

OK

+CCHCLOSE: 0,0

//stop SSL Service

AT+CCHSTOP

OK

+CCHSTOP: 0

3.6 Access to SSL/TLS server in transparent mode

Following commands shows how to access to a SSL/TLS server with not verifying the server in transparent mode.

It needs to configure the sending and receiving mode to 1(the transparent mode).
Only the session 0 is support the transparent mode.

```
//Set the transparent mode
AT+CCHMODE=1

OK

//Enable reporting +CHSEND result
AT+CCHSET=1
OK
// start SSL service, activate PDP context
AT+CCHSTART
OK

+CCHSTART: 0
// Set the first SSL context to be used in the SSL connection
AT+CCHSSLCFG=0,0
OK
//connect to SSL/TLS server
AT+CCHOPEN=0, ”www.baidu.com”, 443,2
CONNECT 115200
//send data to server
GET / HTTP/1.1
Host: www.baidu.com
User-Agent: MAUI htp User Agent
Proxy-Connection: keep-alive
Content-Length: 0


//report the received data from server
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: no-cache
Connection: Keep-Alive
Content-Length: 227
Content-Type: text/html
Date: Tue, 04 Sep 2018 06:26:03 GMT
Etag: "5b7b7f40-e3"
Last-Modified: Tue, 21 Aug 2018 02:56:00 GMT
P3p: CP=" OTI DSP COR IVA OUR IND COM "
Pragma: no-cache
Server: BWS/1.1
```

Set-Cookie: BD_NOT_HTTPS=1; path=/; Max-Age=300
Set-Cookie: BIDUPSID=F19D0F1E532ED84CE275BC1006F91F9E; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: PSTM=1536042363; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com
Strict-Transport-Security: max-age=0
X-Ua-Compatible: IE=Edge,chrome=1

<html>
<head>
<script>
location.replace(location.href.replace("https://","http://"));
</script>
</head>
<body>
<noscript><meta http-equiv="refresh" content="0;url=http://www.baidu.com/"></noscript>
</body>
</html>
//switch to command mode
+++
OK
//Disconnect from the Service
AT+CCHCLOSE=0
OK

CLOSED
//stop SSL Service
AT+CCHSTOP
OK

+CCHSTOP: 0

3.7 Download certificate into module

Following commands shows how to download certificate into module.

//download file with ASCII coding file name

AT+CCERTDOWN=”client_key.der”,1702

>-----BEGIN RSA PRIVATE KEY-----

MIIEowIBAAKCAQEAlwuz/TNa+foGBG6rXpWE1Wnuc+GN9vS7MRenKOH+z2UfGuaV

BSb8VYFCgoL4RnWLwXAcLlaqw88zlCN89EK6IydaAwNmI/U6nu3oPsVkn8r9+sOX

yh9VD01DmSU349QWJvRgt1ocsFI1VTdd6RDkVtu7FdKv4XC5WHcOD7yrEIsVa7+G

Qbnm5cCCz8E75HH8vHZAOFeaV3HvIHnh/1RZ+jh4ysyhEmFNOFCn3r9v2yu4kPRX

43xEsB13Ue4HgSbnT+Q7LlEK+dfsmUBoSpsS2NAmQOiqGrmmYygT3/V/lSX54hit

gli5bvg9DuNHYBwh2C+4nyZF95pMj2dEJf4jNwIDAQABAoIBAAJ9ze06QKDo79p4

3NjFjJhck/NTYB0XslK/+iDhgWt4VogCD6kzGGxsomU2tdOrsq9xIvXcthpeu5IQ

98mrpBhaWNC96JxlOh9O+0q1xNAh8AiH22QZGjUTaC8Jfx+B6w+fbkz37os1/+00

6ZajkbChFTfp7r7ANj5wUEoQKZ4vNpLJxLWDk6uH4ZMNveWcBaZQ21TUg9ZmoskK

EJ2ZEr/3kOSBgi2B6F50zyL8f1mbqPahHNLqtrndV5/Lr4n74TqZXRwt5Cl9GrBv

tYXDHc+5Y7e1TUlXV00AMDIk+3cVR8m8Oa20tSdXjcw2iUk9brxb4uxreOouGfPW

5lO+q1ECgYEA4Kkok17DVx5FiapFQvJ2Jqi2/WhzDncuBGbZtcLZnwRVfkPn3cBZ

JGNwxYyfEdwItPvTYQYh6Qg81XRdSRfF43GzkQXNmkPOdZM0x3tFwzV6K5Fg7aeR

g50UddaA9MraCltOgK++7C6BvA3ImXciK4VWeSZOmDW99Y6mgf92RdkCgYEArB2u

/Id72LGQBmx0Z+36Hf1dxo6RQ+dB+m6XBMR8iuB/jGO/5PHdFoKoF2qa9Yj2W1+X

B29Xmc1HS6GTvkDIsN5JXNO7fDmlAxd5whbwDdcmv3VEt8xJ2UeACIawjKtVcFoH

LRNlvDBttWVvlCZg+9HfVpuPm14oFxN/HtSXt48CgYACxDJ6thUDspy6mD0oGOI5

kaRHNI0OJYuMhFOz+EVDvwLqfh2RzneKiiruU8/1oVb+G4e7zx6FxxMwsbEgYEmQ

hmrmo0Kn3qPhMMHanvr572Oku7KM2p5hF4MT/GM0IHdU31D1JrTcJap1TVomAaCL

FqY88arQFwFSz8HfIe0r6QKBgCbQLtTdzKzqJdt8+6cwQFYg+9O59MJGVVefNskp

chhzVfAX0n9Tl5Lq9fMJ5FX4g+3JGargjfWuGCTTFBk0TM2t4wde7AmwiiivU5LU

T2Afo6pLTKrSE9k+yX2iug+O156VfsbIeAm/Ng5RCJ91JCvFgULro6/axNmnWORf

9rK7AoGBAIK4edrX1MjerCsLu3y9Dy4pAx6ER6ei4xpkO25U8wUcqqc+YD2m2xlA

DjqROITeaxXkmPIyRKAXVarhk8LmXT/oDFUAPsTqUZ9LBrviqtMi+G2OFPbdKDwe

ZBNAgwFpFlUVoi0UYnZF8rBq0tepqivrayEWdKKfMMJjq+l72SxD

-----END RSA PRIVATE KEY-----

OK

//download file with not ASCII coding file name

AT+CCERTDOWN={non-ascii}"262378344532443B262378353334453B2E70656D",1918

>-----BEGIN CERTIFICATE-----

MIIFRDCCAyygAwIBAgIIZmPau7FelQswDQYJKoZIhvcNAQELBQAwQDELMAkGA1UE

BhMCU0kxGzAZBgNVBAoMEnN0YXRlLWluc3RpdHV0aW9uczEUMBIGA1UEAwwLVGF4

IENBIFRlc3QwHhcNMTUwNzIzMTUyOTA1WhcNMzUwNzIzMTUyOTA1WjBAMQswCQYD

VQQGEwJTSTEbMBkGA1UECgwSc3RhdGUtaW5zdGl0dXRpb25zMRQwEgYDVQQDDAtU

YXggQ0EgVGVzdDCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBALmH3XNA

KDgN8+G2jX4W/a7LTER10VbRhkGeuc9zyOuj9gigYXLno4Im/S4iXMcCs1IxgSsj

NJ1YMOje4qgHbFKQwWV588VDw7/fiMMZIXvFjHfladdHASEDMT53bKX3HIdJZ/iL

6xhpJ/+C/I8dnWcMZUkeP+9BUAni/I2xrHaAVlli0aS6uc/DjO7b4Gj1Vl4FGIHo

DlH+LmWz26P2gg2xnpWgIxXzs5sN8nYErwu+6h/9xREHco8PPCAZb5HZhqoIzYzk

N1S1Do6qAzt/wJM0mhWOWHt9fhp/RoYQ5ZFClZmgd1cJcr6S6U7ebAQ+yYRsIWU5

+FLYZ4Zlt3ZAHNWyraMee/kFsaGcO21cwE+tPDOIn41B8XvfaXApQt4+TejZWzoH

V0ojA+9H8V+wCFVMJssViFOzuS6SlEZ/xzslo+B//cfUkq/PnWLJHEy4BJXsj4+F

CvliZ7Lq3B/RcQmBjmTRQ0mxahiMGrrQW4TLjUYgY8IfwKfMfwFwVwUyk5br9Grs

UX7jy7+Xx17Qed4p0jjOC7KutzRIGr6ULSk11qpd5IHeIwzSOaTXk6rAzZYupPH5

KvY65mdRfq0C0cB2bMvk9m9lyeLfZz5+L9XDLlodTdwOeWaKvjFErT8WSEkpHxtG

q13TVgicoxsHC2K+8hpFjpaz69ZCmTzj4/17AgMBAAGjQjBAMB0GA1UdDgQWBBQz

zVr7CUfHAeY2KCb1gXy3jjX3sjAPBgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQE

AwIBBjANBgkqhkiG9w0BAQsFAAOCAgEAR9xtbaNa/jSAAyqe3aq88GG7rCyxROGH

BPcakfMmhx1cLYdcY5ATXL/n67eo+S+1g7e/sK3fVXav5qWs9oUEhAOgcOACMohu

JlBbMq2Qp8lxdpiRWCcyiY1vGQcHcZ02oey/c06fBZE4iqJdYAhYhsBB5H+idtwJ

s6Lade4wqG58hWCNKBxU+KWDckGGX5CxsfU7gdYgjyKq0ow60qQWi4H8pD+WO1Bn

rvISkAT7vMk2BOz+YlCKZmuq0h3PCkK5T6xA01fUZCaeze0RozFaekDBEHK0bc1D

My3SKbB3cjdcMzmV8sVdxnNOTxlrP7+BinctxT3q3Va96kTmwI5pD0x6KOwC7Urr

53ubhI3U2XBAzkk14lDLU+7tqBqhDWwIMN0NyW1MRTF8JB9Rz+4yCcDWMOT/FZg7

C60RrcnaO/0GETDz6XI6zedBXo1Q/rJTtXMOr8iVnc+joZyO2ImOuTwP3C7M3Bnp

gFHqDtD48n9PV9prhbD4fYPyMe/3rshtBcpGAy2cGjpsP28pkvP8lwBaP8pnpxvQ

7d3oiCBzznaOHjhm8+8C53b/1txzj/LP/4ZzIynsOhxy4cihEPhAg1MKUY9qnbw9

9Q6EKrCSqk3TPqiWrTtu4pxyiEiquCHk8n+HX5cVhxUkaEShdx4bjgvKB7JRF2T2

ST1lrKEM2DY=

-----END CERTIFICATE-----

OK

//list certificate files

AT+CCERTLIST

+CCERTLIST: "中华.pem"

+CCERTLIST: "client_key.der"

OK