SIM7500_SIM7600_SIM7800 Series_SSL_AT Command Manual_V1.00

LTE Module
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
This document contains proprietary technical information which is the property of SIMCom Limited., copying of this document and giving it to others and the using or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights reserved in the event of grant of a patent or the registration of a utility model or design. All specification supplied herein are subject to change without notice at any time.

Copyright © Shanghai SIMCom Wireless Solutions Ltd. 2018
## Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Chapter</th>
<th>What is new</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.00</td>
<td>2018-09-28</td>
<td></td>
<td>New version</td>
</tr>
</tbody>
</table>
Contents

Version History ................................................................................................................................................ 2

Contents ............................................................................................................................................................ 3

1 Introduction .............................................................................................................................................. 4
1.1 The SSL Context Management AT Commands ....................................................................................... 4
1.2 The process of Using SSL AT Commands .............................................................................................. 4

2 Description of AT Command ..................................................................................................................... 5
2.1 SSL Context Management AT .................................................................................................................. 5
2.1.1 AT+CSSLCFG  Configure the SSL Context ................................................................................... 5
2.1.2 AT+CCERTDOWN  Download certificate into the module ........................................................... 9
2.1.3 AT+CCERTLIST  List certificates .............................................................................................. 9
2.1.4 AT+CCERTDELE  Delete certificates ....................................................................................... 10
2.2 SSL Services AT ........................................................................................................................................ 10
2.2.1 AT+CCHSET  Configure the report mode of sending and receiving data .................................... 10
2.2.2 AT+CCHMODE  Configure the mode of sending and receiving data ...........................................11
2.2.3 AT+CCHSTART  Start SSL service .............................................................................................. 12
2.2.4 AT+CCHSTOP  Stop SSL service ................................................................................................ 13
2.2.5 AT+CCHADDR  Get the IPv4 address ......................................................................................... 13
2.2.6 AT+CCHSSLCFG  Set the SSL context ....................................................................................... 13
2.2.7 AT+CCHOPEN  Connect to server ............................................................................................... 14
2.2.8 AT+CCHCLOSE  Disconnect from server .................................................................................. 15
2.2.9 AT+CCHSEND  Send data to server .......................................................................................... 16
2.2.10 AT+CCHRECV  Read the cached data that received from the server ........................................ 17
2.3 Command result codes and unsolicited codes ........................................................................................... 18
2.3.1 Command result <err> codes .......................................................................................................... 18
2.3.2 Unsolicited result codes .................................................................................................................. 19

3 Example ....................................................................................................................................................... 19
3.1 Access to TCP server ................................................................................................................................. 20
3.2 Access to SSL/TLS server (not verify server and client) ........................................................................... 22
3.3 Access to SSL/TLS server (only verify the server) ................................................................................... 24
3.4 Access to SSL/TLS server (verify server and client) ................................................................................. 26
3.5 Access to SSL/TLS server (only verify the client) .................................................................................... 28
3.6 Access to SSL/TLS server in transparent mode ......................................................................................... 31
3.7 Download certificate into module........................................................................................................... 32

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 The SSL Context Management AT Commands

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+CCERTDELETE.

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

<table>
<thead>
<tr>
<th>Test Command</th>
<th>AT+CSSLCFG=?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>+CSSLCFG: &quot;sslversion&quot;,(0-9),(0-4)</td>
</tr>
<tr>
<td></td>
<td>+CSSLCFG: &quot;authmode&quot;,(0-9),(0-3)</td>
</tr>
<tr>
<td></td>
<td>+CSSLCFG: &quot;ignorelocaltime&quot;,(0-9),(0,1)</td>
</tr>
<tr>
<td></td>
<td>+CSSLCFG: &quot;negotiatetime&quot;,(0-9),(10-300)</td>
</tr>
<tr>
<td></td>
<td>+CSSLCFG: &quot;cacert&quot;,(0-9),(5-128)</td>
</tr>
<tr>
<td></td>
<td>+CSSLCFG: &quot;clientcert&quot;,(0-9),(5-128)</td>
</tr>
<tr>
<td></td>
<td>+CSSLCFG: &quot;clientkey&quot;,(0-9),(5-128)</td>
</tr>
<tr>
<td>OK</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Read Command</th>
<th>AT+CSSLCFG?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>+CSSLCFG: 0,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt;ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;</td>
</tr>
<tr>
<td></td>
<td>+CSSLCFG: 1,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt;ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;</td>
</tr>
<tr>
<td></td>
<td>+CSSLCFG: 2,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt;ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;</td>
</tr>
<tr>
<td></td>
<td>+CSSLCFG: 3,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt;ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;</td>
</tr>
<tr>
<td></td>
<td>+CSSLCFG: 4,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt;ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;</td>
</tr>
<tr>
<td></td>
<td>+CSSLCFG: 5,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt;ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;</td>
</tr>
<tr>
<td>Write Command</td>
<td>Response</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>/<em>Query the configuration of the specified SSL context</em>/</td>
<td>+CSSLCFG:&lt;ssl_ctx_index&gt;,&lt;sslversion&gt;,&lt;authmode&gt;,&lt;ignoreltime&gt;,&lt;negotiatetime&gt;,&lt;ca_file&gt;,&lt;clientcert_file&gt;,&lt;clientkey_file&gt;</td>
</tr>
<tr>
<td>AT+CSSLCFG=&lt;ssl_ctx_index&gt;</td>
<td>OK</td>
</tr>
<tr>
<td>Write Command</td>
<td></td>
</tr>
<tr>
<td>/<em>Configure the version of the specified SSL context</em>/</td>
<td></td>
</tr>
<tr>
<td>AT+CSSLCFG=&quot;sslversion&quot;,&lt;ssl_ctx_index&gt;,&lt;sslversion&gt;</td>
<td>OK</td>
</tr>
<tr>
<td>Write Command</td>
<td></td>
</tr>
<tr>
<td>/<em>Configure the authentication mode of the specified SSL context</em>/</td>
<td></td>
</tr>
<tr>
<td>AT+CSSLCFG=&quot;authmode&quot;,&lt;ssl_ctx_index&gt;,&lt;authmode&gt;</td>
<td>OK</td>
</tr>
<tr>
<td>Write Command</td>
<td></td>
</tr>
<tr>
<td>/<em>Configure the ignore local time flag of the specified SSL context</em>/</td>
<td></td>
</tr>
<tr>
<td>AT+CSSLCFG=&quot;ignorelocaltime&quot;,&lt;ssl_ctx_index&gt;,&lt;ignoreltime&gt;</td>
<td>OK</td>
</tr>
<tr>
<td>Write Command</td>
<td></td>
</tr>
<tr>
<td>/<em>Configure the negotiate timeout value of the specified SSL context</em>/</td>
<td></td>
</tr>
<tr>
<td>AT+CSSLCFG=&quot;negotiatetime&quot;,&lt;ssl_ctx_index&gt;</td>
<td>OK</td>
</tr>
</tbody>
</table>
### _index>,<negotiatetime> 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**

### Defined Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ssl_ctx_index&gt;</td>
<td>The SSL context ID. The range is 0-9.</td>
</tr>
<tr>
<td>&lt;sslversion&gt;</td>
<td>The SSL version. The default value is 4.</td>
</tr>
<tr>
<td></td>
<td>0 – SSL3.0</td>
</tr>
<tr>
<td></td>
<td>1 – TLS1.0</td>
</tr>
<tr>
<td></td>
<td>2 – TLS1.1</td>
</tr>
<tr>
<td></td>
<td>3 – TLS1.2</td>
</tr>
<tr>
<td></td>
<td>4 – All</td>
</tr>
<tr>
<td>&lt;authmode&gt;</td>
<td>The authentication mode, the default value is 0.</td>
</tr>
<tr>
<td></td>
<td>0 – no authentication.</td>
</tr>
<tr>
<td></td>
<td>1 – server authentication. It needs the root CA of the server.</td>
</tr>
<tr>
<td></td>
<td>2 – server and client authentication. It needs the root CA of the server,</td>
</tr>
<tr>
<td></td>
<td>the cert and key of the client.</td>
</tr>
<tr>
<td></td>
<td>3 – client authentication and no server authentication. It needs the</td>
</tr>
<tr>
<td></td>
<td>cert and key of the client.</td>
</tr>
<tr>
<td>&lt;ignoreltime&gt;</td>
<td>The flag to indicate how to deal with expired certificate, the default</td>
</tr>
<tr>
<td></td>
<td>value is 1.</td>
</tr>
<tr>
<td></td>
<td>0 – care about time check for certification.</td>
</tr>
<tr>
<td></td>
<td>1 – ignore time check for certification</td>
</tr>
</tbody>
</table>

The configured version should be support by server. So you should use the default value if you can’t confirm the version which the server supported.

When set the value to 0, it need to set the right current date and time by AT+CCLK when need SSL certification.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>&lt;negotiatetime&gt;</strong></td>
<td>The timeout value used in SSL negotiate stage. The range is 10-300 seconds. The default value is 300.</td>
</tr>
<tr>
<td><strong>&lt;ca_file&gt;</strong></td>
<td>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 128 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</td>
</tr>
<tr>
<td><strong>&lt;clientcert_file&gt;</strong></td>
<td>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 128 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</td>
</tr>
<tr>
<td><strong>&lt;clientkey_file&gt;</strong></td>
<td>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 128 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</td>
</tr>
</tbody>
</table>
### 2.1.2 AT+CCERTDOWN  Download certificate into the module

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AT+CCERTDOWN</strong></td>
<td>Download certificate into the module</td>
</tr>
</tbody>
</table>

**Test Command**

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CCERTDOWN=?.</td>
<td>+CCERTDOWN: (5-128),(1-10240) OK</td>
</tr>
</tbody>
</table>

**Write Command**

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CCERTDOWN=&lt;filename&gt;,&lt;len&gt;</td>
<td>a) If it can be download: &gt; &lt;input data here&gt; b) If failed: ERROR</td>
</tr>
</tbody>
</table>

#### 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 128 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 (\x4E2D;\x534E; pem), then input the hexadecimal (26237834452443B26237835334453B2E70656D) of UTF8 coding. |
| **<len>** | The length of the file data to send. The range is from 1 to 10240 bytes. |

### 2.1.3 AT+CCERTLIST  List certificates

<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AT+CCERTLIST</strong></td>
<td>[+CCERTLIST:&lt;filename&gt; [+CCERTLIST:&lt;filename&gt;] ...]</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>AT+CCERTDELE Delete certificate from the module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write Command</td>
</tr>
<tr>
<td>AT+CCERTDELE=&lt;filename&gt;</td>
</tr>
<tr>
<td>Response</td>
</tr>
<tr>
<td>a) If delete successfully:</td>
</tr>
<tr>
<td>OK</td>
</tr>
<tr>
<td>b) If failed:</td>
</tr>
<tr>
<td>ERROR</td>
</tr>
</tbody>
</table>

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 128 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 (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

<table>
<thead>
<tr>
<th>Test Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CCHSET=?</td>
<td>+CCHSET: (0,1),(0,1)</td>
</tr>
<tr>
<td></td>
<td>OK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Read Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CCHSET?</td>
<td>+CCHSET: &lt;report_send_result&gt;,&lt;recv_mode&gt;</td>
</tr>
<tr>
<td></td>
<td>OK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Write Command</th>
<th>Response</th>
</tr>
</thead>
</table>
| AT+CCHSET=<report_send_result>[,<recv_mode>] | 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 elect 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.
### Write Command

**AT+CCHMODE=<mode>**

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OK</strong></td>
<td>a) If successfully:</td>
</tr>
<tr>
<td></td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>b) If failed:</td>
</tr>
<tr>
<td></td>
<td>ERROR</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Response</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OK</strong></td>
<td>a) If start SSL service successfully:</td>
</tr>
<tr>
<td></td>
<td>+CCHSTART: 0</td>
</tr>
<tr>
<td></td>
<td>b) If start SSL service successfully:</td>
</tr>
<tr>
<td></td>
<td>+CCHSTART: 0</td>
</tr>
<tr>
<td></td>
<td>c) If failed:</td>
</tr>
<tr>
<td></td>
<td>ERROR</td>
</tr>
<tr>
<td></td>
<td>d) If failed:</td>
</tr>
<tr>
<td></td>
<td>ERROR</td>
</tr>
</tbody>
</table>

| 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.

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CCHSTOP</td>
<td></td>
</tr>
</tbody>
</table>

a) If stop SSL service successfully:
+CCHSTOP: 0
OK
b) If stop SSL service successfully:
OK
+c) 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.

<table>
<thead>
<tr>
<th>Execute Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CCHADDR</td>
<td></td>
</tr>
</tbody>
</table>

+CCHADDR: < ip_address>
OK

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 the CCHOPEN operation is finished.
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.

<table>
<thead>
<tr>
<th>AT+CCHSSLCFG Set the SSL context</th>
<th>Test Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CCHSSLCFG=?</td>
<td></td>
<td>+CCHSSLCFG: (0,1),(0-9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AT+CCHSSLCFG?</th>
<th>Read Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+CCHSSLCFG: &lt;session_id&gt;,&lt;ssl_ctx_index&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+CCHSSLCFG: &lt;session_id&gt;,&lt;ssl_ctx_index&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OK</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AT+CCHSSLCFG=&lt;session_id&gt;,&lt;ssl_ctx_index&gt;</th>
<th>Write Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a) If successfully:</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>b) If failed:</td>
<td>ERROR</td>
</tr>
</tbody>
</table>

**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+CCHOPEN Connect to server

AT+CCHOPEN is used to connect to 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 1st parameter of AT+CCHOPEN) SSL context when connecting to the server.

<table>
<thead>
<tr>
<th>AT+CCHOPEN Connect to server</th>
<th>Test Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CCHOPEN=?</td>
<td></td>
<td>+CCHOPEN: (0,1),”ADDRESS”, (1-65535),(1-2),(1-65535)]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AT+CCHOPEN=&lt;session_id&gt;, &quot;&lt;host&gt;&quot;, &lt;port&gt;[&lt;client_type&gt;,&lt;bind_port&gt;]</th>
<th>Write Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a) If connect successfully:</td>
<td>+CCHOPEN: &lt;session_id&gt;,0</td>
</tr>
</tbody>
</table>

**AT+CCHOPEN SIM7500_SIM7600_SIM7800 Series_SSL_AT Command Manual** 14 / 36
b) If connect successfully:
OK

+CCHOPEN: <session_id>,0

c) If connect successfully in transparent mode:
CONNECT [<text>]

d) If failed:
OK

+CCHOPEN: <session_id>,<err>

e) If failed:
ERROR

f) If failed in transparent mode:
CONNECT FAIL

Defined Values

<table>
<thead>
<tr>
<th>&lt;session_id&gt;</th>
<th>The session index to operate. It’s from 0 to 1.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;host&gt;</td>
<td>The server address, maximum length is 256 bytes.</td>
</tr>
<tr>
<td>&lt;port&gt;</td>
<td>The server port which to be connected, the range is from 1 to 65535.</td>
</tr>
</tbody>
</table>
| <client_type>| The type of client:  
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 |

2.2.8 AT+CCHCLOSE  Disconnect from server

AT+CCHCLOSE is used to disconnect from the server.

AT+CCHCLOSE  Disconnect from the Server

<table>
<thead>
<tr>
<th>Write Command</th>
<th>Response</th>
</tr>
</thead>
</table>
| AT+CCHCLOSE=<session_id> | a) If successfully:  
+CCHCLOSE: <session_id>,0  
OK |
b) If successfully:

OK

+CCHCLOSE: <session_id>,0

c) If successfully in transparent mode:

OK

CLOSED

d) If failed:

ERROR

Defined Values

<table>
<thead>
<tr>
<th>&lt;session_id&gt;</th>
<th>The session index to operate. It’s from 0 to 1.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;err&gt;</td>
<td>The result code: 0 is success. Other values are failure. Please refer to chapter 2.3.1.</td>
</tr>
</tbody>
</table>

2.2.9 AT+CCHSEND   Send data to server

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

<table>
<thead>
<tr>
<th>AT+CCHSEND</th>
<th>Send Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Command</td>
<td></td>
</tr>
<tr>
<td>AT+CCHSEND=?</td>
<td></td>
</tr>
<tr>
<td>Response:</td>
<td>+CCHSEND: (0,1),(1-2048)</td>
</tr>
<tr>
<td></td>
<td>OK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Read Command</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CCHSEND?</td>
<td></td>
</tr>
<tr>
<td>Response:</td>
<td>+CCHSEND: 0,&lt;unsent_len_0&gt;,1,&lt;unsent_len_1&gt;</td>
</tr>
<tr>
<td></td>
<td>OK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Write Command</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CCHSEND=&lt;session_id&gt;,&lt;len&gt;</td>
<td></td>
</tr>
<tr>
<td>Response</td>
<td>a) if parameter is right:</td>
</tr>
<tr>
<td></td>
<td>&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;input data here&gt;</td>
</tr>
<tr>
<td></td>
<td>When the total size of the inputted data reaches &lt;len&gt;, TA will report the following code.</td>
</tr>
</tbody>
</table>
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. |

2.2.10 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.

AT+CCHRECV  Receive 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

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

<table>
<thead>
<tr>
<th>Result codes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Operation succeeded</td>
</tr>
<tr>
<td>1</td>
<td>Alerting state (reserved)</td>
</tr>
<tr>
<td>2</td>
<td>Unknown error</td>
</tr>
<tr>
<td>3</td>
<td>Busy</td>
</tr>
<tr>
<td>4</td>
<td>Peer closed</td>
</tr>
<tr>
<td>5</td>
<td>Operation timeout</td>
</tr>
<tr>
<td>6</td>
<td>Transfer failed</td>
</tr>
<tr>
<td>7</td>
<td>Memory error</td>
</tr>
<tr>
<td>8</td>
<td>Invalid parameter</td>
</tr>
<tr>
<td>9</td>
<td>Network error</td>
</tr>
<tr>
<td>10</td>
<td>Open session error</td>
</tr>
<tr>
<td>11</td>
<td>State error</td>
</tr>
<tr>
<td>12</td>
<td>Create socket error</td>
</tr>
<tr>
<td>13</td>
<td>Get DNS error</td>
</tr>
<tr>
<td>14</td>
<td>Connect socket error</td>
</tr>
<tr>
<td>15</td>
<td>Handshake error</td>
</tr>
<tr>
<td>16</td>
<td>Close socket error</td>
</tr>
<tr>
<td>17</td>
<td>Nonet</td>
</tr>
<tr>
<td>18</td>
<td>Send data timeout</td>
</tr>
</tbody>
</table>
2.3.2 Unsolicited result codes

### Unsolicited codes

<table>
<thead>
<tr>
<th>Code Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+CCHEVENT: &lt;session_id&gt;,RECV EVENT</td>
<td>In manual receiving mode, when new data of a connection arriving to the module, this unsolicited result code will be reported to MCU.</td>
</tr>
<tr>
<td>+CCH_RECV_CLOSED: &lt;session_id&gt;,&lt;err&gt;</td>
<td>When receive data occurred any error, this unsolicited result code will be reported to MCU.</td>
</tr>
<tr>
<td>+CCH_PEER_CLOSED: &lt;session_id&gt;</td>
<td>The connection is closed by the server.</td>
</tr>
</tbody>
</table>

### 3 Example

Before all SSL related operations, we should ensure the following:

a) ensure GPRS network is available:

```plaintext
AT+CSQ
+CSQ: 23,0
OK
AT+CREG?
+CREG: 0,1
OK
AT+CGREG?
+CGREG: 0,1
OK
```

b) PDP context Enable:
// Specify the parameter value of the PDP context corresponding to cid
AT+CGSOCKCONT=1,"IP","CMNET"
OK
AT+CGPADDR
+CGPADDR: 1,10.49.14.68 //ensure the first PDP context get a IP address
+CGPADDR: 4,0.0.0.0
OK
Note: usually CSOCKAUTH and CSOCKSETPN parameter are kept default if not care about.

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">
<center><h1>302 Found</h1></center>
<hr><center>7a367f7b87705e16b985e34ca59b8ae8b1d28d47
</body>
</html>

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

+CCHCLOSE: 0
//stop SSL Service
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.

```c
// 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
```
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
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-c3"
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>
//Disconnect from the Service
AT+CCHCLOSE=0
OK
+CCHCLOSE: 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
//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
//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"
```
//switch to command mode
+++ OK

//Disconnect from the Service
AT+CCHCLOSE=0
OK
CLOS
//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
BSh8VYFCgoL4RnWLwXAcLlaqw88zlCN89EK6IydaAwNmUI/U6nu3oPsVkn8r9+sOX
yh9VD01DmSU349QWJvRgt1ocsFI1VTdd6RDkVtu7FdKv4XC5WHcOD7yrElIsVa7+G
Qbnm5CcZz8E75H8hvHZAOFeav3Hv1Hnh/1RZ+jh4ysyhEmFNOFCn3r9v2yuy4kPRX
43xEsB13Ue4HgSbnT+Q7LIEK+dsfUM0oSpss2NAamQOiQgrmmYygT3/V/IXS54hit
gli5bv9DuNYBwv2C+4nyZF95pMj2dEJf4jNw1DAQABaoIBAAJ9ze06QKDo79p4
3NfJfhJhck/NTYB0Xssl/+iDhgWt4VogCD6kzGGxsmU2tdOrsq9xIvXcthpeu5IQ
98mmpBhaWNc96Jx1Oh9O+0q1xNAh8AiiH2QZGjUTac8Jfx+B6w+fbkz37os1/+00
6ZajkbChFTpf7r7An5wUEOKQZ4vNpLjxlWDdk6uH4ZMNvWcBaZQ21TuG9ZmoskK
EJ2Zer/3kOSBgi2B6F50yZ1fmbpqPahHNLqrtrdV5/Lr4n74TqZXWrt5C9GrBv
tYXDHc/+5Y7e1TUIXV00AMDIIk+3cVR8m8Oa20tSdXjcw2iU9kbrxb4uxreOouGfPW
5l0+q1ECgYEA4Kkok17DVx5FiapFQvJ2qj2/WhzDncuBGBztcLZnwRfKp3n3BZ
JGNwxYyfEdwt1PtPVTYQYh6Qg81RXRdSRIF43GzqKXNmkP0dZM0x3tfWzV6K5Fg7aeR
g50Udda9MraCltOgK++7c6BvA31mXciK4VWe5ZQomDW99Y6mgf92RdkGcYEArB2u
/ld72LQBMx0Z+36Hflxd0x6Rq+db+m6XBM8riuB/jG0/5PHdFoK0F2qa9yJ2w1+W
B29Xmc1HS6GTVkDlsN5XNO7DmlAxD5whbwDdcmv3VEt8xj2UeACIawjKtVcFoH
LRNlVBtBttWvICzg+9HfVPumP14oFxn/HStx48CgYACxJD6thUDspy6mD0oGO15
kaRHN10JYUmfOzo+EVDyvwLqfh2RzneKiu ruU8/10vB+G4e7zx6FxxMwsbEgYEmQ
hmrm0Kn3qPhMMHanvr572Oku7KM2p5hF4MT/GRM0lHdU31D1JrTcJap1TVomAaCL
FqY88arQFwFSz8Hfl0e6QKBgChQltTdzKzqJdt8+6cwQFYGg+9O59MJGVVefNskp
chzVFAX0n9T15Lq9fMJ5FX4g+3JGargjfWuGCTTFBk0TMt4wde7Amwiiu5LU
T2Af6pLTKrSE9k+yX2iug+O156VsfbeAm/NG5RCJ91NCvFgULro6/axNmmWOrf
9rK7AoGBAIK4edrX1MjerCSLu3y9Dy4pAxER6eie4xpkO25U8wUccqc+YDm2x1A
DjqROITeaxXkmPlyRKAXVarhk8LmXT/oDFUApsTqUZLBrviqtMi+G2OFpbdKDwe
ZBNAgwFpFlUVo0UYnZFr8Bq0tepqvrayEWdKKfMMJjq+172Sxd
-----END RSA PRIVATE KEY-----
OK

// download file with not ASCII coding file name

AT+CCERTDOWN={non-ascii}"262378344532443B2623783533453B2E70656D",1918

>-----BEGIN CERTIFICATE-----
MIIFRDCCAyygAwIBAgIIZmPau7FeIqswDQYJbKoZIhvcNAQELBQAwQDELMAkGA1UE
BhMCU0kxGzAZBgNVMEQwNXRJLWluc3RpdHw0aW9uczEUMBIGA1UEAwwLVGF4IENBIFRc
3QwHhcNMTUwNzIzMTUyOTA1WhecNMzUwNzIzMTUyOTA1WjBAMQswCQYD
VQQGeWJTSTebMBkGA1UECgwSc3RhGUTawWzdGl0dXRpb25zMRQwEgYDVQQD
YXggQ0EgVGzdDCAiIwDQYJbKoZIhvcNAQEBBQADggIBADCCAgggHPADCAgoCggIBALmH3
KQgN8+G2jX4w/a7LTER10VbRkGeue9zyOuj9gigYXLo4lM/S4iXMcs11xgSs
NJ1YMOje4qgHbFKqwWV588VDw7/iIMMZIXvFjHfladdHASEDMT53bKX3HidJZ/iL
6xhpJ/+C/I8dnWzMCZUkeP+9BUAni/I2xrHaAVFl0s6u6c/DjO77b4Gj1V1fFGIHo
DIH+LmWz26P2gg2xnWgIxZxs5sNnYerw+6h/9xREHco8PPCAZb5HHzqoIzYzk
NIS1Do6qAz/t/WM0mWOWH0fhp/ReYQ5ZFCI7md1cJcro6S67ebAQ+yrRsIWU5
+FLYZ4Zht3ZAHNWYraMee/kFsaGe021cwe+tPDOIn41B8XvaXApQt4+TecjZWoH
V0ojA+9H8V+wCFVMJssVF0zuS6SIEZ/xzls/+B//cfUkq/PnWLJHEy4Bjxsj4+F
CviZ7Lq3/Bp/RQmBjTRQ0mxahiMGrrQW4TLjYugY8lfWkFmfWfWVwUyk5br9Gr
UX7jy7+Xx17Qed4p0jOC7KutzRIR6UL5k11qd5IHelwzSOaTXk6rAzZYupPH5
KvY65mdRfq0C0b2Bmvk9m9yefLzZ5+l9XDLIdodTdwOeWaKvJferT8WSEkpHxtG
q13TVgicoxsHC2K+8hpFjpa69ZCmTzj4/17AgMBAAGjQjBAMB0GA1UDgUQWBBQz
zVr7CUfHaesY2KCh1gxy3jxIsjAPBgvNHRRMBAf8EBTADAQH/MA4GA1UDgUWmE/wQ
AwIBBjANBgkgkqhiG9w0BAQsFAAOCAgEAR9xtbaNa/jSAAyqe3aq88GG7rCyxROG
BPcakFmnhx1cLydC5ATXl/n67eo+S+1g7e/sK3fVXav5qW5s9oUEhAOgeOACMo
JlBbMqZ8p8IdprWRCCyiV1gqQcHz02oey/06fBZEx4iqJdYAhYhsBB5H+idtwJ
s6Lade4wqG58hWCNKBxU+KWDekCGG5CsfU7gdYqjyKq0ow60qWWi4H8pD+WO1Bn
rviSkAT7vMk2BOZ+yI1CKZmuq0h3PCkK5T6x0A1fUZCaeze0RozFaekDBEK0be1D
My3SKbB3cjdMzmV8sVdxnNOTxrlP7+BinctxT3q3Va96kTmw15p0x6KOwC7Urr
-----END CERTIFICATE-----

OK

//list certificate files

AT+CCERTLIST

+CCERTLIST: "&x4E2D;&x534E;.pem"

+CCERTLIST: "client_key.der"

OK