SIM7672X & SIM7652X Series_SSL_Application Note

LTE Module

SIMCom Wireless Solutions Limited
SIMCom Headquarters Building, Building 3, No. 289 Linhong Road, Changning District, Shanghai P.R. China
Tel: 86-21-31575100
support@simcom.com
www.simcom.com

Document Title:	SIM7672X & SIM7652X Series_SSL_Application Note
Version:	1.00
Date:	2023.05.22
Status:	Released

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

SIMCom Wireless Solutions Limited
SIMCom Headquarters Building, Building 3, No. 289 Linhong Road, Changning District, Shanghai P.R. China
Tel: +86 21 31575100
Email: simcom@simcom.com

For more information, please visit:
https://www.simcom.com/technical_files.html

For technical support, or to report documentation errors, please visit:
https://www.simcom.com/online_questions.html or email to: support@simcom.com

About Document

Version History

Version	Date	Owner	Description
V1.00	2023.05.22		New version

Scope

Based on module AT command manual, this document will introduce SSL application process. Developers could understand and develop application quickly and efficiently based on this document. This document applies to SIM7672X Series, SIM7652X Series.

Contents

1 Introduction

1.1 Purpose of the document

Based on module AT command manual, this document will introduce SSL application process. Developers could understand and develop application quickly and efficiently based on this document.

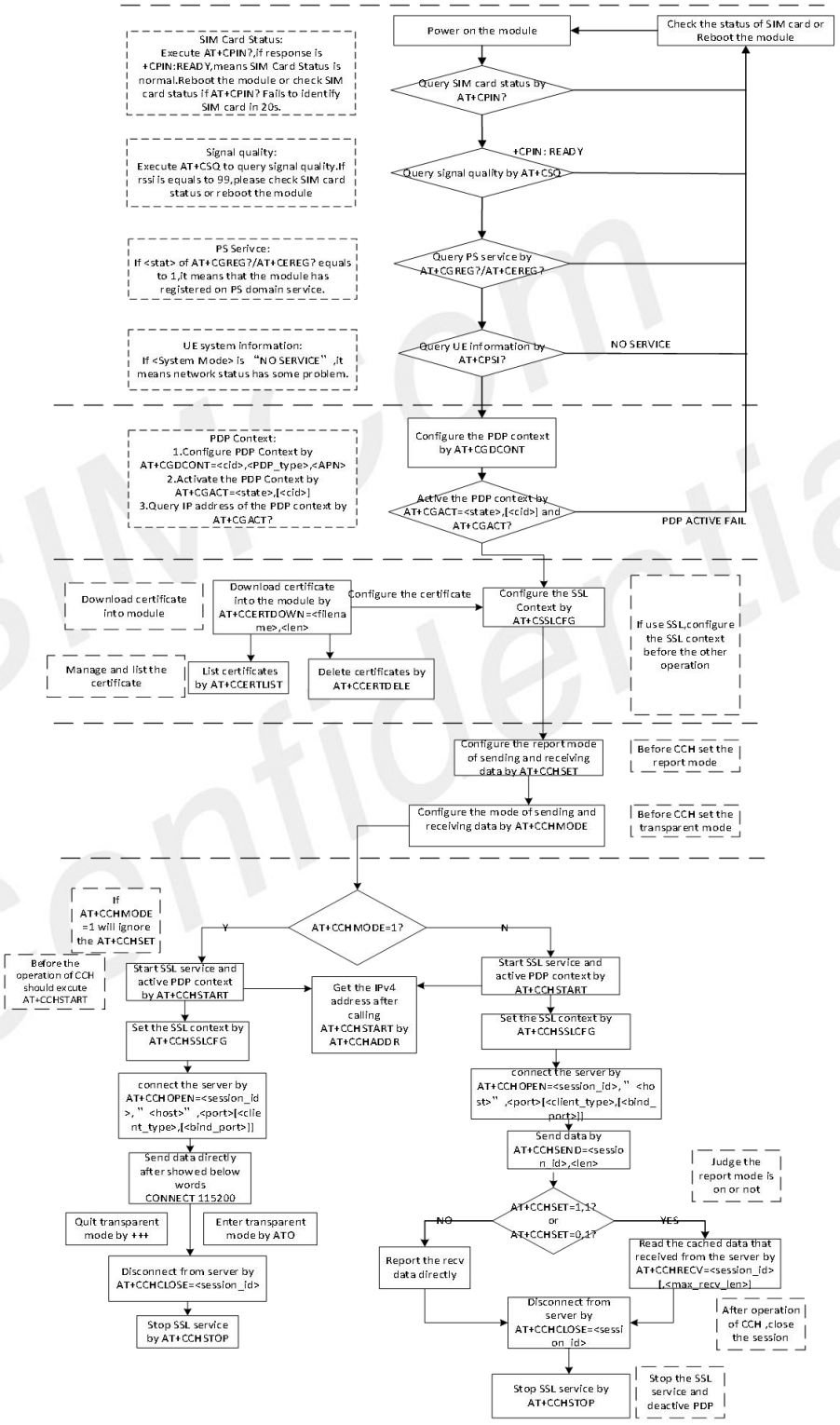
1.2 Related documents

[1] SIM7672X & SIM7652X Series_AT Command Manual.

1.3 Conventions and abbreviations

PDP Packet Data Protocol;
SSL Security Socket Layer;
URC Unsolicited result codes;
DNS Domain Name Server;

1.4 The process of Using SSL AT Commands



1.5 Error Handling

1.5.1 Executing SSL AT Commands Fails

If it is failed to open SSL connection, please check the following aspects:
1. Query the status of the specified PDP context by ***AT+CGACT?*** command to check whether the specified PDP context has been activated.
2. Please check the SSL configuration by ***AT+CSSLCFG?*** command, especially the SSL version and cipher suite.
3. When the CCHXXX: <err> is not 0, it indicates an error code replied from CCH server.
For more details, please refer to SIM7672X & SIM7652X Series_AT Command Manual.

2 AT Commands for SSL

2.1 Overview of AT Commands for SSL

Command	Description
AT+CSSLCFG	Configure the SSL Context
AT+CCERTDOWN	Download certificate into the module
AT+CCERTLIST	List certificates
AT+CCERTDELE	Delete certificates
AT+CCHSET	Configure the report mode of sending and receiving data
AT+CCHMODE	Configure the mode of sending and receiving data
AT+CCHSTART	Start SSL service
AT+CCHSTOP	Stop SSL service
AT+CCHADDR	Get the IPv4 address
AT+CCHSSLCFG	Set the SSL context
AT+CCHCFG	Configure the Client Context
AT+CCHOPEN	Connect to server
AT+CCHCLOSE	Disconnect from server
AT+CCHSEND	Send data to server
AT+CCHRECV	Read the cached data that received from the server
AT+CCERTMOVE	Move the cert from file system to cert content

For more detailed information, please refer to SIM7672X & SIM7652X Series_AT Command Manual.

2.2 Detailed Description of AT Commands for SSL

2.2.1 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: 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",<**	Response 1)If successfully: **OK** 2)If failed:

ssl_ctx_index>,<sslversion>	**ERROR**
Write Command /*Configure the authentication mode of the specified SSL context*/ **AT+CSSLCFG="authmode",<ssl_ctx_index>,<authmode>**	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Write Command /*Configure the ignore local time flag of the specified SSL context*/ **AT+CSSLCFG="ignorelocaltime",<ssl_ctx_index>,<ignoreltime>**	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Write Command /*Configure the negotiate timeout value of the specified SSL context*/ **AT+CSSLCFG="negotiatetime",<ssl_ctx_index>,<negotiatetime>**	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Write Command /*Configure the server root CA of the specified SSL context*/ **AT+CSSLCFG="cacert",<ssl_ctx_index>,<ca_file>**	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Write Command /*Configure the client certificate of the specified SSL context*/ **AT+CSSLCFG="clientcert",<ssl_ctx_index>,<clientcert_file>**	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Write Command /*Configure the client key of the specified SSL context*/ **AT+CSSLCFG="clientkey",<ssl_ctx_index>,<clientkey_file>**	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Write Command /*Configure the enableSNI flag of the specified SSL context */ **AT+CSSLCFG="enableSNI",<ssl_ctx_index>,<enableSNI_flag>**	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms

Reference	-

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. (If the server does not need to authenticate the client, it is equivalent to value 1.) 3 client authentication and no server authentication. It needs the cert and key of the client. ((If the server does not need to authenticate the client, it is equivalent to value 0.)
<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 SIM76XX: The length of filename is from 5 to 55 bytes. There are two ways to download certificate files to module: 1. By ***AT+CCERTDOWN.*** 2. By FTPS or HTTPS commands. Please refer to Chapter 4.1.1 of this document.
<clientcert_file>	The client cert file name of SSL context. The file name must have type like ".pem" or ".der". The SIM76XX: The length of filename is from 5 to 55 bytes.

	There are two ways to download certificate files to module: 1. By ***AT+CCERTDOWN***. 2. By FTPS or HTTPS commands. Please refer to Chapter 4.1.1 of this document.
<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 55 bytes. There are two ways to download certificate files to module: 1. By ***AT+CCERTDOWN***. 2. By FTPS or HTTPS commands. Please refer to Chapter 4.1.1 of this document.
<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.

Examples

```
AT+CSSLCFG=?
+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
AT+CSSLCFG?
+CSSLCFG: 0,4,0,1,300,"","","",0
+CSSLCFG: 1,4,0,1,300,"","","",0
+CSSLCFG: 2,4,0,1,300,"","","",0
+CSSLCFG: 3,4,0,1,300,"","","",0
+CSSLCFG: 4,4,0,1,300,"","","",0
+CSSLCFG: 5,4,0,1,300,"","","",0
+CSSLCFG: 6,4,0,1,300,"","","",0
+CSSLCFG: 7,4,0,1,300,"","","",0
+CSSLCFG: 8,4,0,1,300,"","","",0
+CSSLCFG: 9,4,0,1,300,"","","",0

OK
AT+CSSLCFG="authmode",0,0
```

```
OK
AT+CSSLCFG=6
+CSSLCFG: 6,4,0,1,300,"","","",0

OK
```

2.2.2 AT+CCERTDOWN Download certificate into the module

AT+CCERTDOWN Download certificate into the module	
Test Command **AT+CCERTDOWN=?**	Response **+CCERTDOWN: (5-55),(1-10240)** **OK**
Write Command **AT+CCERTDOWN=<filename>,<len>**	Response 1)If it can be download: **>** **<input data here>** **OK** 2)If failed: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

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 55 bytes.
<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.

Examples

```
AT+CCERTDOWN=?
+CCERTDOWN: (5-108),(1-10240)                    // The SIM76XX response.
```

```
OK
AT+CCERTDOWN="ls.pem",1970
>

OK
```

2.2.3 AT+CCERTLIST List certificates

AT+CCERTLIST List certificates	
Execute Command AT+CCERTLIST	Response **[+CCERTLIST: <file_name>** **[+CCERTLIST: <file_name>]** **…** **]** **OK**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

Defined Values

<filename>	The certificate/key files which has been downloaded to the module.

Examples

```
AT+CCERTLIST
+CCERTLIST: "ls.pem"

OK
```

2.2.4 AT+CCERTDELE Delete certificates

AT+CCERTDELE Delete certificates	
Write Command	Response

AT+CCERTDELE=<filename>	1) If remove the file successfully: **OK** 2) Else **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

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 55 bytes.

Examples

```
AT+CCERTDELE="ls.pem"
OK
```

2.2.5 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 data	
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 1)If successfully: **OK** 2)If failed: **ERROR**
Parameter Saving Mode	-

Max Response Time	120000ms
Reference	-

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, the default value is 0: 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).

Examples

```
AT+CCHSET=?
+CCHSET: (0,1),(0,1)

OK
AT+CCHSET?
+CCHSET: 0,0

OK
AT+CCHSET=1,1
OK
```

2.2.6 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***.

AT+CCHMODE Configure the mode of sending and receiving data	
Test Command **AT+CCHMODE=?**	Response **+CCHMODE: (0,1)**

	OK
Read Command **AT+CCHMODE?**	Response **+CCHMODE: <mode>** **OK**
Write Command **AT+CCHMODE=<mode>**	Response 1)If successfully: **OK** 2)If failed: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

Defined Values

<mode>	The mode value: 0 Normal 1 Transparent mode

Examples

```
AT+CCHMODE=?
+CCHMODE: (0,1)

OK
AT+CCHMODE?
+CCHMODE: 0

OK
AT+CCHMODE=1
OK
```

NOTE

There is only one session in the transparent mode, it's the first session.

2.2.7 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 1)If start SSL service successfully: **OK** **+CCHSTART: 0** 2)If failed: **ERROR** 3)If failed: **ERROR** **+CCHSTART: <err>**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

Defined Values

<err>	The result code, please refer to chapter 4.1.1 of this document.

Examples

```
AT+CCHSTART
OK

+CCHSTART: 0
```

2.2.8 AT+CCHSTOP Stop SSL service

AT+CCHSTOP is used to stop SSL service.

AT+CCHSTOP Stop SSL service

Execute Command **AT+CCHSTOP**	Response 1)If stop SSL service successfully: **OK** **+CCHSTOP: 0** 2)If failed: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

Defined Values

<err>	The result code, please refer to chapter 4.1.1 of this document.

Examples

```
AT+CCHSTOP
OK

+CCHSTOP: 0
```

2.2.9 AT+CCHADDR Get the IPv4 address

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

AT+CCHADDR Get the IPv4 address	
Execute Command **AT+CCHADDR**	Response 1)if successfully, response **+CCHADDR: <ip_address>** **OK** 2)if pdp has not been activated, response **ERROR**
Parameter Saving Mode	-
Max Response Time	12000ms
Reference	-

Defined Values

<ip address>	A string parameter that identifies the IPv4 address after PDP activated.

Examples

```
AT+CCHADDR
+CCHADDR: 10.43.71.130

OK
```

2.2.10 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*.

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 1)If successfully: **OK** 2)If failed: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

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***.

Examples

```
AT+CCHSSLCFG=?
+CCHSSLCFG: (0,1),(0-9)

OK
AT+CCHSSLCFG?
+CCHSSLCFG: 0,
+CCHSSLCFG: 1,

OK
AT+CCHSSLCFG=0,1
OK
```

NOTE

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***

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.

2.2.11 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 **AT+CCHCFG=?**	Response **+CCHCFG: "sendtimeout",(0-1),(60-150)** **+CCHCFG: "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 1)If successfully: **OK** 2)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 1)If successfully: **OK** 2)If failed: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

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***.

Examples

```
AT+CCHCFG=?
+CCHCFG: "sendtimeout",(0-1),(60-150)
+CCHCFG: "sslctx",(0-1),(0-9)

OK
AT+CCHCFG?
+CCHCFG: 0,150,0
+CCHCFG: 1,150,0

OK
```

```
AT+CCHCFG="sendtimeout",0,120
OK
AT+CCHCFG="sslctx",0,3
OK
```

2.2.12 AT+CCHOPEN Connect to server

AT+CCHOPEN is used to connect 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 1)If connect successfully: **OK** **+CCHOPEN: <session_id>,0** 2)If connect successfully in transparent mode: **CONNECT [<text>]** 3)If failed: **OK** **+CCHOPEN: <session_id>,<err>** 4)If failed: **ERROR** 5)If failed in transparent mode: **CONNECT FAIL**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

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 command.
<err>	The result code: 0 is success. Other values are failure. Please refer to chapter 4.1.1 of this document.

Examples

```
AT+CCHOPEN=?
+CCHOPEN: (0,1),"ADDRESS",(1-65535)[,(1-2)[,(1-65535)]]

OK
AT+CCHOPEN=0,"183.230.174.137",6043,1
OK

+CCHOPEN: 0,0
AT+CCHOPEN?
+CCHOPEN: 0,"183.230.174.137",6043,1,
+CCHOPEN: 1,"",,,

OK
```

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+CCHOPEN***) SSL context when connecting to the server.

2.2.13 AT+CCHCLOSE Disconnect from server

AT+CCHCLOSE is used to disconnect from the server.

AT+CCHCLOSE Disconnect from server	
Write Command AT+CCHCLOSE=<session_id>	Response 1)If successfully: **OK** **+CCHCLOSE: <session_id>,0** 2)If successfully in transparent mode: **OK** **CLOSED** 3)If failed: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

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 4.1.1 of this document.

Examples

```
AT+CCHCLOSE=0
OK

+CCHCLOSE: 0,0
```

2.2.14 AT+CCHSEND Send data to server

AT+CCHSEND Send data to server	
Test Command	Response

AT+CCHSEND=?	**+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 1)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** 2)If parameter is wrong or other errors occur: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

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.

Examples

```
AT+CCHSEND=?
+CCHSEND: (0,1),(1-2048)

OK
AT+CCHSEND?
+CCHSEND: 0,0,1,0

OK
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
```

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

AT+CCHRECV Read the cached data that received from the 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 1)if parameter is right and there are cached data: **OK** **[+CCHRECV: DATA,<session_id>,<len>** **…** **+CCHRECV: DATA,<session_id>,<len>** **…]** **+CCHRECV: <session_id>,<err>** 2)if parameter is not right or any other error occurs: **+CCHRECV: <session_id>,<err>** **ERROR** 3)others: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

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. The value ranges from 0 to 2048.

	0 means it will receive all data from the current cache. The default value is 0 and it will receive all 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 4.1.1 of this document.

Examples

```
AT+CCHRECV?
+CCHRECV: LEN,3072,0

OK
AT+CCHRECV=0
OK

+CCHRECV: DATA,0,1024
HTTP/1.1 200 OK
Bdpagetype: 1
Bdqid: 0x9821f6dd000060aa
Cache-Control: private
Connection: keep-alive
Content-Type: text/html;charset=utf-8
Date: Tue, 24 Mar 2020 02:27:10 GMT
Expires: Tue, 24 Mar 2020 02:26:31 GMT
P3p: CP=" OTI DSP COR IVA OUR IND COM "
P3p: CP=" OTI DSP COR IVA OUR IND COM "
Server: BWS/1.1
Set-Cookie: BAIDUID=F0CD980BA0927350B147AB1064A3423D:FG=1; expires=Thu, 31-Dec-37
23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: BIDUPSID=F0CD980BA0927350B147AB1064A3423D; expires=Thu, 31-Dec-37 23:55:55
GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: PSTM=1585016830; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647;
path=/; domain=.baidu.com
Set-Cookie: BAIDUID=F0CD980BA0927350739AA64356C3CB13:FG=1; max-age=31536000;
expires=Wed, 24-Mar-21 02:27:10 GMT; domain=.baidu.com; path=/; version=1; comment=bd
Set-Cookie: BDSVRTM=0; path=/
Set-Cookie: BD_HOME=1; path=/
Set-Cookie: H_PS_PSSID=30972_1467_21116_30823; path=/; domain=.baidu.com
Traceid
```

```
+CCHRECV: DATA,0,1024
: 1585016830040414772210962314397044727978
Vary: Accept-Encoding
Vary: Accept-Encoding
X-Ua-Compatible: IE=Edge,chrome=1
Transfer-Encoding: chunked

b5e
<!DOCTYPE html><!--STATUS OK--><html><head><meta http-equiv="Content-Type"
content="text/html;charset=utf-8"><meta http-equiv="X-UA-Compatible"
content="IE=edge,chrome=1"><meta content="always" name="referrer"><meta
name="theme-color" content="#2932e1"><link rel="shortcut icon" href="/favicon.ico"
type="image/x-icon" /><link rel="search" type="application/opensearchdescription+xml"
href="/content-search.xml" title=" " /><link rel="icon" sizes="any" mask
href="//www.baidu.com/img/baidu_85beaf5496f291521eb75ba38eacbd87.svg"><link
rel="dns-prefetch" href="//dss0.bdstatic.com"/><link rel="dns-prefetch"
href="//dss1.bdstatic.com"/><link rel="dns-prefetch" href="//ss1.bdstatic.com"/><link
rel="dns-prefetch" href="//sp0.baidu.com"/><link rel="dns-prefetch" href="//sp1.baidu.com"/><link
rel="dns-prefetch" href="//sp2.baidu.com"/><title>?
+CCHRECV: DATA,0,1024
</title><style type="text/css" id="css_index"
index="index">body,html{height:100%}html{overflow-y:auto}body{font:12px
arial;background:#fff}body,form,li,p,ul{margin:0;padding:0;list-style:none}#fm,body,form{position:
relative}td{text-align:left}img{border:0}a{text-decoration:none}a:active{color:#f60}input{border:0;p
adding:0}.clearfix:after{content:'\20';display:block;height:0;clear:both}.clearfix{zoom:1}#wrapper{p
osition:relative;min-height:100%}#head{padding-bottom:100px;text-align:center;*z-index:1}#ftCon{
height:50px;position:absolute;text-align:left;width:100%;margin:0
auto;z-index:0;overflow:hidden}#ftConw{display:inline-block;text-align:left;margin-left:33px;line-he
ight:22px;position:relative;top:-2px;*float:right;*margin-left:0;*position:static}#ftConw,#ftConw
a{color:#999}#ftConw{text-align:center;margin-left:0}.bg{background-image:url(http://ss.bdimg.co
m/static/superman/img/icons-5859e577e2.png);background-repeat:no-repeat;_background-image:u
rl(http://ss.bdimg.com/static/superman/img/icon
+CCHRECV: 0,0

+CCHEVENT: 0,RECV EVENT
```

NOTE

If connection is closed by server, the cached data will not be cleaned.

2.2.16 AT+CCERTMOVE Move the cert from file system to cert content

AT+CCERTMOVE Move the cert from file system to cert content	
Test Command **AT+CCERTMOVE=?**	Response **+CCERTMOVE: "FILENAME"** **OK**
Write Command **AT+CCERTMOVE=<filename>**	Response 1)if parameter is right and the file need to move is exist: **OK** 2)if parameter is not right or any other error occurs: **ERROR** 3)others: **ERROR**
Parameter Saving Mode	-
Max Response Time	120000ms
Reference	-

Defined Values

<filename>	The file name must have type like ".pem" or ".der". The length of filename is from 5 to 55 bytes.

Examples

```
AT+CCERTMOVE=”baidu.der”
OK
```

3 SSL Examples

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 Download certificate into module

Following commands shows how to download certificate into module.

```
AT+CCERTDOWN="client_key.der",1702

>-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAlwuz/TNa+foGBG6rXpW
E1Wnuc+GN9vS7MRenKOH+z2UfGuaV
BSb8VYFCgoL4RnWLwXAcLlaqw88zlCN89E
K6lydaAwNml/U6nu3oPsVkn8r9+sOX
yh9VD01DmSU349QWJvRgt1ocsFI1VTdd6RD
kVtu7FdKv4XC5WHcOD7yrElsVa7+G
Qbnm5cCCz8E75HH8vHZAOFeaV3HvlHnh/1R
Z+jh4ysyhEmFNOFCn3r9v2yu4kPRX
43xEsB13Ue4HgSbnT+Q7LIEK+dfsmUBoSps
S2NAmQOiqGrmmYygT3/V/lSX54hit
gli5bvg9DuNHYBwh2C+4nyZF95pMj2dEJf4jN
```

//Download file with not ASCII coding file name
//This command can also be used to write other certificates besides the “client_key”

```
wIDAQABAoIBAAJ9ze06QKDo79p4
3NjFjJhck/NTYB0XsIK/+iDhgWt4VogCD6kzG
GxsomU2tdOrsq9xIvXcthpeu5lQ
98mrpBhaWNC96JxlOh9O+0q1xNAh8AiH22Q
ZGjUTaC8Jfx+B6w+fbkz37os1/+00
6ZajkbChFTfp7r7ANj5wUEoQKZ4vNpLJxLWD
k6uH4ZMNveWcBaZQ21TUg9ZmoskK
EJ2ZEr/3kOSBgi2B6F50zyL8f1mbqPahHNLqt
rndV5/Lr4n74TqZXRwt5Cl9GrBv
tYXDHc+5Y7e1TUlXV00AMDlk+3cVR8m8Oa2
0tSdXjcw2iUk9brxb4uxreOouGfPW
5lO+q1ECgYEA4Kkok17DVx5FiapFQvJ2Jqi2/
WhzDncuBGbZtcLZnwRVfkPn3cBZ
JGNwxYyfEdwltPvTYQYh6Qg81XRdSRfF43G
zkQXNmkPOdZM0x3tFwzV6K5Fg7aeR
g50UddaA9MraCltOgK++7C6BvA3lmXciK4V
WeSZOmDW99Y6mgf92RdkCgYEArB2u
/ld72LGQBmx0Z+36Hf1dxo6RQ+dB+m6XBM
R8iuB/jGO/5PHdFoKoF2qa9Yj2W1+X
B29Xmc1HS6GTvkDlsN5JXNO7fDmlAxd5whb
wDdcmv3VEt8xJ2UeAClawjKtVcFoH
LRNlvDBttWVvlCZg+9HfVpuPm14oFxN/HtSXt
48CgYACxDJ6thUDspy6mD0oGOl5
kaRHNl0OJYuMhFOz+EVDvwLqfh2RzneKiiru
U8/1oVb+G4e7zx6FxxMwsbEgYEmQ
hmrmo0Kn3qPhMMHanvr572Oku7KM2p5hF4
MT/GM0lHdU31D1JrTcJap1TVomAaCL
FqY88arQFwFSz8Hfle0r6QKBgCbQLtTdzKzq
Jdt8+6cwQFYg+9O59MJGVVefNskp
chhzVfAX0n9Tl5Lq9fMJ5FX4g+3JGargjfWuG
CTTFBk0TM2t4wde7AmwiiivU5LU
T2Afo6pLTKrSE9k+yX2iug+O156VfsbleAm/N
g5RCJ91JCvFgULro6/axNmnWORf
9rK7AoGBAIK4edrX1MjerCsLu3y9Dy4pAx6E
R6ei4xpkO25U8wUcqqc+YD2m2xlA
DjqROlTeaxXkmPlyRKAXVarhk8LmXT/oDFU
APsTqUZ9LBrviqtMi+G2OFPbdKDwe
ZBNAgwFpFlUVoi0UYnZF8rBq0tepqivrayEWd
KKfMMJjq+l72SxD
-----END RSA PRIVATE KEY-----
OK
AT+CCERTLIST                                    //List certificate files
+CCERTLIST: "client_key.der"
OK
```

3.2 Access to TCP server

Following commands shows how to communicate with a TCP server.

```
AT+CCHSET=1                                  //Enable reporting +CCHSEND result
OK
AT+CCHSTART
OK

+CCHSTART: 0
AT+CCHOPEN=0,"www.baidu.com",80,1            //Connect to TCP server
OK

+CCHOPEN: 0,0
AT+CCHSEND=0,121                             //Send data to server
>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=D6F6D0D297CCAE39BD45C68399
6696C7; 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>7a367f7b87705e16b985e34ca59
b8ae8b1d28d47
Time : Tue Aug 21 10:55:16 CST
2018</center>
</body>
</html>
AT+CCHCLOSE=0                          //Disconnect from the Service
OK

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

+CCHSTOP: 0
```

3.3 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.

AT+CSSLCFG="sslversion",0,4	//Set SSL version for the first SSL context
OK	
AT+CSSLCFG="authmode",0,0	//Set the authentication mode(not verify server) of the first SSL context
OK	
AT+CCHSET=1	//Enable reporting +CCHSEND result
OK	
AT+CCHSTART	//Start SSL service, activate PDP context
OK	
+CCHSTART: 0	
AT+CCHSSLCFG=0,0	//Set the first SSL context to be used in the SSL connection
OK	
AT+CCHOPEN=0, "www.baidu.com",443,2	//Connect to SSL/TLS server
OK	
+CCHOPEN: 0,0	
AT+CCHSEND=0,121	//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**	
OK	
+CCHSEND: 0,0	
+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**	//Report the received data from server

```
Server: BWS/1.1
Set-Cookie: BD_NOT_HTTPS=1; path=/;
Max-Age=300
Set-Cookie:
BIDUPSID=D95046B2B3D5455BF01A622DB8
DED9EA; 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("ht
tps://","http://"));
    </script>
</head>
<body>
    <noscript><meta http-equiv="refresh"
content="0;url=http://www.baidu.com/"></nos
cript>
</body>
</html>
AT+CCHCLOSE=0                          //Disconnect from the Service
OK

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

+CCHSTOP: 0
```

3.4 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 the root CA of the server, and then it will connect to the server

successfully.

AT+CSSLCFG="sslversion",0,4	//Set SSL version for the first SSL context
OK	
AT+CSSLCFG="authmode",0,1	//Set the authentication mode(verify server) of the first SSL context
OK	
AT+CSSLCFG="cacert",0,"ca_cert.pem"	//Set the server root CA of the first SSL context
OK	
AT+CCHSET=1	//Enable reporting +CCHSEND result
OK	//Start SSL service, activate PDP context
AT+CCHSTART	
OK	
+CCHSTART: 0	
AT+CCHSSLCFG=0,0	//Set the first SSL context to be used in the SSL connection
OK	
AT+CCHOPEN=0,"www.baidu.com",443,2	//Connect to SSL/TLS server
OK	
+CCHOPEN: 0,0	
AT+CCHSEND=0,121	//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**	
OK	
+CCHSEND: 0,0	
+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 "**	//Report the received data from server

```
Pragma: no-cache
Server: BWS/1.1
Set-Cookie: BD_NOT_HTTPS=1; path=/;
Max-Age=300
Set-Cookie:
BIDUPSID=D95046B2B3D5455BF01A622DB8
DED9EA; 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("ht
tps://","http://"));
    </script>
</head>
<body>
    <noscript><meta http-equiv="refresh"
content="0;url=http://www.baidu.com/"></nos
cript>
</body>
</html>
AT+CCHCLOSE=0                          //Disconnect from the Service
OK

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

+CCHSTOP: 0
```

3.5 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 and the root CA of the server, the right client certificate and key, and then it will connect to the server successfully.

AT+CSSLCFG="sslversion",0,4	//Set SSL version for the first SSL context
OK	
AT+CSSLCFG="authmode",0,2	//Set the authentication mode(verify server and client) of the first SSL context
OK	
AT+CSSLCFG="cacert",0,"ca_cert.pem"	//Set the server root CA of the first SSL context
OK	
AT+CSSLCFG="clientcert",0,"cert.pem"	//Set the client certificate of the first SSL context
OK	
AT+CSSLCFG="clientkey",0,"key_cert.pem"	//Set the client key of the first SSL context
OK	
AT+CCHSET=1	//Enable reporting +CCHSEND result
OK	
AT+CCHSTART	//Start SSL service, activate PDP context
OK **+CCHSTART: 0**	
AT+CCHSSLCFG=0,0	//Set the first SSL context to be used in the SSL connection
OK	
AT+CCHOPEN=0,"www.baidu.com",443,2	//Connect to SSL/TLS server
OK **+CCHOPEN: 0,0**	
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**	//Send data to server
OK **+CCHSEND: 0,0**	
+CCHRECV: DATA,0,917 **HTTP/1.1 200 OK** **Accept-Ranges: bytes** **Cache-Control: no-cache** **Connection: Keep-Alive** **Content-Length: 227**	//Report the received data from server

```
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=D95046B2B3D5455BF01A622DB8
DED9EA; 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("ht
tps://","http://"));
    </script>
</head>
<body>
    <noscript><meta http-equiv="refresh"
content="0;url=http://www.baidu.com/"></nos
cript>
</body>
</html>
AT+CCHCLOSE=0                          //Disconnect from the Service
OK

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

+CCHSTOP: 0
```

3.6 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.

AT+CSSLCFG="sslversion",0,4	//Set SSL version for the first SSL context
OK	
AT+CSSLCFG="authmode",0,3	//Set the authentication mode(only verify client) of the first SSL context
OK	
AT+CSSLCFG="clientcert",0,"cert.pem"	//Set the client certificate of the first SSL context
OK	
AT+CSSLCFG="clientkey",0,"key_cert.pem"	//Set the client key of the first SSL context
OK	
AT+CCHSET=1	//Enable reporting +CCHSEND result
OK	
AT+CCHSTART	//Start SSL service, activate PDP context
OK **+CCHSTART: 0**	
AT+CCHSSLCFG=0,0	//Set the first SSL context to be used in the SSL connection
OK	
AT+CCHOPEN=0,"www.baidu.com",443,2	//Connect to SSL/TLS server
OK **+CCHOPEN: 0,0**	
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**	//Send data to server
OK **+CCHSEND: 0,0**	
+CCHRECV: DATA,0,917 **HTTP/1.1 200 OK**	//Report the received data from server

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

+CCHCLOSE: 0,0
AT+CCHSTOP                             //Stop SSL Service
```

```
OK

+CCHSTOP: 0
```

3.7 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.

```
AT+CCHMODE=1                          //Set the transparent mode
OK
AT+CCHSET=1                           //Enable reporting +CCHSEND result
OK
AT+CCHSTART                           //Start SSL service, activate PDP context
OK

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


HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: no-cache
Connection: Keep-Alive
Content-Length: 227
Content-Type: text/html               //Report the received data from server
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=F19D0F1E532ED84CE275BC1006F
91F9E; 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("ht
tps://","http://"));
    </script>
</head>
<body>
    <noscript><meta http-equiv="refresh"
content="0;url=http://www.baidu.com/"></nos
cript>
</body>
</html>
+++                                    //Switch to command mode
OK
AT+CCHCLOSE=0                          //Disconnect from the Service
OK

CLOSED
AT+CCHSTOP                             //Stop SSL Service
OK

+CCHSTOP: 0
```

NOTE

The appeal sample server is for demonstration purposes only, not for commercial purpose.

4 Appendix

4.1 Result codes and unsolicited codes

4.1.1 Command result <err> codes

<err>	Meaning
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	No net
18	Send data timeout
19	Not set certificates

4.1.2 Unsolicited result codes

URC	Meaning
+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.
+CCH:CCH STOP	CCH stopped caused by network error.