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 WIRELESS SOLUTIONS LIMITED. COPYING, TO OTHERS AND USING THIS DOCUMENT, ARE FORBIDDEN WITHOUT EXPRESS AUTHORITY BY SIMCOM. OFFENDERS ARE LIABLE TO THE PAYMENT OF INDEMNIFICATIONS. ALL RIGHTS RESERVED BY SIMCOM IN THE PROPRIETARY TECHNICAL INFORMATION, INCLUDING BUT NOT LIMITED TO REGISTRATION GRANTING OF A PATENT, A UTILITY MODEL OR DESIGN. ALL SPECIFICATION SUPPLIED HEREIN ARE SUBJECT TO CHANGE WITHOUT NOTICE AT ANY TIME.
About Document

Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Owner</th>
<th>What is new</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.00</td>
<td>2020.3.23</td>
<td>Ning.Lv</td>
<td>First Release</td>
</tr>
<tr>
<td>V1.01</td>
<td>2021.11.25</td>
<td>Xianxiang.Ma</td>
<td>Update format</td>
</tr>
</tbody>
</table>

Scope

This document applies to the SIMCom SIM820X series, SIM821X series, SIM826X series and SIM83XX series.
Contents

About Document............................................................................................................................................. 3

Version History.................................................................................................................................................. 3
Scope................................................................................................................................................................. 3

Contents.......................................................................................................................................................... 4

1 Introduction..................................................................................................................................................... 5

1.1 Purpose of the document.......................................................................................................................... 5
1.2 Related documents..................................................................................................................................... 5
1.3 Conventions and abbreviations.................................................................................................................. 5

2 MQTT Introduction......................................................................................................................................... 6

2.1 Characteristic............................................................................................................................................... 6
2.2 Request Method.......................................................................................................................................... 6

3 AT Commands for MQTT(S)......................................................................................................................... 8

4 Bearer Configuration....................................................................................................................................... 9

4.1 PDN Auto-activation.................................................................................................................................... 9

5 MQTT(S) Examples......................................................................................................................................... 11

5.1 MQTT Function......................................................................................................................................... 11
5.1.1 Access to MQTT server without SSL/TLS.......................................................................................... 11
5.1.2 Access to MQTT server without checking UTF8 coding.................................................................... 12
5.2 MQTTS Function....................................................................................................................................... 13
5.2.1 Connect to SSL/TLS MQTT server (not verify server)........................................................................ 13
5.2.2 Access to SSL/TLS MQTT server (only verify the server)................................................................. 15
5.2.3 Access to SSL/TLS MQTT server (verify server and client)............................................................ 16
1 Introduction

1.1 Purpose of the document

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

1.2 Related documents


1.3 Conventions and abbreviations

In this document, the GSM engines are referred to as following term:
- ME (Mobile Equipment);
- MS (Mobile Station);
- TA (Terminal Adapter);
- DCE (Data Communication Equipment) or facsimile DCE (FAX modem, FAX board);

In application, controlling device controls the GSM engine by sending AT Command via its serial interface. The controlling device at the other end of the serial line is referred to as following term:
- TE (Terminal Equipment);
- DTE (Data Terminal Equipment) or plainly "the application" which is running on an embedded system;
2 MQTT Introduction

MQTT (Message Queuing Telemetry Transport) is a lightweight broker-based publish/subscribe messaging protocol. It is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium.

2.1 Characteristic

- **Support client/server mode;**
  - The publish/subscribe message pattern to provide one-to-many message distribution and decoupling of applications
  - A messaging transport that is agnostic to the content of the payload
  - The use of TCP/IP to provide basic network connectivity
  - Three qualities of service for message delivery
  - A small transport overhead (the fixed-length header is just 2 bytes), and protocol exchanges minimised to reduce network traffic
  - A mechanism to notify interested parties to an abnormal disconnection of a client using the Last Will and Testament feature

2.2 Request Method

According to the MQTT standard, MQTT provides a variety of request methods.
CONNECT, SUBSCRIBE, PUBLISH, UNSUBSCRIBE, DISCONNECT, PINGREQ

<table>
<thead>
<tr>
<th>No</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONNECT</td>
<td>When a TCP/IP socket connection is established from a client to a server, a protocol level session must be created using a CONNECT flow.</td>
</tr>
</tbody>
</table>
The SUBSCRIBE message allows a client to register an interest in one or more topic names with the server. Messages published to these topics are delivered from the server to the client as PUBLISH messages. The SUBSCRIBE message also specifies the QoS level at which the subscriber wants to receive published messages.

A PUBLISH message is sent by a client to a server for distribution to interested subscribers. Each PUBLISH message is associated with a topic name (also known as the Subject or Channel). This is a hierarchical name space that defines taxonomy of information sources for which subscribers can register an interest. A message that is published to a specific topic name is delivered to connected subscribers for that topic.

An UNSUBSCRIBE message is sent by the client to the server to unsubscribe from named topics.

The DISCONNECT message is sent from the client to the server to indicate that it is about to close its TCP/IP connection. This allows for a clean disconnection, rather than just dropping the line.

The PINGREQ message is an "are you alive?" message that is sent from a connected client to the server.
# 3 AT Commands for MQTT(S)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CMQTTSTART</td>
<td>Start MQTT service</td>
</tr>
<tr>
<td>AT+CMQTTSTOP</td>
<td>Stop MQTT service</td>
</tr>
<tr>
<td>AT+CMQTTACCQ</td>
<td>Acquire a MQTT client</td>
</tr>
<tr>
<td>AT+CMQTREL</td>
<td>Release a MQTT client</td>
</tr>
<tr>
<td>AT+CMQTTPASSLCFG</td>
<td>Set the SSL context</td>
</tr>
<tr>
<td>AT+CMQTTWILLTOPIC</td>
<td>Input the topic of will message</td>
</tr>
<tr>
<td>AT+CMQTTWILLMSG</td>
<td>Input the will message</td>
</tr>
<tr>
<td>AT+CMQTTCOMNECT</td>
<td>Connect to a MQTT server</td>
</tr>
<tr>
<td>AT+CMQTTDISC</td>
<td>Disconnect from server</td>
</tr>
<tr>
<td>AT+CMQTTTPROPIDC</td>
<td>Input the publish message topic</td>
</tr>
<tr>
<td>AT+CMQTTTPAYLOAD</td>
<td>Input the publish message body</td>
</tr>
<tr>
<td>AT+CMQTTPUB</td>
<td>Publish a message to server</td>
</tr>
<tr>
<td>AT+CMQTTSUBTOPIC</td>
<td>Input a subscribe message topic</td>
</tr>
<tr>
<td>AT+CMQTTSUB</td>
<td>Subscribe a message to server</td>
</tr>
<tr>
<td>AT+CMQTUNSUBTOPIC</td>
<td>Input a unsubscribe message topic</td>
</tr>
<tr>
<td>AT+CMQTUNSUB</td>
<td>Unsubscribe a message to server</td>
</tr>
<tr>
<td>AT+CMQTTCFG</td>
<td>Configure the MQTT Context</td>
</tr>
</tbody>
</table>

For detail information, please refer to “SIM82XX_SIM83XX Series_AT Command Manual".
4 Bearer Configuration

Usually module will register PS service automatically.

4.1 PDN Auto-activation

```plaintext
AT+CPIN?
+CPIN: READY // Check Status of SIM Card
OK
AT+CSQ
+CSQ: 27,99 // Check RF Signal
OK
AT+CGREG?
+CGREG: 0,1 // Check Status of PS Service
OK
AT+CEREG?
+CEREG: 0,1
OK
AT+COPS?
+COPS: 0,0,"CHN-UNICOM",13 // Check Information of Operator
OK
AT+CPSI?
+CPSI: NR5G_SA,Online,460-00,0x161816,13190066179,476,NR5G_BAND41,504990,-1130,-140,30 // Check Information of Network
OK
AT+CGDCONT=1, "IP", "CMNET" // Set PDP context Parameters
OK
AT+CGDCONT?
+CGDCONT: 1,"IPV4","CMNET","0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0","0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0",0,0,0,0 // Check Information of PDP Context
```
5 MQTT(S) Examples

5.1 MQTT Function

5.1.1 Access to MQTT server without SSL/TLS

//Example of Access to MQTT server without SSL/TLS.

```
AT+CMQTTSTART
OK

+CMQTTSTART: 0
AT+CMQTTACCQ=0,"client test0"
OK
AT+CMQTTWILLTOPIC=0,10
>0123456789
OK
AT+CMQTTWILLMSG=0,6,1
>qwerty
OK
AT+CMQTTCONNECT=0,"tcp://test.mosquitto.org:1883",60,1
OK

+CMQTTCONNECT: 0,0
AT+CMQTTSUB=0,10,1
>simcomtest
OK

+CMQTTSUB: 0,0
AT+CMQTTTOPIC=0,10
> simcomtest
OK
AT+CMQTTPAYLOAD=0,9
>mqtt_test
OK
AT+CMQTTTPUB=0,1,60
OK

Set the payload for the PUBLISH message
Set the topic for the PUBLISH message
Subscribe one topic from the server
Connect to a MQTT server
Set the will message for the CONNECT message
Set the will topic for the CONNECT message
Acquire one client which will connect to a MQTT server not SSL/TLS
Start MQTT service, activate PDP context
```

www.simcom.com
+CMQTTSTART: 0
OK

//Example of Access to MQTT server without checking UTF8 coding.

AT+CMQTTSTART
OK

+CMQTTSTART: 0
AT+CMQTTACCQ=0,"client test0"
OK

AT+CMQTTSTOP
OK

5.1.2 Access to MQTT server without checking UTF8 coding

//Example of Access to MQTT server without checking UTF8 coding.

AT+CMQTTSTART
OK

+CMQTTSTART: 0
AT+CMQTTACCQ=0,"client test0"
OK

AT+CMQTTSTOP
OK

+CMQTTPUB: 0,0

+CMQTTRXSTART: 0,10,9
+CMQTTRXTOPIC: 0,10
simcomtest
+CMQTTRXPAYLOAD: 0,9
mqtt_test
+CMQTTRXEND: 0

AT+CMQTTSUBTOPIC=0,9,1
>123456789
OK

AT+CMQTTSUB=0
OK

+CMQTTSUB: 0,0
AT+CMQTTUNSUB=0,9,0
>simcommsg
OK

+CMQTTUNSUB: 0,0
AT+CMQTTDISC=0,120
OK

+CMQTTDISC: 0,0
AT+CMQTTREL=0
OK

AT+CMQTTSTOP
OK

+CMQTTSTOP: 0

Receive publish message from server

Set one topic for the SUBSCRIBE message

Subscribe a message

Unsubscribe one topic from the server

Disconnect from server

Release the client

Stop MQTT Service

Access to MQTT server without checking UTF8 coding
### 5.2 MQTTS Function

#### 5.2.1 Connect to SSL/TLS MQTT server (not verify server)

```c
// Example of Access to a MQTT server without verifying the server
AT+CMQTTSSTART OK
```

Connect to a MQTT server

```
+CMQTTCONNECT=0,"tcp://test.mosquitto.org:1883",60,1
OK

+CMQTTCONNECT: 0,0
AT+CMQTTSUB=0,9,1
>盼望盼望盼望
OK

+CMQTTSUB: 0,0
AT+CMQTTTPUB=0,1,60
OK

set the topic for the PUBLISH message

```
+CMQTTPUB: 0,0

+CMQTTRXSTART: 0,9,0
+CMQTTRXTOPIC: 0,9
盼望盼望盼望
+CMQTTRXEND: 0
AT+CMQTTTPUB=0,1,60
OK

Receive publish message from server

```
+CMQTTPUB: 0,0

+CMQTTTPUB=0,1,60
OK

Set the topic for the PUBLISH message

```
+CMQTTDISC=0,120
OK

Disconnect from server

```
+CMQTTPUB=0,0

AT+CMQTTDISC=0,120
OK

release the client

```
+CMQTTREL=0
OK

Release the client

```
AT+CMQTTSTOP
OK

Stop MQTT Service

```
+CMQTTSTOP: 0
```
+CMQTTSTART: 0
AT+CMQTACCQ=0,"client test0",1
OK
AT+CMQTTWILLTOPIC=0,10
>0123456789
OK
AT+CMQTTWILLMSG=0,6,1
> qwerty
OK
AT+CMQTTCONNECT=0,"tcp://test.mosquitto.org:8883",60,1
OK

+CMQTTCONNECT: 0,0
AT+CMQTTTOPIC=0,13
> ddrrrggghhhk
OK
AT+CMQTTTPAYLOAD=0,60
>0123456789012345678901234567890123456789012345678901234567890123456789
OK
AT+CMQTTPUB=0,1,60
OK

+CMQTTPUB: 0,0
AT+CMQTTSUBTOPIC=0,9,1
>123456789
OK
AT+CMQTTSUB=0
OK

+CMQTTSUB: 0,0
AT+CMQTTSUB=0,9,1
>simcommsmsg
OK

+CMQTTSUB: 0,0
AT+CMQTSubmitSUB=0,9,0
>simcommsmsg
OK

+CMQTTUNSUB: 0,0
AT+CMQTTDISC=0,120
OK

Acquire one client which will connect to a SSL/TLS MQTT server
Set the will topic for the CONNECT message
Set the will message for the CONNECT message
Connect to a MQTT server
Set the topic for the PUBLISH message
Set the payload for the PUBLISH message
Publish a message
Set one topic for the SUBSCRIBE message
Subscribe a message
Subscribe one topic from the server
Unsubscribe one topic from the server
Disconnect from server
5.2.2 Access to SSL/TLS MQTT server (only verify the server)

//Example of Access to SSL/TLS MQTT server (only verify the server)
AT+CSSLCFG="sslversion",0,4
OK
AT+CSSLCFG="authmode",0,1
OK
AT+CSSLCFG="cacert",0,"server_ca.pem"
OK
AT+CMQTTSTART
OK

+CMQTTSTART: 0
AT+CMQTTACCQ=0,"client test0",1
OK
AT+CMQTTPSSSFCFG=0,0
OK
AT+CMQTTPWILLTOPIC=0,10
>0123456789
OK
AT+CMQTTPWILLMSG=0,6,1
>qwerty
OK
AT+CMQTTCONNECT=0,"tcp://mqtts_server:port",60,1
OK

+CMQTTCONNECT: 0,0
AT+CMQTTPTOPIC=0,13
>dddrrrggghhhk
OK
AT+CMQTTTPPAYLOAD=0,60
>0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

AT+CMQTTSTOP
OK

Stop MQTT Service

+CMQTTSTOP: 0

Release the client
### 5.2.3 Access to SSL/TLS MQTT server (verify server and client)

```
AT+CSSLCFG="sslversion",0,4  # Set the SSL version of 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
```
OK
AT+CSSLCFG="clientkey",0,"key_cert.pem" Ok
AT+CMQTSTART Ok

+CMQTSTART: 0
AT+CMQTSTART=0,"client test0",1 Ok
AT+CMQTSSLCFG=0,0 Ok
AT+CMQTSTWILLTOPIC=0,10 >0123456789 Ok
AT+CMQTSTWILLMSG=0,6,1 >qwerty Ok
AT+CMQTCONNECT=0,"tcp://hooleeping.com:8883",60,1 Connect to a MQTT server

+CMQTCONNECT: 0,0
AT+CMQTSSSSTOPIC=0,13 >dddrрггггшк Ok
AT+CMQTTPSLOAD=0,60 >01234567890123456789012345678901234567890123456789 Ok
AT+CMQTTPUB=0,1,60 Publish a message

+CMQTTPUB: 0,0
AT+CMQTTPSUBTOPIC=0,9,1 Set one topic for the SUBSCRIBE message
>123456789 Ok
AT+CMQTTPSUB=0 Subscribe a message

+CMQTTPSUB: 0,0
AT+CMQTTPSUB=0,9,1 Subscribe one topic from the server
>simcommsg Ok

+CMQTTPSUB: 0,0
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CMQTUNSUB=0,9,0</td>
<td>Unsubscribe one topic from the server</td>
</tr>
<tr>
<td>&gt;simcommsg</td>
<td></td>
</tr>
<tr>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>+CMQTUNSUB: 0,0</td>
<td>Disconnect from server</td>
</tr>
<tr>
<td>AT+CMQTTDISC=0,120</td>
<td></td>
</tr>
<tr>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>+CMQTTDISC: 0,0</td>
<td>Release the client</td>
</tr>
<tr>
<td>AT+CMQTREL=0</td>
<td></td>
</tr>
<tr>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>AT+CMQTSTOP</td>
<td>Stop MQTT Service</td>
</tr>
<tr>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>+CMQTSTOP: 0</td>
<td></td>
</tr>
</tbody>
</table>