

SimpleLink™ Wi-Fi® AT Command User's Guide

The SimpleLink™ Wi-Fi® Internet-on-a chip™ family of devices from Texas Instruments™ provides a suite of integrated protocols for Wi-Fi and internet connectivity to dramatically simplify the implementation of internet-enabled devices and applications.

This document describes the AT command protocol for SimpleLink, which is a widely used method to configure and control embedded networking systems due to its simplicity, textual parameter representation, and inherent flexibility.

Contents

1	Supported Platforms	4
2	Architecture Overview	4
3	Getting Started	5
4	Commands Summary	5
5	Protocol Syntax	7
6	Command Description.....	8

List of Tables

1	Device Commands	5
2	Socket Commands	5
3	WLAN Commands.....	6
4	File System Commands	6
5	Network Application Commands.....	6
6	Network Configuration Commands.....	6
7	Network Utility Commands	7
8	Asynchronous Events	7
9	AT+Start Starts the NWP	8
10	AT+Stop Stops the NWP.....	8
11	AT+Get Getting Device Configurations.....	9
12	AT+Set Setting Device Configurations.....	10
13	AT+Test Test Command.....	10
14	AT+Socket Create an End-Point for Communication.....	10
15	AT+Close Close Socket	11
16	AT+Accept Accept a Connection on a Socket.....	11
17	AT+Bind Assign a Name to a Socket.....	11
18	AT+Listen Listen for Connections on a Socket.....	11
19	AT+Connect Initiate a Connection on a Socket	12
20	AT+Select Monitor Socket Activity	12
21	AT+SetSockOpt Set Socket Options	13
22	AT+GetSockOpt Get Socket Options.....	15
23	AT+Recv Read Data From TCP Socket	15
24	AT+RecvFrom Read Data From Socket	16
25	AT+Send Write Data to TCP Socket	16
26	AT+SendTo Write Data to Socket	16

27	AT+WlanConnect Connect to WLAN Network as a Station	17
28	AT+WlanDisconnect Disconnect the Connection	17
29	AT+WlanProfileAdd Add Profile	18
30	AT+WlanProfileGet Get Profile.....	18
31	AT+WlanProfileDel Delete Profile	18
32	AT+WlanPolicySet Set Policy Values	19
33	AT+WlanPolicyGet Get Policy Values	20
34	AT+WlanScan Gets the WLAN Scan Operation Results.....	20
35	AT+WlanSetMode WLAN Set Mode.....	21
36	AT+WlanSet Setting WLAN Configurations	21
37	AT+ WlanGet Getting WLAN Configurations	23
38	AT+FileOpen Open File in Storage Device	25
39	AT+FileClose Close File in Storage Device	25
40	AT+FileCtl Controls Various File System Operations	26
41	AT+FileDel Delete File From Storage Device	27
42	AT+FileGetFilelist Get a List of Files	27
43	AT+FileGetInfo Get Information About a File.....	27
44	AT+FileRead Read a Block of Data From a File in Storage Device	27
45	AT+FileWrite Write Block of Data to a File in Storage Device	28
46	AT+NetAPPStart Starts a Network Application.....	29
47	AT+NetAPPStop Stops a Network Application	29
48	AT+NetAPPGetHostByName Get Host IP by Name.....	29
49	AT+NetAPPGetHostByService Get Host IP by Service.....	30
50	AT+NetAPPSet Setting Network Application Configurations	30
51	AT+NetAPPGet Getting Network Applications Configurations	32
52	AT+NetAPPSend Sends Network Application Response or Data Following a Network Application Request Event.....	32
53	AT+NetAPPRecv Receives Data From the Network Processor Following a Network Application Response Event.....	33
54	AT+NetAPPping Send Ping to Network Hosts	33
55	AT+NetAPPGetServiceList Get Service List.....	34
56	AT+NetAPPRegisterService Register a New mDNS Service	34
57	AT+NetAPPUnRegisterService Unregister mDNS Service	35
58	AT+NetCfgSet Setting Network Configurations	36
59	AT+NetCfgGet Getting Network Configurations.....	37
60	AT+NetUtilGet Getting Utilities Configurations	39
61	AT+NetUtilCmd Performing Utilities-Related Commands.....	40
62	+EventFatalError Fatal Error Event for Inspecting Fatal Error.....	41
63	+EventGeneral General Asynchronous Event for Inspecting General Events	41
64	+EventWlan WLAN Asynchronous Event	42
65	+EventNetApp Network Application Asynchronous Event.....	43
66	+EventSock Socket Asynchronous Event.....	43
67	+EventMqtt MQTT Asynchronous Event	44
68	AT+MqttCreate MQTT Client Create	45
69	AT+MqttDelete MQTT Client Delete	46
70	AT+MqttConnect MQTT Client Connect to Broker	46
71	AT+MqttDisconnect MQTT Client Disconnect From Broker	46
72	AT+MqttPublish MQTT Client Send Message to Broker	47
73	AT+MqttSubscribe MQTT Client Subscribe for Topic	47

74	AT+MqttUnsubscribe MQTT Client Unsubscribe for Topic	47
75	AT+MqttSet MQTT Client Set Option	48
76	AT+HttpCreate Http Client Create	49
77	AT+HttpDestroy Http Client Delete	49
78	AT+HttpConnect Http Client Connect to Host.....	49
79	AT+HttpDisconnect Http Client Disconnect From Host	49
80	AT+HttpSendReq Http Client Send Request to Host.....	50
81	AT+HttpReadResBody Http Client Read Response Body From Host.....	50
82	AT+HttpSetHeader Http Client Set Header	51
83	AT+HttpGetHeader Http Client Get Header	52
84	AT+HttpSetOpt Http Client Set Option	53
85	AT+HttpSetProxy Http Client Set Proxy Address	53

Trademarks

SimpleLink, Internet-on-a chip, Texas Instruments are trademarks of Texas Instruments.
 Wi-Fi, Wi-Fi Direct are registered trademarks of Wi-Fi Alliance.
 All other trademarks are the property of their respective owners.

1 Supported Platforms

Hardware platforms that support the AT command library are:

- CC3220R
- CC3220S
- CC3220SF

2 Architecture Overview

SimpleLink Wi-Fi AT Command consists of two main modules:

- AT Commands Application
 - The application is one of the following application demos:
 - The AT_Commands application provides control by the AT Commands on the local device.
 - The Serial_wifi application provides control by the AT Commands on the local and the remote device.
 - The user-customized application is based on the two previous applications.
- AT Command Core
 - The core includes the command parser, execution, and return status.
 - The AT Command Core should already be compiled into the library.

The following API communicate between the two modules:

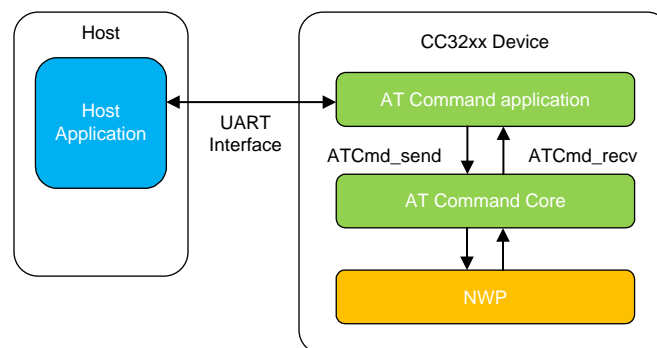
- *ATCmd_create* creates the AT Command core task and initializes the RX event queue.
- *ATCmd_send* transmits string from the AT Command application to the AT Command Core. The function takes one parameter, *Buffer*, which stores the sent string.
- *ATCmd_rcv* transmits a string from the AT Command Core to the AT Command application.

The function takes two parameters:

- *Buffer* stores the received string.
- *Nonblock variant* set to 0 for *waits forever* on the RX queue, otherwise set to 1.

All send and receive buffers should be allocated by the AT Commands application.

Figure 1 shows the basic architecture.



Copyright © 2017, Texas Instruments Incorporated

Figure 1. Basic Architecture Scheme

3 Getting Started

The following describes the procedure to build the AT Command Core. For building and executing the application binary file, refer to the *README.html* file that is located in each AT Command application. Ensure that the AT Command library includes in the application linking list.

The AT Command Core is prebuilt into the library “atcmd.a” per two OS (TI-RTOS and FreeRTOS) and per three compilers (CCS, GCC, and IAR). In the case where changes must be made to the core and you need to recompile it, there are two ways to build it:

- For CCS (TI-RTOS or FreeRTOS), import the CCS project located under `{SDK ROOT}\source\ti\net\atcmd\ccs` and build the library.

NOTE: Pay attention to choose the appropriate product number.

- For all other favorites (including CCS), open the command prompt line under the directory `{SDK ROOT}\source\ti\net\atcmd`, and execute *gmake* from the XDC tool root directory. To clean all outputs, execute *gmake clean*.

4 Commands Summary

Table 1. Device Commands

Command	Definition
AT+Start	Starts the network processor (NWP)
AT+Stop	Stops the NWP
AT+Get	Gets device configurations
AT+Set	Sets device configurations
AT+Test	Test command

Table 2. Socket Commands

Command	Definition
AT+Socket	Create an endpoint for communication
AT+Close	Close socket
AT+Accept	Accept a connection on a socket
AT+Bind	Assign a name to a socket
AT+Listen	Listen for connections on a socket
AT+Connect	Initiate a connection on a socket
AT+Select	Monitor socket activity
AT+SetSockOpt	Set socket options
AT+GetSockOpt	Get socket options
AT+Recv	Read data from TCP socket
AT+RecvFrom	Read data from socket
AT+Send	Write data to TCP socket
AT+SendTo	Write data to socket

Table 3. WLAN Commands

Command	Definition
AT+WlanConnect	Connect to WLAN network as a station
AT+WlanDisconnect	Disconnect connection
AT+WlanProfileAdd	Add profile
AT+WlanProfileGet	Get profile
AT+WlanProfileDel	Delete profile
AT+WlanPolicySet	Set policy values
AT+WlanPolicyGet	Get policy values
AT+WlanScan	Gets the WLAN scan operation results
AT+WlanSetMode	WLAN set mode
AT+WlanSet	Setting WLAN configurations
AT+WlanGet	Getting WLAN configurations

Table 4. File System Commands

Command	Definition
AT+FileOpen	Open file in storage device
AT+FileClose	Close file in storage device
AT+FileCtl	Controls various file system operations
AT+FileDel	Delete file from storage device
AT+FileGetFilelist	Get list of a files
AT+FileGetInfo	Get information of a file
AT+FileRead	Read block of data from a file in storage device
AT+FileWrite	Write block of data to a file in storage device

Table 5. Network Application Commands

Command	Definition
AT+NetAPPStart	Starts a network application
AT+NetAPPStop	Stops a network application
AT+NetAPPGetHostByName	Get host IP by name
AT+NetAPPGetHostByService	Host IP by service
AT+NetAPPSet	Setting network applications configurations
AT+NetAPPGet	Getting network applications configurations
AT+NetAPPSend	Sends Network Application response or data following a Network Application request event
AT+NetAPPRecv	Receives data from the network processor following a Network Application response event
AT+NetAPPping	Send ping to network hosts
AT+NetAPPGetServiceList	Get service list
AT+NetAPPRegisterService	Register a new mDNS service
AT+NetAPPUnRegisterService	Unregister mDNS service

Table 6. Network Configuration Commands

Command	Definition
AT+NetCfgSet	Setting network configurations
AT+NetCfgGet	Getting network configurations

Table 7. Network Utility Commands

Command	Definition
AT+NetUtilGet	Getting utilities configurations
AT+NetUtilCmd	Performing utilities-related commands

Table 8. Asynchronous Events

Command	Definition
+EventFatalError	Fatal Error event for inspecting fatal error
+EventGeneral	General asynchronous event for inspecting general events
+EventWlan	WLAN asynchronous event
+EventNetApp	Network Application asynchronous event
+EventSock	Socket asynchronous event

5 Protocol Syntax

5.1 Commands

Syntax:

AT<command name>=<param1>, <param2>, ..., <paramX>

- Commands that contain parameters should include an equal sign (=) between the command name and the first parameter.
- Commands that contain parameters should include a comma mark (,) as a delimiter between them—comma delimiters are mandatory.
- In case the parameter is defined as "ignore" or "optional", it could be left empty but the comma delimiter should be mentioned—it looks like two conjunction delimiters (,,).
- Parameters that are left empty must be treated as 0 or NULL (according to the parameter type), and in case it was not defined as "ignore" or "optional", an error should be raised.
- String parameters containing spaces must be enclosed with quotes (" ").
- String parameters containing a comma delimiter (,) must be enclosed with quotes (" ").
- Numeric value parameters could be one of the following:
 - Decimal
 - Hexadecimal—must have a prefix of zero x notation (0x)
- Numeric array parameters could be enclosed with square brackets ([]).
- Numeric array parameters could be one of the following:
 - IPv4 address—contains four numeric values (8 bits each) with a point mark (.) as a delimiter between them enclosed with or without square brackets—x.x.x.x or [x.x.x.x]
 - IPv6 address—contains four numeric values (32 bit each) with a colon mark (:) as a delimiter between them enclosed with or without square brackets—x:x:x:x or [x:x:x:x]
 - MAC address—contains six numeric values (8 bit each) with a colon mark (:) as a delimiter between them enclosed with or without square brackets—x:x:x:x:x or [x:x:x:x:x]
- Bitmask parameters should contain values with a vertical bar (|) as delimiter between them enclosed with or without square brackets—x|x|x or [x|x|x]
- The AT command handler allows for the AT commands to be entered in uppercase or lowercase with spaces between the arguments.
- Data parameter should be one of the following formats:
 - Binary format
 - Base64 format—binary to text encoding

5.2 Command Return Status

Command return status could be one of the following cases:

- Command that returns values:

```
<command name>: <value1>, ..., <valueX>
```

- Command that returns success:

```
OK
```

- Command that returns failure:

```
ERROR:<error description>, <error code>
```

Command return status should include a colon mark (:) between the command name and the first value.

Command return status that contains list values should include a semicolon mark (;) as a delimiter between the list members.

5.3 Asynchronous Event

The events may arrive at any time. Asynchronous events are always built in the following format:

```
<event name>: <event ID>, <value1>, ..., <valueX>
```

The event should include a colon mark (:) between the event name and the event ID.

6 Command Description

6.1 Device Commands

Table 9. AT+Start Starts the NWP

Request:	Response:
AT+Start	OK
Arguments: none	Arguments: none

Table 10. AT+Stop Stops the NWP

Request:	Response:
AT+Stop = [Timeout]	OK
Arguments: Timeout: Stop timeout in milliseconds should be used to give the device time to finish any transmission or reception that is not completed when the function was called. <ul style="list-style-type: none"> • 0: Enter to hibernate immediately • 0xFFFF: Host waits for the response from the device before hibernating, without timeout protection • 0 <Timeout[msec] <0xFFFF: Host waits for the response from the device before hibernating, with a defined timeout protection This timeout defines the maximum time to wait. The NWP response can be sent earlier than this timeout. 	Arguments: none

Table 11. AT+Get Getting Device Configurations

Request:		Response:
AT+Get = [ID],[Option]		+Get:[Value1],...,[ValueX] OK
Arguments:		Arguments:
ID	Option	Return Values
<i>Status</i>	<i>Device</i>	Value1: bitmask: General error
	<i>WLAN</i>	Value1: bitmask: <ul style="list-style-type: none"> • WLANASYNCONNECTEDRESPONSE • WLANASYNCDISCONNECTEDRESPONSE • STA_CONNECTED • STA_DISCONNECTED • P2P_DEV_FOUND • CONNECTION_FAILED • P2P_NEG_REQ_RECEIVED • RX_FILTERS
	<i>BSD</i>	Value1: bitmask: <ul style="list-style-type: none"> • TX_FAILED
	<i>NETAPP</i>	Value1: bitmask: <ul style="list-style-type: none"> • IPACQUIRED • IPACQUIRED_V6 • IP_LEASED • IP_RELEASED • IPV4_LOST • DHCP_ACQUIRE_TIMEOUT • IP_COLLISION • IPV6_LOST
<i>General</i>	<i>Version</i>	<ul style="list-style-type: none"> • Value1: Chip ID • Value2: FW Version (x.x.x.x) • Value3: PHY Version (x.x.x.x) • Value4: NWP Version (x.x.x.x) • Value5: ROM Version
	<i>Time</i>	<ul style="list-style-type: none"> • Value1: Hour = Current hours • Value2: Minute = Current minutes • Value3: Second = Current seconds • Value4: Day = Current Date, 1–31 • Value5: Month = Current Month, 1–12 • Value6: Year = Current year
	<i>Persistent</i>	Value1: <ul style="list-style-type: none"> • 1: Enable • 0: Disable
<i>IOT</i>	<i>UDID</i>	16 bytes

Table 12. AT+Set Setting Device Configurations

Request:			Response:
AT+Set = [ID],[Option],[Value1],...,[ValueX]			OK
Arguments:			
ID	Option	Value	
General	<i>Persistent</i> sets the default system-wide configuration persistence mode. In case true, all APIs that follow <i>system configured</i> persistence (see persistence attribute noted per API) shall maintain the configured settings. In case false, all calls to APIs that follow <i>system configured</i> persistence shall be volatile. Configuration should revert to default after reset or power recycle.	Value1: <ul style="list-style-type: none"> • 1: Enable • 0: Disable 	
	<i>Time</i> sets the device time and date	<ul style="list-style-type: none"> • Value1: Hour = Current hours • Value2: Minute = Current minutes • Value3: Second = Current seconds • Value4: Day = Current Date, 1–31 • Value5: Month = Current Month, 1–12 • Value6: Year = Current year 	

Table 13. AT+Test Test Command

Request:	Response:
AT+Test	OK
Arguments: none	Arguments: none

6.2 Socket Commands

Table 14. AT+Socket Create an End-Point for Communication

Request:	Response:
AT+Socket = [Domain],[Type],[Protocol]	+Socket: [socket] OK
Arguments: <ul style="list-style-type: none"> • Domain: Specifies the protocol family of the created socket: <ul style="list-style-type: none"> – INET: For network protocol IPv4 – INET6: For network protocol IPv6 – RF: For starting transceiver mode • Type: Specifies the communication semantic: <ul style="list-style-type: none"> – STREAM: Reliable stream-oriented service or Stream Sockets – DGRAM: Datagram service or Datagram Sockets – RAW: Raw protocols atop the network layer • Protocol: Specifies a particular transport to be used with the socket: <ul style="list-style-type: none"> – TCP – UDP – RAW – SEC 	Arguments: socket: Socket descriptor that will be used in the socket commands described in Table 15 through Table 26 .

Table 15. AT+Close Close Socket

Request:	Response:
AT+Close = [socket]	+Close: [socket] OK
Arguments: socket: Socket descriptor received from AT+Socket command	

Table 16. AT+Accept Accept a Connection on a Socket

Request:	Response:
AT+Accept = [socket],[family]	OK +Accept: [New Socket],[Family],[Port],[Address]
Arguments: <ul style="list-style-type: none"> • socket: Socket descriptor received from AT+Socket command • family: Specifies the protocol family of the created socket: <ul style="list-style-type: none"> – INET: For network protocol IPv4 – INET6: For network protocol IPv6 	<ul style="list-style-type: none"> • NewSocket: New connected socket • Family: internet protocol (AF_INET) • Port: Address port • Address: Peer socket address

Table 17. AT+Bind Assign a Name to a Socket

Request:	Response:
AT+Bind = [Socket],[Family],[Port],[Address]	OK
Arguments: <ul style="list-style-type: none"> • Socket: Socket descriptor received from AT+Socket command • Family: Specifies the protocol family of the created socket: <ul style="list-style-type: none"> – INET: For network protocol IPv4 – INET6: For network protocol IPv6 • Port: Address port • Address: Local socket address 	

Table 18. AT+Listen Listen for Connections on a Socket

Request:	Response:
AT+Listen = [socket],[backlog]	OK
Arguments: <ul style="list-style-type: none"> • socket: Received from AT+Socket command • backlog: Listen 	

Table 19. AT+Connect Initiate a Connection on a Socket

Request:	Response:
AT+Connect = [Socket],[Family],[Port],[Address]	OK +Connect : [Port], [Address]
Arguments: <ul style="list-style-type: none"> • Socket: Received from AT+Socket command • Family: internet protocol: <ul style="list-style-type: none"> – INET: For network protocol IPv4 – INET6: For network protocol IPv6 • Port: Address port • Address: Peer socket address (“x.x.x.x”) 	

Table 20. AT+Select Monitor Socket Activity

Request:	Response:
AT+Select = [nfds],[readsds],[timeout sec],[timeout usec]	OK +Select: [readsds]
Arguments: <ul style="list-style-type: none"> • nfds: The highest-numbered file descriptor in any of the three sets (read, write, and except) • readsds: Socket descriptors as bit list (for example, 0 2 for monitoring socket 0 and socket 2) • timeout sec: Time in seconds is an upper bound on the amount of time elapsed before select() returns. 0 means return immediately. • timeout usec: Time in microseconds 	Arguments: readsds: Socket descriptors list for read monitoring and accept monitoring

Table 21. AT+SetSockOpt Set Socket Options

Request:		Response:
AT+SetSockOpt = [sd],[Level],[Option],[Value1],...,[ValueX]		OK
Arguments: sd: Socket descriptor		
Level: Defines the protocol level for this option	Option	Value
SOCKET	KEEPALIVE Enable or disable periodic keep alive. Keeps TCP connections active by enabling the periodic transmission of messages	Value1: <ul style="list-style-type: none"> 1: Enable 0: Disable
	KEEPALIVETIME Set keep alive timeout	Value1: Timeout in seconds
	RX_NO_IP_BOUNDARY Enable or disable RX IP boundary	Value1: <ul style="list-style-type: none"> 1: Enable 0: Disable
	RCVTIMEO Sets the timeout value that specifies the maximum amount of time an input function waits until it completes	<ul style="list-style-type: none"> Value1: Seconds Value2: Microseconds. 10000 microseconds resolution
	RCVBUF Sets TCP maximum receive window size	Value1: Size in bytes
	NONBLOCKING Sets socket to nonblocking	Value1: <ul style="list-style-type: none"> 1: Enable 0: Disable
	SECMETHOD Sets method to TCP secured socket	Value1 security method: <ul style="list-style-type: none"> SSLV3: Security method SSL v3 TLSV1: Security method TLS v1 TLSV1_1: Security method TLS v1_1 TLSV1_2: Security method TLS v1_2 SSLV3_TLSV1_2: Use highest possible version from SSLv3–TLS 1.2
	SECURE_MASK Sets specific ciphers as OR bitmask to TCP secured socket (default value: all ciphers)	Value1: Cipher type: <ul style="list-style-type: none"> SSL_RSA_WITH_RC4_128_SHA SSL_RSA_WITH_RC4_128_MD5 TLS_RSA_WITH_AES_128_CBC_SHA TLS_DHE_RSA_WITH_AES_128_CBC_SHA TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA TLS_ECDHE_RSA_WITH_RC4_128_SHA TLS_RSA_WITH_AES_128_CBC_SHA256 TLS_RSA_WITH_AES_256_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA TLS_RSA_WITH_AES_128_GCM_SHA256 TLS_RSA_WITH_AES_256_GCM_SHA384 TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256

Table 21. AT+SetSockOpt Set Socket Options (continued)

Request:		Response:
SOCKET (continued)	<i>SECURE_FILES_CA_FILE_NAME</i> Map secured socket to CA file by name	Value1: File name
	<i>SECURE_FILES_PRIVATE_KEY_FILE_NAME</i> Map secured socket to private key by name	Value1: File name
	<i>SECURE_FILES_CERTIFICATE_FILE_NAME</i> Map secured socket to certificate file by name	Value1: File name
	<i>SECURE_FILES_DH_KEY_FILE_NAME</i> Map secured socket to Diffie Hellman file by name	Value1: File name
	<i>CHANGE_CHANNEL</i> Sets channel in transceiver mode	Value1: Channel number (range is 1–13)
	<i>SECURE_ALPN</i> Sets the ALPN list	Value1: The parameter is a bit map consist of or of the following values: H1 H2 H2C H2_14 H2_16 FULL_LIST
	<i>LINGER</i> Socket lingers on close pending remaining send and receive packets	<ul style="list-style-type: none"> • Value1: <ul style="list-style-type: none"> – 1: Enable – 0: Disable • Value2: Linger time in seconds
	<i>SECURE_EXT_CLIENT_CHLNG_RESP</i> Set with no parameter to indicate that the client uses external signature using Network Application request	Value1: Ignore
	<i>SECURE_DOMAIN_NAME_VERIFICATION</i> Set a domain name, to check in SSL client connection	Value1: Domain name
IP	<i>MULTICAST_TTL</i> Set the time-to-live value of outgoing multicast packets for this socket	Value1: Number of hops
	<i>ADD_MEMBERSHIP</i> UDP socket, join a multicast group	<ul style="list-style-type: none"> • Value1: IPv4 multicast address to join • Value2: Multicast interface address
	<i>DROP_MEMBERSHIP</i> UDP socket, leave a multicast group	<ul style="list-style-type: none"> • Value1: IPv4 multicast address to join • Value2: Multicast interface address
	<i>RAW_RX_NO_HEADER</i> Raw socket remove IP header from received data	Value1: <ul style="list-style-type: none"> • 1: Remove header • 0: Keep header
	<i>HDRINCL</i> RAW socket only, the IPv4 layer generates an IP header when sending a packet unless this option is enabled on the socket	Value1: <ul style="list-style-type: none"> • 1: Enable • 0: Disable
	<i>RAW_IPV6_HDRINCL</i> RAW socket only, the IPv6 layer generates an IP header when sending a packet unless this option is enabled on the socket	Value1: <ul style="list-style-type: none"> • 1: Enable • 0: Disable

Table 21. AT+SetSockOpt Set Socket Options (continued)

Request:		Response:
PHY	PHY_RATE Set WLAN PHY transmit rate on RAW socket	Value1: Rate
	PHY_TX_POWER RAW socket, set WLAN PHY TX power	Value1: Power range is 1–15
	PHY_NUM_FRAMES_TO_TX RAW socket, set number of frames to transmit in transceiver mode	Value1: Number of frames
	PHY_PREAMBLE RAW socket, set WLAN PHY preamble for long or short	Value1: Preamble value
	PHY_TX_INHIBIT_THRESHOLD RAW socket, set WLAN TX inhibit threshold (CCA).	Value1: Threshold value: <ul style="list-style-type: none"> • MIN • LOW • DEFAULT • MED • HIGH • MAX
	PHY_TX_TIMEOUT RAW socket, changes the TX timeout (lifetime) of transceiver frames	Value1: Time in milliseconds, maximum value is 10 ms
	PHY_ALLOW_ACKS RAW socket, enable sending ACKs in transceiver mode	Value1: <ul style="list-style-type: none"> • 1: Enable • 0: Disable

Table 22. AT+GetSockOpt Get Socket Options

Request:	Response:
AT+GetSockOpt = [sd],[level],[option]	+GetSockOpt: [value1],...,[valueX] OK
Arguments: <ul style="list-style-type: none"> • sd: Socket handle • level: Defines the protocol level for this option (see Table 21) • option: Defines the option name to interrogate (see Table 21) 	Arguments: value1,...,valueX (see the AT+SetSockOpt command in Table 21)

Table 23. AT+Recv Read Data From TCP Socket

Request:	Response:
AT+Recv = [sd],[format],[length]	OK +Recv: [sd],[format],[length],[data]
Arguments: <ul style="list-style-type: none"> • sd: Socket handle • format: Data format: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • length: Maximum number of bytes to receive 	

Table 24. AT+RecvFrom Read Data From Socket

Request:	Response:
AT+RecvFrom = [sd],[family],[port],[addr],[format],[length]	OK +RecvFrom: [sd],[format],[length],[data]
Arguments: <ul style="list-style-type: none"> • sd: Socket handle • family: internet protocol <ul style="list-style-type: none"> – INET: For network protocol IPv4 – INET6: For network protocol IPv6 • port: Address port (16 bits) • addr: internet address (32 bits) • format: Data format: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • length: Maximum number of bytes to receive 	

Table 25. AT+Send Write Data to TCP Socket

Request:	Response:
AT+Send = [sd],[format],[length],[data]	OK
Arguments: <ul style="list-style-type: none"> • sd: Socket handle • format: Data format: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • length: Number of bytes to send • data: Data to send 	

Table 26. AT+SendTo Write Data to Socket

Request:	Response:
AT+SendTo = [sd],[family],[port],[addr],[format],[length],[data]	OK
Arguments: <ul style="list-style-type: none"> • sd: Socket handle • family: internet protocol: <ul style="list-style-type: none"> – INET: For network protocol IPv4 – INET6: For network protocol IPv6 • port: Address port (16 bits) • addr: internet address (32 bits) • format: Data format: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • length: Maximum number of bytes to receive • data: Data to send 	

6.3 WLAN Commands

Table 27. AT+WlanConnect Connect to WLAN Network as a Station

Request:	Response:
AT+WlanConnect = [SSID],[BSSID],[SecurityType],[SecurityKey],[SecurityExtUser],[SecurityExtAnonUser],[SecurityExtEapMethod]	OK
Arguments: <ul style="list-style-type: none"> • SSID: Name of the Access Point • BSSID: Access Point MAC address (Optional) • SecurityType: Security type: <ul style="list-style-type: none"> – OPEN – WEP – WEP_SHARED – WPA_WPA2 – WPA_ENT – WPS_PBC – WPS_PIN • SecurityKey: Password (Optional in case it is not needed) • SecurityExtUser: Enterprise user name parameters (Ignored in case WPA_ENT was not selected) • SecurityExtAnonUser: Enterprise anonymous user name parameters (Ignored in case WPA_ENT was not selected) • SecurityExtEapMethod: Extensible Authentication Protocol (Ignored in case WPA_ENT was not selected): <ul style="list-style-type: none"> – TLS – TTLS_TLS – TTLS_MSCHAPv2 – TTLS_PSK – PEAP0_TLS – PEAP0_MSCHAPv2 – PEAP0_PSK – PEAP1_TLS – PEAP1_PSK 	

Table 28. AT+WlanDisconnect Disconnect the Connection

Request:	Response:
AT+WlanDisconnect	OK
Arguments: none	

Table 29. AT+WlanProfileAdd Add Profile

Request:	Response:
AT+WlanProfileAdd = [SSID],[BSSID],[SecurityType],[SecurityKey],[SecurityExtUser],[SecurityExtAnonUser],[SecurityExtEapMethod],[Priority]	+WlanProfileAdd: [index] OK
Arguments: <ul style="list-style-type: none"> • SSID: Name of the Access Point • BSSID: Access Point MAC address (Optional) • SecurityType: Security type: <ul style="list-style-type: none"> – OPEN – WEP – WEP_SHARED – WPA_WPA2 – WPA_ENT – WPS_PBC – WPS_PIN • SecurityKey: Password (Optional in case it is not needed) • SecurityExtUser: Enterprise user name parameters (Ignored in case WPA_ENT was not selected) • SecurityExtAnonUser: Enterprise anonymous user name parameters (Ignored in case WPA_ENT was not selected) • SecurityExtEapMethod: Extensible Authentication Protocol (Ignored in case WPA_ENT was not selected): <ul style="list-style-type: none"> – TLS – TTLS_TLS – TTLS_MSCHAPv2 – TTLS_PSK – PEAP0_TLS – PEAP0_MSCHAPv2 – PEAP0_PSK – PEAP1_TLS – PEAP1_PSK • Priority: Profile priority: <ul style="list-style-type: none"> – Lowest priority: 0 – Highest priority: 15 	Arguments: index: Profile stored index

Table 30. AT+WlanProfileGet Get Profile

Request:	Response:
AT+WlanProfileGet = [index]	+WlanProfileGet: [SSID],[BSSID],[SecurityType],[SecurityExtUser],[SecurityExtAnonUser],[SecurityExtEapMethod],[priority] OK
Arguments: index: Profile stored index received from +WlanProfileAdd	Arguments: See the AT+WlanProfileAdd command in Table 29 .

Table 31. AT+WlanProfileDel Delete Profile

Request:	Response:
AT+ WlanProfileDel = [index]	OK
Arguments: index: Number of profile to delete received from +WlanProfileAdd To delete all profiles, use index = 0xFF	

Table 32. AT+WlanPolicySet Set Policy Values

Request:			Response:
AT+WlanPolicySet = [Type],[Option],[Value]			OK
Type	Option	Value	
CONNECTION Defines options available to connect to the AP (Options could be set as bit masked). No option selected = disable all	<i>Auto</i> Reconnect to one of the stored profiles each time the connection fails or the device is rebooted	Ignore	
	<i>Fast</i> Establish a fast connection to AP	Ignore	
	<i>P2P</i> Automatically connect to the first P2P device available	Ignore	
	<i>Auto_Provisioning</i> Start the provisioning process after a long period of disconnection when profiles exist	Ignore	
SCAN Defines system scan time interval. An interval is 10 minutes. After settings scan interval, an immediate scan is activated	<i>Hidden_SSID</i>	Scan interval in seconds	
	<i>No_Hidden_SSID</i>	Scan interval in seconds	
	<i>Disable_Scan</i>	Ignore	
PM Defines a power management policy for Station mode	<i>Normal</i>	Ignore	
	<i>Low_Latency</i>	Ignore	
	<i>Low_Power</i>	Ignore	
	<i>Always_On</i>	Ignore	
	<i>Long_Sleep</i>	Maximum sleep time in milliseconds	
P2P Defines P2P negotiation policy parameters for P2P role	<ul style="list-style-type: none"> • CLIENT Indicates that the device is forced to be CLIENT • GROUP_OWNER Indicates that the device is forced to be P2P GO • NEGOTIATE Indicates that the device can be either CLIENT or GO, depending on the Wi-Fi Direct® negotiation tiebreaker 	<ul style="list-style-type: none"> • ACTIVE When the remote peer is found after the discovery process, the device immediately sends the negotiation request to the peer device. • PASSIVE When the remote peer is found after the discovery process, the device passively waits for the peer to start the negotiation, and only responds after. • RAND_BACKOFF When the remote peer is found after the discovery process, the device triggers a random timer (from 1 to 6 seconds). During this period, the device passively waits for the peer to start the negotiation. If the timer expires without negotiation, the device immediately sends the negotiation request to the peer device. 	

Table 33. AT+WlanPolicyGet Get Policy Values

Request:	Response:
AT+WlanPolicyGet = [Type]	+WlanPolicyGet: [Option],[Value] OK
Arguments: <ul style="list-style-type: none"> • Type: Type of policy. The options are: <ul style="list-style-type: none"> – CONNECTION Get connection policy – SCAN Get scan policy – PM Get power management policy – P2P Get P2P policy 	Arguments: <ul style="list-style-type: none"> • Option: See the AT+WlanPolicySet command in Table 32 • Value: See the AT+WlanPolicySet command in Table 32

Table 34. AT+WlanScan Gets the WLAN Scan Operation Results

Request:	Response:
AT+WlanScan = [Index],[Count]	+WlanScan: [SSID],[BSSID],[RSSI],[Channel],[Security_Type],[Hidden_SSID], [Cipher],[Key_Mgmt]; OK
Arguments: <ul style="list-style-type: none"> • Index: Starting index identifier (range 0–29) for getting scan results. • Count: How many entries to fetch; maximum is 30 	Arguments: <ul style="list-style-type: none"> • SSID: Wireless LAN identifier • BSSID: MAC address of the wireless access point • Channel • RSSI: Relative received signal strength in a wireless environment • Security_Type: <ul style="list-style-type: none"> – OPEN – WEP – WPA – WPA2 – WPA_WPA2 • Hidden_SSID: <ul style="list-style-type: none"> – 1: Hidden – 0: Not hidden • Cipher: <ul style="list-style-type: none"> – None – WEP40 – WEP104 – TKIP – CCMP – TKIP_CCMP • Key_Mgmt: <ul style="list-style-type: none"> – None – 802_1_X – PSK

Table 35. AT+WlanSetMode WLAN Set Mode

Request:	Response:
AT+WlanSetMode = [Mode]	OK
Arguments: <ul style="list-style-type: none"> Mode: WLAN mode to start the device: <ul style="list-style-type: none"> STA: For WLAN station mode AP: For WLAN Access Point mode P2P: For WLAN P2P mode 	

Table 36. AT+WlanSet Setting WLAN Configurations

Request:	Response:	
AT+WlanSet = [ID],[Option],[Value1],...,[ValueX]	OK	
ID	Option	
	Value	
AP	<i>SSID</i> Set SSID for AP mode	String up to 32 characters
	<i>CHANNEL</i> Set channel for AP mode	Channel in the range of [1–11]
	<i>HIDDEN_SSID</i> Set Hidden SSID Mode for AP mode	<ul style="list-style-type: none"> 0: Disabled 1: Send empty (length = 0) SSID in beacon and ignore probe request for broadcast SSID 2: Clear SSID (ASCII 0), but keep the original length (this may be required with some clients that do not support empty SSID) and ignore probe requests for broadcast SSID
	<i>SECURITY</i> Set Security type for AP mode	<ul style="list-style-type: none"> OPEN: Open security WEP: WEP security WPA_WPA2: WPA security
	<i>PASSWORD</i> Set Password for AP mode (for WEP or for WPA)	Password for WPA: 8–63 characters Password for WEP: 5 or 13 characters (ASCII)
	<i>MAX_STATIONS</i> Set Max AP stations	1...4 Note: can be less than the number of currently connected stations
	<i>MAX_STA_AGING</i> Set Max station aging time	Number of seconds
	<i>ACCESS_LIST_MODE</i> Set AP access list mode	<ul style="list-style-type: none"> DISABLE DENY_LIST: Set Black List Mode
	<i>ACCESS_LIST_ADD_MAC</i> Add MAC address to the AP access list	MAC address: 6 characters
	<i>ACCESS_LIST_DEL_MAC</i> Delete MAC address from the AP access list	MAC address: 6 characters
<i>ACCESS_LIST_DEL_IDX</i> Delete MAC address from index in the AP access list	Index	

Table 36. AT+WlanSet Setting WLAN Configurations (continued)

Request:		Response:
GENERAL	<i>COUNTRY_CODE</i> Set Country Code for AP mode	Two characters country code
	<i>STA_TX_POWER</i> Set STA mode TX power level	Number between 0–15, as dB offset from maximum power (0 sets maximum power)
	<i>AP_TX_POWER</i> Set AP mode TX power level	Number between 0–15, as dB offset from maximum power (0 sets maximum power)
	<i>INFO_ELEMENT</i> Set Info Element for AP mode	<ul style="list-style-type: none"> • Value1: Index of the info element • Value2: Role: <ul style="list-style-type: none"> – AP – P2P • Value3: Info element ID • Value4: Organization unique ID first Byte • Value5: Organization unique ID second Byte • Value6: Organization unique ID third Byte • Value7: Info element (maximum 252 chars)
	<i>SCAN_PARAMS</i> Set scan parameters	<ul style="list-style-type: none"> • Value1: Channel mask • Value2: RSSI threshold
	<i>SUSPEND_PROFILES</i> Set suspended profiles mask	Suspended bitmask
	<i>DISABLE_ENT_SERVER_AUTH</i> This option enables to skip server authentication and is valid for one use, when manually connection to an enterprise network	<ul style="list-style-type: none"> • 1: Disable the server authentication • 0: Enable
P2P	<i>DEV_TYPE</i> Set P2P Device type	Device type is published under P2P I.E (maximum length of 17 characters)
	<i>CHANNEL_N_REGS</i> Set P2P Channels	<ul style="list-style-type: none"> • Value1: Listen channel (either 1/6/11 for 2.4 GHz) • Value2: Listen regulatory class (81 for 2.4 GHz) • Value3: Operating channel (channel 1, 6, or 11 for 2.4 GHz) • Value4: Operating regulatory class (81 for 2.4 GHz)
RX_FILTER	<i>STATE</i> Enable or disable filters	Filter Bitmap array (16 bytes in format xx:xx)
	<i>SYS_STATE</i> Enable or disable system filters	Filter Bitmap array (4 bytes in format xx:xx)
	<i>REMOVE</i> Remove filters	Filter Bitmap array (16 bytes in format xx:xx)
	<i>STORE</i> Save the filters as persistent	null

Table 37. AT+ WlanGet Getting WLAN Configurations

Request:		Response:
AT+WlanGet = [ID],[Option]		+WlanGet: [Value1],...,[ValueX] OK
Arguments:		Arguments: See the AT+WlanSet command in Table 36 .
ID	Option	
AP	<i>SSID</i> Get SSID for AP mode	
	<i>CHANNEL</i> Get channel for AP mode	
	<i>HIDDEN_SSID</i> Get Hidden SSID Mode for AP mode	
	<i>SECURITY</i> Get Security type for AP mode	
	<i>PASSWORD</i> Get Password for AP mode (for WEP or for WPA)	
	<i>MAX_STATIONS</i> Get Max AP allowed stations	
	<i>MAX_STA_AGING</i> Get AP aging time in seconds	
	<i>ACCESS_LIST_NUM_ENTRIES</i> Get AP access list number of entries	
<i>ACCESS_LIST</i> Get the AP access list from start index	The start index in the access list	
GENERAL	<i>COUNTRY_CODE</i> Get Country Code for AP mode	
	<i>STA_TX_POWER</i> Get STA mode TX power level	
	<i>AP_TX_POWER</i> Get AP mode TX power level	
	<i>SCAN_PARAMS</i> Get scan parameters	
P2P	<i>CHANNEL_N_REGS</i> Get P2P Channels	
RX_FILTER	<i>STATE</i> Retrieves the filters enable/disable status	
	<i>SYS_STATE</i> Retrieves the system filters enable or disable status	

Table 37. AT+ WlanGet Getting WLAN Configurations (continued)

Request:		Response:
Connection	Ignore	<ul style="list-style-type: none"> • Value1: Role: <ul style="list-style-type: none"> – sta – ap – p2p • Value2: Status: <ul style="list-style-type: none"> – disconnected – station_connected – p2pcl_connected – p2pgo_connected – ap_connected_stations • Value3: Security: <ul style="list-style-type: none"> – open – wep – wpa_wpa2 – wps_pbc – wps_pin – wpa_ent – wep_shared • Value4: SSID Name • Value5: BSSID • Value6: Device name (relevant to P2P Client only)

6.4 File System Commands

Table 38. AT+FileOpen Open File in Storage Device

Request:	Response:
AT+FileOpen = [Filename],[Options],[File size]	+FileOpen:[FileID],[Secure Token] OK
Arguments: <ul style="list-style-type: none"> • Filename: Full path File Name • Options: Bitmask depend in option: <ul style="list-style-type: none"> – READ: Read a file (no bitmask) – WRITE: Open for write for an existing file (optionally bitmask with CREATE) – CREATE: Open for creating a new file (optionally bitmask with WRITE or OVERWRITE) – OVERWRITE: Opens an existing file (optionally bitmask with CREATE) /* Creation flags bitmask with CREATE */ – CREATE_FAILSAFE: Fail safe – CREATE_SECURE: Secure file – CREATE_NOSIGNATURE : Relevant to secure file only – CREATE_STATIC_TOKEN: Relevant to secure file only – CREATE_VENDOR_TOKEN: Relevant to secure file only – CREATE_PUBLIC_WRITE: Relevant to secure file only, the file can be opened for write without Token – CREATE_PUBLIC_READ: Relevant to secure file only, the file can be opened for read without Token • File size: Maximum file size is defined in bytes (mandatory only for the CREATE option and is ignored for other options) 	

Table 39. AT+FileClose Close File in Storage Device

Request:	Response:
AT+FileClose = [FileID],[CertificateFileName],[Signature]	OK
Arguments: <ul style="list-style-type: none"> • FileID: Assigned from AT+FileOpen • CertificateFileName: Certificate file with full path (Optional) • Signature: The signature is SHA-1, the certificate chain may include SHA-256 (Optional) 	

Table 40. AT+FileCtl Controls Various File System Operations

Request:				Response:	
AT+FileCtl = [Command],[Secure_Token],[Filename],[Data]				+FileCtl:[NewSecureToken],[OutputData] OK	
Arguments:				Arguments:	
Command	Token	Filename	Data	Token	Output Data
<i>RESTORE</i> Return to factory default	Ignore	Ignore	<i>FACTORY_IMAGE</i> The system will be back to the production image. <i>FACTORY_DEFAULT</i> Return to factory default	Ignore	Ignore
<i>ROLLBACK</i> Roll-back file	Token assigned from AT+FileOpen	Filename to roll back	Ignore	New secure token	Ignore
<i>COMMIT</i> Commit file	Token assigned from AT+FileOpen	Filename to commit	Ignore	New secure token	Ignore
<i>RENAME</i> Rename file	Token assigned from AT+FileOpen	Filename to rename	New file name	Ignore	Ignore
<i>GET_STORAGE_INFO</i> Get storage information	Ignore	Ignore	Ignore	Ignore	<ul style="list-style-type: none"> • DeviceBlockSize • DeviceBlocks Capacity • NumOfAllocated Blocks • NumOfReserved Blocks • NumOfReserved BlocksFor Systemfiles • LargestAllocated GapInBlocks • NumOfAvailable Blocks • ForUserFiles • MaxFsFiles • IsDevelopment FormatType • Bundlestate • MaxFsFilesReservedForSysFiles • ActualNumOf UserFiles • ActualNumOf SysFiles NumOfAlerts • NumOfAlerts Threshold • FATWrite Counter
<i>BUNDLE_ROLLBACK</i> Roll back a bundle	Ignore	Ignore	Ignore	Ignore	Ignore
<i>BUNDLE_COMMIT</i> Commit a bundle	Ignore	Ignore	Ignore	Ignore	Ignore

Table 41. AT+FileDel Delete File From Storage Device

Request:	Response:
AT+FileDel = [FileName],[SecureToken]	OK
Arguments: <ul style="list-style-type: none"> • FileName: Full path File Name • SecureToken: Token assigned from AT+FileOpen (optional) 	

Table 42. AT+FileGetFilelist Get a List of Files

Request:	Response:
AT+FileGetFileList	+FileGetFileList: [FileName],[FileMaxSize],[Properties],[FileAllocatedBlocks] OK
Arguments:	Arguments: <ul style="list-style-type: none"> • FileName: File name • FileMaxSize: Maximum file size • Properties: Info flag bitmask • FileAllocatedBlocks: Allocated blocks

Table 43. AT+FileGetInfo Get Information About a File

Request:	Response:
AT+FileGetInfo = [FileName],[SecureToken]	+FileGetInfo: [Flags],[File Size],[Allocated Size], [Tokens],[Storage Size],[Write Counter] OK
Arguments: <ul style="list-style-type: none"> • FileName: Full path file name • SecureToken: token assigned from AT+FileOpen (optional) 	

Table 44. AT+FileRead Read a Block of Data From a File in Storage Device

Request:	Response:
AT+FileRead = [FileID],[Offset],[Format],[Length]	+FileRead:[format],[NumberOfReadBytes],[ReceivedData] OK
Arguments: <ul style="list-style-type: none"> • FileID: Assigned from AT+FileOpen • Offset: Offset to specific read block • Format: Data format: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • Length: Number of bytes to read 	

Table 45. AT+FileWrite Write Block of Data to a File in Storage Device

Request:	Response:
AT+FileWrite = [FileID],[Offset],[Format],[Length],[Data]	+FileWrite:[NumberOfWrittenBytes] OK
Arguments: <ul style="list-style-type: none"> • FileID: Assigned from AT+FileOpen • Offset: Offset to specific block to be written • Format: Data format: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • Length: Number of bytes to write • Data: Transmitted data to the storage device 	

6.5 Network Application Commands

Activate networking applications, such as:

- HTTP Server
- DHCP Server
- Ping
- DNS
- mDNS

Table 46. AT+NetAPPStart Starts a Network Application

Request:	Response:
AT+NetAPPStart = [APP Bitmap]	OK
Arguments: <ul style="list-style-type: none"> • APP Bitmap: Application bitmap, could be one or a combination of the following with OR (" ") between them: <ul style="list-style-type: none"> – HTTP_SERVER – DHCP_SERVER – MDNS – DNS_SERVER 	

Table 47. AT+NetAPPStop Stops a Network Application

Request:	Response:
AT+NetAPPStop = [APP Bitmap]	OK
Arguments: <ul style="list-style-type: none"> • APP Bitmap: Application bitmap, could be one or a combination of the following with OR (" ") between them: <ul style="list-style-type: none"> – HTTP_SERVER – DHCP_SERVER – MDNS – DNS_SERVER 	

Table 48. AT+NetAPPGetHostByName Get Host IP by Name

Request:	Response:
AT+NetAPPGetHostByName = [HostName],[Family]	OK +NetAPPGetHostByName: [HostName],[Host IP address]
Arguments: <ul style="list-style-type: none"> • HostName • Family: Protocol Family: <ul style="list-style-type: none"> – INET: For network protocol IPv4 – INET6: For network protocol IPv6 	Arguments: <ul style="list-style-type: none"> • HostName • Host IP address: IP address according to the family (IPv4 or IPv6)

Table 49. AT+NetAPPGetHostByService Get Host IP by Service

Request:	Response:
AT+NetAPPGetHostByService = [ServiceName],[Family]	OK +NetAPPGetHostByService: [ServiceName],[Port],[HostIPAddress],[Text]
Arguments: <ul style="list-style-type: none"> • ServiceName: Service name can be full or partial • Family: Protocol Family: <ul style="list-style-type: none"> – INET: For network protocol IPv4 – INET6: For network protocol IPv6 	Arguments: <ul style="list-style-type: none"> • ServiceName • Port: Service port • HostIPAddress: Host IP address (IPv4 or IPv6) • Text: Text of the service full or partial

Table 50. AT+NetAPPSet Setting Network Application Configurations

Request:			Response:
AT+NetAPPSet = [App ID],[Option],[Value1],...,[ValueX]			OK
Arguments:			
App ID	Option	Values	
DHCP_SERVER	BASIC	<ul style="list-style-type: none"> • Value1: Lease time (in seconds) of the IP Address • Value2: First IP Address for allocation • Value3: Last IP Address for allocation 	
HTTP_SERVER	PRIM_PORT_NUM	Value1: port number	
	AUTH_CHECK	Value1: <ul style="list-style-type: none"> • 1: Authentication enable • 0: Authentication disable 	
	AUTH_NAME	Value1: Authentication name (maximum length is 20 bytes)	
	AUTH_PASSWORD	Value1: Authentication password (maximum length is 20 bytes)	
	AUTH_REALM	Value1: Authorization realm (maximum length is 20 bytes)	
	ROM_PAGES_ACCESS	Value1: <ul style="list-style-type: none"> • 1: Access enable • 0: Access disable 	
	SECOND_PORT_NUM	Value1: port number	
	SECOND_PORT_EN	Value1: <ul style="list-style-type: none"> • 1: Enable • 0: Disable 	
	PRIM_PORT_SEC_EN	Value1: <ul style="list-style-type: none"> • 1: Enable • 0: Disable 	
	PRIV_KEY_FILE	Value1: File name (maximum length is 96 bytes)	
	DEV_CERT_FILE	Value1: File name (maximum length is 96 bytes)	
	CA_CERT_FILE	Value1: File name (maximum length is 96 bytes)	
	TMP_REGISTER_SERVICE	Value1: Service name for MDNS (maximum length is 80 bytes)	
TMP_UNREGISTER_SERVICE	Value1: Service name for MDNS (maximum length is 80 bytes)		

Table 50. AT+NetAPPSet Setting Network Application Configurations (continued)

Request:		Response:
<i>MDNS</i>	<i>CONT_QUERY</i>	Value1: Service name (maximum length is 80 bytes)
	<i>QEVETN_MASK</i>	Value1: Event mask: <ul style="list-style-type: none"> • ipp • deviceinfo • http • https • workstation • guid • h323 • ntp • objective • rdp • remote • rtsp • sip • smb • soap • ssh • telnet • tftp • xmpp • raop
	<i>TIMING_PARAMS</i>	<ul style="list-style-type: none"> • Value1: Period in ticks (100 ticks = 1 second) • Value2: Repetitions • Value3: Telescopic factor • Value4: Retransmission interval • Value5: Maximum period interval • Value6: Maximum time
<i>DEVICE</i>	<i>URN</i>	Value1: device name (maximum length is 33 bytes)
	<i>DOMAIN</i>	Value1: domain name (maximum length is 63 bytes)
<i>DNS_CLIENT</i>	<i>TIME</i>	<ul style="list-style-type: none"> • Value1: Maximum response time in milliseconds • Value2: Number of retries

Table 51. AT+NetAPPGet Getting Network Applications Configurations

Request:		Response:
AT+NetAPPGet = [App ID],[Option]		+NetAPPGet: [return values] OK
Arguments:		Arguments: See AT+NetAPPSet command values
App ID	Option	
DHCP_SERVER	BASIC	
HTTP_SERVER	PRIM_PORT_NUM	
	AUTH_CHECK	
	AUTH_NAME	
	AUTH_PASSWORD	
	AUTH_REALM	
	ROM_PAGES_ACCESS	
	SECOND_PORT_NUM	
	SECOND_PORT_EN	
	PRIM_PORT_SEC_EN	
	MDNS	CONT_QUERY
QEVETN_MASK		
TIMING_PARAMS		
DEVICE	URN	
	DOMAIN	
DNS_CLIENT	TIME	

Table 52. AT+NetAPPSend Sends Network Application Response or Data Following a Network Application Request Event

Request:	Response:
AT+NetAPPSend = [Handle],[Flags],[Format],[Length],[Data]	OK
Arguments: <ul style="list-style-type: none"> • Handle: Handle to send the data to. Should match the handle received in the Network Application request event • Flags: Bitmask: <ul style="list-style-type: none"> – CONTINUATION: More data will arrive in subsequent calls to AT+NetAPPSend – METADATA: Define data as metadata, otherwise data is payload – ACCUMULATION: The network processor should accumulate the data chunks and will process it when it is completely received • Format: Data format: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • Length: Number of bytes to send • Data: Data to send. Can be just data payload or metadata (depends on flags) 	

Table 53. AT+NetAPPRecv Receives Data From the Network Processor Following a Network Application Response Event

Request:	Response:
AT+NetAPPRecv = [Handle],[Format],[Length]	OK +NetAPPRecv:[Handle],[Flags],[Format],[Length],[Data]
Arguments: <ul style="list-style-type: none"> • Handle: Handle to receive data from. Should match the handle receive in the Network Application request event • Format: Data format: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • Length: Number of bytes to receive 	Arguments: <ul style="list-style-type: none"> • Handle • Flags: Can have the following value: • CONTINUATION: More data is pending in the network processor. Application should continue reading the data by calling AT+NetAPPRecv again • Format: Data format: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format • Length: Number of bytes received • Data: Data received

Table 54. AT+NetAPPPing Send Ping to Network Hosts

Request:	Response:
AT+NetAPPPing = [Family],[Destination],[Size],[Delay],[Timeout],[Max],[Flags]	OK +NetAPPPing: [PacketsSent],[PacketsReceived],[RoundTime]
Arguments: <ul style="list-style-type: none"> • Family: <ul style="list-style-type: none"> – INET: For network protocol IPv4 – INET6: For network protocol IPv6 • Destination: Destination IP address. For stopping an ongoing ping activity, set destination to 0 • Size: Size of ping, in bytes • Delay: Delay between pings, in milliseconds • Timeout: Timeout for every ping in milliseconds • Max: Maximum number of ping requests <ul style="list-style-type: none"> – 0: Forever • Flags: <ul style="list-style-type: none"> – Set to 0: Ping reports back once all requested pings are done – Set to 1: Ping reports back after every ping – Set to 2: Ping stops after the first successful ping and reports back for the successful ping, as well as any preceding failed pings 	

Table 55. AT+NetAPPGetServiceList Get Service List

Request:	Response:
AT+NetAPPGetServiceList = [IndexOffset],[MaxServiceCount],[Flags]	+NetAPPGetServiceList:[ServiceInfo1];...:[ServiceInfoX] OK
Arguments: <ul style="list-style-type: none"> • IndexOffset: The start index in the peer cache that from it the first service is returned • MaxServiceCount: The maximum services that can be returned if existed or if not exceed the maximum index in the peer cache • Flags: Which service to use (means which types of service to fill): <ul style="list-style-type: none"> – FULL_IPV4_WITH_TEXT – FULL_IPV4 – SHORT_IPV4 – FULL_IPV6_WITH_TEXT – FULL_IPV6 – SHORT_IPV6 	Arguments: ServiceInfo: Depends on flag type: <ul style="list-style-type: none"> • SHORT_IPV4 <ul style="list-style-type: none"> – ip – port • FULL_IPV4 • FULL_IPV6 <ul style="list-style-type: none"> – ip – port – service name – service host name • FULL_IPV4_WITH_TEXT • FULL_IPV6_WITH_TEXT <ul style="list-style-type: none"> – ip – port – service name – service host name – service text

Table 56. AT+NetAPPRegisterService Register a New mDNS Service

Request:	Response:
AT+NetAPPRegisterService = [ServiceName],[Text],[Port],[TTL],[Options]	OK
Arguments: <ul style="list-style-type: none"> • ServiceName: The service name • Text: The description of the service • Port: The port on this target host port • TTL: The TTL of the service • Options: Bitwise parameters: <ul style="list-style-type: none"> – <i>IS_UNIQUE_BIT</i>: Service is unique per interface (means that the service needs to be unique) – <i>IPV6_IPV4_SERVICE</i>: Add this service to IPv6 interface, if exist (default is IPv4 service only) – <i>IPV6_ONLY_SERVICE</i>: Add this service to IPv6 interface, but remove it from IPv4 (only IPv6 is available) – <i>UPDATE_TEXT</i>: For update text fields (without reregistering the service) – <i>IS_NOT_PERSISTENT</i>: For setting a nonpersistent service 	

Table 57. AT+NetAPPUnRegisterService Unregister mDNS Service

Request:	Response:
AT+NetAPPUnRegisterService = [ServiceName],[Options]	OK
Arguments: <ul style="list-style-type: none"> • ServiceName: Full service name • Options: Bitwise parameters: <ul style="list-style-type: none"> – <i>IS_UNIQUE_BIT</i>: Service is unique per interface (means that the service needs to be unique) – <i>IPV6_IPV4_SERVICE</i>: Add this service to IPv6 interface, if exist (default is IPv4 service only) – <i>IPV6_ONLY_SERVICE</i>: Add this service to IPv6 interface, but remove it from IPv4 (only IPv6 is available) – <i>UPDATE_TEXT</i>: For update text fields (without reregistering the service) – <i>IS_NOT_PERSISTENT</i>: For setting a nonpersistent service 	

6.6 Network Configuration Commands

The Network Configuration Commands control the configuration of the device addresses (that is, IP and MAC addresses).

Table 58. AT+NetCfgSet Setting Network Configurations

Request:			Response:
AT+NetCfgSet = [ConfigId],[ConfigOpt],[Value1],...,[ValueX]			OK
Arguments:			
ConfigId	ConfigOpt	Value	
<i>IF</i>	<i>STATE</i> Enable or disable modes (bitmask)	<ul style="list-style-type: none"> • <i>IPV6_STA_LOCAL</i>: Enable ipv6 local • <i>IPV6_STA_GLOBAL</i>: Enable ipv6 global • <i>DISABLE_IPV4_DHCP</i>: Disable ipv4 DHCP • <i>IPV6_LOCAL_STATIC</i>: Enable ipv6 local static • <i>IPV6_LOCAL_STATELESS</i>: Enable ipv6 local stateless • <i>IPV6_LOCAL_STATEFUL</i>: Enable ipv6 local stateful • <i>IPV6_GLOBAL_STATIC</i>: Enable ipv6 global static • <i>IPV6_GLOBAL_STATEFUL</i>: Enable ipv6 global stateful • <i>DISABLE_IPV4_LLA</i>: Disable LLA feature • <i>ENABLE_DHCP_RELEASE</i>: Enables DHCP release • <i>IPV6_GLOBAL_STATELESS</i>: Enable ipv6 global stateless • <i>DISABLE_FAST_RENEW</i>: Fast renew disabled 	
<i>SET_MAC_ADDR</i> Setting MAC address to the Device	Ignore value	New MAC address	
<i>IPV4_STA_ADDR</i> Setting IP address	<i>STATIC</i> Setting a static IP address	<ul style="list-style-type: none"> • Value1: IP address • Value2: Subnet mask • Value3: Default gateway address • Value4: DNS server address 	
	<i>DHCP</i> Setting IP address by DHCP	Ignore value	
	<i>DHCP_LLA</i> Setting DHCP LLA	Ignore value	
	<i>RELEASE_IP_SET</i> Setting release IP before disconnect enables sending a DHCP release frame to the server	Ignore value	
	<i>RELEASE_IP_OFF</i> Setting release IP before disconnect disables sending a DHCP release frame to the server	Ignore value	
<i>IPV4_AP_ADDR</i> Setting a static IP address to the device working in AP mode	<i>STATIC</i> Setting a static IP address	<ul style="list-style-type: none"> • Value1: IP address • Value2: Subnet mask • Value3: Default gateway address • Value4: DNS server address 	

Table 58. AT+NetCfgSet Setting Network Configurations (continued)

Request:			Response:
IPV6_ADDR_LOCAL	<i>STATIC</i> Setting a IPv6 Local static address	IP address	
	<i>STATELESS</i> Setting a IPv6 Local stateless address	Ignore value	
	<i>STATEFUL</i> Setting a IPv6 Local stateful address	Ignore value	
IPV6_ADDR_GLOBAL	<i>STATIC</i> Setting a IPv6 Global static address Value1 : IP address Value2: DNS Server IP <i>STATEFUL</i>	<ul style="list-style-type: none"> • Value1: IP address • Value2: DNS Server IP 	
	<i>STATEFUL</i> Setting a IPv6 Global stateful address	Ignore value	
<i>AP_STATION_DISCONNECT</i> Disconnect AP station by MAC address	Ignore value	AP MAC address	
<i>IPV4_DNS_CLIENT</i> Set secondary DNS address	Ignore value	Secondary DNS Server address	

Table 59. AT+NetCfgGet Getting Network Configurations

Request:	Response:
AT+NetCfgGet = [ConfigId]	+NetCfgGet:[Value1],...,[ValueX] OK
Arguments: ConfigId: Configuration ID:	Arguments:
<i>GET_MAC_ADDR</i> Get the device MAC address	Value1: MAC address
<i>IPV4_STA_ADDR</i> Get IP address from WLAN station or P2P client	<ul style="list-style-type: none"> • Value1: Address option: <ul style="list-style-type: none"> – DHCP – DHCP_LLA – STATIC • Value2: Address • Value3: Subnet mask • Value4: Gateway • Value5: DNS
<i>IPV4_AP_ADDR</i> Get static IP address for AP or P2P go	
<i>IF</i> Get interface bitmap	Value1: State (bitmask): <ul style="list-style-type: none"> • ipv6_sta_local • ipv6_sta_global • disable_ipv4_dhcp • ipv6_local_static • ipv6_local_stateless • ipv6_local_stateful • ipv6_global_static • ipv6_global_stateful • disable_ipv4_lls • enable_dhcp_release • ipv6_global_stateless • disable_fast_renew

Table 59. AT+NetCfgGet Getting Network Configurations (continued)

Request:	Response:
<i>IPV6_ADDR_LOCAL</i> Get IPV6 Local address	<ul style="list-style-type: none"> • Value1: Address option: <ul style="list-style-type: none"> – stateless – stateful – STATIC • Value2: Address
<i>IPV6_ADDR_GLOBAL</i> Get IPV6 Global address	
<i>AP_STATIONS_CONNECTED</i> Get AP number of connected stations	Value1: Number of connected stations
<i>AP_STATIONS_INFO</i> Get AP full list of connected stations	[address1],[MAC address1],[name1]; ...; [addressX],[MAC addressX],[nameX]
<i>IPV4_DNS_CLIENT</i> Set secondary DNS address	Value1: DNS second server address
<i>IPV4_DHCP_CLIENT</i> Get DHCP Client info	<ul style="list-style-type: none"> • Value1: Address • Value2: Subnet mask • Value3: Gateway • Value4: DNS 1 • Value5: DNS 2 • Value6: DHCP server • Value7: Lease time • Value8: Time to renew • Value9: DHCP State: <ul style="list-style-type: none"> – unknown – disabled – enabled – bound – renew – rebind

6.7 Network Utility Commands

Networking related commands and configuration.

Table 60. AT+NetUtilGet Getting Utilities Configurations

Request:		Response:
AT+NetUtilGet = [ID],[Option]		+NetUtilGet: [Value1],...,[ValueX] OK
Arguments:		Arguments:
ID Identifier of the specific "get" operation to perform	Option	Value
<i>public_key</i>	<ul style="list-style-type: none"> • 0: Binary data format • 1: Base64 data format (binary to text encoding) 	<ul style="list-style-type: none"> • Value1: Public key format: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format • Value2: Public key length (maximum length is 255 bytes or 370 bytes in base64 format) • Value3: Public key
<i>true_random</i>	Number of random numbers (maximum is 172 numbers)	List of random numbers

Table 61. AT+NetUtilCmd Performing Utilities-Related Commands

Request:		Response:
AT+NetUtilCmd = [Cmd],[Value1],...,[ValueX]		+NetUtilCmd:[Value1],...,[ValueX] OK
Arguments:		Arguments:
Cmd	Option	Value
sign_msg Create a digital signature using the ECDSA algorithm	<ul style="list-style-type: none"> • Value1: Key index: • Value2: Data format: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • Value3: Data length (maximum length is 1500 bytes) • Value4: Data 	<ul style="list-style-type: none"> • Value1: Signature format: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • Value2: Signature length (maximum length is 255 bytes) • Value3: Signature
verify_msg verify a digital signature using the ECDSA algorithm	<ul style="list-style-type: none"> • Value1: Key index • Value2: Data and signature format: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • Value3: Data length (maximum length is 1500 bytes) • Value4: Signature length • Value5: Data and signature (signature concatenate to end of data) 	Value1: Success or failure
temp_keys Create or remove a temporary ECC key pair with the SECP256R1 curve	<ul style="list-style-type: none"> • Value1: Key index • Value2: Action: <ul style="list-style-type: none"> – <i>create</i> – <i>remove</i> 	
install_op Install or uninstall a key pair in one of the crypto utilities key pair management mechanism	<ul style="list-style-type: none"> • Value1: Key index • Value2: Action: <ul style="list-style-type: none"> – <i>install</i> – <i>uninstall</i> • Value3: Key Algorithm (ignored for uninstall action): <ul style="list-style-type: none"> – <i>none</i> – <i>ec</i> • Value4: EC Named Curve identifier (optional for Key Algorithm none) (ignored for uninstall action): <ul style="list-style-type: none"> – <i>none</i> – <i>secp256r1</i> • Value5: Certification file name (ignored for uninstall action) • Value6: Key file name (ignored for uninstall action) 	

6.8 Asynchronous Events

Table 62. +EventFatalError Fatal Error Event for Inspecting Fatal Error

Response:	
+EventFatalError:[EventID],[Value1],...,[ValueX]	
Arguments:	
EventID	Value
<i>DEVICE_ABORT</i> Indicates a severe error occurred and the device stopped	<ul style="list-style-type: none"> Value1: An indication of the abort type Value2: The abort data
<i>NO_CMD_ACK</i> Indicates that the command sent to the device had no ACK	Value1: An indication of the CMD opcode
<i>CMD_TIMEOUT</i> Indicates that the command got a timeout while waiting for its asynchronous response	Value1: An indication of the asynchronous event opcode
<i>DRIVER_ABORT</i> Indicates a severe error occurred in the driver	null
<i>SYNC_LOSS</i> Indicates a sync loss with the device	null

Table 63. +EventGeneral General Asynchronous Event for Inspecting General Events

Response:	
+EventGeneral:[EventID],[Value1],...,[ValueX]	
Arguments:	
EventID	Value
<i>RESET_REQUEST</i>	<ul style="list-style-type: none"> Value1: An error code indication from the device Value2: The sender originator: <ul style="list-style-type: none"> WLAN NETCFG NETAPP SECURITY OTHER
<i>ERROR</i>	<ul style="list-style-type: none"> Value1: An error code indication from the device Value2: The sender originator

Table 64. +EventWlan WLAN Asynchronous Event

Response:	
+EventWlan:[EventID],[Value1],...,[ValueX]	
Arguments:	
EventID	Value
<i>CONNECT</i> STA connection indication event	<ul style="list-style-type: none"> Value1: SSID name Value2: BSSID
<i>P2P_CONNECT</i> P2P client connection indication event	<ul style="list-style-type: none"> Value1: SSID name Value2: BSSID Value3: Go Device Name
<i>DISCONNECT</i> STA client disconnection event	<ul style="list-style-type: none"> Value1: SSID name Value2: BSSID Value3: Reason
<i>P2P_DISCONNECT</i> P2P client disconnection event	<ul style="list-style-type: none"> Value1: SSID name Value2: BSSID Value3: Reason Value4: Go Device Name
<i>STA_ADDED</i> AP connected STA	Value1: MAC address
<i>STA_REMOVED</i> AP disconnected STA	Value1: MAC address
<i>P2P_CLIENT_ADDED</i> P2P(Go) connected P2P(Client)	<ul style="list-style-type: none"> Value1: MAC address Value2: Go Device Name Value3: Own SSID
<i>P2P_CLIENT_REMOVED</i> P2P(Go) disconnected P2P(Client)	<ul style="list-style-type: none"> Value1: MAC address Value2: Go Device Name Value3: Own SSID
<i>P2P_DEVFOUND</i>	<ul style="list-style-type: none"> Value1: Go Device Name Value2: MAC address Value3: WPS Method
<i>P2P_REQUEST</i>	<ul style="list-style-type: none"> Value1: Go Device Name Value2: MAC address Value3: WPS Method
<i>P2P_CONNECTFAIL</i> P2P only	Value1: Status
<i>PROVISIONING_STATUS</i>	Value1: Status
<i>PROVISIONING_PROFILE_ADDED</i>	<ul style="list-style-type: none"> Value1: Status Value2: SSID name

Table 65. +EventNetApp Network Application Asynchronous Event

Response:	
+EventNetApp:[EventID],[Value1],...,[ValueX]	
Arguments:	
EventID	Value
<i>IPV4_ACQUIRED</i>	<ul style="list-style-type: none"> Value1: IP address Value2: Gateway Value3: DNS
<i>IPV6_ACQUIRED</i>	<ul style="list-style-type: none"> Value1: IP address Value2: DNS
<i>ip_collision</i>	<ul style="list-style-type: none"> Value1: IP address Value2: DHCP MAC Value3: DNS
<i>IP_LEASED</i> AP or P2P go DHCP lease event	<ul style="list-style-type: none"> Value1: IP address Value2: Lease time Value3: MAC
<i>IP_RELEASED</i> AP or P2P go DHCP IP release event	<ul style="list-style-type: none"> Value1: IP address Value2: MAC Value3: Reason
<i>IPV4_LOST</i>	Value1: Status
<i>dhcp_ipv4_acquire_timeout</i>	Value1: Status
<i>IPV6_LOST</i>	Value1: IP lost

Table 66. +EventSock Socket Asynchronous Event

Response:	
+EventSock:[EventID],[Value1],...,[ValueX]	
Arguments:	
EventID	Value
<i>TX_FAILED</i>	<ul style="list-style-type: none"> Value1: sd Value2: Status
<i>ASYNC_EVENT</i>	<ul style="list-style-type: none"> Value1: sd Value2: Type: <ul style="list-style-type: none"> SSL_ACCEPT RX_FRAG_TOO_BIG OTHER_SIDE_CLOSE_SSL CONNECTED_SECURED WRONG_ROOT_CA Value3: Error value

Table 67. +EventMqtt MQTT Asynchronous Event

Response:	
+EventMqtt:[EventID],[Value1],...,[ValueX]	
Arguments:	
EventID	Value
<i>operation</i>	<ul style="list-style-type: none"> • Value1: operation ID: <ul style="list-style-type: none"> – Connack: connection acknowledge Value2: 16 bits: <ul style="list-style-type: none"> • 8 MSBs: Acknowledge Flags • 8 LSBs: return code: <ul style="list-style-type: none"> • 0: Connection Accepted • 1: Connection Refused, unacceptable protocol version • 2: Connection Refused, identifier rejected • 3: Connection Refused, Server unavailable • 4: Connection Refused, bad user name or password • 5: Connection Refused, not authorized – Puback: publish acknowledge Value2: Packet Identifier from the PUBLISH Packet that is being acknowledged – Suback: subscribe acknowledge Value2: Packet Identifier from the SUBSCRIBE Packet that is being acknowledged Value3 to ValueX: return code per topic: <ul style="list-style-type: none"> • 0: Success, Maximum QoS 0 • 1: Success, Maximum QoS 1 • 2: Success, Maximum QoS 2 • 128: Failure – Unsuback: unsubscribe acknowledge Value2: Packet Identifier from the UNSUBSCRIBE Packet that is being acknowledged
<i>recv</i>	<ul style="list-style-type: none"> • Topic: topic string • QoS: Quality of service type: <ul style="list-style-type: none"> – QoS 0 – QoS 1 – QoS 2 • Retain: <ul style="list-style-type: none"> – 0: message should not be retained – 1: message should be retained • Duplicate: <ul style="list-style-type: none"> – 0: first attempted to send the message – 1: might be re-delivery of an earlier attempt to send the message • Message Format: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • Message length: number of bytes to send • Message: message to send
<i>disconnect</i>	

6.9 MQTT Client Commands

MQTT client commands and configuration.

Table 68. AT+MqttCreate MQTT Client Create

Request:	Response:
AT+MqttCreate = [client ID],[flags],[address],[port],[method],[cipher],[private key],[Certificate],[CA],[DH key],[protocol],[blocking send],[data format]	+ MqttCreate: [index] OK
Arguments:	Arguments:
<ul style="list-style-type: none"> • client ID • flags: bitmask of the following: <ul style="list-style-type: none"> – ip4: IPv4 connection – ip6: IPv6 connection – url: Server address is an URL and not IP address – sec: Connection to server must be secure (TLS) – skip_domain_verify: skip domain name verification – skip_cert_verify: skip certificate catalog verification – skip_date_verify: skip date verification • address: server address (ip or url) • port: address port (16 bits) • method: security method (mandatory only in case of secure connection): <ul style="list-style-type: none"> – SSLV3: Security method SSL v3 – TLSV1: Security method TLS v1 – TLSV1_1: Security method TLS v1_1 – TLSV1_2: Security method TLS v1_2 – SSLV3_TLSV1_2: Use highest possible version from SSLv3–TLS 1.2 • cipher: security cipher as OR bitmask (optional), (default value: all ciphers): <ul style="list-style-type: none"> – SSL_RSA_WITH_RC4_128_SHA – SSL_RSA_WITH_RC4_128_MD5 – TLS_RSA_WITH_AES_256_CBC_SHA – TLS_DHE_RSA_WITH_AES_256_CBC_SHA – TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA – TLS_ECDHE_RSA_WITH_RC4_128_SHA – TLS_RSA_WITH_AES_128_CBC_SHA256 – TLS_RSA_WITH_AES_256_CBC_SHA256 – TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA – TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA – TLS_RSA_WITH_AES_128_GCM_SHA256 – TLS_RSA_WITH_AES_256_GCM_SHA384 – TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 – TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 – TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 – TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 – TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 – TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 – TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 – TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 – TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 	index: client handle

Table 68. AT+MqttCreate MQTT Client Create (continued)

Request:	Response:
<ul style="list-style-type: none"> • private key: private key file name (Optional) • certificate: certificate file name (Optional) • CA :certificate authority file name (mandatory only in case of secure connection) • DH key: Diffie Hellman file name (Optional) • protocol: MQTT protocol: <ul style="list-style-type: none"> – v3_1: protocol v3.1 – v3_1_1: protocol v3.1.1 • blocking send: <ul style="list-style-type: none"> – 0: do not wait for server response – 1: wait for response • data format: set format globally to all MQTT commands and events: <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) 	index: client handle

Table 69. AT+MqttDelete MQTT Client Delete

Request:	Response:
AT+MqttDelete = [index]	OK
Arguments:	Arguments:
index: client handle received from At+MqttCreate	

Table 70. AT+MqttConnect MQTT Client Connect to Broker

Request:	Response:
AT+MqttConnect = [index]	OK
Arguments:	Arguments:
index: client handle received from At+MqttCreate	

Table 71. AT+MqttDisconnect MQTT Client Disconnect From Broker

Request:	Response:
AT+MqttDisconnect = [index]	OK
Arguments:	Arguments:
index: client handle received from At+MqttCreate	

Table 72. AT+MqttPublish MQTT Client Send Message to Broker

Request:	Response:
AT+MqttPublish = [index],[topic],[QoS],[retain],[message length],[message]	OK
Arguments:	Arguments:
<ul style="list-style-type: none"> • index: client handle received from At+MqttCreate • topic: topic string • QoS: Quality of service type: <ul style="list-style-type: none"> – QoS 0 – QoS 1 – QoS 2 • retain: <ul style="list-style-type: none"> – 0: message should not be retained – 1: message should be retained • message length: number of bytes to send • message: message to send in format according to previous configuration in At+MqttCreate (Data format field) 	

Table 73. AT+MqttSubscribe MQTT Client Subscribe for Topic

Request:	Response:
AT+MqttSubscribe = [index],[number of topics],[topic1][QoS1],[persistent1],..., [topicX][QoSX],[persistentX]	OK
Arguments:	Arguments:
<ul style="list-style-type: none"> • index: client handle received from At+MqttCreate • number of topics: maximum 4 topics • topic: topic string • QoS: Quality of service type: <ul style="list-style-type: none"> – QoS 0 – QoS 1 – QoS 2 • persistent (optional for future use) 	

Table 74. AT+MqttUnsubscribe MQTT Client Unsubscribe for Topic

Request:	Response:
AT+MqttUnsubscribe = [index],[number of topics],[topic1],[persistent1],..., [topicX],[persistentX]	OK
Arguments:	Arguments:
<ul style="list-style-type: none"> • index: client handle received from At+MqttCreate • number of topics: maximum 4 topics • topic: topic string • persistent (optional for future use) 	

Table 75. AT+MqttSet MQTT Client Set Option

Request:		Response:
AT+MqttSet = [index],[option],[value1],...,[valueX]		OK
Arguments:		Arguments:
index: client handle received from At+MqttCreate		
Option	Value	
<i>user</i>	Value1: User name string	
<i>password</i>	Value1: Password string	
<i>will</i>	<ul style="list-style-type: none"> • Value1: Topic: will topic string • Value2: QoS: Quality of service type: <ul style="list-style-type: none"> – QoS 0 – QoS 1 – QoS 2 • Value3: Retain: <ul style="list-style-type: none"> – 0: will message should not be retained – 1: will message should be retained • Value4: Message length: number of bytes contain in will message • Value5: Message: will message to send in format according to previous configuration in At+MqttCreate (Data format field) 	
<i>keepalive</i>	Value1: keep alive time in seconds (16 bits)	
<i>clean</i>	Value1: <ul style="list-style-type: none"> • 0: Persistent connection • 1: Enable clean connection 	

6.10 HTTP Client Commands

HTTP client commands and configuration.

Table 76. AT+HttpCreate Http Client Create

Request:	Response:
AT+HttpCreate	+HttpCreate: [index] OK
Arguments:	Arguments:
	index: client handle

Table 77. AT+HttpDestroy Http Client Delete

Request:	Response:
AT+HttpDestroy = [index]	OK
Arguments:	Arguments:
index: client handle received from At+HttpCreate	

Table 78. AT+HttpConnect Http Client Connect to Host

Request:	Response:
AT+HttpConnect = [index],[host],[flags],[private key],[certificate],[ca]	OK
Arguments:	Arguments:
<ul style="list-style-type: none"> • index: client handle received from At+HttpCreate • host: host name • flags: bitmask: <ul style="list-style-type: none"> – ignore_proxy – host_exist • private key: private key file name (optional) • certificate: client certificate file name (optional) • ca: root ca file name (optional) 	

Table 79. AT+HttpDisconnect Http Client Disconnect From Host

Request:	Response:
AT+HttpDisconnect = [index]	OK
Arguments:	Arguments:
index: client handle received from At+HttpCreate	

Table 80. AT+HttpSendReq Http Client Send Request to Host

Request:	Response:
AT+HttpSendReq = [index],[method],[uri],[flags],[format],[length],[data]	+HttpSendReq: [status] OK
Arguments:	Arguments:
<ul style="list-style-type: none"> • index: client handle received from At+HttpCreate • method: <ul style="list-style-type: none"> – get – post – head – options – put – del – connect • uri: request uri string • flags: bitmask: <ul style="list-style-type: none"> – chunk_start: Sets the client's request state into chunked body – chunk_end: Sets the client's request state out of chunked body and sends last chunk – drop_body: Flushes the response body • format: request data format (mandatory only in case of methods post or put) <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • length: length of request data (mandatory only in case of methods post or put) • data: request data (mandatory only in case of methods post or put) 	Status: case of success status = 200, else failure

Table 81. AT+HttpReadResBody Http Client Read Response Body From Host

Request:	Response:
AT+HttpReadResBody = [index],[format],[length]	+HttpReadResBody: [index],[flag],[format],[length],[body] OK
Arguments:	Arguments:
<ul style="list-style-type: none"> • index: client handle received from At+HttpCreate • format: request data format <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • length: maximum length of body 	<ul style="list-style-type: none"> • index: client handle • flag: more data flag • format: request data format <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • length: maximum length of body • body: received data

Table 82. AT+HttpSetHeader Http Client Set Header

Request:	Response:
AT+HttpSetHeader = [index],[option],[flags],[format],[length],[data]	OK
Arguments:	Arguments:
<ul style="list-style-type: none"> • index: client handle received from At+HttpCreate • option: <ul style="list-style-type: none"> – res_age – res_allow – res_cache_control – res_connection – res_content_encoding – res_content_language – res_content_length – res_content_location – res_content_range – res_content_type – res_date – res_etag – res_expires – res_last_modified – res_location – res_proxy_auth – res_retry_after – res_server – res_set_cookie – res_trailer – res_tx_encoding – res_upgrade – res_vary – res_via – res_www_auth – res_warning – req_accept – req_accept_charset – req_accept_encoding – req_accept_language – req_allow – req_auth – req_cache_control – req_connection – req_content_encoding – req_content_language – req_content_location – req_content_type – req_cookie – req_date – req_expect – req_forwarded – req_from – req_host – req_if_match – req_if_modified_since 	<ul style="list-style-type: none"> • index: client handle • flag: more data flag • format: request data format <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • length: maximum length of body • body: received data

Table 82. AT+HttpSetHeader Http Client Set Header (continued)

Request:	Response:
<ul style="list-style-type: none"> • option: <ul style="list-style-type: none"> – req_if_none_match – req_if_range – req_if_unmodified_since – req_origin – req_proxy_auth – req_range – req_te – req_tx_encoding – req_upgrade – req_user_agent – req_via – req_warning • flags: bitmask: <ul style="list-style-type: none"> – not_persistent: Header Field added is not persistent – persistent: Header Field added is persistent • format: data format <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • length: length of data (optional) • data: (optional) 	

Table 83. AT+HttpGetHeader Http Client Get Header

Request:	Response:
AT+HttpGetHeader = [index],[option],[format],[length]	+HttpGetHeader:[index],format],[length],[data] OK
Arguments:	Arguments:
<ul style="list-style-type: none"> • index: client handle received from At+HttpCreate • option: see option in AT+HttpSetHeader command (Table 82) • format: data format <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • length: maximum length of data 	<ul style="list-style-type: none"> • index: client handle • format: data format <ul style="list-style-type: none"> – 0: Binary data format – 1: Base64 data format (binary to text encoding) • length: current length of data • data: received value

Table 84. AT+HttpSetOptHttp Client Set Option

Request:		Response:
AT+HttpSetOpt = [index],[option],[value]		OK
Arguments:		Arguments:
Index: client handle received from At+HttpCreate		
Option	Value	
<i>redirect_feature</i>	<ul style="list-style-type: none"> • 0: disable redirect feature • 1: enable redirect feature 	
<i>res_filter_clear</i>	<ul style="list-style-type: none"> • 1: clear response filter to default (all enabled) 	
<i>redirect_tls_downgrade</i>	<ul style="list-style-type: none"> • 0: disable the option for tls downgrade • 1: enable the option for tls downgrade 	

Table 85. AT+HttpSetProxy Http Client Set Proxy Address

Request:	Response:
AT+HttpSetProxy = [family],[port],[address]	OK
Arguments:	Arguments:
<ul style="list-style-type: none"> • family: Internet Protocol <ul style="list-style-type: none"> – INET: for network protocol IPv4 – INET6: for network protocol IPv6 • port: proxy port • address: proxy server address 	

Revision History

Changes from May 18, 2018 to April 30, 2019 (from A Revision (May 2018) to B Revision)	Page
• Table 11 : Deleted WLAN_STA_CONNECTED	9
• Added TKIP_CCMP Cipher to Gets the WLAN Scan Operation Results table.	20
• Table 85 : Changed AT+HttpSetOpt to AT+HttpSetProxy	53

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2019, Texas Instruments Incorporated