# AT Command Reference

# Table of Contents

# 1.    DEFINITIONS

In this document, the following naming conventions are used:

- DCE (Data Communications Equipment): the Wi-Fi module.
- DTE (Data Terminal Equipment): the terminal that issues commands to the Wi-Fi module.

## 2. NOTE

- In the command description below or while using the AT commands with the module, there are some parameters which are placeholder for future releases. Please refer to this document to find the list of parameters which are supported. If a parameter is not mentioned in this document but gets printed while reading the configuration, it is just a placeholder for a future release.

# 3.    Command Reference: General/Miscellaneous

## 3.1    Command Echo Control E

### 3.1.1    Description

This command controls if characters received from the DTE are echoed back to the DTE.

### 3.1.2    Command Syntax

`ATE[<N>]`

| Parameter Name | Type | Description |
|---|---|---|
| <N> | Integer | 0.    Turn of character echo<br>1.    Turn on character echo |

### 3.1.3    Response Syntax

| Response | Description |
|---|---|
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 3.1.4    Examples:

| ATE0 | Turn off character echo |
|---|---|
| OK | Command completed |

## 3.2    Request Manufacturer Identification +GMI

### 3.2.1    Description

This command retrieves the manufacturers information from the DCE.

The format of the returned data is platform dependent, the information returned for a WINC-based platform is as given below:

### 3.2.2    Command Syntax

`AT+GMI`

### 3.2.3    Response Syntax

| Response | Description |
|---|---|
| +GMI:<MAN_ID> | Information response |
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

| Parameter Name | Type | Description |
|---|---|---|
| <MAN_ID> | String | Manufacturers ID/Name |

### 3.2.4    Examples:

| | |
|---|---|
| AT+GMI | Query the manufacturers ID/name |
| +GMI:"Microchip" | Manufacturers information |
| OK | Command completed |

## 3.3    Request Model Identification +GMM

### 3.3.1    Description

This command retrieves the model information from the DCE.

The format of the returned data is platform dependent, the information returned for a WINC-based platform is as given below:

### 3.3.2    Command Syntax

AT+GMM

### 3.3.3    Response Syntax

| Response | Description |
|---|---|
| +GMM:<MODEL_ID> | Information response |
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

| Parameter Name | Type | Description |
|---|---|---|
| <MODEL_ID> | String | Model Information |

### 3.3.4    Examples:

| | |
|---|---|
| AT+GMM | Query the model information |
| +GMM:"WINC1500" | Model information |
| OK | Command completed |

## 3.4    Request Revision Identification +GMR

### 3.4.1    Description

This command retrieves version information from the DCE.

The format of the returned data is platform dependent, the information returned for a WINC-based platform is as given below:

### 3.4.2 Command Syntax

`AT+GMR`

### 3.4.3 Response Syntax

| Response | Description |
|---|---|
| +VER:<CHIP_ID>,<FW_VER>,<DRV_VER>,<RB_FW> | Information response |
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

| Parameter Name | Type | Description |
|---|---|---|
| <CHIP_ID> | String | ID / version of the Wi-Fi chip |
| <FW_VER> | String | The version of the firmware (software running on Wi-Fi IC) |
| <DRV_VER> | String | Any additional versioning info (e.g. of driver to application layer) |
| <RB_FW> | String | Rollback firmware version |

### 3.4.4 Examples:

| | |
|---|---|
| AT+GMR | Query the revision information |
| +GMR:"15x0","19.6.1","19.6.1","19.5.3" | Version information |
| OK | Command completed |

## 3.5 Low Power +LOWPOWER

### 3.5.1 Description

The LOWPOWER command can be used to put the device in Low Power Mode.

Once the device has been put in Low Power Mode, a GPIO needs to be pressed to bring the device out of Low Power Mode. When the device comes out of low power mode, the device needs to be reconfigured with the Home AP details to start connection with the Home AP.

### 3.5.2 Command Syntax

AT+LOWPOWER[=<MODE>]

| Parameter Name | Type | Description |
|---|---|---|
| <MODE> | Integer | Low Power Mode<br><br>4 Put device in low power mode<br><br>**Note:** 0-3 are placeholders for features not supported in this release. |

### 3.5.3 Response Syntax

| Response | Description |
|---|---|

| | |
|---|---|
| `ERROR:<ERROR_CODE>` | Error response |

## 3.6 Read Certificate +READCERT

### 3.6.1 Description

The READCERT command can be used for reading the certificate from the TNG (ECC608).

Once the device has been put in Low Power Mode, a GPIO needs to be pressed to bring the device out of Low Power Mode. When the device comes out of low power mode, the device needs to be reconfigured with the Home AP details to start connection with the Home AP.

### 3.6.2 Command Syntax

AT+READCERT[=<TYPE>]

| Parameter Name | Type | Description |
|---|---|---|
| <TYPE> | Integer | Certificate Type<br><br>1. Device Certificate<br>2. Root Certificate<br>3. Last Certificate Loaded by the Host |

### 3.6.3 Response Syntax

| Response | Description |
|---|---|
| `OK` | Successful response |
| `ERROR:<ERROR_CODE>` | Error response |

## 3.7 Load Certificate +LOADCERT

### 3.7.1 Description

The LOADCERT command can be used to loading a server certificate into the device for secured TCP Connection with that Server.

The certificate is loaded in Binary Mode. Once the LOADCERT command is given, the UART switches to binary mode and waits for LEN bytes of the certificate, or +++ to exit the binary mode. The Certificate can be in PEM or DER Mode.

### 3.7.2 Command Syntax

AT+LOADCERT[=<LEN>][<NAME>]

| Parameter Name | Type | Description |
|---|---|---|
| <LEN> | Integer | Length of the certificate in bytes (Max length = 1500) |
| <NAME> | String | Certificate Name |

### 3.7.3 Response Syntax

| Response | Description |
|---|---|

| OK | Successful response |
|---|---|
| +LOADCERT:<RESPONSE> | 0: Success; 1: Failure |

# 4. Command Reference: Scanning

## 4.1 Configure Scan Behavior +WSCNC

### 4.1.1 Description

This command is used to modify or query the configuration of the scanning function.

### 4.1.2 Command Syntax

| Command | Description |
|---|---|
| AT+WSCNC[=<param_id>] | Read configuration |
| AT+WSCNC=<param_id>,<param_val> | Set configuration |

### 4.1.3 Supported Parameters

| ID | Name & Default Value | Type | Description |
|---|---|---|---|
| 1 | <SCAN_TIME> | Integer | The time in milliseconds to wait for probe responses |
| 2 | <PASV_LISTEN> | Integer | The time in milliseconds to wait for beacons |

### 4.1.4 Response Syntax

| Response | Description |
|---|---|
| +WSCNC:<param_id>,<param_val> | Read response |
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 4.1.5 Examples:

| AT+WSCNC | Query the current active scan configuration settings |
|---|---|
| +WSCNC:1,30 | 30ms wait |
| +WSCNC:2,300 | 300ms duration for passive scans |
| OK | Command completed |

## 4.2 Active Scanning Command +WSCNA

### 4.2.1 Description

This command is used to actively scan for infrastructure networks in range of the DCE.

This form of scanning causes the device to transmit energy on each channel being scanned.

### 4.2.2 Command Syntax

AT+WSCNA=<CHANNEL>[,<SSID>]

| Parameter Name | Type | Description |
|---|---|---|
| <CHANNEL> | Integer | The channel to scan, a value of 255 scans all available channels |
| <SSID> | String | Scan for a specific SSID, confirms presence of a cloaked network |

### 4.2.3 Response Syntax

| Response | Description |
|---|---|
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 4.2.4 AEC Syntax (scan indications)

+WSCNIND:<RSSI>,<SEC_TYPE>,<CHANNEL>,<BSSID>,<SSID>

| Parameter Name | Type | Description |
|---|---|---|
| <RSSI> | Integer | Received Signal Strength (higher is better) |
| <SEC_TYPE> | Integer | Security Type |
| | | 0      No security (open network) |
| | | 1      WEP |
| | | 2      WPA Personal (pre-shared key) |
| | | 3      WPA2 Personal (pre-shared key) |
| <CHANNEL> | Integer | Channel number of detected device |
| <BSSID> | String | BSSID of detected device |
| <SSID> | String | SSID of detected device |

### 4.2.5 AEC Syntax (scan completed)

+WSCNDONE

### 4.2.6 AEC Syntax (scan failed)

+WSCNERROR

### 4.2.7 Example:

| | |
|---|---|
| AT+WSCNA=255 | Scan all channels |
| OK | |
| +WSCNIND:-57,0,1,"98:FC:11:41:E7:7C","test-network" | Scan indication: open network, channel 1, SSID test-network |
| +WSCNIND:-62,0,6,"24:F2:7F:A5:8B:81", [466F6F20C2A920626172] | Open network, channel 6, with UTF-8 SSID Foo © Bar |
| +WSCNDONE | Indication of completion of scanning activities |

## 4.3 Passive Scanning Command +WSCNP

### 4.3.1 Description

This command is used to passively scan for infrastructure networks in range of the DCE.

This form of scanning causes the device to tune to the required channel(s), turn on its receiver, and listen for beacons from nearby devices. No energy is transmitted on the scanned channel(s) during this operation.

### 4.3.2 Command Syntax

```
AT+WSCNP=<CHANNEL>[,<SSID>]
```

| Parameter Name | Type | Description |
|---|---|---|
| <CHANNEL> | Integer | The channel to scan, a value of 255 scans all available channels |
| <SSID> | Integer | Scan for a specific SSID |

### 4.3.3 Response Syntax

| Response | Description |
|---|---|
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 4.3.4 AEC Syntax (scan indications)

+WSCNIND:<RSSI>,<SEC_TYPE>,<CHANNEL>,<BSSID>,<SSID>

| Parameter Name | Type | Description |
|---|---|---|
| <RSSI> | Integer | Received Signal Strength (higher is better) |
| <SEC_TYPE> | Integer | Security Type |
| | | 0        No security (open network) |
| | | 1        WEP |
| | | 2        WPA Personal (pre-shared key) |
| | | 3        WPA2 Personal (pre-shared key) |
| | | 4        Enterprise |
| <CHANNEL> | Integer | Channel number of detected device |
| <BSSID> | String | BSSID of detected device |
| <SSID> | String | SSID of detected device |

### 4.3.5 AEC Syntax (scan completed)

+WSCNDONE

### 4.3.6 AEC Syntax (scan failed)

+WSCNERROR

### 4.3.7 Examples:

| | |
|---|---|
| `AT+WSCNP=255` | Scan all channels |
| `OK` | |
| `+WSCNIND:-57,0,1,"98:FC:11:41:E7:7C","test-network"` | Scan indication: open network, channel 1, SSID test-network |
| `+WSCNIND:-62,4,1,"24:F2:7F:A5:8B:80","enterprise-secure"` | Scan indication: Enterprise, channel 1, SSID enterprise-secure |
| `+WSCNIND:-62,0,6,"24:F2:7F:A5:8B:81",`<br>`[466F6F20C2A920626172]` | Open network, channel 6, with UTF-8 SSID Foo © Bar |
| `+WSCNIND:-70,1,11,"24:F2:7F:A5:8B:81",[]` | Open network, channel 11, with hidden SSID |
| `+WSCNDONE` | Scanning completed |

# 5.  Command Reference: WLAN Configuration

## 5.1  Wi-Fi Station Configuration +WSTAC

### 5.1.1  Description

This command is used to read / set the DCE's Wi-Fi station mode configuration.

### 5.1.2  Command Syntax

| Command | Description |
|---|---|
| `AT+WSTAC[=<param_id>]` | Read configuration |
| `AT+WSTAC=<param_id>,<param_val>` | Set configuration |

### 5.1.3  Supported Parameters

When not connected to an AP these parameters are available for configuration.

| ID | Name & Default Value | Type | Description |
|---|---|---|---|
| 1 | <SSID> | String | Network Name (Mandatory parameter) |
| 2 | <SEC_TYPE>: 0 | Integer | Security Type<br><br>0  No security (open network)<br><br>1  WEP<br><br>2  WPA Personal (pre-shared key)<br><br>3  WPA2 Personal (pre-shared key)<br><br>4  WPA2 and WPA3 Mixed<br><br>5  WPA3 |
| 3 | <CREDENTIALS>: "" | String | Credentials for connecting to the network:<br><br>SEC_TYPE — Credentials format (ASCII or hex string)<br><br>0  "" (empty string)<br><br>1  Any ASCII or hex string<br><br>2  Of format "KEYINDEX*PASSWORD"<br><br>3  Of format "USERNAME*PASSWORD" |
| 4 | <CHANNEL>: 255 (any channel) | Integer | The channel the network must reside on. |
| 12 | <NTP_SVR>: "" | String | The address/name of NTP server |

| 13 | <NTP_STATIC>: 0 | Integer | NTP configuration mode | |
|---|---|---|---|---|
| | | | 0 | DHCP – can be set via DHCP **(Placeholder for future release)** |
| | | | 1 | Static – cannot be set by DHCP |

### 5.1.4 Response Syntax

| Response | Description |
|---|---|
| +WSTAC:<param_id>,<param_val> | Read response |
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

## 5.2 Wi-Fi Station, Start and Stop +WSTA, +WSTALU & +WSTALD Link Notifications

### 5.2.1 Description

This command is used to enable the DCE's station mode functionality.

### 5.2.2 Command Syntax

| Command | Description |
|---|---|
| AT+WSTA | Read State of STA function |
| AT+WSTA=<STATE> | Set State of STA function |

| Parameter Name | Type | Description | |
|---|---|---|---|
| <STATE> | Integer | State of the Wi-Fi station feature | |
| | | 0 | Disable |
| | | 1 | Use configuration from +WSTAC command |

### 5.2.3 Response Syntax

| Response | Description |
|---|---|
| +WSTA:<ASSOC_ID>,<STATE>[,<SSID>,<RSSI>,<SEC_TYPE>] | Read response |
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

| Parameter Name | Type | Description |
|---|---|---|
| <ASSOC_ID> | Integer | Association ID |
| <SSID> | String | Network Name |

| <RSSI> | Integer | Received Signal Strength (higher is better) |
|---|---|---|
| | | **Placeholder for future release (always reported as 0 in this version)** |
| <SEC_TYPE> | Integer | Security Type |
| | | 0          No security (open network) |
| | | 1          WEP (**Not Supported in PIC32MZW1**) |
| | | 2          WPA Personal (pre-shared key) |
| | | 3          WPA2 Personal (pre-shared key) |
| | | 4          Enterprise |

### 5.2.4 AEC Syntax (Link established)

+WSTALU: <ASSOC_ID>,<BSSID>,<CHANNEL>

| Parameter Name | Type | Description |
|---|---|---|
| <ASSOC_ID> | Integer | Association ID |
| <BSSID> | String | The BSSID of the Access Point the DCE has connected to |
| <CHANNEL> | Integer | The channel number of network |

### 5.2.5 AEC Syntax (Link lost)

+WSTALD: <ASSOC_ID>

| Parameter Name | Type | Description |
|---|---|---|
| <ASSOC_ID> | Integer | Association ID |

### 5.2.6 AEC Syntax (Indication of STA Automatic Address Assignment)

+WSTAAIP: <ASSOC_ID>,<IP_ADDRESS>

| Parameter Name | Type | Description |
|---|---|---|
| <ASSOC_ID> | Integer | Association ID |
| <IP_ADDRESS> | String | IP address assigned |

### 5.2.7 AEC Syntax (Connection Error)

+WSTAERR: <ERROR_CODE>

| Parameter Name | Type | Description |
|---|---|---|
| <ERROR_CODE> | Integer | Error Code |

### 5.2.8 Examples:

```
AT+WSTAC=1,"MyAP"                              Configure AP connection parameters

OK

AT+WSTAC=2,1
```

| | |
|---|---|
| `OK` | |
| `AT+WSTAC=3,"MyAPPSK"` | |
| `OK` | |
| `AT+WSTA=1` | Connect to AP |
| `OK` | |
| `+WSTALU:1,"00:01:02:03:04:05",6` | Link up: connected on channel 6 |
| `+WSTAAIP:1,"19.168.0.20"` | IP address received via auto configuration (DHCP/SLAAC) |
| `AT+WSTA` | |
| `+WSTA:1,1,"MyAP",-50,0` | |
| `OK` | |
| `+WSTALD` | Link down |

## 5.3 Wi-Fi Hotspot (AP) Configuration +WAPC

### 5.3.1 Description

This command is used to read or set the DCE's hotspot access point configuration.

### 5.3.2 Command Syntax

| Command | Description |
|---|---|
| `AT+WAPC[=<param_id>]` | Read configuration |
| `AT+WAPC=<param_id>,<param_val>` | Set configuration |

### 5.3.3 Supported Parameters

When not configured as an AP these parameters are available for configuration.

| ID | Name & Default Value | Type | Description |
|---|---|---|---|
| 1 | <SSID>: DEMO_AP | String | Network Name |
| 2 | <SEC_TYPE>: 0 | Integer | Security Type<br>0    No security (open network)<br>1    WEP<br>2    WPA Personal (pre-shared key)<br>3    WPA2 Personal (pre-shared key)<br>4    Enterprise |

| | | | |
|---|---|---|---|
| 3 | <CREDENTIALS>: "" | String | Credentials required for connecting to the network of the security type specified:<br><br>**SEC_TYPE** / **Credentials format**<br>0 — "" (empty string)<br>1 — Any ASCII or hexadecimal string<br>2 — Any ASCII or hexadecimal string of format "KEYINDEX*PASSWORD"<br>3 — Any ASCII or hexadecimal string of format "USERNAME*PASSWORD" |
| 4 | <CHANNEL>: 6 | Integer | The channel of the network to connect to |
| 5 | <HIDDEN>: 0 | Integer | Visibility of the network<br>0 — Not hidden, SSID is broadcast in beacons<br>1 — Hidden, SSID is not broadcast |

When configured as an AP the parameter entries are 'Read Only' and cannot be modified.

### 5.3.4    Response Syntax

| Response | Description |
|---|---|
| +WAPC:<param_id>,<param_val> | Read response |
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

## 5.4    Wi-Fi Hotspot (Soft AP) Start/Stop/Status +WAP, +WAPSC & +WAPSD Link Indications

### 5.4.1    Description

This command is used to enable the DCE's hotspot access point functionality.

### 5.4.2    Command Syntax

| Command | Description |
|---|---|
| AT+WAP | Read Status of AP function |
| AT+WAP=<STATE> | Set State of AP function |

| Parameter Name | Type | Description |
|---|---|---|
| | | |

| <STATE> | Integer | State of the hotspot feature | |
|---|---|---|---|
| | | 0 | Disable |
| | | 1 | Enable |

### 5.4.3 Response Syntax

| Response | Description |
|---|---|
| +WAP:<STATE>[,<BSSID>,<RSSI>] | Query state response |
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

| Parameter Name | Type | Description |
|---|---|---|
| <BSSID> | String | The BSSID of the STA connecting to the AP |
| <RSSI> | Integer | Received Signal Strength (higher is better) |

### 5.4.4 AEC Syntax (Station Connected)

+WAPSC: <ASSOC_ID>,<BSSID>

| Parameter Name | Type | Description |
|---|---|---|
| <ASSOC_ID> | Integer | Association ID |

### 5.4.5 AEC Syntax (Station Disconnected)

+WAPSD: <ASSOC_ID>,<BSSID>

| Parameter Name | Type | Description |
|---|---|---|
| <ASSOC_ID> | Integer | Association ID |

### 5.4.6 AEC Syntax (DHCP Server Indication of Connected STA Address Assignment)

+WAPAIP: <ASSOC_ID>,<IP_ADDRESS>

| Parameter Name | Type | Description |
|---|---|---|
| <ASSOC_ID> | Integer | Association ID |

Examples:

| | |
|---|---|
| AT+WAPC=1,"MyAPSSID" | AP SSID |
| OK | |
| AT+WAPC=2,2 | WEP security |
| OK | |
| AT+WAPC=3,"MyAPPassword" | WEP key |
| OK | |

| | |
|---|---|
| `AT+WAPC=4,11` | Channel |
| `OK` | |
| `AT+WAP=1` | |
| `OK` | |
| `+WAPSC:2,"00:01:02:03:04:05"` | STA connected |
| `+WAPAIP:2,"192.168.0.100"` | STA assigned IP address |
| `+WAPSD:2,"00:01:02:03:04:05"` | STA disconnected |

## 5.5    Configure Provisioning +WPROVC

### 5.5.1    Description

This command is used to modify or query the configuration of the provisioning function.

When using the provisioning method, the SSID used for the SoftAP can be set using the +WAPC command. The SSID needs to be set before the device is put in Provisioning Mode.

By default the device is in STA mode. To put the device in provisioning mode, the "AT+WPROV=1" command needs to be given. Once the device enters the provisioning mode, it opens a socket on the provisioning port and waits for a TCP client to connect. Any TCP client can be used to connect to the device.

Once a TCP connection has been setup with the device under provisioning mode, the AT+WSTAC commands can be given (refer to the AT+WSTAC section to know about the format in which commands need to be given) to configure the device for Home Router details. Once all provisioning commands are given, the "AT+WPROV=0" command needs to be given to make the device exit the provisioning mode. As soon as the device receives the "AT+WPROV=0" command over the TCP connection, it would stop its SoftAP interface. Hence the TCP client will lose connection with the device. "+WPROVEXIT" will be sent on the UART to indicate to the Host MCU that now it can give the "AT+WSTA=1" command to start connection with the Home Router.

Note: In case the scan results need to be retrieved over the TCP connection, then the "AT+WSCNA" or "AT+WSCNP" command can be used to retrieve the scan results from the device. When the scan command is given on the TCP Client, the scan results will be sent by the device to the TCP Client.

### 5.5.2    Command Syntax

| Command | Description |
|---|---|
| AT+WPROVC[=<param_id>] | Read configuration |
| AT+WPROVC=<param_id>,<param_val> | Set configuration |

### 5.5.3    Supported Parameters

| ID | Name & Default Value | Type | Description |
|---|---|---|---|
| 1 | <MODE> | Integer | Mode in which provisioning would be done. In existing release, only mode 2 (Socket based provisioning) is supported |
| 2 | <PIN> | String | Pin number to be used for WPS. **Not supported in existing release.** |

| 3 | <PROV_PORT > | Integer | Port number on which a TCP server will wait for provisioning data. Default port number is set to 7777. |
|---|---|---|---|

### 5.5.4 Response Syntax

| Response | Description |
|---|---|
| +WPROVC:<param_id>,<param_val> | Read response |
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 5.5.5 Examples:

| | |
|---|---|
| AT+WPROVC | Query the current provisioning configuration settings |
| +WPROVC:1,2 | Mode is set to 2 (Socket based provisioning). |
| +WPROVC:2,[] | Not used in this release |
| +WPROVC:3,7777 | Provisioning Port number is set to 7777 |
| OK | |
| AT+WPROVC=3,8888 | Set the provisioning port to 8888 |
| OK | |

## 5.6 Start / Stop Provisioning +WPROV

### 5.6.1 Description
The WPROV command can be used to start / stop the provisioning mode.

### 5.6.2 Command Syntax
AT+WPROV[=<STATE>]

| Parameter Name | Type | Description | |
|---|---|---|---|
| <STATE> | Integer | Provisioning feature state | |
| | | 0 | Stop |
| | | 1 | Start |
| | | | |

### 5.6.3 Response Syntax

| Response | Description |
|---|---|
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

# 6. Command Reference: MQTT

## 6.1 MQTT Configuration +MQTTC

### 6.1.1 Description
This command is used to read or set the MQTT configuration.

### 6.1.2 Command Syntax

| Command | Description |
|---|---|
| AT+MQTTC[=<param_id>] | Read configuration |
| AT+MQTTC=<param_id>,<param_val> | Set configuration |

### 6.1.3 Supported Parameters

| ID | Name & Default Value | Type | Description |
|---|---|---|---|
| 1 | <BROKER_ADDR>: "" | String | Broker domain name or IPv4 address |
| 2 | <BROKER_PORT>: 8883 | Integer | Broker listening TCP port |
| 3 | <CLIENT_ID >: "<Device ID>" | String | MQTT Client ID |
| 4 | <USERNAME>: "" | String | Username |
| 7 | <TLS_CONF_IDX>: 1 | Integer | Enable (1)/ Disable(0) TLS |

### 6.1.4 Response Syntax

| Response | Description |
|---|---|
| +MQTTC:<param_id>,<param_val> | Read response |
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

## 6.2 MQTT Connection +MQTTCONN

### 6.2.1 Description
This is used to connect to an MQTT broker or request current connection status.

### 6.2.2 Command Syntax
AT+MQTTCONN[=<CLEAN>]

| Parameter Name | Type | Description |
|---|---|---|

| <CLEAN> | Integer | **Placeholder for future release. Ignored in this release.** |
| | | Clean session flag: |
| | | 0 Reuse existing session if possible |
| | | **(Placeholder for future release)** 1 Request clean session |

### 6.2.3 Response Syntax

| Response | Description |
|---|---|
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 6.2.4 AEC Syntax

+MQTTCONNACK:<CONNACK_FLAGS>,<RETURN_CODE>

| Parameter Name | Type | Description |
|---|---|---|
| <CONNACK_FLAGS> | Integer | Connect Acknowledge Flags |
| <RETURN_CODE> | Integer | Connect Return code |

### 6.2.5 AEC Syntax

+MQTTCONN:<CONN_STATE>

| Parameter Name | Type | Description |
|---|---|---|
| <CONN_STATE> | Integer | MQTT connected state: |
| | | 0 Not connected 1 Connected |

### 6.2.6 Examples:

| AT+MQTTCONN=1 | Connect to broker |
|---|---|
| OK | Command completed |
| +MQTTCONNACK:0,0 | Connection acknowledgement |
| +MQTTCONN:1 | Connected |

## 6.3 MQTT Subscribe +MQTTSUB

### 6.3.1 Description

This is used to subscribe to an MQTT topic.

### 6.3.2 Command Syntax

AT+MQTTSUB=<TOPIC_NAME>,<MAX_QOS>

| Parameter Name | Type | Description |
|---|---|---|
| <TOPIC_NAME> | String | Name of topic to subscribe to |

| | | |
|---|---|---|
| <MAX_QOS> | Integer | Max QoS (Valid values are 0, 1 and 2) |

### 6.3.3 Response Syntax

| Response | Description |
|---|---|
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 6.3.4 AEC Syntax

+MQTTSUB:<REASON_CODE>

| Parameter Name | Type | Description |
|---|---|---|
| <REASON_CODE> | Integer | Result of unsubscribe request:<br><br>0.   Request succeeded<br>1.   Request failed |

### 6.3.5 Examples:

| | |
|---|---|
| AT+MQTTSUB="Topic/name" | Subscribe to a topic |
| OK | Command completed |
| +MQTTSUB:0 | Subscription successful |

## 6.4 MQTT Unsubscribe +MQTTUNSUB

### 6.4.1 Description

This is used to unsubscribe from an MQTT topic.

### 6.4.2 Command Syntax

AT+MQTTUNSUB=<TOPIC_NAME>

| Parameter Name | Type | Description |
|---|---|---|
| <TOPIC_NAME> | String | Name of topic to unsubscribe from |

### 6.4.3 Response Syntax

| Response | Description |
|---|---|
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 6.4.4 AEC Syntax

+MQTTUNSUB:<RESULT>

| Parameter Name | Type | Description |
|---|---|---|

| | | |
|---|---|---|
| <RESULT> | Integer | Result of unsubscribe request: <br> 1. Request failed <br> 2. Request succeeded |

### 6.4.5 Examples:

| | |
|---|---|
| `AT+MQTTUNSUB="Topic/name"` | Unsubscribe from topic |
| `OK` | Command completed |
| `+MQTTUNSUB:1` | Unsubscribe was successful |

## 6.5 MQTT Publish +MQTTPUB

### 6.5.1 Description

This is used to publish a message.

### 6.5.2 Command Syntax

AT+MQTTPUB=<DUP>,<QOS>,<RETAIN>,<TOPIC_NAME>,<TOPIC_PAYLOAD>

| Parameter Name | Type | Description |
|---|---|---|
| <DUP> | Integer | **Placeholder for future release. Always set to 0 in this release.** <br> Duplicate flag: <br> 0. First attempt to send message <br> 1. Subsequent attempt to send message |
| <QOS> | Integer | QoS: <br> 0. At most once <br> 1. At least once <br> 2. Exactly once |
| <RETAIN> | Integer | Retain flag: <br> 0. Do not retain message on server <br> 1. Retain message on server |
| <TOPIC_NAME> | String | Name of topic to send message to |
| <TOPIC_PAYLOAD> | String | Content of message to send |

### 6.5.3 Response Syntax

| Response | Description |
|---|---|
| `OK` | Successful response |
| `ERROR:<ERROR_CODE>` | Error response |

### 6.5.4 AEC Syntax (QoS=1)

+MQTTPUBACC

### 6.5.5    AEC Syntax (QoS=2)
+MQTTPUBCOMP

### 6.5.6    AEC Syntax (Error)
+MQTTPUBERR

### 6.5.7    Examples:

| | |
|---|---|
| `AT+MQTTPUB=0,1,0,"Topic/name","123456"` | Publish to a topic, QoS is 1 |
| `OK` | Command completed |
| `+MQTTPUBACC` | Publish acknowledged |

## 6.6    MQTT Disconnect +MQTTDISCONN

### 6.6.1    Description
This is used to disconnect from a broker.

<REASON_CODE> parameters is for MQTT V5 only.

### 6.6.2    Command Syntax
AT+MQTTDISCONN[=<REASON_CODE>]

| Parameter Name | Type | Description |
|---|---|---|
| <REASON_CODE> | Integer | Reason code describing reason for disconnecting. |

### 6.6.3    Response Syntax

| Response | Description |
|---|---|
| `OK` | Successful response |
| `ERROR:<ERROR_CODE>` | Error response |

### 6.6.4    AEC Syntax
+MQTTCONN:<CONN_STATE>

| Parameter Name | Type | Description |
|---|---|---|
| <CONN_STATE> | Integer | MQTT connected state: 0 Not connected 1 Connected |

### 6.6.5    Examples:

| | |
|---|---|
| `AT+MQTTDISCONN` | Disconnect |
| `OK` | Command completed |
| `+MQTTCONN:0` | Not connected |

# 7. Command Reference: Network Stack

## 7.1 Create Socket +SOCKO

### 7.1.1 Description
This command is used to open a new socket.

### 7.1.2 Command Syntax
`AT+SOCKO=<PROTOCOL>`

| Parameter Name | Type | Description |
|---|---|---|
| <PROTOCOL> | Integer | The protocol to use |
| | | 1    UDP |
| | | 2    TCP |

### 7.1.3 Response Syntax

| Response | Description |
|---|---|
| `+SOCKO:<SOCK_ID>` | Socket open response with new ID |
| `OK` | Successful response |
| `ERROR:<ERROR_CODE>` | Error response |

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | ID of the socket opened |

### 7.1.4 Examples:
(TCP):

| | |
|---|---|
| `AT+SOCKO=2` | Open a new listening socket, TCP |
| `+SOCKO:1` | Socket opened, ID is 1 |
| `OK` | Command completed |

(UDP):

| | |
|---|---|
| `AT+SOCKO=1` | Open a new listening socket, UDP |
| `+SOCKO:5` | Socket opened, ID is 5 |
| `OK` | Command completed |

## 7.2 Close Socket +SOCKCL

### 7.2.1 Description

This command is used to close a socket.

The AEC signals that the socket was closed by the remote address or due to a failure.

Note: Receiving the AEC indicates a shutdown of the socket at the remote end, the socket will still exist locally, `AT+SOCKCL` must still be sent to remove the socket locally.

### 7.2.2 Command Syntax

`AT+SOCKCL=<SOCK_ID>`

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | ID of the socket to close |

### 7.2.3 Response Syntax

| Response | Description |
|---|---|
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 7.2.4 AEC Syntax

`+SOCKCL:<SOCK_ID>`

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | The socket ID which has closed |

### 7.2.5 Examples:

| | |
|---|---|
| AT+SOCKCL=2 | Close socket ID 2 |
| OK | Command completed |
| +SOCKCL:4 | Socket ID 4 was closed remotely |
| AT+SOCKCL=4 | Close socket ID 2 locally |
| OK | Command completed |

## 7.3 Bind a Listening Socket +SOCKBL

### 7.3.1 Description

This command is used to bind a socket to a local port.

A locally bound socket allows network services to be implemented. Any remote device may send data to this socket and the DCE will issue a data ready indication. For TCP sockets a connection must have been established prior to sending and receiving data.

### 7.3.2 Command Syntax

`AT+SOCKBL=<SOCK_ID>,<LCL_PORT>`

| Parameter Name | Type | Description |
|---|---|---|
| | | |

| <SOCK_ID> | Integer | ID of the socket to bind |
|-----------|---------|--------------------------|
| <LCL_PORT> | Integer | The port number to use |

### 7.3.3 Response Syntax

| Response | Description |
|----------|-------------|
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 7.3.4 AEC Syntax (TCP Connection Established) +SOCKIND

+SOCKIND:<SOCK_ID>,<LCL_ADDR>,<LCL_PORT>,<RMT_ADDR>,<RMT_PORT>

| Parameter Name | Type | Description |
|----------------|------|-------------|
| <SOCK_ID> | Integer | The ID of the passive listening socket which received connection |
| <LCL_ADDR> | String | The address of the local end of the connection |
| <LCL_PORT> | Integer | The port number of the local end of the connection |
| <RMT_ADDR> | String | The address of the remote end of the connection |
| <RMT_PORT> | Integer | The port number of the remote end of the connection |

### 7.3.5 Examples:

(TCP):

| AT+SOCKBL=2,6000 | Bind a listening socket, port 6000 |
|------------------|-------------------------------------|
| OK | Command completed |
| : | |
| +SOCKIND:4,"192.168.0.100",6000,"192.168.0.1",12345 | New incoming TCP connection creating socket ID is 4 |

(UDP):

| AT+SOCKBL=1,6000 | Bind a listening socket, port 6000 |
|------------------|-------------------------------------|
| OK | Command completed |

## 7.4 Bind a Remote Socket +SOCKBR

### 7.4.1 Description

This command is used to bind a socket to a remote address.

### 7.4.2 Command Syntax

AT+SOCKBR=<SOCK_ID>,<RMT_ADDR>,<RMT_PORT>

| Parameter Name | Type | Description |
|----------------|------|-------------|

| <SOCK_ID> | Integer | ID of the socket to bind |
|---|---|---|
| <RMT_ADDR> | String | The address of the remote device |
| <RMT_PORT> | Integer | The port number on the remote device |

### 7.4.3    Response Syntax

| Response | Description |
|---|---|
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 7.4.4    AEC Syntax (TCP Bind Success) +SOCKIND

+SOCKIND:<SOCK_ID>,<LCL_ADDR>,<LCL_PORT>,<RMT_ADDR>,<RMT_PORT>

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | The ID of the bound socket |
| <LCL_ADDR> | String | The address of the local end of the connection |
| <LCL_PORT> | Integer | The port number of the local end of the connection |
| <RMT_ADDR> | String | The address of the remote end of the connection |
| <RMT_PORT> | Integer | The port number of the remote end of the connection |

### 7.4.5    AEC Syntax (TCP Bind Failed) +SOCKERR

+SOCKERR:<SOCK_ID>,<STATUS_CODE>[,<STATUS_MSG>]

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | The ID of the bound socket |
| <STATUS_CODE> | Integer | Numeric status code, see Error! Reference source not found. Status Response CodesError! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found. |
| <STATUS_MSG> | String | Descriptive text detailing the error |

### 7.4.6    Examples:

(UDP):

| AT+SOCKBR=1,"192.168.0.1",6000 | Bind a connection to remote host, port 6000 |
|---|---|
| OK | Command completed |

(TCP):

| AT+SOCKBR=4,"192.168.0.1",9000 | Bind a connection to remote host, port 9000 |
|---|---|
| OK | Command completed |

```
+SOCKIND:4,"192.168.0.100",12345,"192.168.0.1",   Connection established
9000
```

## 7.5 Bind a Multicast Socket (+SOCKBM)

### 7.5.1 Description

These commands are used to bind a UDP socket to a multicast group.

### 7.5.2 Command Syntax

`AT+SOCKBM=<SOCK_ID>,<MCAST_ADDR>,<MCAST_PORT>`

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | The socket ID to change |
| <MCAST_ADDR> | String | The address of the multicast group |
| <MCAST_PORT> | Integer | The port of the multicast group |

### 7.5.3 Response Syntax

| Response | Description |
|---|---|
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 7.5.4 Examples:

| | |
|---|---|
| `AT+SOCKBM=7,"224.0.0.1",6000` | Add socket to multicast group |
| `OK` | Command completed |

## 7.6 Upgrade Socket to TLS +SOCKTLS

Description

This command is used to enable TLS on a socket.

### 7.6.1 Command Syntax

`AT+SOCKTLS=<SOCK_ID>,<TLS_CONF_IDX>`

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | ID of the socket |
| <TLS_CONF_IDX> | Integer | TLS certificate configuration index to use (see +TLSC) |

### 7.6.2 Response Syntax

| Response | Description |
|---|---|
| +SOCKTLS:<SOCK_ID> | Socket information |

| OK | Successful response |
|---|---|
| ERROR:<ERROR_CODE> | Error response |

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | Socket ID |

### 7.6.3 AEC Syntax (TLS Succeed) +SOCKTLS

+SOCKTLS:<SOCK_ID>

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | The ID of the bound socket |

### 7.6.4 AEC Syntax (TLS Failed) +SOCKERR

+SOCKERR:<SOCK_ID>,<STATUS_CODE>[,<STATUS_MSG>]

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | The ID of the bound socket |
| <STATUS_CODE> | Integer | Numeric status code, see Error! Reference source not found. Status Response CodesError! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found. |
| <STATUS_MSG> | String | Descriptive text detailing the error |

### 7.6.5 Examples:

| AT+SOCKTLS=1 | Command: list all open sockets |
|---|---|
| OK | Command completed successfully |
| +SOCKTLS:1 | TCP socket completed TLS negotiation |

## 7.7 List Current Sockets +SOCKLST

### 7.7.1 Description

This command is used to present a list of the DCE's open sockets/connections.

### 7.7.2 Command Syntax

AT+SOCKLST[=<SOCK_ID>]

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | ID of the socket |

### 7.7.3 Response Syntax

| Response | Description |
|---|---|

| | |
|---|---|
| +SOCKLST:<SOCK_ID>,<PROTOCOL>,<RMT_ADDR>,<RMT_PORT>,<LCL_PORT> | Socket information |
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | Socket ID |
| <PROTOCOL> | Integer | The protocol in use on the socket<br><br>    1     UDP<br><br>    2     TCP |
| <RMT_ADDR> | String | The destination address associated with the socket (if any) |
| <RMT_PORT> | Integer | The destination port associated with the socket (if any) |
| <LCL_PORT> | Integer | The local port associated with this socket |

### 7.7.4    Examples:

| | |
|---|---|
| AT+SOCKLST | Command: list all open sockets |
| +SOCKLST:1,2,"",0,6000 | TCP listening socket on port 6000 |
| +SOCKLST:2,2,"192,168.0.1",4000,12345 | TCP connection with 192.168.0.1:4000 local port 12345 |
| +SOCKLST:3,1,"192.168.0.2",4000,12345 | UDP connection with 192.168.0.1:4000 local port 12345 |
| OK | Command completed successfully |

## 7.8    Send Data to Network +SOCKWR

### 7.8.1    Description

This command is used to send data over a socket.

For TCP sockets the socket must be connected to a remote address/port.

### 7.8.2    Command Syntax

AT+SOCKWR=<SOCK_ID>,<LENGTH>[,<DATA>]

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | The socket ID to transmit the data through |
| <LENGTH> | Integer | The length of the data to send (1 – 1500 bytes) |

| | | |
|---|---|---|
| <DATA> | String | The data to send in either ASCII or hexadecimal string format. If omitted the DCE will enter raw binary mode and will remain in that mode until the specified length of binary data has been received from the DTE. |

### 7.8.3 Response Syntax

| Response | Description |
|---|---|
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 7.8.4 Examples:

Terminal mode:

| | |
|---|---|
| AT+SOCKWR=1,7,"hello2u" | Send hello2u to 192.168.0.1:6000 via socket ID 1 |
| OK | |
| AT+SOCKWR=1,7,[68656c6c6f3275] | Send data given in hexadecimal string format |
| OK | 7 bytes successfully queued for transmission |

Non-Terminal mode, length unknown:

| | |
|---|---|
| AT+SOCKWR=1,0 | Prepare to send an unknown number of bytes (non-terminal mode) |
| #One fine day+++ | '#' followed by 12 bytes of data |
| OK | Command completed successfully |

Non-Terminal mode, length known:

| | |
|---|---|
| AT+SOCKWR=1,5 | Prepare to send an known number of bytes (non-terminal mode) |
| #hello | '#' followed by 5 bytes of data |
| OK | Command completed successfully |

## 7.9 Send Data to Specific Address (UDP only) +SOCKWRTO

### 7.9.1 Description

This command is used to send data to an arbitrary destination using the connectionless UDP protocol.

### 7.9.2 Command Syntax

AT+SOCKWRTO=<SOCK_ID>,<RMT_ADDR>,<RMT_PORT>,<LENGTH>[,<DATA>]

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | The socket ID to transmit the data through |
| <RMT_ADDR> | String | The destination address |

| <RMT_PORT> | Integer | The destination port |
|---|---|---|
| <LENGTH> | Integer | The length of the data to send (1 – 1500 bytes) |
| <DATA> | String | The data to send in either ASCII or hexadecimal string format. If omitted the DCE will enter raw binary mode and will remain in that mode until the specified length of binary data has been input. |

### 7.9.3 Response Syntax

| Response | Description |
|---|---|
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 7.9.4 Examples:

Terminal mode:

| | |
|---|---|
| AT+SOCKWRTO=1,"192.168.0.1",6000,7,"hello2u" | Send hello2u to 192.168.0.1:6000 via socket ID 1 |
| OK | |
| AT+SOCKWRTO=1,"192.167.0.1",6000,7,[68656c6c6f3275] | Send data given in hexadecimal string format |
| OK | 7 bytes successfully queued for transmission |

Non-Terminal mode, length unknown:

| | |
|---|---|
| AT+SOCKWRTO=1,"192.168.0.1",6000,0 | Prepare to send an unknown number of bytes (non-terminal mode) |
| #One fine day+++ | '#' followed by 12 bytes of data |
| OK | Command completed successfully |

Non-Terminal mode, length known:

| | |
|---|---|
| AT+SOCKWRTO=1,"192.168.0.1",6000,5 | Prepare to send an known number of bytes (non-terminal mode) |
| #hello | '#' followed by 5 bytes of data |
| OK | Command completed successfully |

## 7.10 Receive Data from Network +SOCKRXU, +SOCKRXT and +SOCKRD

### 7.10.1 Description

Two AECs present notification of data received by the DCE:

- +SOCKRXU indicates UDP data has been received.
- +SOCKRXT indicates TCP data has been received.

The DTE is responsible for retrieving the datagram/stream data via the +SOCKRD command.

For TCP sockets the DCE will indicate, via +SOCKRXT, the number of bytes of data which are currently available for reading via the +SOCKRD command. The DCE may issue multiple +SOCKRXT AECs as data is received. When requesting data via the +SOCKRD command the DTE may receive less data than request, the number of bytes provided by the DCE will be declared in the +SOCKRD response before the data is presented. The DTE may request less data than that declared by the +SOCKRXT AEC as being available, the remaining data will be available for subsequent reading.

For UDP sockets the DCE will indicate, via +SOCKRXU, the number of bytes of data which were received in the oldest datagram received by the DCE. Only a single +SOCKRXU will be issued by the DCE even if subsequent UDP datagrams are received. Reading data from the UDP socket via the +SOCKRD command will read and discard the current datagram, if less data is requested than was indicated by the +SOCKRXU AEC the remaining unread data in the datagram will be discarded.

### 7.10.2 AEC Syntax (TCP)

+SOCKRXT:<SOCK_ID>,<LENGTH>

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | The socket ID on which there is data to read |
| <LENGTH> | Integer | The number of bytes received |

### 7.10.3 AEC Syntax (UDP)

+SOCKRXU:<SOCK_ID>,<RMT_ADDR>,<RMT_PORT>,<LENGTH>

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | The socket ID on which there is data to read |
| <RMT_ADDR> | String | The sender's IP address |
| <RMT_PORT> | Integer | The sender's port number |
| <LENGTH> | Integer | The number of bytes received |

### 7.10.4 Command Syntax (+SOCKRD - read socket data)

AT+SOCKRD=<SOCK_ID>,<OUTPUT_MODE>,<LENGTH>

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | The socket number to read data from |
| <OUTPUT_MODE > | Integer | The format the DTE wishes to receive the data:<br><br>1    ASCII or hex String<br><br>2    Binary |
| <LENGTH> | Integer | The number of bytes the DTE wishes to read |

### 7.10.5 Response Syntax

| Response | Description |
|---|---|
| +SOCKRD:<SOCK_ID>,<LENGTH>,<DATA> | Socket read |
| OK | Successful response |

| ERROR:<ERROR_CODE> | Error response |
|---|---|

| Parameter Name | Type | Description |
|---|---|---|
| <SOCK_ID> | Integer | The socket ID the data is being retrieved from |
| <LENGTH> | Integer | The number of bytes being retrieved |
| <DATA> | Variable | The data in the format requested, either ASCII/hex string or binary |

### 7.10.6 Examples:

| +SOCKRXT:1,7 | AEC indicating TCP data available on socket ID 1, length 7 bytes |
|---|---|
| AT+SOCKRD=1,1,999 | Command – read 999 bytes, ASCII/hex string format, socket ID 1 |
| +SOCKRD:1,7,[466F6FC2A9626172] | Response – 7 bytes. Data is Foo©Bar |
| OK | Command completed |
| +SOCKRXU:4,"1.2.3.4",6000,7 | AEC: UDP data from 1.2.3.4:6000 available on socket 4, 7 bytes |
| AT+SOCKRD=4,2,999 | Command – read 999 bytes, binary format, socket ID 4 |
| +SOCKRD:1,7,<br><br>#<7_bytes_binary> | Response – 7 bytes from socket 4 to be presented in binary<br><br>Response – 8 bytes binary data (includes leading '#' marker) |
| OK | Command completed |

## 7.11 TLS Configuration +TLSC

### 7.11.1 Description

This command is used to read or set the TLS configuration. Multiple TLS configurations are possible, each configuration has an index starting with 1 for the first configuration. The index must be specified when using this command.

### 7.11.2 Command Syntax

| Command | Description |
|---|---|
| AT+TLSC=<conf_idx>[,<param_id>] | Read configuration |
| AT+TLSC=<conf_idx>,<param_id>,<param_val> | Set configuration |

### 7.11.3 Supported Parameters

| ID | Name & Default Value | Type | Description |
|---|---|---|---|

| 1 | <CA_CERT_NAME>: "" | String | CA certificate name |
|---|---|---|---|
| 2 | <CERT_NAME>: "" | String | Certificate name |
| 3 | <PRI_KEY_NAME>: "" | String | Private key name |
| 4 | <PRI_KEY_PASSWORD>: "" | String | Private key password |
| 5 | <SERVER_NAME>: "" | String | Server name |

### 7.11.4 Response Syntax

| Response | Description |
|---|---|
| +TLSC:<param_id>,<param_val> | Read response |
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

Note: The user needs to ensure that the certificates are a part of the image. Please check "8. Appendix A: Establishing a secured connection using AT Commands" for more details.

## 7.12 DNS Name Resolution +DNSRESOLV

### 7.12.1 Description
This command is used to resolve domain names via DNS.

### 7.12.2 Command Syntax
AT+DNSRESOLV=<TYPE><DOMAIN_NAME>

| Parameter Name | Type | Description |
|---|---|---|
| <TYPE> | Integer | Type of record:<br>1. A<br>2. AAAA |
| <DOMAIN_NAME> | String | Domain name to resolve |

### 7.12.3 Response Syntax

| Response | Description |
|---|---|
| OK | Successful response |
| ERROR:<ERROR_CODE> | Error response |

### 7.12.4 AEC Syntax (Success) +DNSRESOLV
+DNSRESOLV:<TYPE>,<DOMAIN_NAME>[,<QUERY_RESP>]

| Parameter Name | Type | Description |
|---|---|---|

| <TYPE> | Integer | Type of record: <br> 1. A Record <br> 2. AAAA Record |
|---|---|---|
| <DOMAIN_NAME> | String | Original domain name requested |
| <QUERY_RESP> | String | Query response received |

### 7.12.5 AEC Syntax (Failure) +DNSERROR

+DNSERROR:<ERROR_CODE>

| Parameter Name | Type | Description |
|---|---|---|
| <ERROR_CODE> | Integer | Error response |

### 7.12.6 Examples:

| | |
|---|---|
| `AT+DNSRESOLV=1,"www.example.com"` | Request A record for www.example.com |
| `OK` | Command completed |
| `+DNSRESOLV:1,"www.example.com","192.168.0.1"` | A record response for www.example.com |

## 7.13 Ping Link Check +PING

### 7.13.1 Description

This command sends a ping (ICMP Echo Request) to the target address.

### 7.13.2 Command Syntax

AT+PING=<TARGET_ADDR>,[<IP_PROTOCOL_VERSION>]

| Parameter Name | Type | Description |
|---|---|---|
| <TARGET_ADDR> | String | IP address or domain name of target |
| <IP_PROTOCOL_VERSION> | Integer | IP protocol version: <br> 1. IPv4 <br> 2. IPv6 |

### 7.13.3 Response Syntax

| Response | Description |
|---|---|
| `OK` | Successful response |
| `ERROR:<ERROR_CODE>` | Error response |

### 7.13.4 AEC Syntax

+PING:<IP_ADDRESS>,<RTT>

| Parameter Name | Type | Description |
|---|---|---|
| <IP_ADDRESS> | String | IP address of the target responding |

| <RTT> | Integer | Round trip time (in milliseconds) |
|-------|---------|-----------------------------------|

### 7.13.5 Examples:

| | |
|---|---|
| `AT+PING="www.example.com"` | Ping the host www.example.com |
| `OK` | Command completed |

# 8.    Appendix A: Establishing a secured connection using AT Commands

In case the user wants to establish a connection with a Server, one needs to ensure that the server certificate needed for verification of the server during the TLS negotiations is loaded into the device.

When establishing a secured TCP connection:

1.  Once the Wifi connection has been established, load a TLS Certificates using AT+LOADCERT:
    a.  *a. AT+LOADCERT=1500, "app_client_cert_der_mosquitto_org"*
2.  Establish the normal TCP connection with the server first
    a.  *AT+SOCKO=2*
    b.  *AT+SOCKBR=1,"5.196.95.208",8883*
3.  Once the connection has been established, to start the secured connection
    a.  *AT+SOCKTLS=1,1*
4.  Check more details in the section 7.11.  TLS Configuration +TLSC and 7.6.  Upgrade Socket to TLS +SOCKTLS on the usage of respective AT Commands.