xE910 – WE866 Bundling Application Note

80541NT11528A Rev.17 – 2018-01-30

TELIT
TECHNICAL
DOCUMENTATION

[04.2016]

SPECIFICATIONS ARE SUBJECT TO CHANGE WITHOUT NOTICE

NOTICE

While reasonable efforts have been made to assure the accuracy of this document, Telit assumes no liability resulting from any inaccuracies or omissions in this document, or from use of the information obtained herein. The information in this document has been carefully checked and is believed to be reliable. However, no responsibility is assumed for inaccuracies or omissions. Telit reserves the right to make changes to any products described herein and reserves the right to revise this document and to make changes from time to time in content hereof with no obligation to notify any person of revisions or changes. Telit does not assume any liability arising out of the application or use of any product, software, or circuit described herein; neither does it convey license under its patent rights or the rights of others.

It is possible that this publication may contain references to, or information about Telit products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that Telit intends to announce such Telit products, programming, or services in your country.

COPYRIGHTS

This instruction manual and the Telit products described in this instruction manual may be, include or describe copyrighted Telit material, such as computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and its licensors certain exclusive rights for copyrighted material, including the exclusive right to copy, reproduce in any form, distribute and make derivative works of the copyrighted material. Accordingly, any copyrighted material of Telit and its licensors contained herein or in the Telit products described in this instruction manual may not be copied, reproduced, distributed, merged or modified in any manner without the express written permission of Telit. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit, as arises by operation of law in the sale of a product.

COMPUTER SOFTWARE COPYRIGHTS

The Telit and 3rd Party supplied Software (SW) products described in this instruction manual may include copyrighted Telit and other 3rd Party supplied computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and other 3rd Party supplied SW certain exclusive rights for copyrighted computer programs, including the exclusive right to copy or reproduce in any form the copyrighted computer program. Accordingly, any copyrighted Telit or other 3rd Party supplied SW computer programs contained in the Telit products described in this instruction manual may not be copied (reverse engineered) or reproduced in any manner without the express written permission of Telit or the 3rd Party SW supplier. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit or other 3rd Party supplied SW, except for the normal non-exclusive, royalty free license to use that arises by operation of law in the sale of a product.

USAGE AND DISCLOSURE RESTRICTIONS

I. License Agreements

The software described in this document is the property of Telit and its licensors. It is furnished by express license agreement only and may be used only in accordance with the terms of such an agreement.

II. Copyrighted Materials

Software and documentation are copyrighted materials. Making unauthorized copies is prohibited by law. No part of the software or documentation may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, without prior written permission of Telit

III. High Risk Materials

Components, units, or third-party products used in the product described herein are NOT fault-tolerant and are NOT designed, manufactured, or intended for use as on-line control equipment in the following hazardous environments requiring fail-safe controls: the operation of Nuclear Facilities, Aircraft Navigation or Aircraft Communication Systems, Air Traffic Control, Life Support, or Weapons Systems (High Risk Activities"). Telit and its supplier(s) specifically disclaim any expressed or implied warranty of fitness for such High Risk Activities.

IV. Trademarks

TELIT and the Stylized T Logo are registered in Trademark Office. All other product or service names are the property of their respective owners.

V. Third Party Rights

The software may include Third Party Right software. In this case you agree to comply with all terms and conditions imposed on you in respect of such separate software. In addition to Third Party Terms, the disclaimer of warranty and limitation of liability provisions in this License shall apply to the Third Party Right software.

TELIT HEREBY DISCLAIMS ANY AND ALL WARRANTIES EXPRESS OR IMPLIED FROM ANY THIRD PARTIES REGARDING ANY SEPARATE FILES, ANY THIRD PARTY MATERIALS INCLUDED IN THE SOFTWARE, ANY THIRD PARTY MATERIALS FROM WHICH THE SOFTWARE IS DERIVED (COLLECTIVELY “OTHER CODE”), AND THE USE OF ANY OR ALL THE OTHER CODE IN CONNECTION WITH THE SOFTWARE, INCLUDING (WITHOUT LIMITATION) ANY WARRANTIES OF SATISFACTORY QUALITY OR FITNESS FOR A PARTICULAR PURPOSE.

NO THIRD PARTY LICENSORS OF OTHER CODE SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND WHETHER MADE UNDER CONTRACT, TORT OR OTHER LEGAL THEORY, ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE OTHER CODE OR THE EXERCISE OF ANY RIGHTS GRANTED UNDER EITHER OR BOTH THIS LICENSE AND THE LEGAL TERMS APPLICABLE TO ANY SEPARATE FILES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The AZ C project integrates the lwIP stack. lwIP is a small independent implementation of the TCP/IP protocol suite that has been initially developed by Adam Dunkels. lwIP is licenced under the BSD licence:

APPLICABILITY TABLE

PRODUCTS

- UE910 SERIES
- HE910 SERIES
- LE910 V2 SERIES
- UE866 SERIES
- WE866

HSPA:

Application Version

WE.00.109-12

Stream Version

12.00.xx8-B016

LTE:

Application Version

WE.00.209-13

Stream Version

20.00.xx4

CONTENTS

1. INTRODUCTION

1.1. Scope

This document gives an overview of xE910 in bundle with WE866 It explains how to connect a xE910 module + WE866 Short range module

1.2. Audience

This document is intended only for Telit customers that wants to set up a bundling application between xE910 + WE866.

1.3. Contact Information, Support

For general contact, technical support services, technical questions and report documentation errors contact Telit Technical Support at:

- TS-EMEA@telit.com
- TS-AMERICAS@telit.com
- TS-APAC@telit.com

Alternatively, use:

http://www.telit.com/support

For detailed information about where you can buy the Telit modules or for recommendations on accessories and components visit:

http://www.telit.com

Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.

Telit appreciates feedback from the users of our information.

1.4. Text Conventions

Danger – This information MUST be followed or catastrophic equipment failure or bodily injury may occur.

Caution or Warning – Alerts the user to important points about integrating the module, if these points are not followed, the module and end user equipment may fail or malfunction.

Tip or Information – Provides advice and suggestions that may be useful when integrating the module.

All dates are in ISO 8601 format, i.e. YYYY-MM-DD.

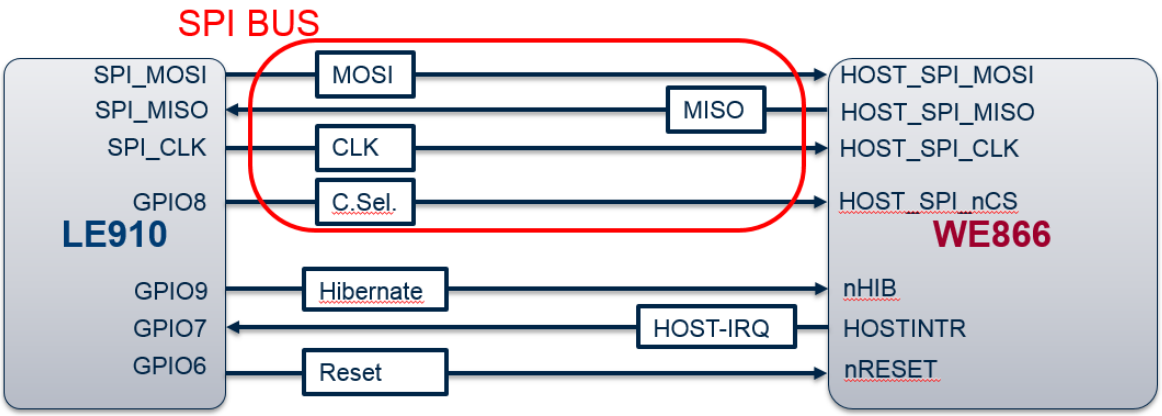
1.5. Related Documents

1. Telit AppZone C User Guide, 80496ST10722A
2. Telit EVK2 User Guide, 1VV0300704 Rev. 19
3. WE866 Hardware user guide
4. Telit USB Driver Installer User Guide, 1V0301164 Rev. 7
5. Telit AT Commands Reference Guide, 80000ST10025a
6. Telit 3G Modules AT Commands Reference Guide, 80378ST10091A
7. Telit LE910 V2 Series AT Commands Reference Guide, 80446ST10707A

2. BUNDLE OVERVIEW

2.1. Overview

Telit Bundle solution integrates Cellular and WiFi technologies, allowing customers to manage both through a single module, with a single communication port and a single communication protocol.



Telit WE866 Bundle will make the connection between Cellular and WiFi easy.

Warning – on **UE866-EU only GPIO6 and GPIO7 can be used for HOST-IRQ signal**

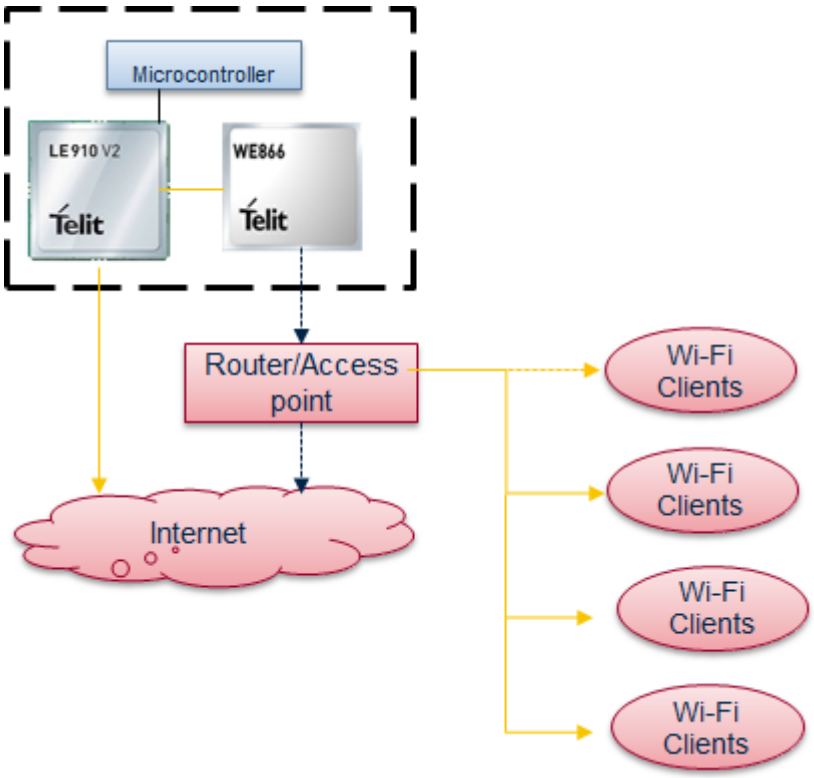
---

3. SW ARCHITECTURE

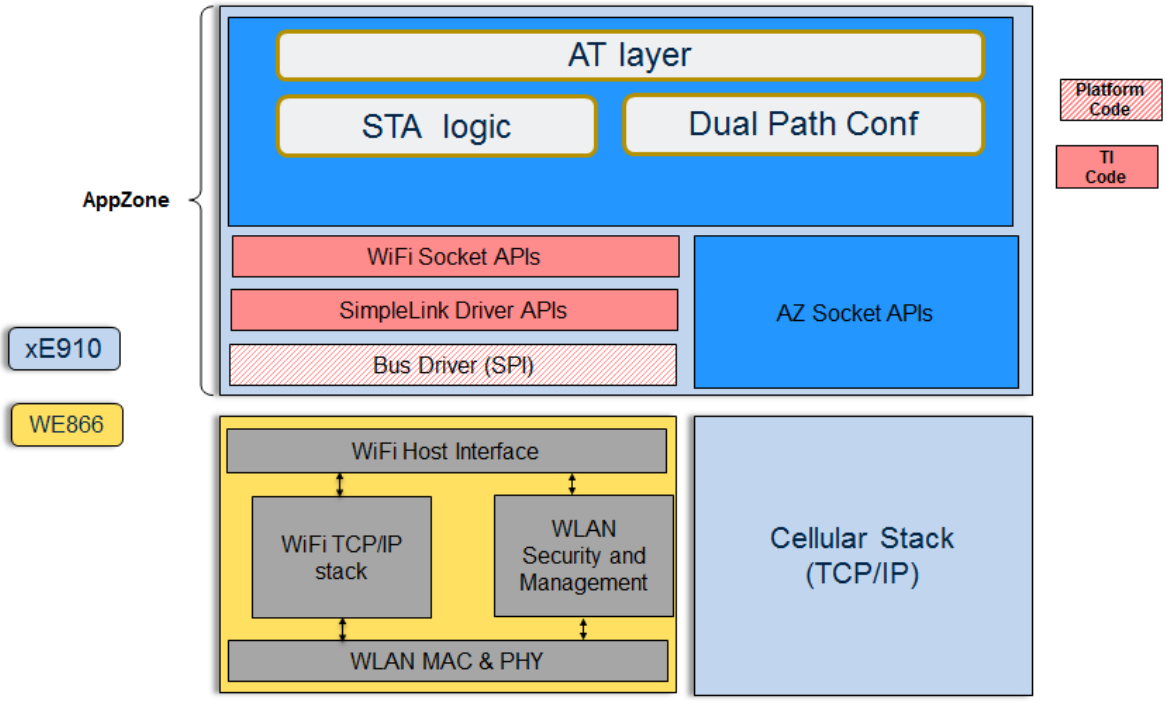
3.1. STATION

In this scenario the external MCU can select if the data can be sent to WiFi network or the cellular one.

In this case for the WiFi interface the module is acting as a station

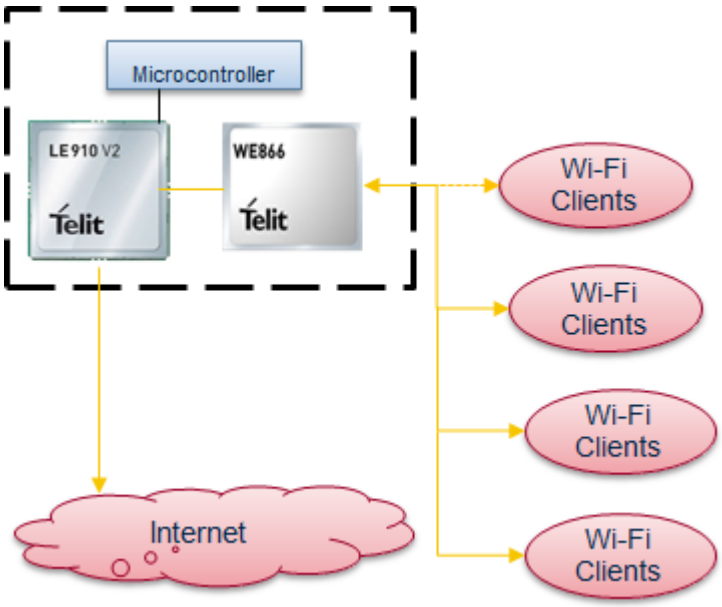


Below it is described the SW architecture running on the xE910 module.

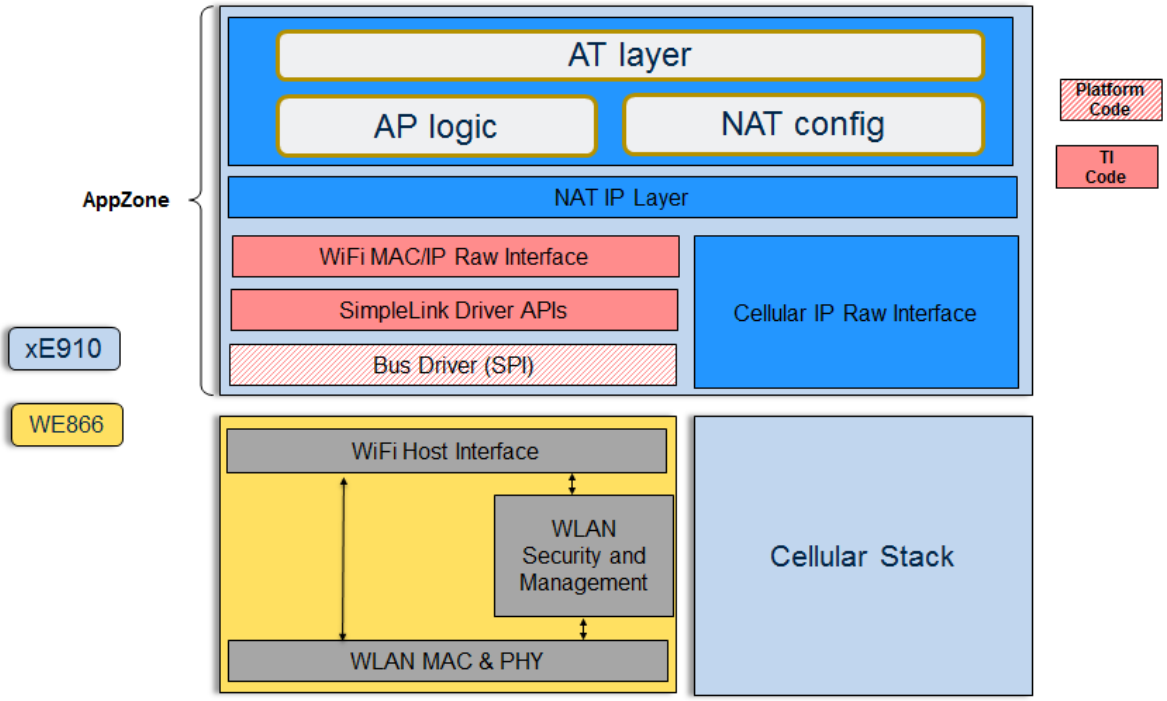


3.2. HOTSPOT

The WiFi module broadcasts a local network for stations to connect to. Stations can open connections (e.g. HTTP, SMTP, …) and send data. The cellular forward the IP packets to the cellular radio (NAT). **AZ C example in under implementation for HE910 and LE910 V2.**



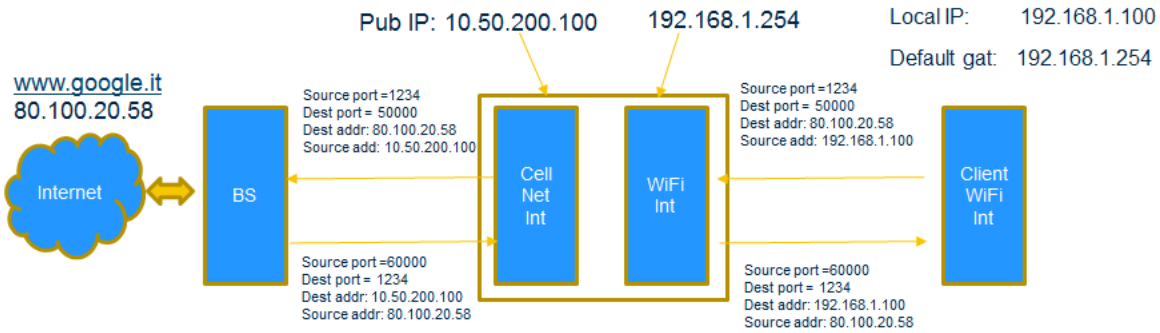
In this case the SW architecture is below



The WE866 is configured in MAC raw mode. All the packet MAC header include will be routed to the cellular module.

On AppZone framework a NAT layer of the IP packet has been implemented and they are routed through the cellular network. On the AppZone layer an AT layer has been implemented in order to manage both cellular and WiFi interface in order to enable/disable the HOTSPOT functionality.

Below an example of natting has been showed



Warning – On LTE platform, AT#MTUSIZE=1500 **MUST be sent before starting the hot spot functionality** using AT#WIFISTART=1 command (see section 5.1)

4. SOFTWARE SETUP

Before starting any of the next steps please download from download zone the XFP and TATC tool.

Moreover it is strongly suggest to read the AppZone user guide at the below link

http://www.telit.com/fileadmin/AppZone_Guide/az-c-user-guide-V3/index.html

4.1. Module firmware flashing

Before start the flashing you have to install the USB module drivers.

Download the XFP tool and the USB driver from download zone

To flash the module, you will need the XFP tool, and possibly the USB drivers if you want to flash using USB cable.

You can find the XFP tool here:

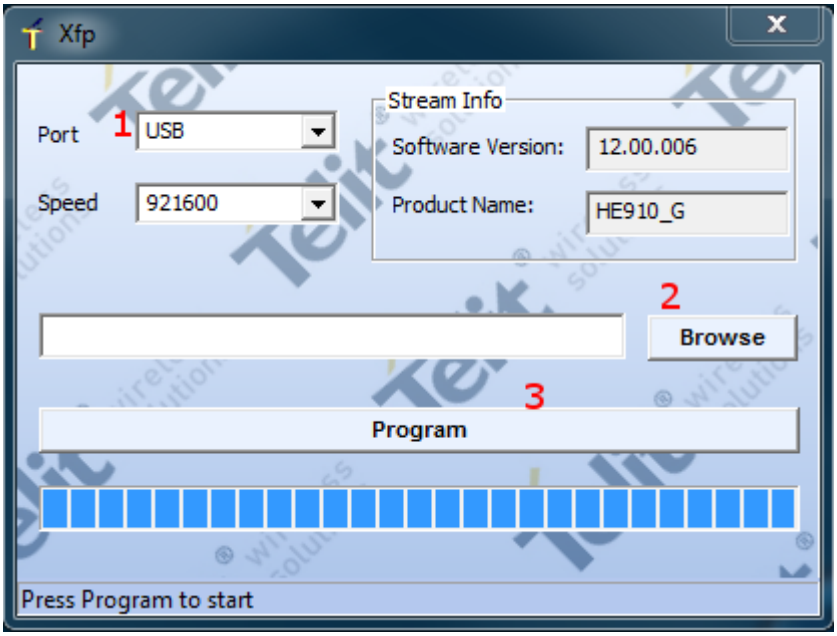
https://tcloud.telit.com/public.php?service=files&t=8f9a292cfedeebf88de75816167548b5

And the Telit_USB_Driver_Win_Desktop_UF.00.05 here:

https://tcloud.telit.com/public.php?service=files&t=56ee1f87e5eb1f3ef45e81143b64a5b1

(You can select to install the drivers for all devices from the installer prompt, or just the family of the module you are using)

Once you have the software installed, you can run XFP. You will see something like below



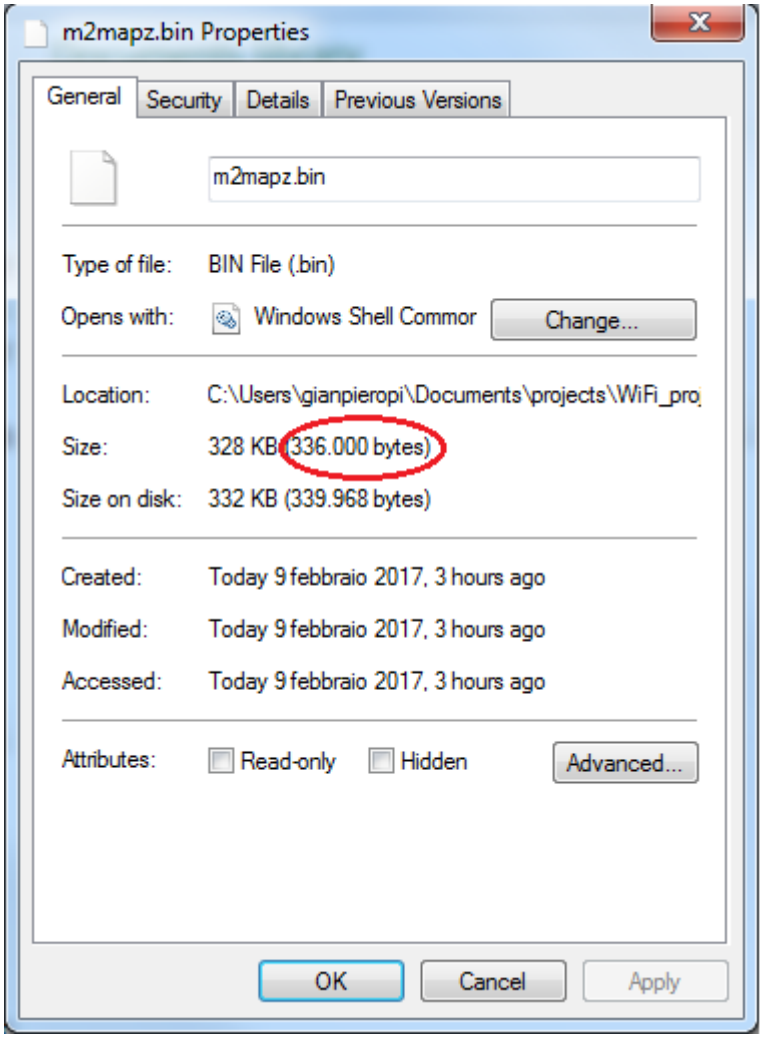
Steps to follow (refer to the image above):

1) Select the USB connection and 921600 bps speed (if not present, simply type it manually)
2) Browse to the .bin file of the firmware stream.
2bis) Be sure that the module is turned OFF (for example, press the Reset/Hw shutdown button, or unplug the power supply) and the USB cable is connected to the interface
3) Click on the Program button, the tool will search for the module USB interface.
4) Turn on the module pressing the **ON OFF** button; the tool will link it and start the flashing procedure. It will take a few minutes.
5) Once it is finished, the module will turn off. Turn it on again, its memory will be formatted on the first startup, and then the module will turn OFF again.
6) Turn the module on again. It is now ready.

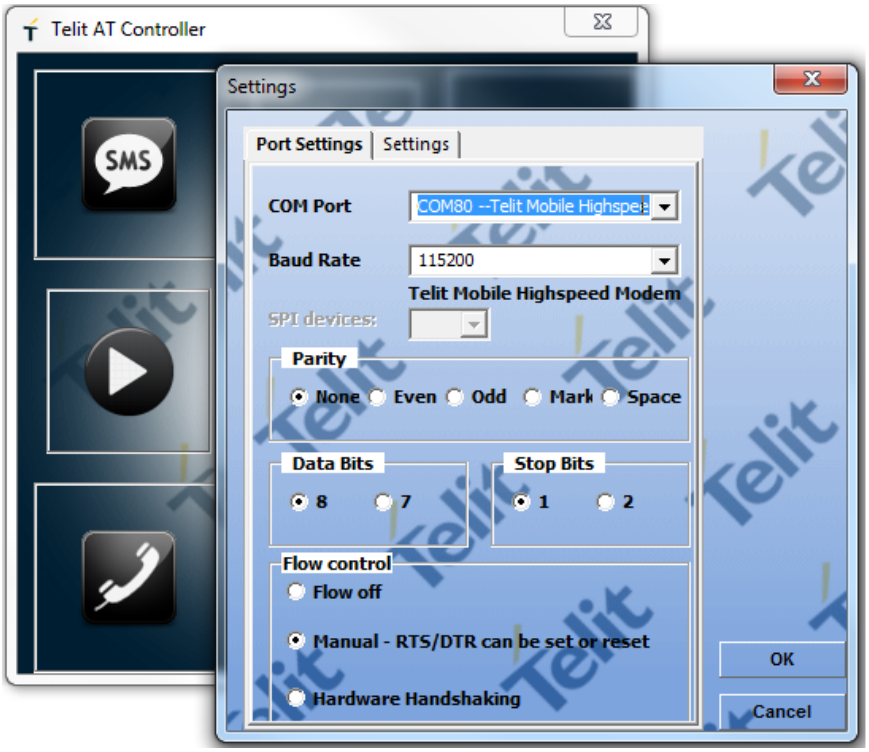
4.2. WiFi AppZone Application programming.

Before starting the load of the application the first step is to have the dimension of the application firmware in bytes.

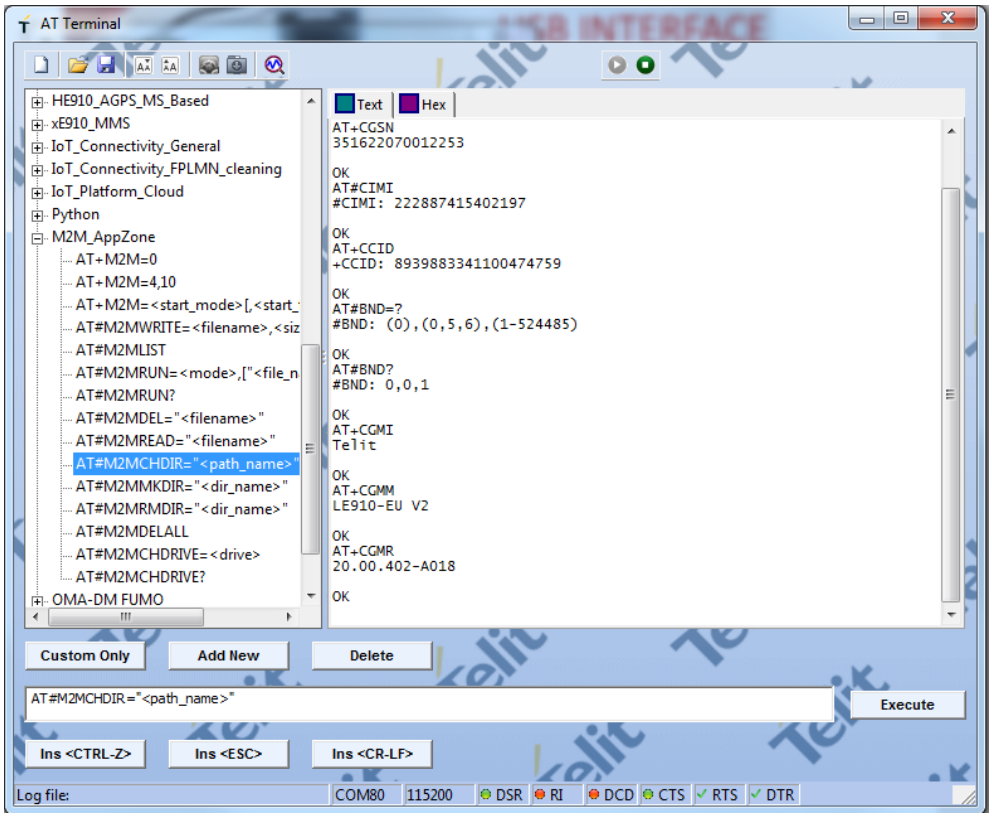
In order to have it, using the file browser go in the folder where the application is located and push right click on the application and as show in the picture memorize the size of the binary. In the example it is 336000 and filename is m2mapz.bin



After that, open TATC tool and connect it to the USB0 of the module and go to the tab M2M_AppZone. See below

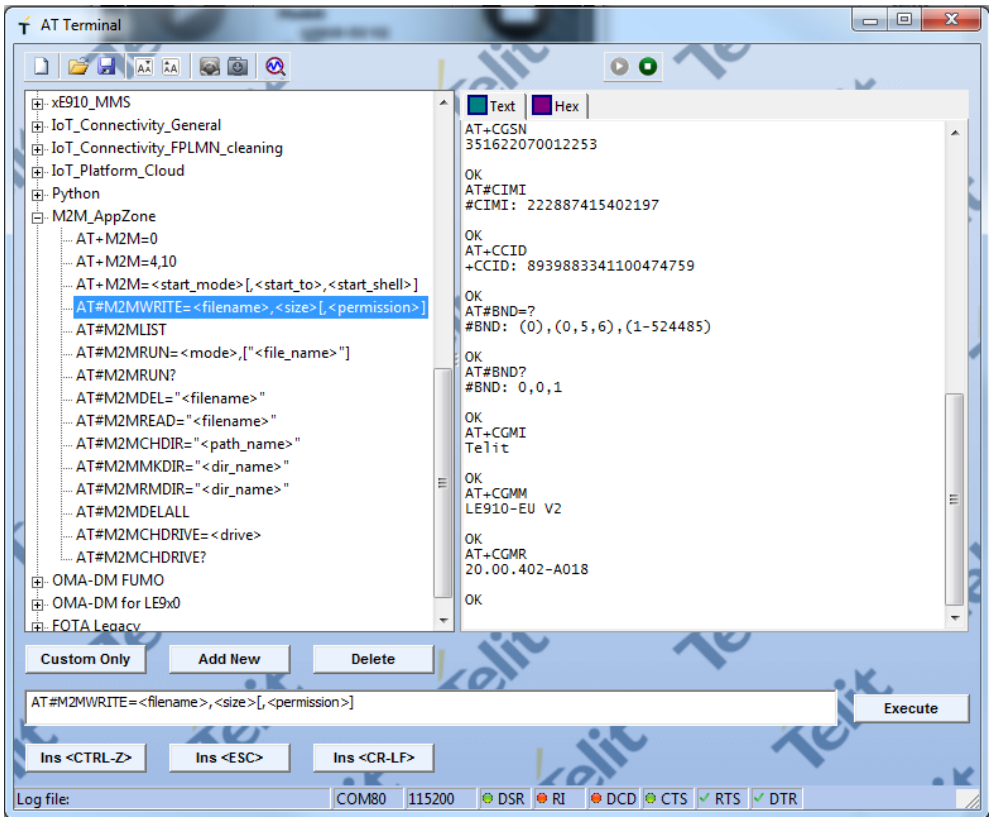


Select the AT#M2MCHDIR and complete the command with AT#M2MCHDIR=”MOD”. This will mode in the MOD directory of the file system

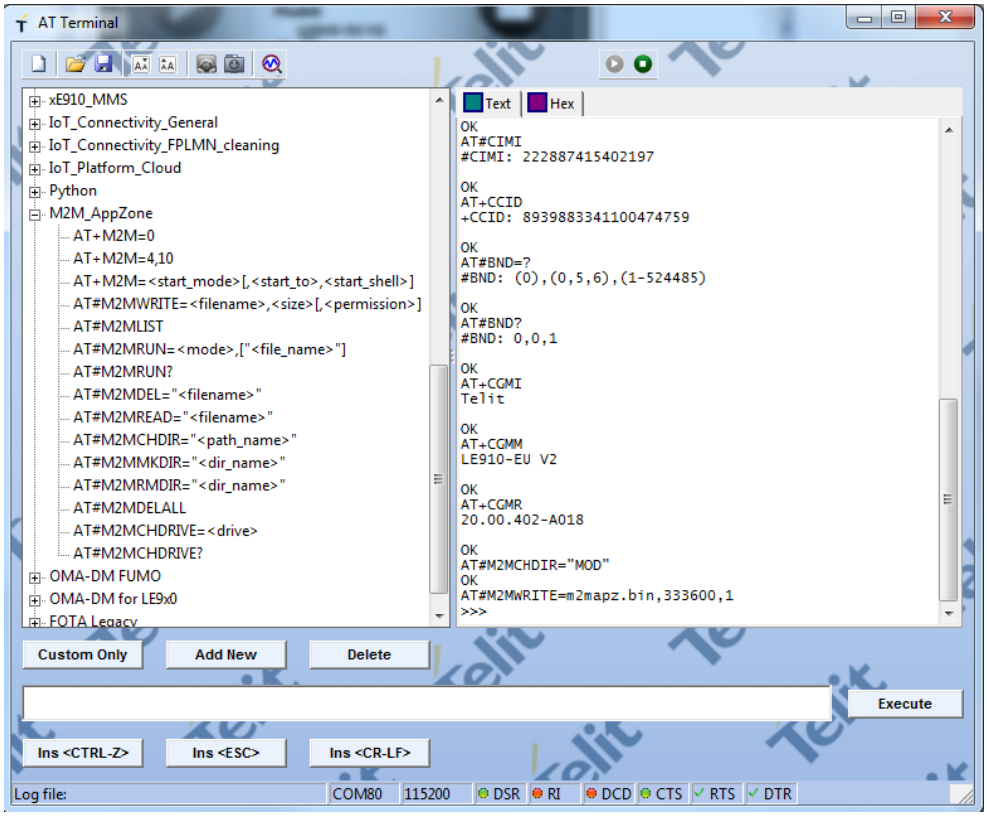


Now select the AT#M2MWRITE command and complete with AT#M2MWRITE=<file name>,<size>,<permission>.

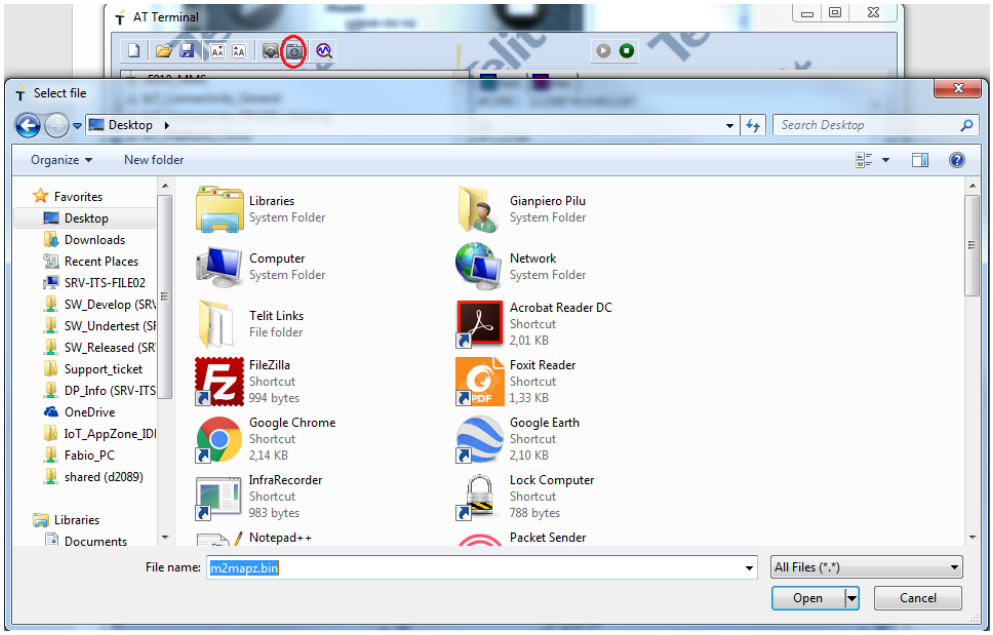
In this example AT#M2MWRITE=m2mapz.bin,336000,1



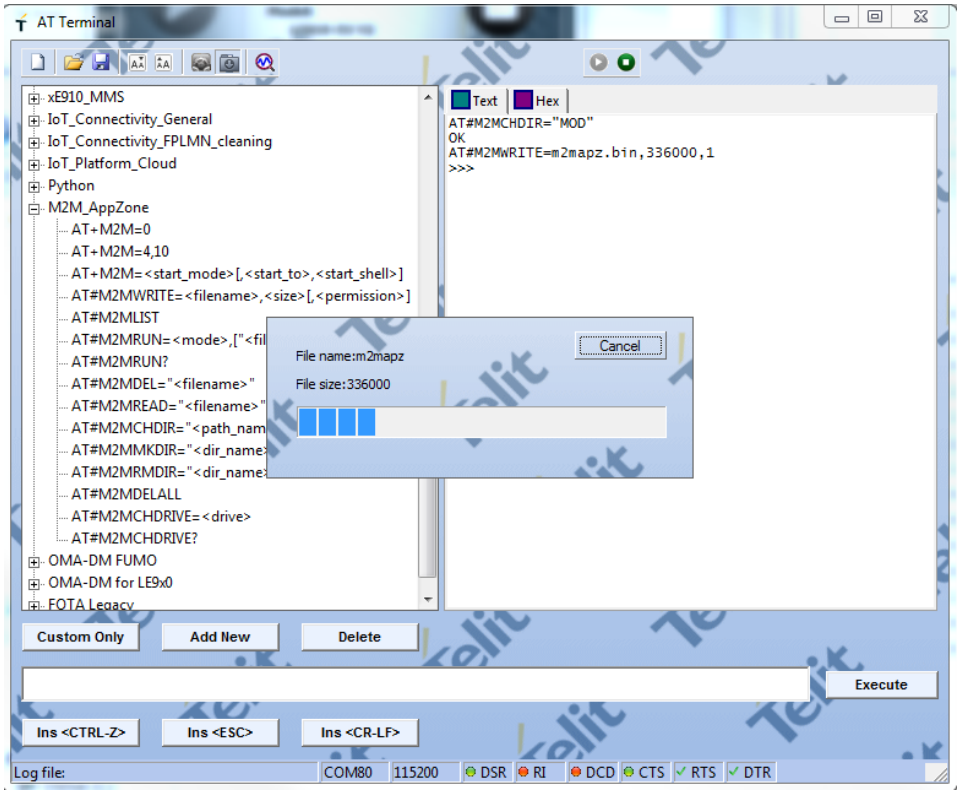
After you send the AT#M2MWRITE command, a >>> prompt will be showed like the image below:



Push the transfer button indicated with a circle red in next image select the application to download and push ok and the transfer will start.

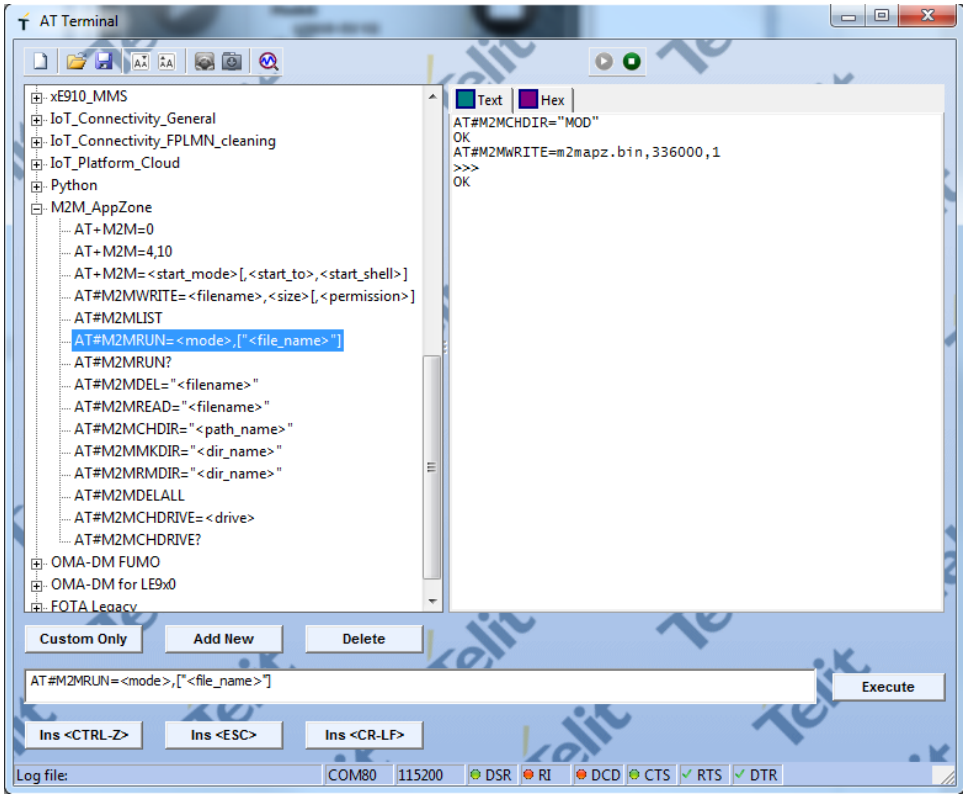


The transfer pop-up will appear



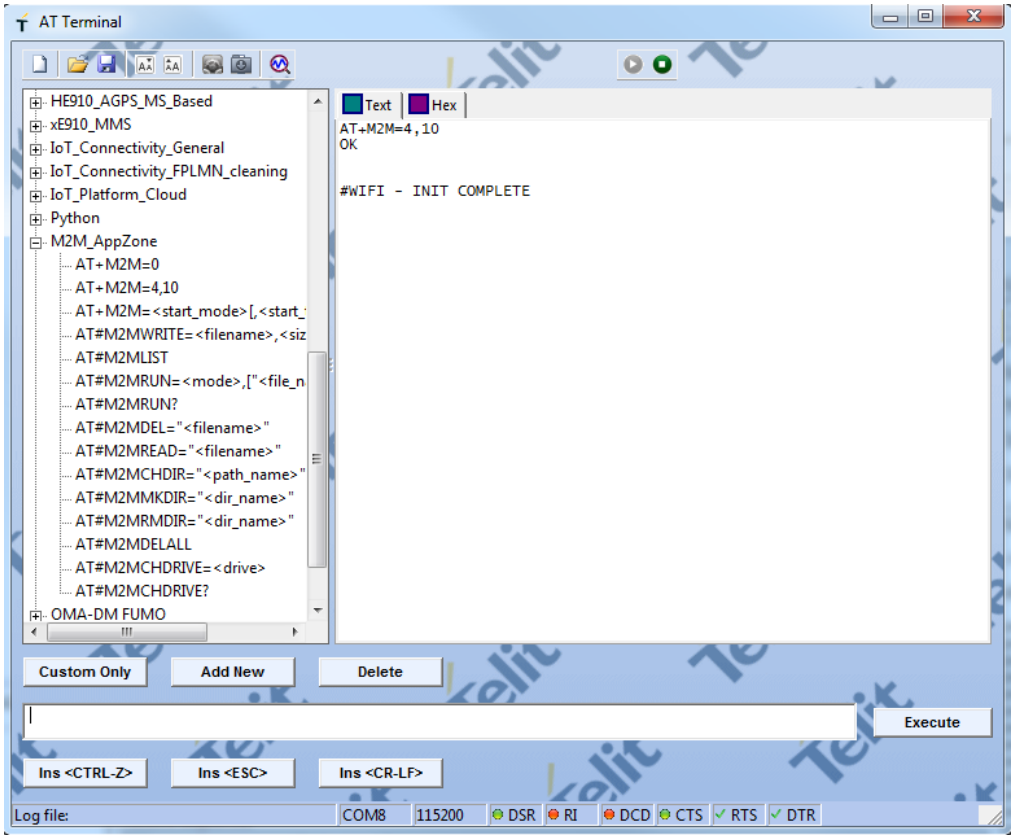
After the download will finish , the application need to be set as the application that need to start at the next start-up.

Select the AT#M2MRUN and complete with AT#M2MRUN=2, <filename> , in the example is AT#M2MRUN=2,m2mapz.bin



Now the application can start and for this select AT+M2M command and complete it with AT+M2M=4,10 this means that the application will start 10 seconds after start-up.

See below the next image.



Now the application started and you can use it in STATION or HOTSPOT module.

Follow the instruction in the next paragraph for the details

Warning – If the **CS1801a** bundling board is used, the following AT commands must be sent at the first boot to preset ONOFF signal and enable for level shifter

AT#GPIO=3,1,1,1

AT#GPIO=4,0,1,1

Note: The application trace log outputs on USB3.

5. AT COMMANDS

5.1. General commands

5.1.1. #WIFISTART

#WIFISTART – enable/disable WLAN	
AT#WIFISTART=<mode>	Set command enable/disable WLAN Parameters: **<mode> -** int type**,** status mode. 0 – disable 1 – enable Note: enable WLAN will start it as Station mode by default unless configure prior to start (**AT#WIFIMODE**) or configuration was stored in NVM with **AT#WIFISAV**. Note: The command will return immediately and generate URC messages depending on the configuration and events. Immediate response can be: AP mode **INITIALIZING AP..<CRLF><CRLF>OK<CRLF>** Station Mode **CONNECTING TO AP..<CRLF><CRLF>OK<CRLF>** AP Mode URCs: #WIFI: WAITING STATIONS (AcT **X**) Will be printed when the AP is ready to accept connections from stations. **X** value is the Access Technology in use for Cellular connection #WIFI: A Station Connected to AP #WIFI: A Station disconnected from AP #WIFI: AP OFF Station Mode URCs: #WIFI: STA Connected to external AP #WIFI: STA OFF
AT#WIFISTART?	Read command returns the currently WLAN status in the format: #WIFISTART: <mode> **Where:** **<mode>** 0 – WLAN off 1 – WLAN on
AT#WIFISTART=?	Test command returns the supported range of values for parameter **<mode>**.

5.1.2. #WIFIMODE

#**WIFIMODE** – change mode: access point/client	
AT#WIFIMODE=<mode> [,<auto-start>]	Set command change WLAN mode Parameters: **<mode> -** operation mode. 0 – Access point mode 1 – Station mode <auto-start> - auto start configuration. 0 – No autostart 1 – AutoStart at next reboot (equivalent to call AT#WIFISTART=1 Note: to apply AutoStart configuration, AT#WIFISAV command must be called
AT#WIFIMODE?	Read command returns the currently mode status in the format: **#WIFIMODE: <mode>,<auto-start>** With the same ranges and values of set command.
AT#WIFIMODE=?	Test command returns the supported range of values for parameters <**mode**> and **<auto-start>.**

5.1.3. #WIFISCAN

#**WIFISCAN**– Shows a list of available networks	
AT#WIFISCAN=<mode>	Set command will return the parameters for available WiFi networks. Parameters: **<mode>** 0 – Scan visible networks only 1 – Scan also hidden networks #WIFISCAN: SSID: <**SSID**> SEC TYPE: <**security_type**> \tBSSID: <**BSSID**> \tRSSI: <**RSSI**>dBm <**Hidden**> for every retrieve Network, where: <**SSID**>: Visible Access Point SSID <**security_type**>: One of “OPEN” “WEP” “WPA” “WPA2” <**BSSID**>: Access Point MAC address <**RSSI**>: AP received signal level in dBm <**Hidden**> “(HIDDEN)” if the WiFi networks has a hidden SSID, empty string otherwise.
AT#WIFISCAN	Execution command returns the same output of set command with mode= 0 (do not print Hidden networks)

AT#WIFISCAN=?	Test command returns the supported range of values for <**mode**> parameter.

5.1.4. #WIFIPWR

#**WIFIPWR** – Set Tx power level	
AT#WIFIPWR=<mode>, <dB>	Set command sets the Tx power for either AP and STATION modes. Parameters: **<mode> -** 0 – Access point mode 1 – Client (Station) mode <**dB**> - Number between 0-15, as dB offset from maximum power - 0 will set maximum power
AT#WIFIPWR?	Read command returns the current Tx power **#WIFIPWR**: **< mode >,< dB >** With the same ranges and values of set command.
AT#WIFIPWR=?	Test command reports supported values for parameters < **mode** > and < **dB** >.

5.1.5. #WIFIGPIO

#**WIFIGPIO** – change the Cellular -WiFi communication GPIOs	
AT#WIFIGPIO=<nHib>, <nReset>, <HIrq>, <SPI_nCS>	Set command changes GPIOs configuration Parameters: **<nHib>** - WiFi module hibernate pin **<nReset>** - WiFi module reset pin <**HIrq**> - WiFi Host Interrupt signal **<SPI_nCS>** - SPI chip select signal Default values are the ones for bundling board: **#WIFIGPIO:** 9,6,7,8 Notes: To keep the previous value, set a parameter to 0 e.g. to change SPI_nCS only: **AT#WIFIGPIO=0,0,0,3**
AT#WIFIGPIO?	Read command returns the currently GPIOs configuration in the format: **#WIFIGPIO:** nHib,nReset,HIrq,SPI_nCS,
AT#WIFIGPIO=?	Test command returns the supported range of values for all GPIOs.

5.1.6. #WIFISWVER

#WIFISWVER – Software version	
AT#WIFISWVER	Execution command returns the WiFi application software version in format **WE.MM.ppm** Where **MM** is the major release number **m** is the minor release number **pp** is the platform (10: 3G, 20: 4G)
AT#WIFISWVER?	Read command returns WiFi application software version, as the execution command

5.1.7. #WIFISAV

#WIFISAV – Store current configuration	
AT#WIFISAV	Execution will store the current configuration in NVM. Parameters are stored in groups: 1) All of AP and STA CFG commands, AT#WIFIMODE parameters 2) Port Forwarding entries currently set. 3) DHCP IP reservation entries currently set. 4) Socket Options (see #WSCFG and #WSCFGEXT commands)
AT#WIFISAV=<op> **[,<file>]**	Set command will execute the requested operation Parameters: **<op>**: only 0 is supported (delete Application configuration group) <**file**>: select which configuration file to be deleted. Admitted values: 0 – Delete ALL configuration groups 1 – Delete Application configuration group only 2 – Delete Port Forwarding configuration group only 3 – Delete DHCP IP reservation configuration group only 4 – Delete socket options group only
At#WIFISAV=?	Test command reports supported values for the parameters.

5.1.8. #WE

#**WE** – change the echo level setup	
AT#WE=<level>	Set command changes echo level configuration Parameters: **<level>** **0 –** disable echo for custom commands (as #WSSEND 1 – enable echo for custom commands Notes: Setting is not stored in NVM
AT#WE	Execution command disables echo (same as **#WE=0**)
AT#WE=?	Test command returns the supported range of values for parameter

5.2. Access Point mode commands

5.2.1. #WIFIAPCLIENTS

#**WIFIAPCLIENTS**– List of connected clients	
AT#WIFIAPCLIENTS	Execution command retrieves the list of connected clients (up to 4 possible) and returns it in the formats **#WIFIAPCLIENTS - Connected Stations Info:** **No Station Connected** If no station is connected. **#WIFIAPCLIENTS - Connected Stations Info:** **[Host: <HostName> |]<MacAddress>[| Leased IP: <LeasedIp>]** If one or more (up to 4) stations are connected. Where <**HostName**> is the Station hostname, provided to DHCP server during connection <**MacAddress**> is Station MAC Address <**LeasedIp**> is the IP address the DHCP server leased to connected station. Note: -In case HostName is not provided, the first part of the row will not be printed -In case the station is using static IP, the Leased IP part of the row will not be printed.

5.2.2. #WIFIAPMAC

#**WIFIAPMAC**– Retrieve the WiFi MAC address	
AT#WIFIAPMAC	Execution command retrieves the WiFi Module MAC address in the format: **#WIFIAPMAC: XX:XX:XX:XX:XX:XX**

5.2.3. #WIFIAPPHYCFG

#WIFIAPPHYCFG– Set Physical configuration of AP	
AT#WIFIAPPHYCFG=<SSID>, **<Security>,** **<PWD>,** **<Region>,** **<Channel>** **[,<Hidden>]**	Set command sets the Access Point's Physical Parameters: Parameters: **< SSID > -** Alphanumeric string containing up to 32 bytes (default: **Telit_<last 4 cellular IMEI cyphers>**) **< Security >** Integer values 0-2 0 – Open 1 – WEP 2 – WPA & WPA2 (factory default) **< PWD > -** Alphanumeric characters WPA: 8 to 63 characters WEP: 5 to 13 characters (default: **password**) **< Region > -** String. Available options: "US" "EU" (factory default) "JP" **< Channel > -** Integer values. Ranges depend on Region: **"US": 1-11** **"EU": 1-13** **"JP": 1-14** (default: 6) **<Hidden>** - AP SSID option 0 – Visible (factory default) 1 – Hidden with length 0 2 – Hidden with original length
AT#WIFIAPPHYCFG?	Read command returns the current Physical Parameters: **#WIFIAPPHYCFG**: **< SSID** >,**< Security** >,**< PWD** >,**< Region** >,**< Channel** >,**<Hidden>**
AT#WIFIAPPHYCFG=?	Test command reports supported values for the parameters.

5.2.4. #WIFIAPNETCFG

#WIFIAPNETCFG – Set Network configuration of AP (WiFi will reboot to complete configuration)	
AT#WIFIAPNETCFG=<ip> **[,< DNS >** **[,<ip_start >** **[,<ip_last >** **[,<ip_lease_time >]]]]**	Set command sets the Access Point's Network Parameters: **< ip > -** IPV4 address, string dotted format (xxx.xxx.xxx.xxx) **< DNS >** - IPV4 DNS Server, string dotted format (default: 0) **< ip_start> -** DHCP start address last octet (1-255) (default: 10) **<ip_last> -** DHCP last address last octet (1-255) (default: 20) **<ip_lease_time> -** IP address lease time (seconds) (default: 3600) 32 bits unsigned integer. **Notes**: **< DNS** >- if not set, cellular DNS will be used

	The DSN address is updated only after a AT#WIFISTART=1 is executed, so the Read command will return the previously set value. WiFi Network mask is set to 255.255.255.0
AT#WIFIAPNETCFG?	Read command returns the current Network Parameters: **#WIFIAPNETCFG**: < **ip** >,< **DNS** >,< **ip_start** >,< **ip_last** > ,< **ip_lease_time** >
AT#WIFIAPNETCFG=?	Test command reports supported values for the parameters.

5.2.5. #WIFIAPPDPCFG

#**WIFIAPPDPCFG** – Set AP mode	
AT#WIFIAPPDPCFG=<apn> **[,<username>]** **[,<password>]** **[,<cid>]**	Execution command set the following parameter in the AP mode: Parameters: <**apn**> - PDP context APN to be used in the connection, string type, max length: 139 bytes <**username**> - PDP context username to be used in the connection, string type, max length: 139 <**password**> - PDP context password to be used in the connection, string type, max length: 139 <**cid**> PDP context ID to be used for the connection. Defaults to AppZone internal CID. Refer to AT+CGDCONT=? Command for CID range on cellular platform in use.
AT#WIFIAPPDPCFG?	Read command returns the current cellular's parameters and IP address: **#WIFIAPPDPCFG**: <**apn**>,<**username**>,<****>,<**cid**>,"**<IP_address>"** Where <**IP_address**> will be 0.0.0.0 if cellular connection is not active, the current IP address otherwise
AT#WIFIAPPDPCFG=?	Test command reports supported values for the parameters.

5.2.6. #WIFIAPMAXSTA

#**WIFIAPMAXSTA**– Set Physical configuration of AP	
AT#WIFIAPMAXSTA= **<max_sta>**	Set command sets the Access Point's max number of supported connected Stations at a time. Parameters: < **max_sta** > - Integer value (1-4), indicating the new value to be used. Cannot be set while AP is running.
AT#WIFIAPPHYCFG?	Read command returns the current allowed maximum connected stations

	#**WIFIAPPHYCFG**: <**max_sta**>
AT#WIFIAPPHYCFG=?	Test command reports supported values for the parameter.

5.2.7. #WIFIPFWD

#**WIFIPFWD** – Set Port Forwarding configuration	
AT#WIFIPFWD=<opcode>, **<proto>,** **<in_ip>,** **<in_port>** **[,<out_ip>** **[,<out_port>]]**	Execution command will manage port forwarding entries: Parameters: <**opcode**> - operation to be performed: 0: add entry 1: remove existing entry <**proto**> - protocol for the operation 0: TCP 1: UDP <**in_ip**> - input IP address on the cellular network (the remote host IP). can be 0 or a valid IP in the format xxx.xxx.xxx.xxx <**in_port**> the input port to be forwarded in the local network. valid values: 1-65535 <**out_ip**> - local IP address in the WiFi network for the destination node. Incoming packets matching the input parameters will be redirected to this IP address. It must be a valid IP in the format xxx.xxx.xxx.xxx. Can be omitted when removing an existing entry <**out_port**> local destination port for the redirect. If omitted, <**in_port>** will be used. Valid values: 1-65535
AT#WIFIPFWD?	Read command returns the current port forwarding configuration in the format: **#WIFIPFWD**: **TCP ENTRIES** **xxx.xxx.xxx.xxx:mmmmm --> yyy.yyy.yyy.yyy:nnnnn** **UDP ENTRIES** **xxx.xxx.xxx.xxx:mmmmm --> yyy.yyy.yyy.yyy:nnnnn** Where: xxx.xxx.xxx.xxx represents the input IP addresses, mmmm is the input port for every entry yyy.yyy.yyy.yyy represents the output IP addresses, nnnnn is the output port for every entry **Note:** if the same input port is used for the same protocol, an error will be returned. To update the entry, first delete it and add it again
AT#WIFIPFWD=?	Test command reports supported values for the parameters.

5.2.8. #WIFIDHCPRES

#**WIFIDHCPRES** – Set DHCP IP Reservation

AT#WIFIDHCPRES=<opcode>, <mac> [,<res_ip>]	Execution command will manage DHCP IP reservation entries: Parameters: <**opcode**> - operation to be performed: 0: add entry 1: remove existing entry <**mac**> - MAC Address in the format xx:xx:xx:xx:xx:xx of the WiFi Station for which reserve a static IP address. <**res_ip**> - static IP address to be reserved for the specified MAC address. It is the least significant byte of the address in the format xxx. Must be in the range 1-254 and must be different from AP IP address. The other 3 bytes of the address will be the same as the configured AP subnet. **Note:** Entries can be added or removed only when AP is not running
AT#WIFIDHCPRES?	Read command returns the current DHCP reservation configuration in the format: **#WIFIDHCPRES**: **xx:xx:xx:xx:xx:xx - yyy.yyy.yyy.yyy** Where: xx:xx:xx:xx:xx:xx represents the MAC addresses and yyy.yyy.yyy.yyy represents the reserved IP addresses for every entry **Note:** if the same IP is used for another entry, an error will be returned
AT#WIFIDHCPRES=?	Test command reports supported values for the parameters.

5.3. Station mode commands

5.3.1. #WIFISTAPHYCFG

#WIFISTAPHYCFG – Set Physical configuration of Station (WiFi will reboot to complete configuration)	
AT#WIFISTAPHYCFG= <ESSID>, <Method> [,<PWD>]	Set command sets the Access Point's Physical Parameters Parameters: < **ESSID** > - Alphanumeric string containing up to 32 bytes (default: **AP**) < **Method** > Integer values 0-2 (default: 2) 0 – Open 1 – WEP 2 – WPA & WPA2 < **PWD** > - Alphanumeric characters. (default: **password**) Len: 8-63 for WPA 5-13 for WEP
AT#WIFISTAPHYCFG?	Read command returns the current Physical Parameters: **#WIFISTAPHYCFG**: "<**ESSID** >",< **Method** >,"< **PWD** >"
AT#WIFISTAPHYCFG=?	Test command reports supported values for the parameters.

5.3.2. #WIFISTANETCFG

#WIFISTANETCFG – Set Network configuration of Station (WiFi will reboot to complete configuration)	
AT#WIFISTANETCFG=<Auto> [,<ip>, <Mask>, [<GateWay>, [<DNS>]]]	Set command sets the Station's Network Parameters all in the dotted format - Parameters: < **Auto** > - 0 – Automatic DHCP settings (factory default) 1 – Manual DHCP settings with the following parameters: < **ip** > - IPV4 address, string dotted format < **Mask** > - IPV4 Net Mask, string dotted format < **GateWay** > - IPV4 GateWay, string dotted format < **DNS** > - IPV4 DNS Server, string dotted format **Note**: < Auto > - if '0', network layer parameters will be automatically retrieved < GateWay > - if not set, <ip> will be used < DNS >- if not set, <ip> will be used
AT#WIFISTANETCFG?	Read command returns the current Network Parameters. If DHCP is enabled and Station is connected, or Static IP configuration is used **#WIFISTANETCFG**: <**dhcp_on**>,< **ip** >,< **Mask** >,<**GateWay** >,< **DNS** >

	Where: <dhcp_on> - 1 - DHCP client is enabled. 0 - Static configuration is used. Otherwise, **#WIFISTANETCFG: DHCP ENABLED**
AT#WIFISTANETCFG=?	Test command reports supported values for the parameters

5.3.3. #WIFISTAPING

#WIFISTAPING – Ping WiFi Gateway or remote server	
#WIFISTAPING=<type> **[,<host>** **[,<interval** **[,<timeout>]** **[,<attempts>]]]]**	Execution command sets socket configuration: Parameters: **< type >** - The ping type 0 - Ping the LAN Gateway 1 - Ping a remote host (using the following parameters) **< host >** - Address of the remote host: IPV4 or host name < **interval** > - Time interval between Pings in milliseconds **<timeout>** - Timeout for every ping. In milliseconds (0-65535) **<attempts>** - number of ping attempts **Note**: if the module is in AP mode, an ERROR will be returned.
#WIFISTAPING=?	Test command reports supported values for the parameters.

5.3.4. #WIFISTASD

#**WIFISTASD** – Send /receive data to/from remote server in Station mode	
AT#WIFISTASD=< socketId> **,<protocol>,** **<rPort>,** **<rAddress>** **[,<closureType>,** **[<lPort>,** **[<connMode>]]]**	Execution command send data to remote server according to the following parameters: Parameters: **< socketId> -** Socket connection identifier 1..5 **< protocol > -** 1 – TCP 2 – UDP **< rPort > -** Port of the remote host **< rAddress** > - Address of the remote host: IPV4 or host name <**closureType**> - UNSUPPORTED **< lPort > -** local port to be used for UDP sockets (ignored for TCP connections) **<connMode> -** Connection mode 0 - online mode connection (default) 1 - command mode connection **Note**: <**connMode**> - only online mode supported < **lPort** > - Ignored for TCP connections.

	< rAddress > - No quotes are needed Note: When the command execution is successful the module will enter in online data mode. The intermediate result code CONNECT is received. After the CONNECT , the socket can be closed using the escape sequence (+++): the module moves back to command mode and a NO CARRIER is returned. Any data incoming from the AT interface will be sent through the socket, and any response from remote host will be showed on the AT interface until the socket is open.
AT#WIFISTASD=?	Test command reports supported values for the parameters. **#WIFISTASD: (1-5),(0-1),(1-65535),(xxx.xxx.xxx.xxx),(0-1),(1-65535),(0-1)**

5.3.5. #WIFISTASCFG

#WIFISTASCFG – Sockets configuration in Station mode	
#WIFISTASCFG=<socketId>, <cid> [,<packet_size> [,<max_idle_to> [,<conn_to>] [,<tx_to>]]]]	Set command sets socket configuration: Parameters: **< socketId >** - Socket connection identifier 1..5 **< cid >** - Context Id – UNSUPPORTED <**packet_size**> - Sending max packet size. UNSUPPORTED **<max_idle_to> -** exchange timeout. If there is no data exchange within this period the connection will be closed. 0 - no timeout 1..65535 - timeout value in seconds (default: 90 seconds) **<conn_to>** - connection timeout. If a connection cannot be established within this timeout period, an error will be raised. UNSUPPORTED, defaults to 10 seconds < **tx_to>** - data sending timeout; UNSUPPORTED
AT#WIFISTASCFG?	Read command returns the current Sockets state: **#WIFISTASCFG**: < **socketId** >, < **cid**>, < **packet_size** >, < **max_idle_to**>, <**conn_to**>, <**tx_to**> For every socketId [1-5]
#WIFISTASCFG=?	Test command reports supported values for the parameters.

5.4. Hot Spot mode commands

5.4.1. #WSD

#**WSD** – Socket dial in Hot Spot mode	
AT#WSD=<socketId> **,<protocol>,** **<rPort>,** **<rAddress>** **[,<closureType>** **[,<lPort>,** **[<connMode>** **[,<interface>]]]]**	Set command opens a socket to a remote host (while in hot spot mode) according to the following parameters: Parameters: **< socketId> -** Socket connection identifier. 1-5 available **< protocol > -** 0 – TCP 1 – UDP **< rPort > -** Port of the remote host **< rAddress > -** Address of the remote host: IPV4 or host name <**closureType**> - for future use, keep to 0 (immediate closure) **< lPort > -** local port to be used for UDP sockets (ignored for TCP connections) **<connMode> -** Connection mode 0 - online mode connection (default) 1 - command mode connection **< interface > -** 1 – Cellular (open socket to remote host) 2 – WiFi (open socket to connected WiFi station) **Notes**: < **lPort** > - Ignored for TCP connections. **< rAddress >** - No quotes are needed If called when in Station mode, an error will be returned. Note: unless otherwise specified, the meaning and default values/ranges for the parameters are the same of **AT#SD** command. Refer to §5. Note: if **<connMode>** is set to **online mode** connection, and the command is successful, the module will enter in online data mode and the intermediate result code **CONNECT** will be displayed. the socket can be suspended using the escape sequence (**+++**): the module moves back to **command mode** and a **OK** response is returned after the suspension. After suspension, it is possible to resume **online mode** it in every moment (unless the socket inactivity timer timeouts, see **#WSCFG**) by using the **#WSO** command with the corresponding < **socketId**>. Any data incoming from the AT interface will be sent through the socket, and any response from remote host will be showed on the AT interface until the socket is open (and until the socket inactivity timer timeouts, see #**WSCFG**). Note: if <**connMode**> is set to **command mode** connection and the command is successful, the socket is opened and the result code **OK** is displayed.

	Note: if there are input data arrived through a connected socket and not yet ready because the module entered **command mode** before reading them (after #**WSD** has been issued with <**connMode**> set to **command mode** connection), these data are buffered and the **WSRING URC** (**WSRING** presentation format depends on the last #**WSCFGEXT** setting) will be received; it is possible to read these data afterwards issuing #**WSRECV**. Under the same hypotheses, It is possible to send data while in command mode issuing #**WSSEND**. Note: closure of the socket (#**WSH**) must be done on the same instance on which the socket was opened through #WSD.
AT#WSD=?	Test command reports supported values for the parameters.

5.4.2. #WSCFG

#WSCFG – Socket configuration in Hot Spot mode	
#WSCFG=<socketId>, **<cid>,** **<packet_size>,** **<max_idle_to>,** **<conn_to>,** **<tx_to>**	Set command sets socket configuration: Parameters: **< socketId >** - Socket connection identifier. 1-5 available **< cid >** - Context Id – unused, keep to 0 <**packet_size**> - packet size to be used by the TCP/UDP/IP stack for data sending. 0 - select automatically default value (300). 1..1500 - packet size in bytes. **<max_idle_to> -** exchange timeout in online mode. If there is no data exchange within this period, the connection will be closed. 0 - no timeout 1..65535 - timeout value in seconds (default: 90 seconds) **<conn_to>** - connection timeout. If a connection cannot be established within this timeout period, an error will be raised. 10..1200 - timeout value in hundreds of milliseconds (default 600) **< tx_to>** - data sending timeout; Unused, keep to 0. Note: if not otherwise specified, parameters have the same meaning and range of AT#SCFG
AT#WSCFG?	Read command returns the current Sockets configuration: **#WSCFG**: < **socketId** >, < **cid**>, < **packet_size** >, < **max_idle_to**>, <**conn_to**>, <**tx_to**> For every available socket id
#WSCFG=?	Test command reports supported values for the parameters.

5.4.3. #WSCFGEXT

#WSCFGEXT – Socket extended configuration in Hot Spot mode	
#WSCFGEXT=<socketId>, **<srMode>,** **<recvDataMode>,** **<keepalive>,** **[,<ListenAutoRsp>** **[,<sendDataMode>]**	Set command sets socket extended configuration: Parameters: **< socketId >** - Socket connection identifier. 1-5 available <**srMode**> - SRing unsolicited mode 0 - Normal (default): WSRING : <connId> where <connId> is the socket connection identifier 1 – Data amount: WSRING : <connId>,<recData> where <recData> is the amount of data received on the socket connection number <connId> 2 - Data view: not available 3 – Data view with UDP datagram informations: not available <**recvDataMode**> - data view mode for received data in command mode(**AT#WSRECV** or <**srMode**> = 2) 0- text mode (default) 1- hexadecimal mode <**keepalive**> - unused, keep to 0 (Deactivated) <**ListenAutoRsp**> - unused, keep to 0 (Deactivated) <**sendDataMode**> - data mode for sending data in command mode (**AT#WSSEND**) 0 - data represented as text (default) 1 - data represented as sequence of hexadecimal numbers (from 00 to FF) Each octet of the data is given as two IRA character long hexadecimal number Note: if not otherwise specified, parameters have the same meaning and range of **AT#SCFGEXT**
AT#WSCFGEXT?	Read command returns the current socket extended configuration for all available sockets, in the format: **#WSCFGEXT**: **< socketId >,<srMode1>,<dataMode1>,<keepalive1>,** **<ListenAutoRsp1>,< sendDataMode><CR><LF>** For every available socket id
#WSCFGEXT=?	Test command reports supported values for the parameters.

5.4.4. #WSRECV

#WSRECV – Socket send in command mode in Hot Spot mode	
#WSRECV=<socketId>, <maxByte>, [<UDPInfo>]	Set command permits the user to read data arrived through a connected socket, but buffered and not yet read because the module entered **command mode** before reading them; the module is notified of these data by a WSRING URC, whose presentation format depends on the last **#WSCFGEXT** setting. Parameters: **< socketId >** - Socket connection identifier. 1-5 available <**maxByte**> - max number of bytes to read 1..1500 <**UDPInfo**> unused. Note: issuing #**WSRECV** when there's no buffered data raises an error. Note: if not otherwise specified, parameters have the same meaning and range of AT#SRECV
#WSRECV=?	Test command reports supported values for parameters
Example	**WSRING URC (<srMode> be 0, <dataMode> be 0)** telling data have just come through connected socket identified by <socketId>=1 and are now buffered WSRING: 1 Read in text format the buffered data AT#WSRECV=1,20 #WSRECV: 1,11 Test string OK **WSRING URC (<srMode> be 1, <dataMode> be 1)** telling 11 bytes data have just come through connected socket identified by <socketId>=1 and are now buffered WSRING: 1,11 Read in hexadecimal format the buffered data AT#WSRECV=1,11 #WSRECV: 1,11 546573742073747269E67 OK

5.4.5. #WSSEND

#WSSEND – Socket send in command mode in Hot Spot mode	
#WSSEND=<socketId>	Set command permits, while the module is in **command mode**, to send data through a connected socket. Parameters: **< socketId >** - Socket connection identifier. 1-5 available The device responds to the command with the prompt <greater_than><space> and waits for the data to send. To complete the operation send **Ctrl-Z** char (**0x1A** hex); to exit without writing the message send **ESC** char (**0x1B** hex). If data are successfully sent, then the response is **OK**. If data sending fails for some reason, an error code is reported Note: the maximum number of bytes to send is 1500 bytes; trying to send more data will cause the surplus to be discarded and lost. Note: it's possible to use #**WSSEND** only if the connection was opened by **#WSD**, else an error is returned. Note: a byte corresponding to **BS** char(**0x08** hex) is treated with its corresponding meaning; therefore previous byte will be cancelled (and BS char itself will not be sent) Note: if not otherwise specified, parameters have the same meaning and range of AT#SSEND
#WSSEND=?	Test command reports supported values for the parameter <socketId>.
Example	Send data through socket number 1 AT#WSSEND=1 >Test<CTRL-Z> OK

5.4.6. #WSO

#WSO – Socket Restore in Hot Spot mode	
#WSO=<socketId>	Set command resumes the direct interface to a socket connection which has been suspended by the escape sequence. Parameters: **< socketId >** - Socket connection identifier. Only 1 available
#WSO=?	Test command reports supported values for the parameters.

5.4.7. #WSL

#WSL – Socket listen in Hot Spot mode	
AT#WSL=<socketId>, **<listenState>,** **<listenPort>** [,<**closureType**> **[,<interface>]]**	Set command opens a TCP socket listening (while in hot spot mode) for an incoming connection on a specified port. Parameters: **< socketId> -** Socket connection identifier. 1-5 available **< listenState > -** 0 – Closes socket listening 1 – Starts socket listening **< listenPort > -** **1**..65535 Local listening port <**closureType**> - for future use, keep to 0 (immediate closure) **< interface > -** 0 – Any interface (Default) 1 – Cellular (open socket to remote host) 2 – WiFi (open socket to connected WiFi station) **Note**: If called when in Station mode, an error will be returned. **Note**: if successful, the command returns **OK**. **Note:** when a TCP connection request comes on the input port, an URC is received: **WSRING : <connId>** **AT#WSA** can be used to accept the connection or **AT#WSH** to refuse it **Note**: unless otherwise specified, the meaning and default values/ranges for the parameters are the same of **AT#SL** command. Refer to §5.
AT#WSL?	Read command returns the currently listening TCP sockets.
AT#WSL=?	Test command reports supported values for the parameters.

5.4.8. #WSLUDP

#WSLUDP – Socket listen in Hot Spot mode	
AT#WSLUDP=<socketId>, **<listenState>,** **<listenPort>** **[,<interface>]**	Set command opens an UDP socket listening (while in hot spot mode) for an incoming connection on a specified port. Parameters: **< socketId> -** Socket connection identifier. 1-5 available **< listenState > -** 0 – Closes socket listening 1 – Starts socket listening **< listenPort > -** **1**..65535 Local listening port **< interface > -** 0 – Any interface (Default) 1 – Cellular (open socket to remote host) 2 – WiFi (open socket to connected WiFi station) **Note**: If called when in Station mode, an error will be returned.

	Note: if successful, the command returns **OK**. **Note:** when a UDP connection request comes on the input port, an URC is received: **WSRING : <connId>** **AT#WSA** can be used to accept the connection or **AT#WSH** to refuse it **Note**: unless otherwise specified, the meaning and default values/ranges for the parameters are the same of **AT#SLUDP** command. Refer to §5.
AT#WSLUDP?	Read command returns the currently listening UDP sockets.
AT#WSLUDP=?	Test command reports supported values for the parameters.

5.4.9. #WSA

#WSA – Socket dial in Hot Spot mode	
AT#WSA=<socketId> [,<connMode>]	Set command accepts an incoming socket connection after an URC **WSRING: <socketId>** (while in hot spot mode) Parameters: **< socketId> -** Socket connection identifier. 1-5 available **<connMode> -** Connection mode 0 - online mode connection (default) 1 - command mode connection **Note:** If called when in Station mode, an error will be returned. **Note**: unless otherwise specified, the meaning and default values/ranges for the parameters are the same of **AT#SA** command. Refer to §5. **Note: the WSRING URC must be a consequence of a #WSL issue.** **Note: setting the command before to having received a WSRING will result in an ERROR indication, giving the information that a connection request has not yet been received.**
AT#WSA=?	Test command reports supported values for the parameters.

5.4.10. #WSH

#WSH – Socket Shutdown in Hot Spot mode	
#WSH=<socketId>	Set command is used to close a socket. Parameters: **< socketId >** - Socket connection identifier. 1-5 available
#WSH=?	Test command reports supported values for the parameters.

5.4.11. #WSS

#WSS – Socket Status in Hot Spot mode	
#WSS[=<socketId>]	Execution command reports the current status of the socket: Parameters: **< socketId >** - Socket connection identifier. 1-5 available The response format is: **#WSS:** **<socketId>,<state>,<locIP>,<locPort>,<remIP>,<remPort>** where: <**socketId**> - socket connection identifier, as before <**state**> - actual state of the socket: 0 - Socket Closed. 1 - Socket with an active data transfer connection. 2 - Socket suspended. 3 - Socket suspended with pending data. 4 - Socket listening. 5 - Socket with an incoming connection. Waiting for the user accept or shutdown command. 6 - Socket resolving DNS. 7 - Socket connecting. <**locIP**> - IP address associated by the local network interface to the socket. <**locPort**> - two meanings: - the listening port if we put the socket in listen mode. (not available) - the local port for the connection if we use the socket to connect to a remote machine. <**remIP**> - it is the remote machine IP address. <**remPort**> - it is the the remote machine port Note: issuing **#WSS<CR>** causes getting information about status of all the sockets; The response format is: **#WSS:** **<socketId1>,<state1>,<locIP1>,<locPort1>,<remIP1>,<remPort1><CR><LF>** **…**

	Note: if not otherwise specified, parameters have the same meaning and range of AT#SS
#WSS=?	Test command reports supported values for the parameters.

5.4.12. #WSI

#WSI – Socket Info in Hot Spot mode	
#WSI[=<socketId>]	Execution command is used to get information about socket data traffic. Parameters: **< socketId >** - Socket connection identifier. 1-5 available The response format is: **#WSI:** **<socketId>,<sent>,<received>,<buff_in>,<ack_waiting>** where: <**socketId**> - socket connection identifier, as before <**sent**> - total amount (in bytes) of sent data since the last time the socket connection identified by <**socketId**> has been opened <**received**> - total amount (in bytes) of received data since the last time the socket connection identified by <**socketId**> has been opened <**buff_in**> - total amount (in bytes) of data just arrived through the socket connection identified by <**socketId**> and currently buffered, not yet read <**ack_waiting**> - total amount (in bytes) of sent and not yet acknowledged data since the last time the socket connection identified by <**socketId**> has been opened Note: not yet acknowledged data are available only for TCP connections; the value <**ack_waiting**> is always 0 for UDP connections. Note: issuing **#WSI<CR>** causes getting information about data traffic of all the sockets; the response format is: **#WSI:** **<socketId1>,<sent1>,<received1>,<buff_in1>,<ack_waiting1>** **<CR><LF>** **…** Note: if not otherwise specified, parameters have the same meaning and range of AT#SI
#WSI=?	Test command reports supported values for parameter **<socketId>**

5.4.13. #WPING

#**WPING** – Send PING request in Hot Spot mode	
AT#WPING=<IPaddr> **[,<retryNum>** **[,<len>** **[,<timeout>** **[,<ttl>]]]]**	This command is used to send Ping Echo request messages and to receive the corresponding Echo Reply. Parameters: **< IPaddr>** - address of the remote host, string type. This parameter can be either: - any valid IP address in the format: "xxx.xxx.xxx.xxx" - any host name to be solved with a DNS query **<retryNum> -** the number of Ping Echo Request to send 1-64 (default 4) **<len>** - the length of Ping Echo Request message 32-1460 (default 32) **<timeout>** - the timeout, in 100 ms units, for each Echo Reply 1-600 (default 50) **<ttl>** - time to live of Echo messages 1-255 (default 128) Once the single Echo Reply message is receive a string like that is displayed: **#WPING: <replyId>,<Ip Address>,<replyTime>,<ttl>** Where: **<replyId>** - Echo Reply number **<Ip Address>** - IP address of the remote host **<replyTime>** - time, in 100 ms units, required to receive the response **<ttl>** - time to live of the Echo Reply message **Note:** when the Echo Request timeout expires (no reply received on time) the response will contain <**replyTime**> set to 600 and <**ttl**> set to 255 **Note:** If called when in Station mode, an error will be returned.
AT#WPING=?	Test command reports supported values for the parameters.

5.4.14. #WSSENDUDP

#WSSENDUDP – Socket send in command mode in Hot Spot mode	
#WSSENDUDP=<socketId>, <rAddress >,<rPort>	Set command permits, while the module is in **command mode**, to send data over UDP to a specific remote host. UDP connection must be previously completed with a first remote host through #WSLUDP/ #WSA. Then, if data is received from this or another host, this command allows to send data to any of them. Like command #WSSEND, the device responds with '> ' and waits for the data to send. The user input management is the same as #WSSEND. Parameters: **< socketId >** - Socket connection identifier. 1-5 available **< rAddress >** - Address of the remote host: IPV4 or host name **<rPort>** - Remote host port If data are successfully sent, then the response is **OK**. If data sending fails for some reason, an error code is reported Note: the maximum number of bytes to send is 1500 bytes; trying to send more data will cause the surplus to be discarded and lost. Note: after WSRING that indicates incoming UDP data and issuing #WSRECV to receive data itself, through #WSS is possible to check last remote host (IP/Port). Note: if not otherwise specified, parameters have the same meaning and range of AT#SSENDUDP
#WSSENDUDP=?	Test command reports supported values for the parameters
Example	Starts listening on <LocPort> on WiFi interface AT#WSLUDP=1,1,<LocPort>,2 OK WSRING: 1 // UDP data from a remote host available AT#WSA=1,1 OK WSRING: 1 AT#WSI=1 #WSI: 1,0,0,23,0 // 23 bytes to read OK AT#WSRECV=1,23 #WSRECV:1,23 message from first host OK

	AT#WSS=1 #WSS: 1,2,<LocIP>,<LocPort>,<RemIP1>,<RemPort1> OK AT#WSSENDUDP=1,<RemIP1>,<RemPort1> >response to first host OK WSRING: 1 // UDP data from a remote host available AT#WSI=1 #WSI: 1,22,23,24,0 // 24 bytes to read OK AT#WSRECV=1,24 #WSRECV:1,24 message from second host OK AT#WSS=1 #WSS: 1,2,<LocIP>,<LocPort>,<RemIP2>,<RemPort2> OK *Remote host has changed, to send a response:* **AT#WSSENDUDP=1,<RemIP2>,<RemPort2>** **>response to second host** **OK**

5.5. Raw (transceiver) mode commands

5.5.1. #WRADIOPHYCFG

#WRADIOPHYCFG – WiFi transceiver mode physical parameters configuration for TX mode	
#WRADIOPHYCFG=<preambleSeq>[,<rate/mod>[,<power_lvl>[,<overrideCCA>[,<CCAThreshold>]]]]	Set command sets transceiver mode physical configuration for TX mode Parameters: **< preambleSeq >** - TX preamble sequence. Range: [0-3] Default: 0 (Long preamble) 0 - Long preamble 1 - Short preamble, 2 - OFDM Mode 3 - Greenfield Mode **< rate/mod >** - TX Testing rate (with the corresponding MODULATION). Range: [1, 21]. Default: 1 (1Mbps DSSS). 1: 1 Mbps (DSSS) 2: 2 Mbps (DSSS) 3: 5.5 Mbps (CCK) 4: 11 Mbps (CCK) 5: **NOT SUPPORTED** 6: 6 Mbps (OFDM) 7: 9 Mbps (OFDM) 8: 12 Mbps (OFDM) 9: 18 Mbps (OFDM) 10: 24 Mbps (OFDM) 11: 36 Mbps (OFDM) 12: 48 Mbps (OFDM) 13: 54 Mbps (OFDM) 14: MCS 0 15: MCS 1 16: MCS 2 17: MCS 3 18: MCS 4 19: MCS 5 20: MCS 6 21: MCS 7 < **power_lvl** > - TX POWER attenuation for Continuous and Packetized testing, 0 being the maximum power and 15 being the minimum power. Range: [0, 15]. Default: 0 **< overrideCCA >** - TX CCA override enable. Range: [0-1] Default: 1 (enabled) **< CCAThreshold**> - In case <**overrideCCA**> is not enabled, Clear Channel Assessment threshold to determine the signal level at which the channel is considered as occupied. Available values: 1. Min: (-88dBm) 2. Low: (-78dBm) **3. Default: (-68dBm)** 4. Med: (-58dBm) 5. High: (-48dBm) 6. Max: (-38dBm) Default: 3
AT#WRADIOPHYCFG?	Read command returns the current transceiver mode physical parameters:

	#WRADIOPHYCFG: <preambleSeq>,<rate/mod >,<power_lvl>,<overrideCCA>,<CCAThreshold>
#WRADIOPHYCFG=?	Test command reports supported values for the parameters. **#WRADIOPHYCFG:** **(0-4),(1-4,6-20),(0-15),(0-1),(1-6)**

5.5.2. #WRADIOTXCFG

#WRADIOTXCFG – WiFi transceiver mode TX parameters configuration	
(CONTINUOUS MODULATED) **#WRADIOTXCFG=<tx_mode>[,<channel>[,<dest_mac_addr>[,<packet_size>[,<data_pattern>[,<duration>]]]]]** *(PACKETIZED)* **#WRADIOTXCFG=<tx_mode>[,<channel>[,<dest_mac_addr>[,<packet_size>[,<data_pattern>[,<packets_amount>[,<delay_period>]]]]]]** *(CW)* **#WRADIOTXCFG=<tx_mode>[,<channel>[,<tone_offset>[,<duration>]]]**	Set command sets transceiver mode TX parameters configuration Parameters sequence depends on <**tx_mode**> value. Parameters: **< tx_mode > -** Transmission Mode. Range: [1-3] Default: 1 (Continuous modulated) 1 - Continuous Modulated 2 – Packetized 3 – CW • If <**tx_mode**> is 1 (Continuous modulated): **< channel >** - TX Wlan channel. Range: [1, 13]. Default: 1. **<dest_mac_addr>** - TX Destination MAC Address in the format xx:xx:xx:xx:xx:xx. Case insensitive. Default: 01:23:45:67:89:AB **< packet_size > -** TX Testing Frame's size, excluding header. Range: [14, 1500]. Default: 1500. **< data_pattern** > - TX data pattern. Default: 0 (All 0) 0: All 0 1: All 1 2: Incremental 3: Decremental **<duration> -** TX Testing duration in seconds. Use 0 for infinite transmission. Range: [0, 65535]. Default: 1 • If <**tx_mode**> is 2 (Packetized): **< channel >** - TX Wlan channel. Range: [1, 13]. Default: 1. **<dest_mac_addr>** - TX Destination MAC Address in the format xx:xx:xx:xx:xx:xx. Case insensitive. Default: 01:23:45:67:89:AB **< packet_size > -** TX Testing Frame's size, excluding header. Range: [14, 1500]. Default: 1500. **< data_pattern** > - TX data pattern. Default: 0 (All 0) 0: All 0 1: All 1 2: Incremental 3: Decremental **< packets_amount > -** TX maximum number of packets. Range:[0, 65535]. 0 for infinite amount. Default: 0

	<delay_period> - TX delay between packets in milliseconds. Range:[100, 1,000,000]. Default: 100 • If <**tx_mode**> is 3 (CW): **< channel >** - TX Wlan channel. Range: [1, 13]. Default: 1. **<tone_offset> -** TX tone offset. A value of N means tone at offset N*312.5kHz. Range: [-25, 25]. Default: 0 **<duration> -** TX Testing duration in seconds. Use 0 for infinite transmission. Range: [0, 65535]. Default: 1
AT#WRADIOTXCFG?	Read command returns the current transceiver mode physical parameters(parameters sequence depending on <tx_mode>: **#WRADIOTXCFG**: **1,<channel>,<dest_mac_address>,<packet_size>,<data_pattern>,<duration>** **#WRADIOTXCFG**: **2,<channel>,<dest_mac_address>,<packet_size>,<data_pattern>,<packets_amount>,<delay_period>** **#WRADIOTXCFG**: **2,<channel>,<tone_offset>,<duration>**
#WRADIOTXCFG=?	Test command reports supported values as a compound value

5.5.3. #WRADIORXCFG

#WRADIORXCFG – WiFi transceiver mode RX mode	
#WRADIORXCFG=<channel>[,<duration><urc_mode>]]]]	Set command sets transceiver mode physical configuration for RX mode Parameters: **< channel > -** RX Wlan channel. Range: [1, 13]. Default: 1. **<duration> -** RX Testing duration in seconds. Use 0 for infinite reception. Range: [0, 65535]. Default: 1 **<urc_mode> -** RX results report mode. Range: [0-1]. Default: 0. 0: None. No RX result will be printed. 1: All. After RX operation completion, a histogram will be printed, showing RSSI and Rate/Modulation for received packets. Refer to AT#WRADIOHIST command for further details.
AT#WRADIORXCFG?	Read command returns the current transceiver mode physical parameters: **#WRADIORXCFG**: <channel>,<duration>,<urc_mode>
#WRADIORXCFG=?	Test command reports supported values for the parameters. **#WRADIORXCFG:** **(1-13),(1-65535),(0-1)**

5.5.4. #WRADIOTEST

#WRADIOTEST – WiFi transceiver mode operation start	
#WRADIOTEST=<mode>	Set command starts transceiver mode in TX or RX Using parameters set with AT#WRADIOTXCFG, AT#WRADIORXCFG and AT#WRADIOPHYCFG Parameters: **< mode > -** Radio test mode. Range: [0-1]. Default 0. 0: Tx 1: Rx Note: in RX mode, if <urc_mode> was enabled with AT#WRADIORXCFG a report will be printed with results of RX operation. If the RX or TX duration was set to 0 (infinite) AT#WRADIOSTOP must be used to stop the operation.
#WRADIOTEST=?	Test command reports supported values for the parameters. **#WRADIOTEST:** **(0-1)**

5.5.5. #WRADIOSTOP

#WRADIOSTOP – WiFi transceiver mode stop running RX/TX	
#WRADIOSTOP	Execution will stop any running RX or TX operation (either with finite or infinite timeout). If no operation is ongoing, ERROR will be returned.

5.5.6. #WRADIOHIST

#WRADIOHIST – WiFi transceiver mode RX print histograms	
#WRADIOHIST=<read>	Set command will force the collect of all data received so far and report it as showed in execution command Parameters: **< read > -** Force reading of live received packets. Only allowed value is 1. Can be used only if AT#WRADIOTEST=1 is running (e.g. RX timeout was set to infinite and AT#WRADIOSTOP was not issued yet). If no RX operation is ongoing, ERROR will be returned.
#WRADIOHIST	Execution will report two histograms, one for RSSI values and one for Rate/modulation for all valid received packets RSSI values are divided in six groups: Above -48 dBm -48 dBm to -55 dBm -56 dBm to -63 dBm -64 dBm to -71 dBm -72 dBm to -79 dBm below -79 dBm

	Rate values are divided in twenty groups 1: 1 Mbps (DSSS) 2: 2 Mbps (DSSS) 3: 5.5 Mbps (CCK) 4: 11 Mbps (CCK) 6: 6 Mbps (OFDM) 7: 9 Mbps (OFDM) 8: 12 Mbps (OFDM) 9: 18 Mbps (OFDM) 10: 24 Mbps (OFDM) 11: 36 Mbps (OFDM) 12: 48 Mbps (OFDM) 13: 54 Mbps (OFDM) 14: MCS 0 15: MCS 1 16: MCS 2 17: MCS 3 18: MCS 4 19: MCS 5 20: MCS 6 21: MCS 7 Both histograms will report the percentage and absolute packet number for every group. Histograms will show data from the last collect (e.g. after #WRADIOTEST=1 completed) If not RX operation was performed before, an ERROR will be returned.
#WRADIOHIST?	Read command acts as execution command

5.6. ME Error Result Code - +CME ERROR: <err>

This is NOT a command, it is the error response to AT commands. (Refer to docs 6 and 7)

Syntax: **+CME ERROR: <err>**

Parameter: **<err>** - error code can be either numeric or verbose (see +CMEE).

The possible values of **<err>** (referring to standard error events) can be found in docs 6 and 7.

An additional set of values related to WiFi events has been defined. The new values are reported in the table below:

Numeric Format	Verbose Format
General Errors	
10001	wifi - already running
10002	wifi - not running
10003	wifi - cannot set
10004	wifi - AP init not complete.
10005	wifi - AP deinit not complete.
10006	wifi - wrong mode
10007	wifi - API abort
10008	wifi - cannot start hot spot mode
Access Point Mode Errors	
10020	wifi - cannot get stations info
10021	wifi - no stations connected
Station Mode Errors	
10030	wifi - station not connected
10031	wifi - station cannot connect
10032	wifi - cannot ping gateway
10033	wifi - cannot ping server
10034	wifi - cannot scan available networks
Station Socket Errors	
10050	wifi - tx error

10051	wifi - rx error
DHCP Server Errors	
10100	wifi - DHCP no free slots
10101	wifi - DCHP - MAC present
10102	wifi - DHCP - IP present
Raw Mode Errors	
10200	wifi - cannot start transceiver
10201	wifi - cannot stop transceiver
10202	wifi - cannot collect RX statistics

5.7. URC messages

Below a list of possible unsolicited notification messages and when they are generated. Errors are received depending on CME level (refer to ME Error Result Code - +CME ERROR: <err>)

Message	Reason
#WIFI - INIT COMPLETE<CRLF>	Issued after startup, indicates the app is ready to receive commands
#WIFI: WAITING STATIONS.. (AcT X)<CRLF>	Hot Spot mode is running and waiting for connections from WiFi stations. X indicates the Access technology of the cellular connection (with the same meaning of **AT+COPS?** command.
#WIFI: AP OFF<CRLF>	Hot Spot mode has been successfully stopped.
#WIFI: A Station Connected<CRLF>	A WiFi station connected to the Hot Spot
#WIFI: A Station Disconnected<CRLF>	A WiFi station disconnected to the Hot Spot
#WIFI: STA Connected to external AP<CRLF>	The WiFi in station mode connected to an external Access Point.
#WIFI: PING AP SUCCESS<CRLF>	In station mode, ping of the WiFi gateway succeeded. (LAN network access)
#WIFI: STA has Internet connection<CRLF>	In station mode, ping to a remote server succeeded. (Internet access)
#WIFI: STA OFF<CRLF>	Station mode disabled (disconnected from external AP)
#WIFI: Transmission done. <CRLF>	Transceiver TX operation ended
#WIFI: RX Complete.<CRLF>	Transceiver RX operation ended
#WIFI: PDP CONTEXT DEACTIVE<CRLF>	The Cellular connection dropped. The hot spot must be stopped and restarted.
#WSRING:	Data was received from a connected socket in command mode. The format of the URC will depend on **#WSCFGEXT** configuration.

6. USE CASES EXAMPLES

6.1. HOT SPOT commands sequence

Remember that you need to have SIM connected in order to have the HOT SPOT working

- Set the AP Mode

AT#WIFIMODE=0

- Configure AP Physical parameters

AT#WIFIAPPHYCFG=<SSID>,<security>,<password>,<region>,<channel>

- Configure the AP network parameter. For security typical is 2

(OPTIONAL) AT#WIFIAPNETCFG=<ip> ,< DNS > ,< ip_start>,<ip_last>,<ip_lease_time>

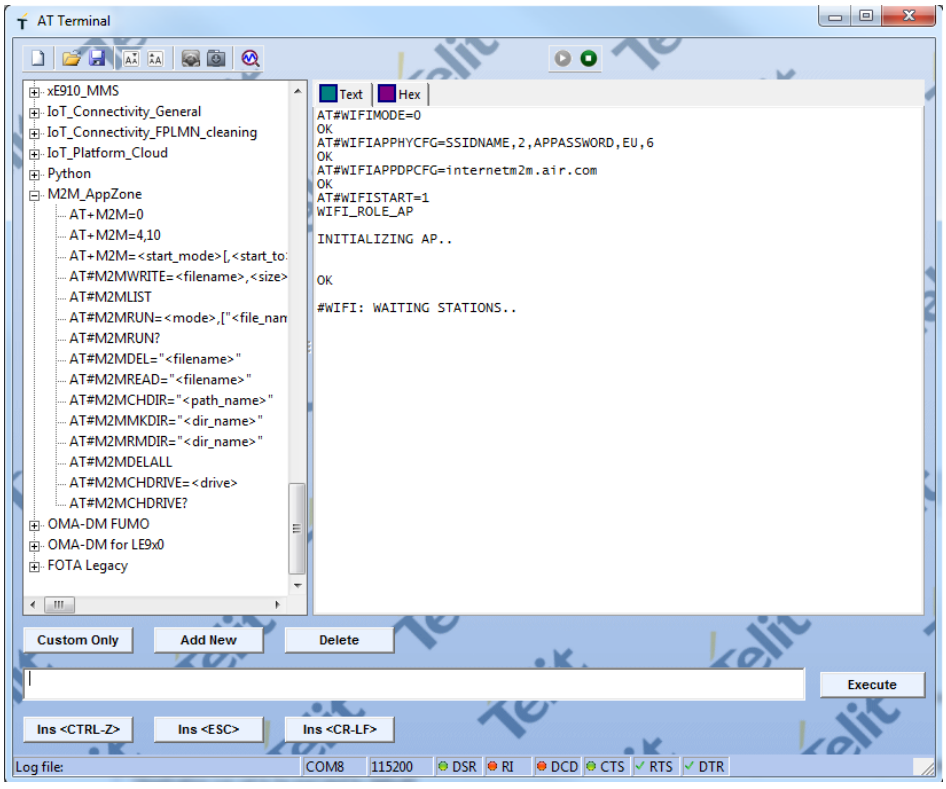
- Configure the AP Cellular interface.

AT#WIFIAPPDPCFG=[<"apn">], <username>, <password>

- Start the AP and wait for station to be connected.

AT#WIFISTART=1

After this command you will the message `#WIFI: WAITING FOR STATIONS ..`as the image below:



6.1.1. HotSpot Mode, Socket commands

The following examples refer to a scenario with LE910 modules using a single APN and a single available IP address. For 3G modules (or 4G modules with 2 or more APNs), standard commands can be used.

Warning – All the commands in this subsection require that AT#WIFISTART=1 has been issued.

6.1.1.1. TCP client socket to a remote host using cellular network

- Server address: “server.address"
- Server Port: 8080

- Online Mode

AT#WSD=1,0,8080,"server.address",0,0,0,1

CONNECT

…send data…

+++

OK

AT#WSH=1

OK

- Command Mode

AT#WSD=1,0,8080,"server.address",0,0,1,1

OK

AT#WSSEND=1

>

…send data…

<ctrl>Z

OK

WSRING: 1

AT#WSRECV=1,10

#WSRECV=1,8

received

OK

AT#WSH=1

OK

6.1.1.2. UDP client socket to a remote host using cellular network

- Server address: “server.address"
- Server Port: 8080

- Online Mode

AT#WSD=1,1,8080,"server.address",0,0,0,1

CONNECT

<…send data…>

+++

OK

AT#WSH=1

OK

- Command Mode

AT#WSD=1,1,8080,"server.address",0,0,1,1

OK

AT#WSSEND=1

>

…send data…

<ctrl>Z

OK

WSRING: 1

AT#WSRECV=1,10

#WSRECV=1,8

received

OK

AT#WSH=1

OK

6.1.1.3. TCP client socket to a connected Wifi station using WiFi network

- Server address: 10.80.1.15
- Server Port: 8080

- Online Mode

AT#WSD=1,0,8080,10.80.1.15,0,0,0,2

CONNECT

…send data…

+++

OK

AT#WSH=1

OK

- Command Mode

AT#WSD=1,0,8080,10.80.1.15,0,0,1,2

OK

AT#WSSEND=1

>

…send data…

<ctrl>Z

OK

WSRING: 1

AT#WSRECV=1,10

#WSRECV=1,8

received

OK

AT#WSH=1

OK

6.1.1.4. TCP listening socket on any interface (cellular or WiFi)

- Local Port: 8080

AT#WSL=1,1,8080

OK

AT#WSS=1

#WSS: 1,4,0.0.0.0,8080

OK

<A remote host with IP address **78.5.99.3** and remote port **54993** tries to connect on cellular interface, whose IP address is **10.240.11.235**>

WSRING: 1

AT#WSS=1

#WSS: 1,5,10.240.11.235,8080,78.5.99.3,54993

OK

- Accept in Online Mode

AT#WSA=1,0

CONNECT

…send data…

+++

OK

AT#WSH=1

OK

- Accept in Command Mode

AT#WSA=1,1

OK

AT#WSS=1

#WSS: 1,2,10.240.11.235,8080,78.5.99.3,54993

OK

AT#WSSEND=1

>

…send data…

<ctrl>Z

OK

WSRING: 1

AT#WSRECV=1,10

#WSRECV=1,8

received

OK

AT#WSH=1

OK

6.1.1.5. TCP listening socket on cellular interface

- Local Port: 8080

AT#WSL=1,1,8080,1

OK

<same steps as above>

6.1.1.6. TCP listening socket on WiFi interface

- Local Port: 8080

AT#WSL=1,1,8080,2

OK

<same steps as above>

6.1.1.7. UDP listening socket on any interface (cellular or WiFi)

- Local Port: 8080

AT#WSLUDP=1,1,8080

OK

AT#WSS=1

#WSS: 1,4,0.0.0.0,8080

OK

<A remote host with IP address **78.5.99.3** and remote port **54993** tries to connect on cellular interface, whose IP address is **10.240.11.235**>

WSRING: 1

AT#WSS=1

#WSS: 1,5,10.240.11.235,8080,78.5.99.3,54993

OK

- Accept in Online Mode

AT#WSA=1,0

CONNECT

…send data…

+++

OK

AT#WSH=1

OK

- Accept in Command Mode

AT#WSA=1,1

OK

AT#WSS=1

#WSS: 1,2,10.240.11.235,8080,78.5.99.3,54993

OK

AT#WSSEND=1

>

…send data…

<ctrl>Z

OK

WSRING: 1

AT#WSRECV=1,10

#WSRECV=1,8

received

OK

AT#WSH=1

OK

6.1.1.8. UDP listening socket on cellular interface

- Local Port: 8080

AT#WSLUDP=1,1,8080,1

OK

<same steps as above>

6.1.1.9. UDP listening socket on WiFi interface

- Local Port: 8080

AT#WSLUDP=1,1,8080,2

OK

<same steps as above>

6.2. STATION AT commands sequence

- Set the Station Mode

AT#WIFIMODE=1

- Scan available networks

AT#WIFISCAN

- Configure Station Physical parameters

AT#WIFISTAPHYCFG=<external SSID>,2,<password>

- Configure the station in DHCP mode

AT#WIFISTANETCFG=0

- Start the Station and connect to the Access Point

AT#WIFISTART=1

- Open a TCP socket in online mode with remote server “modules.telit.com” on port 10510 (echo) and send/receive data

AT#WIFISTASD=1,1,10510,modules.telit.com

(Send +++ to close the socket)

- Additional feature: ping a remote server

AT#WIFISTAPING=1,"www.telit.com",300,2000,2

7. GLOSSARY AND ACRONYMS

	Description
TTSC	Telit Technical Support Centre
USB	Universal Serial Bus
HS	High Speed
DTE	Data Terminal Equipment
UMTS	Universal Mobile Telecommunication System
WCDMA	Wideband Code Division Multiple Access
HSDPA	High Speed Downlink Packet Access
HSUPA	High Speed Uplink Packet Access
UART	Universal Asynchronous Receiver Transmitter
HSIC	High Speed Inter Chip
SIM	Subscriber Identification Module
SPI	Serial Peripheral Interface
ADC	Analog – Digital Converter
DAC	Digital – Analog Converter
I/O	Input Output
GPIO	General Purpose Input Output
CMOS	Complementary Metal – Oxide Semiconductor
MOSI	Master Output – Slave Input
MISO	Master Input – Slave Output
CLK	Clock
MRDY	Master Ready
SRDY	Slave Ready
CS	Chip Select
RTC	Real Time Clock
PCB	Printed Circuit Board
ESR	Equivalent Series Resistance
VSWR	Voltage Standing Wave Radio
VNA	Vector Network Analyzer

8. DOCUMENT HISTORY

Revision	Date	Changes
0	2017-02-02	First issue
1	2017-02-21	Added AT#WIFIGPIO
2	2017-02-24	Added AT#WIFISW
3	2017-03-13	Added AT#WIFISAV, General revision of AT commands Test and set command output
4	2017-04-21	Added CID parameter in AT#WIFIAPPDPCFG command. Updated AT#WIFISTART URC description Updated Stream and Application version compatibilities Added a Warning box for AT#MTUSIZE command in LTE platform
5	2017-05-26	Added port forwarding command AT#WIFIPFWD Modified AT#WIFIGPIO (removed wifi_nCS and ONOFF pins Added AT#GPIO commands in the software setup section 4.2 Added AT#WRADIO commands to manage test mode (transceiver mode)
6	2017-06-08	Modified AT#WRADIOHIST command (added set command description) Modified AT#WIFIPFWD command (error in ip_input description)
7	2017-06-27	Fixed AT#WIFIAPNETCFG description Modified AT#WIFISAV command (added group deletion selection)
8	2017-07-27	Fixed AT#WIFIDHCPRES query description Updated AT#WIFISAV description Updated all commands description Added commands AT#WSD AT#WSCFG AT#WSCFGEXT AT#WSSEND AT#WSRECV AT#WSS

		AT#WSI
9	2017-08-09	Added CME codes Updated WIFISWVER description Updated chapters organization
10	2017-09-08	Removed #WIFISD and #WIFISCFG commands. Added #WSO, #WSL, #WSA and #WE commands, updated #WSD description Added Hot Spot socket commands examples
11	2017-09-14	Added #SLUDP and #WPING commands, updated #WSL description Added #SLUDP example
12	2017-11-06	Updated SocketId parameter description in all #WS* commands Removed DRAFT watermark
13	2017-11-15	Updated AT#WIFIAPPDPCFG command description (query command now reports IP address)
14	2017-11-16	Updated application version
15	2018-01-23	Updated application version, SLUDP description
16	2018-01-25	Updated application and firmware for LTE version, updated #WIFISTART description, added URC messages table
17	2018-01-30	Added At#WSSENDUDP command

SUPPORT INQUIRIES

Link to **www.telit.com** and contact our technical support team for any questions related to technical issues.

www.telit.com

Telit Communications S.p.A.
Via Stazione di Prosecco, 5/B
I-34010 Sgonico (Trieste), Italy

Telit IoT Platforms LLC
5300 Broken Sound Blvd, Suite 150
Boca Raton, FL 33487, USA

Telit Wireless Solutions Inc.
3131 RDU Center Drive, Suite 135
Morrisville, NC 27560, USA

Telit Wireless Solutions Co., Ltd.
8th Fl., Shinyoung Securities Bld.
6, Gukjegeumyung-ro8-gil, Yeongdeungpo-gu
Seoul, 150-884, Korea

Telit Wireless Solutions Ltd.
10 Habarzel St.
Tel Aviv 69710, Israel

Telit Wireless Solutions
Technologia e Servicos Ltda
Avenida Paulista, 1776, Room 10.C
01310-921 São Paulo, Brazil

[04.2016]