



Running AT Commands Remotely Application Note

80000NT10029a Rev.9 – 2015-06-04

Making machines talk.

Running AT Commands Remotely
80000NT10029a Rev.9 – 2015-06-04

APPLICABILITY TABLE

SW Versions	
GC Family (Compact)	
GC864-QUAD	10.00.xx7
GC864-QUAD V2	10.00.xx7
GC864-DUAL V2	10.00.xx7
GE/GL Family (Embedded)	
GE864-QUAD	10.00.xx7
GE864-QUAD V2	10.00.xx7
GE864-QUAD Automotive V2	10.00.xx7
GE864-QUAD ATEX	10.00.xx7
GE864-DUAL V2	10.00.xx7
GE864-GPS	10.00.xx7
GE865-QUAD	10.00.xx7
GL865-DUAL	10.00.xx7
GL865-QUAD	10.00.xx7
GL868-DUAL	10.00.xx7
GE910-QUAD	13.00.xx3
GE910-QUAD AUTO	13.00.xx5
GE910-GNSS	13.00.xx5
GL865-DUAL V3	16.00.xx2
GL865-QUAD V3	16.00.xx2
GL868-DUAL V3	16.00.xx2
GE910-QUAD V3	16.00.xx3
GE866-QUAD	16.00.xx3
GT Family (Terminal)	
GT863-PY	10.00.xx7
GT864-QUAD	10.00.xx7
GT864-PY	10.00.xx7
HE910 Family	
HE910 ¹	12.00.xx4
HE910-D	12.00.xx4
HE910-GL	12.00.xx6
HE910-EUR / HE910-EUD	12.00.xx4
HE910-EUG / HE910-NAG	12.00.xx4
HE910-NAR / HE910-NAD	12.00.xx4
UE/UL Family (Embedded)	
UE910-EUR / UE910-EUD	12.00.xx4
UE910-NAR / UE910-NAD	12.00.xx4
UL865-EUR / UL865-EUD	12.00.xx4
UL865-NAR / UL865-NAD	12.00.xx4
UL865-N3G	12.00.xx4
UE910-N3G	12.00.xx6
UE866-N3G	12.00.xx6

Note: the features described by the present document are provided by the products equipped with the software versions equal or higher than the versions shown in the table.

¹ HE910 is the “type name” of the products marketed as HE910-G & HE910-DG.



Contents

1. INTRODUCTION.....	5
1.1. CONTACT INFORMATION, SUPPORT	5
1.2. TEXT CONVENTIONS	6
1.3. RELATED DOCUMENTS	6
1.4. DOCUMENT HISTORY	7
2. SMSATRUN	8
2.1. SMS FORMATS	8
2.2. ALERTS.....	10
2.3. DESCRIPTION	11
2.4. EXAMPLE	15
2.5. DELAYS AND NOTES	17
2.5.1. <i>Delays</i>	17
2.5.2. <i>Notes</i>	17
3. TCPATRUN	19
3.1. CONFIGURATION	20
3.2. TCPATRUN IN SERVER MODE	21
3.2.1. <i>Description</i>	21
3.2.2. <i>Authentication mode</i>	24
3.3. TCPATRUN IN CLIENT MODE	25
3.3.1. <i>Description</i>	25
3.4. AT COMMANDS IN SEQUENCE	27
3.5. TRANSPARENCY	29
3.6. ALERTS.....	30
3.7. EXAMPLES	31
3.7.1. <i>TCPATRUN in Server Mode</i>	31
3.8. DELAY AND NOTES.....	33
3.8.1. <i>Delays</i>	33
3.8.2. <i>Notes</i>	33
3.9. TCPATRUN AND PYTHON.....	35
3.10. TCPATRUN AND APPZONE	37



DISCLAIMER

The information contained in this document is the proprietary information of Telit Communications S.p.A. and its affiliates ("TELIT"). The contents are confidential and any disclosure to persons other than the officers, employees, agents or subcontractors of the owner or licensee of this document, without the prior written consent of Telit, is strictly prohibited.

Telit makes every effort to ensure the quality of the information it makes available. Notwithstanding the foregoing, Telit does not make any warranty as to the information contained herein, and does not accept any liability for any injury, loss or damage of any kind incurred by use of or reliance upon the information.

Telit disclaims any and all responsibility for the application of the devices characterized in this document, and notes that the application of the device must comply with the safety standards of the applicable country, and where applicable, with the relevant wiring rules.

Telit reserves the right to make modifications, additions and deletions to this document due to typographical errors, inaccurate information, or improvements to programs and/or equipment at any time and without notice. Such changes will, nevertheless be incorporated into new editions of this application note.

Copyright: Transmittal, reproduction, dissemination and/or editing of this document as well as utilization of its contents and communication thereof to others without express authorization are prohibited. Offenders will be held liable for payment of damages. All rights are reserved.

Copyright © Telit Communications S.p.A.



1. Introduction

The information presented in this document is believed to be accurate and reliable. However, no responsibility is assumed by Telit Communications S.p.A. for its use, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent rights of Telit Communications S.p.A. other than for circuitry embodied in Telit products. This document is subject to change without notice.

1.1. Contact Information, Support

For general contact, technical support, to report documentation errors and to order manuals, contact Telit Technical Support Center (TTSC) at:

TS-EMEA@telit.com
TS-NORTHAMERICA@telit.com
TS-LATINAMERICA@telit.com
TS-APAC@telit.com

Alternatively, use:

<http://www.telit.com/en/products/technical-support-center/contact.php>

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

<http://www.telit.com>

To register for product news and announcements or for product questions contact Telit Technical Support Center (TTSC).

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.



1.2. 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.

1.3. Related Documents

- AT Commands Reference guide, 80000ST10025a
- AT Commands Reference guide, 80378ST10091a
-



1.4. Document History

Revision	Date	Changes
ISSUE #0	2009-08-28	Release First ISSUE# 0
ISSUE #1	2010-05-07	Added commands in paragraph 3.9.2 (Notes) Correction of delays table regarding AT+CFUN command in paragraph 3.9.1 Correction in the pictures in paragraph 2.3 “Alerts” moved from paragraph 3.3.1 to paragraph 3.6 and integrated with a note about Idle Activity Timeout
ISSUE #2	2010-05-18	Deletion of note 15 in par. 3.9.2 Notes Shifted paragraph 3.9 in par. 2.5, renamed Delays and Notes. Added par. 3.9 regarding Python use. Changed the images in par 2.3 and in par. 3.1 (before 3.1). Added notes about ATZ command and about Packet Size
ISSUE#3	2010-10-04	Added GL865-DUAL to the applicability table
ISSUE#4	2010-02-10	Added note regarding SMS ATRUN Activation in 2.5.2 Added note regarding the Idle Activity Timeout and the Transparency in 3.6 Edited table in par 3.8.1
ISSUE#5	2012-10-08	General review of the document in accordance with the addition of the products: HE910, GE910.
ISSUE#6	2013-04-16	Updated Application Table.
ISSUE#7	2014-02-21	In the Applicability Table have been added the following products: GE910-GNSS, GE910-QUAD V3, GL865-QUAD V3, UE910-EUR/EUD, UE910-NAR/NAD, UL865-EUR/EUD, UL865-NAR/NAD, UL865-N3G, GE866-QUAD, GE910-QUAD AUTO.
ISSUE#8	2015-01-23	Added par.3.10: TCPATRUN and AppZone Removed HE910-GA from Applicability Table
ISSUE#9	2015-06-04	Added product variants HE910-GL, UE910-N3G, UE866-N3G Added notes in par. 2.3 and 3.2.1



2. SMSATRUN

Telit modules provide the SMSATRUN service allowing the user to run on the module itself AT Commands sent from a mobile via SMS messages, refer to Fig. 1. Examples: the mobile sends the AT Command to the remote module to collect information concerning the current Network Operator on which the module itself is camped, or the AT Command to set the desired GPIO.

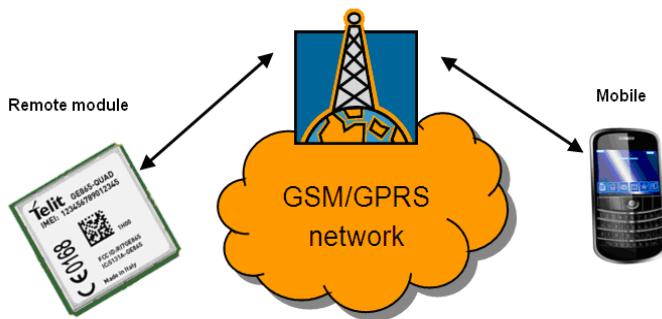


Fig. 1: SMSATRUN Scenario

The next paragraphs describe the AT commands sequence to configure and start the SMSATRUN service and how the service behaves. Please, refer to the AT Commands Reference Guide to have more information about AT commands syntax.

2.1. SMS Formats

The SMSATRUN service supports two SMS formats:

1. Simple ATRUN SMS:

- a) the SMS originating address shall match with a number defined in the white list of the remote module;
- b) the text has to begin with the string “AT” or “HAT”;
- c) the SMS coding alphabet has to be GSM 7 bit (default) or 8 bit;

2. Digest ATRUN SMS:

- a) The SMS User Data has to contain an header comprising a MD5 digest calculated on the useful text with a password belonging to the white list of the remote module;
- b) the text has to begin with the string “AT” or “HAT”;
- c) the SMS coding alphabet has to be 8 bit;



Running AT Commands Remotely 80000NT10029a Rev.9 – 2015-06-04

In the Digest SMS setting the SMS User Data has the following format:

Offset	Size	Value	Description
0	3	0xD0D0D0	RUNAT SMS Code
3	1	0	Transaction Id
4	1	0x11	Segment 1 of 1
5	1		Session Id
6	24		Digest: B64(MD5(B64(MD5(Pwd)):B64(MD5(Useful Text))))
30			Useful Text

Tab. 1: Digest ATRUN SMS User Data Format

- If the text begins with “AT” string the AT command response has to be sent to the sender by one or more response SMS. If the text begins with “HAT” string the AT command response hasn’t to be sent to the sender.
- After the “AT” (or “HAT”) string there are the commands. For example: AT+CGMR;+CGSN or HAT+CGMR;+CGSN.
- The response SMS from the remote module to the sender is coded with the same alphabet and has the same configuration of the received SMS (Simple SMS or Digest SMS).
- The received SMS hasn’t to be concatenated, but the response SMS can be concatenated. In the case of Digest SMS, the MD5 digest is calculated on the portion of text contained in the same SMS; the User Data has the following format:

Offset	Size	Value	Description
0	3	0xD1D1D1	Response RUNAT SMS Code
3	1	0	Transaction Id
4	1	0x11	Segment 1 of 1
5	1		Session Id
6	24		Digest: B64(MD5(B64(MD5(Pwd)):B64(MD5(Useful Text))))
30			Useful Text

Tab. 2: Response Digest ATRUN SMS User Data Format



- The SMSATRUN service is enabled setting to 1 the <mod> parameter of the AT command #SMSATRUN, the AT instance specified by the <muxInstance> parameter of the AT command #SMSATRUNCFG is reserved to run the AT commands received via the validated SMS.
- The SMSATRUN service is disabled setting to 0 the <mod> parameter of the AT command #SMSATRUN, the used AT instance is returned to the previous user.
- The <mod> parameter of the AT command #SMSATRUN and <muxInstance> parameter of the AT command #SMSATRUNCFG are stored in the NVM (Non Volatile Memory). At boot, if the <mod> parameter is set to 1, the AT instance specified in the <muxInstance> is automatically reserved to run the AT commands received via validated SMS.

2.2. Alerts

- If the <mod> parameter of the AT command #SMSATRUN is set to 1, then all received SMS are subjected to a test to validate them as ATRUN SMS.
- If a SMS is recognized as ATRUN SMS, it is handled and silently discarded.
- When an AT instance is reserved for the SMSATRUN service, then every other request to use it, coming from other services, is refused. Only the FOTA (Firmware Over-the-Air) service can take possession of the AT instance reserved to SMS ATRUN service.
- If the TCPATRUN service is activated on the first instance (<muxInstance> = 1 in AT#TCPATRUNCFG), that instance is only dedicated to TCPATRUN service. Use CMUX (multiplexer) to access the second or third instance and deactivate the service to free the instance, refer to the document: Virtual Serial Device, 80000NT10045A.

NOTICE: Using HE910 products could be not necessary to use the CMUX if two ports are available at the same time, refer to the document: HE910 Family Ports Arrangements, 1vv0300971.



2.3. Description

The Fig. 2 summarizes two aspects: the configuration of the SMSATRUN service carried out locally, and an example of SMS messages exchange between the remote module and the mobile.

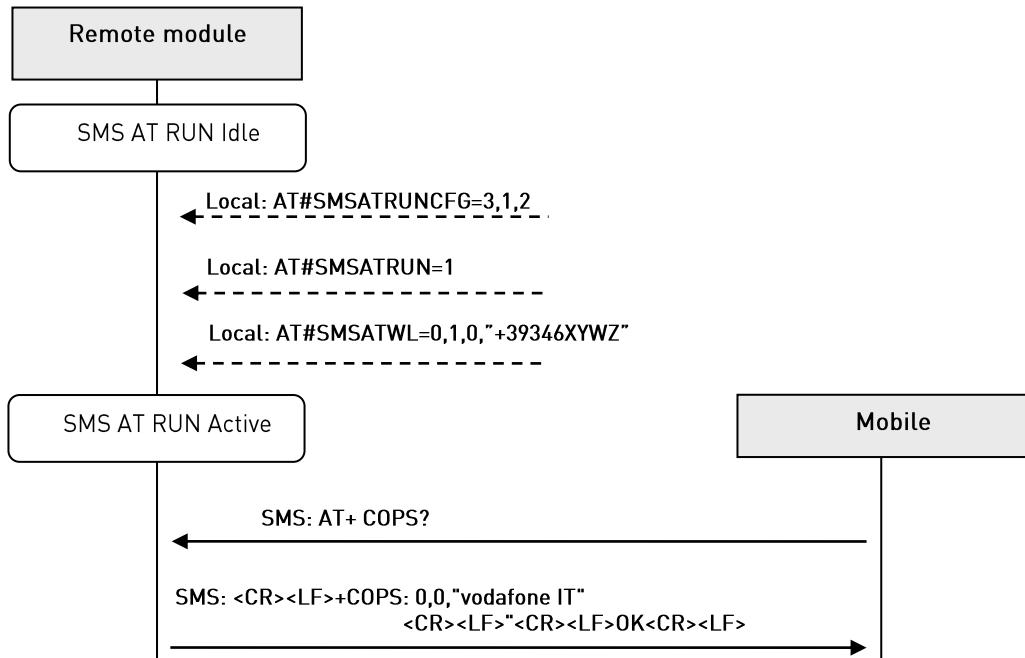


Fig. 2: Local Configuration and SMS messages Exchange

First of all, the SMSATRUN service provided by the remote module is locally configured and enabled:

AT#SMSATRUNCFG=3,1,2

// the service is configured to run on the third instance, the unsolicited is enabled and // the timeout for the response to the AT commands is set to 2 minutes

AT#SMSATRUN=1 // the service is enabled



Running AT Commands Remotely
80000NT10029a Rev.9 – 2015-06-04

The mobile phone number is added in the white list of the remote module by means of the following AT command:

AT#SMSATWL=0,1,0,"+39346XYWZ" //the number +39346XYWZ is added in the first position of the white list

At this point the SMSATRUN service is configured and enabled. The remote module is ready to receive AT commands from the mobile.

Note: this command will return ERROR if it is executed using SMSATRUN digest mode or TCPATRUN server mode

It is assumed that the remote module receives the AT command AT+COPS? encapsulated in an SMS message and executes it using the third instance. After command execution, the remote module sends back the command response to the mobile. The mobile reads the arrived SMS:

+CMTI: "SM",1
AT+CMGR=1
+CMGR: "REC UNREAD","+39346XYWZ","","08/05/28,11:04:37+08"

+COPS: 0,0,"vodafone IT"

OK

Using the "HAT" string, the command response is not sent back.

AT+CMGS=346XYWZ
> HAT+COPS?



Running AT Commands Remotely 80000NT10029a Rev.9 – 2015-06-04

Using the Digest ATRUN SMS format, a password must be inserted in the white list of the remote module via the following AT command:

AT#SMSATWL=0,2,1,"1234567890123456" //the password has to be 16 // characters length and is inserted, in this example, in position 2 of the white list.

A maximum number of 2 passwords can be inserted in the white list of the remote module. In Digest ATRUN SMS format the mobile can also change the SIM because the sender number isn't checked from the remote module.

The mobile can interrogate the remote module, to know the selected network operator, sending the AT+COPS? command using the following standard AT command:

AT+CMGS=53

>079193432900200011000C919343868676110015AD26D0D0D000110157696A437032715633734E762B61727532305A5561773D3D41542B434F50533F

Where the User Data portion is formatted according to Tab. 1 as follows:

Offset	Size	Value	Description
0	3	D0D0D0	RUNAT SMS Code
3	1	00	Transaction Id
4	1	11	Segment 1 of 1
5	1	01	Session Id
6	24	57696A437032715633734E762B61727532305A5561773D3D	Digest: B64(MD5(B64(MD5(Pwd)):B64(MD5(AT+COPS?))))
30		41542B434F50533F	AT+COPS?

The remote module receives the AT+COPS? command encapsulated in an SMS message and executes it using the third instance. After command execution, the remote module sends back the response to the mobile. The mobile reads the arrived SMS:



```
+CMTI: "SM",1
AT+CMGR=1
+CMGR: 0,"",83
0791934329002000040C9193438686761100158050922102818040D1D1D10011014A756E7261377
2306B494563483971563534326E33513D3D0D0A2B434F50533A20302C302C22766F6461666F6E6
5204954220D0A0D0A4F4B0D0A
```

Where the User Data portion is formatted according to Tab. 2 as follows:

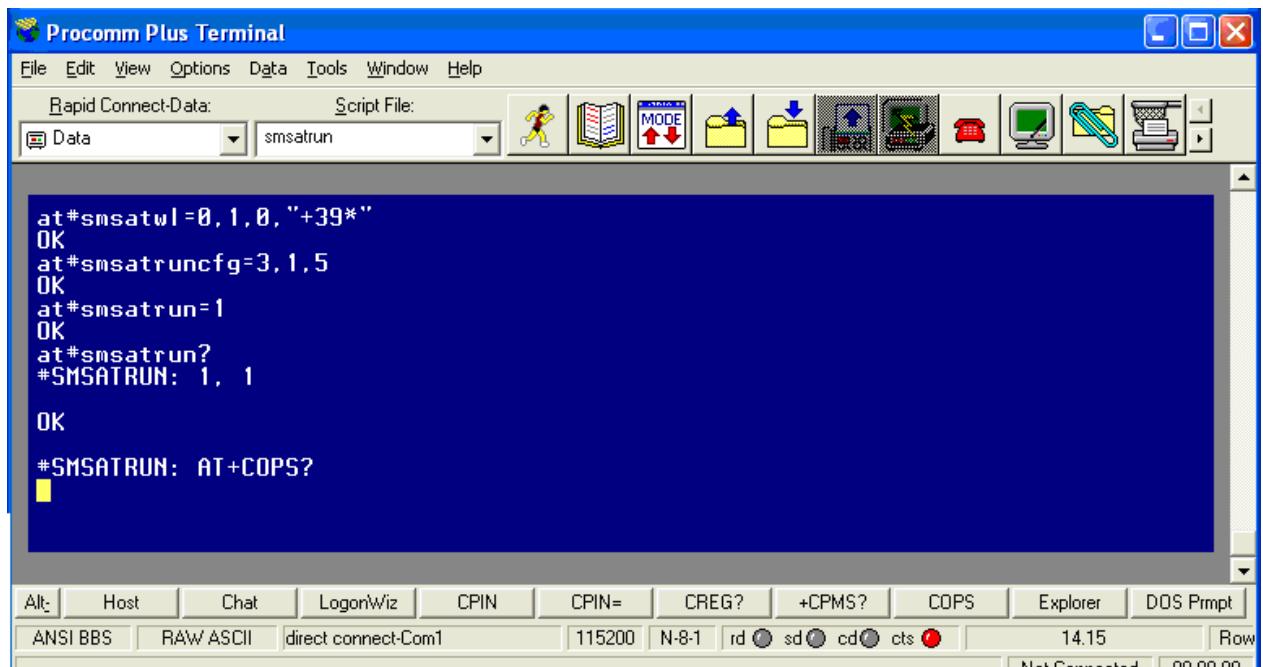
Offset	Size	Value	Description
0	3	D1D1D1	Response RUNAT SMS Code
3	1	00	Transaction Id
4	1	11	Segment 1 of 1
5	1	01	Session Id
6	24	4A756E72613772306B49456348397156353 4326E33513D3D	Digest: B64(MD5(B64(MD5(Pwd)):B64(MD5(<CR><LF>+COPS: 0,0,"vodafone IT"<CR><LF>"<CR><LF>OK<CR><LF>))))
30		0D0A2B434F50533A20302C302C22766F6 461666F6E65204954220D0A0D0A4F4B0D 0A	<CR><LF>+COPS: 0,0,"vodafone IT"<CR><LF>"<CR><LF>OK<CR><LF>



2.4. Example

The Fig. 3 shows the configuration of the SMSATRUN service carried out locally:

- All the numbers starting with +39 are added in the white list by the command #SMSATWL
- the service is configured with the #SMSATRUNCFG command to run on the third instance, the unsolicited is enabled and the timeout for the response to the AT commands is set to 2 minutes
- the service is enabled with the #SMSATRUN command



The screenshot shows a terminal window titled "Procomm Plus Terminal". The menu bar includes File, Edit, View, Options, Data, Tools, Window, Help. The toolbar contains icons for Data, smsatrun, and various communication modes. The main window displays the following AT command session:

```

at#smsatwl=0,1,0,"+39*"
OK
at#smsatruncfg=3,1,5
OK
at#smsatrun=1
OK
at#smsatrun?
#SMSATRUN: 1, 1

OK
#SMSATRUN: AT+COPS?

```

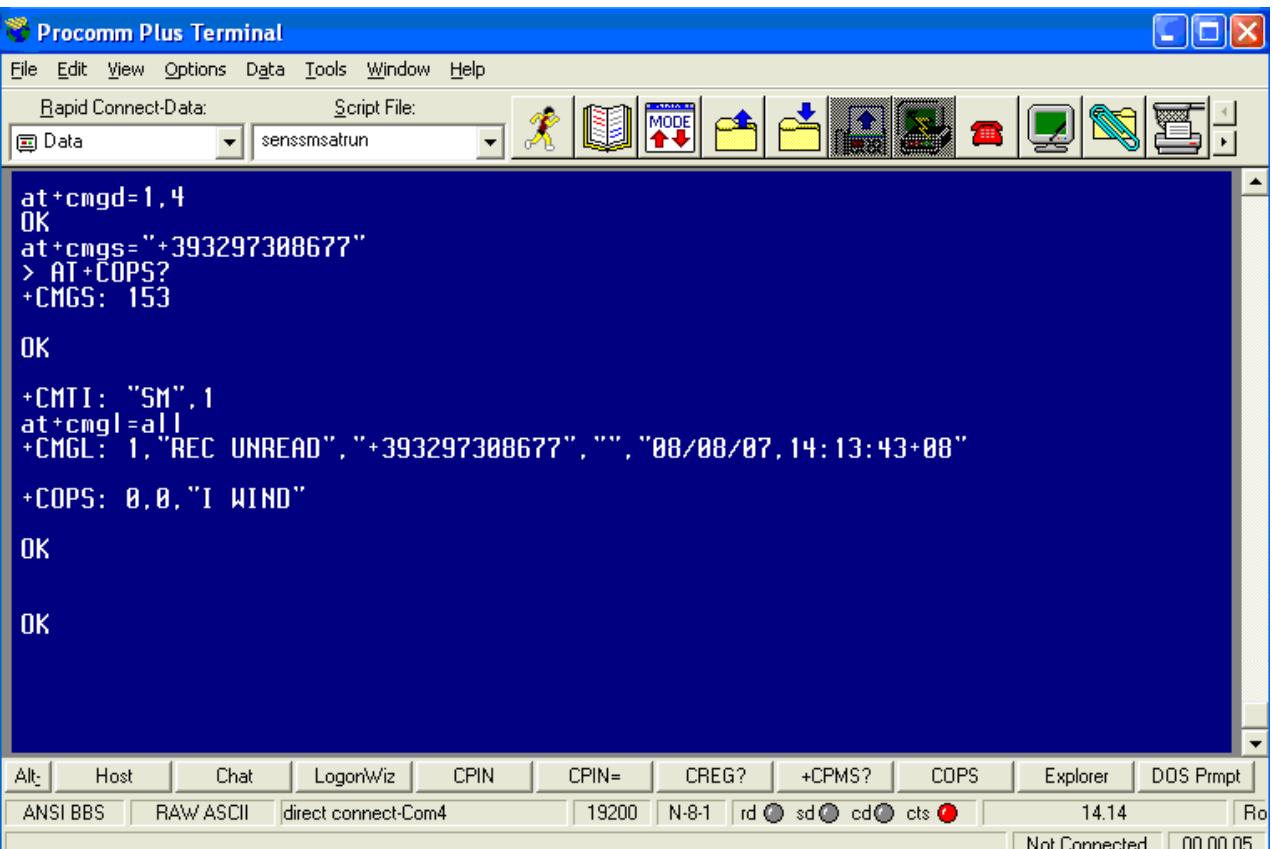
The status bar at the bottom shows the connection parameters: ANSI BBS, RAW ASCII, direct connect-Com1, 115200, N-8-1, rd, sd, cd, cts, 14.15, Row, Not Connected, 00.00.00.

Fig. 3: Configuration of the SMSATRUN service



Running AT Commands Remotely 80000NT10029a Rev.9 – 2015-06-04

The Fig. 4 shows the mobile sending to the remote module the SMS message containing the AT command AT+COPS?. The remote module, after receiving the SMS message, runs automatically the command AT+COPS?, displays the URC #SMSATRUN: AT+COPS?, see Fig. 3, and sends back the AT command response.



```

Procomm Plus Terminal
File Edit View Options Data Tools Window Help
Rapid Connect-Data: Script File:
Data senssmsatrun
at+cmgd=1,4
OK
at+cmgs="+393297308677"
> AT+COPS?
+CMGS: 153

OK

+CMTI: "SM",1
at+cmgl=all
+CMGL: 1, "REC UNREAD", "+393297308677", "", "08/08/07, 14:13:43+08"
+COPS: 0,0,"I WIND"

OK

OK

```

Alt Host Chat LogonWiz CPIN CPIN= CREG? +CPMS? COPS Explorer DOS Prompt
ANSI BBS RAW ASCII direct connect-Com4 19200 N-8-1 rd sd cd cts 14.14 Not Connected 00.00.05 Ro

Fig. 4: SMS messages Exchange



2.5. Delays and Notes

2.5.1. Delays

Some AT commands can fail if executed by means of the SMS ATRUN service due to an interaction of the AT command running with the SMS service. These problems can be avoided inserting a delay before AT command execution. The delay is set through the following command:

AT#ATRUNDELAY=1,5 sets for SMS ATRUN commands a delay of 5 seconds

Once set, the delay is valid till a new AT#ATRUNDELAY is issued. The following table shows the suggested delays (in seconds).

AT+COPS=? AT+CGCLASS AT+CFUN AT#GPRS AT#SIMDET AT#CSURV AT#CSURVC AT#CSURVU AT#CSURVUC AT#CSURVB AT#CSURVBC AT#CSURVP AT#CSURVPC	10
AT#SGACT AT+CGACT AT+CGATT	5

2.5.2. Notes

- 1) To send AT\x and AT^xxx commands it is necessary to set the outgoing SMS's Data Coding Scheme (DCS) so that they are 8 bits coded, because characters '\ and '^' are not included in the standard GSM 7 bit alphabet. AT#SMSMODE must be set to 1, and the SMS text must be in hexadecimal format; the answer SMS received to the sent command is also 8 bit coded (hexadecimal format);
- 2) AT#REBOOT and AT#SHDN commands do not send back the answer SMS before powering down the remote module; AT+CFUN=4, AT+CGCLASS="CG" and AT+COPS=2 commands do not send back the answer SMS before deregistering the module;
- 3) AT#SIMDET=0, AT+CGDATA="PPP",1, AT#FTPPUT, AT\$GPSNMUN=3, ATQ1 and ATQ2 commands cause a block of the SMS exchange mechanism, until the remote module reboots after timeout set with AT#SMSATRUNCFG command expires;
- 4) AT+CMUX=0, AT#BND=3, AT+CFUN=2 and AT+CFUN=4 commands cause an irreversible block of the SMS exchange mechanism;



Running AT Commands Remotely

80000NT10029a Rev.9 – 2015-06-04

- 5) AT+CGCLASS="CG" and AT+CGSMS=0 commands cause an irreversible block of the SMS exchange mechanism, until a new AT+CGCLASS or AT+CGSMS command is sent with TCP ATRUN; AT+CFUN=4 and AT+COPS=2 commands cause an irreversible block of the SMS exchange mechanism, until a new AT+CFUN or AT+COPS command is sent with TCP ATRUN;
- 6) enabling of FDN phonebook in the remote module causes an irreversible block of the SMS exchange mechanism if the “mobile” phone number is not in the FDN list; FDN must then be disabled using TCP ATRUN;
- 7) if ATD command is sent to the remote module to start a data call the response SMS, containing the CONNECT or NO CARRIER result, is received only after the call has been disconnected by the called party;
- 8) AT+CMGS, AT+CMGW, AT#SEMAIL, AT#SSEND, AT#WSCRIPT and all commands that require to insert data after a prompt is received cannot be executed via SMS ATRUN, because the prompt is not sent back in an SMS response;
- 9) The commands of AT#CSURV family will have their response truncated if too long (response text longer than the text that can fit into a concatenated SMS made up of the maximum number of SMS that can be concatenated, i.e. 15);
- 10) The command ATZ returns ERROR if a Data Call is up;
- 11) By default the SMS ATRUN service is disabled. It can be activated either by the AT#SMSATRUN command or receiving a special SMS that can be sent from a Telit server. After the reception of this SMS, the module activates the service and inserts a default password in the white list. An optional phone number can be specified in the SMS body that will be inserted in the white list. For further information please refer to Telit contacts indicated in chapter 1.1.



3. TCPATRUN

Telit modules provide the TCPATRUN service allowing the user to run on the module itself AT Commands sent from a PC via TCP/IP protocol, refer to Fig. 5. Examples: the PC sends the AT Command to the module to collect information concerning the current Network Operator on which the module itself is camped, or the AT Command to set the desired GPIO.

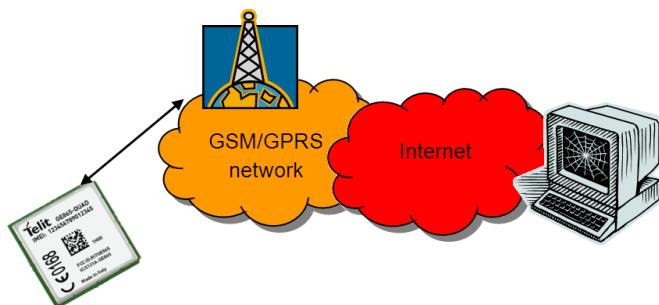


Fig. 5: TCPATRUN Scenario

There are two TCPATRUN service configurations:

Server Mode

The module acts as a server. It is in TCP listen state waiting for an incoming TCP/IP connection from a client.

Client Mode

The module acts as a client, as soon as the service is enabled, the module tries to connect to the server.

The next paragraphs describe the AT commands sequence to configure and start the TCPATRUN service and how the service behaves. Please, refer to the AT Commands Reference Guide to have more information about AT commands syntax.



3.1. Configuration

To start TCPATRUN service, either in server mode or in client mode, it is necessary to open a PDP context via the AT#SGACT command:

AT#SGACT=1,1 for example, here we open the PDP context with id=1

The answer, in case of success, will be the IP address of the module, e.g.:

#SGACT: "212.141.113.92"

It is recommended to use the “AT#SGACTCFG” command to set the context “auto-activation”: in this way the context will be automatically activated in case of switching off/on, deactivation from Network, and SIM removal.

AT#SGACTCFG=1,1 the auto-activation is set on context id=1

Configure the socket parameters through the AT#SCFG command, see chapter 3.6, using the socket identifier (the first parameter). For example:

AT#SCFG=1,1,300,0,600,1

Now, some configuration parameters have to be set for both service configurations using the AT#TCPATRUNCFG command:

AT#TCPATRUNCFG =1,2,1024,12345,"212.141.125.127",1,5,1,5,2

The first parameter is the identifier of the socket that will be used; the second one is the instance (in the module the AT interface can use three instances) that will be dedicated to the TCPATRUN; the third one is the TCP port for listen (used in server mode); the fourth one and the fifth one are respectively the TCP port and IP address of the host (for the client mode); the sixth one is for enabling the unsolicited that advises of the connection/disconnection from server; the seventh one is the timeout for an AT command execution; the eighth one selects the authentication mode (in server mode); the ninth one is the number of attempts to connect that the module will do in case of automatic re-start of the service (in client mode); the last one is the delay (in minutes) from one attempt and the other (in client mode).



3.2. TCPATRUN in Server Mode

3.2.1. Description

The TCPATRUN service in server mode is enabled locally by the AT#TCPATRUNL command. It configures the module in listen state on a determined port, so that a client can connect to the IP address and port of the server (module). Once connected, the client can issue AT commands to the module.

Note that the TCPATRUN service in server mode is useful when its address is a static IP address.

In general, the TCP access to the module is controlled by a firewall. In the case of TCP ATRUN service, this firewall can be set (and saved in memory) by the following command:

AT#TCPATRUNFRWL=1,"212.141.112.216","255.255.255.255"

The two strings represent the IP address and mask to manage the **clients access**.

Note: this command will return **ERROR if it is executed using SMSATRUN digest mode or TCPATRUN server mode**

The service is also protected by the authentication parameters: Username and Password. These parameters are set (and saved in memory) through the following command:

AT#TCPATRUNAUTH=1,"testuser","testpassw"

The user can choose between two kinds of authentication procedure, as we will see in the next paragraph.

Finally, the TCPATRUN service can be enabled using:

AT#TCPATRUNL =1

See the examples showed in chapter 3.7.1.



Running AT Commands Remotely
80000NT10029a Rev.9 – 2015-06-04

Hereafter are same clarifications regarding the configuration parameters used in the AT#TCPATRUNCFG command:

- The socket 1 is in listen state on port 1024, and the IP address is the one given by the AT#SGACT command.
- The client (PC) can connect to the server (module) if its address is included in the firewall. The client can use Telnet or a dedicated program to open the socket and communicate with the module. Once connected, username and password have to be transmitted in a way that depends on the authentication mode set with AT#TCPATRUNCFG.
- After authentication, an AT command can be sent by the client, the module will parse the received command and send back the response.
- For example, the client can interrogate the module about all the neighbor cells with the serving cell of the module itself using the AT#MONI=7 and AT#MONI commands.
- If the TCPATRUN service is not disabled, it automatically starts at every start-up of the module as long as the PDP context auto-activation has been set by AT#SGACTCFG. The same happens in case of a deactivation from Network and in case of SIM removal.



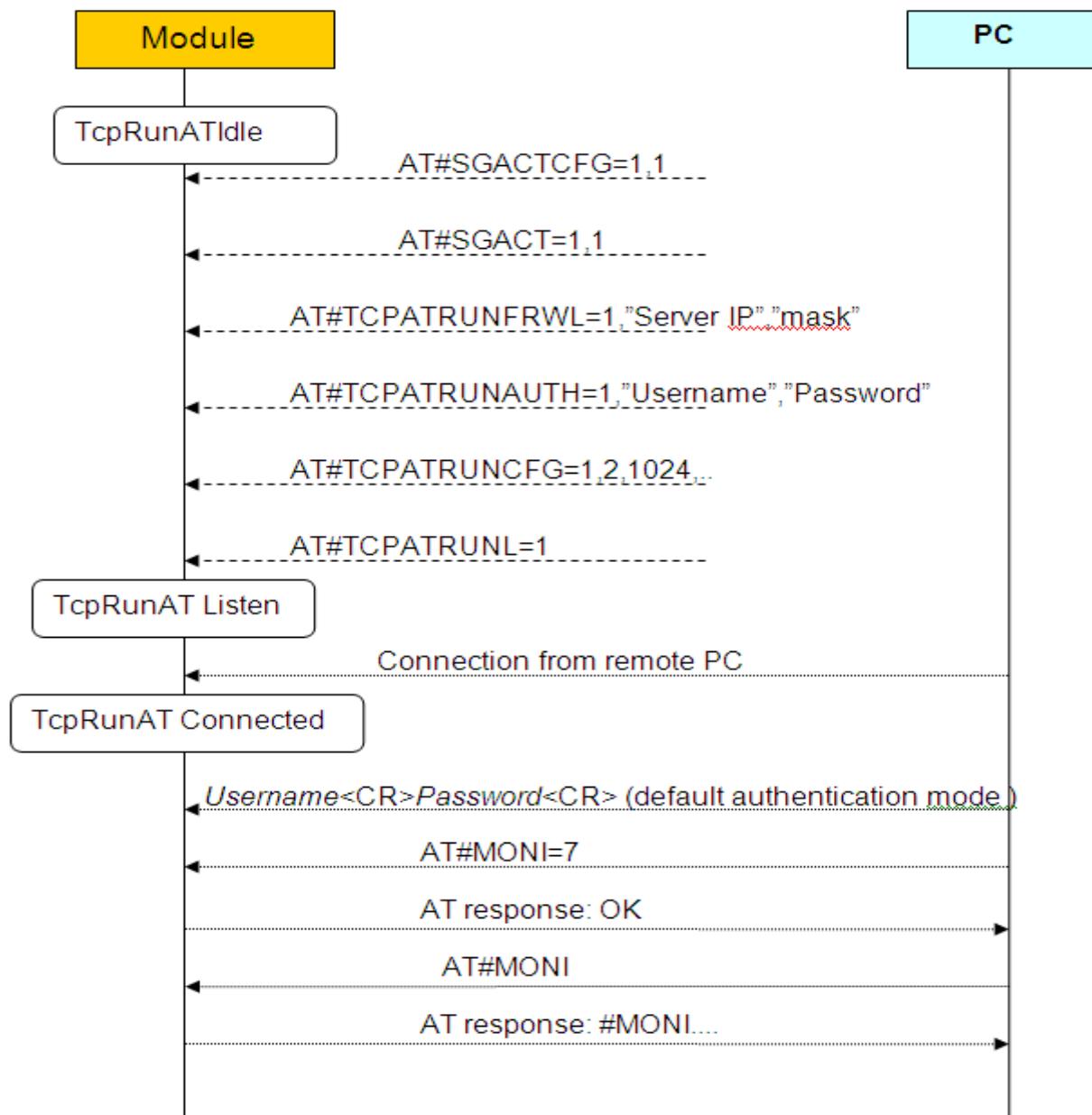


Fig. 6: TCPATRUN in Server Mode



3.2.2. Authentication mode

The AT#TCPATRUNCFG command allows the user to choose between two authentication modes: “automatic” and “step-by-step”, see next subchapters.

3.2.2.1. Automatic mode (default)

Once the server-client connection is opened and before issuing any AT commands, username and password have to be transmitted to the server (module) in the following way: *userid<CR>passw<CR>* (where <CR> means Carriage Return and is equal to 0x0d)

The couple *Username – Password* must be included in the authentication parameters list using the AT#TCPATRUNAUTH command, otherwise the server (module) will immediately close the connection.

If the server timeout expires before receiving one of the two parameters, the server (module) closes the connection.

3.2.2.2. Step-by-step

Once the server-client connection is opened, the module waits for the Username and then, if the Username has been received and accepted (*userid<CR>*), waits for the Password. When the Password has been received and accepted (*passw<CR>*), the module transmits a “Login successful” message to the client.

Username: *userid*
Password: *passw*
Login successful.

From this moment, AT commands sent by the client (PC) are parsed by the module. If the couple *Username – Password* is not included in the authentication parameters list, or if the timeout expires before receiving a parameter, the server (module) closes the connection.



3.3. TCPATRUN in Client Mode

3.3.1. Description

The Client Mode is configured by the AT#TCPATRUND command entered into the module locally:

AT#TCPATRUND = 1

If the command response is “OK”, the client (module) opens the client-server connection, see Fig. 7. The server (Host, PC) in order to accept the client connection request must be in the listen status running a dedicated application configured to use the Host IP address and the Host port specified in the AT#TCPATRUNCFG, see chapter 3.1. After connection establishment, the client (module) is ready to receive AT commands.

If the TCPATRUN service is not disabled, it starts automatically at every start-up of the module as long as the PDP context auto-activation has been set by AT#SGACTCFG. The same happens in case of a deactivation from Network and in case of SIM removal. This means that the module tries automatically to connect to the Host IP (the one set through AT#TCPATRUNCFG), and if connection is not successful it will retry. The number of attempts and the delay between one and the other will be those set through AT#TCPATRUNCFG.



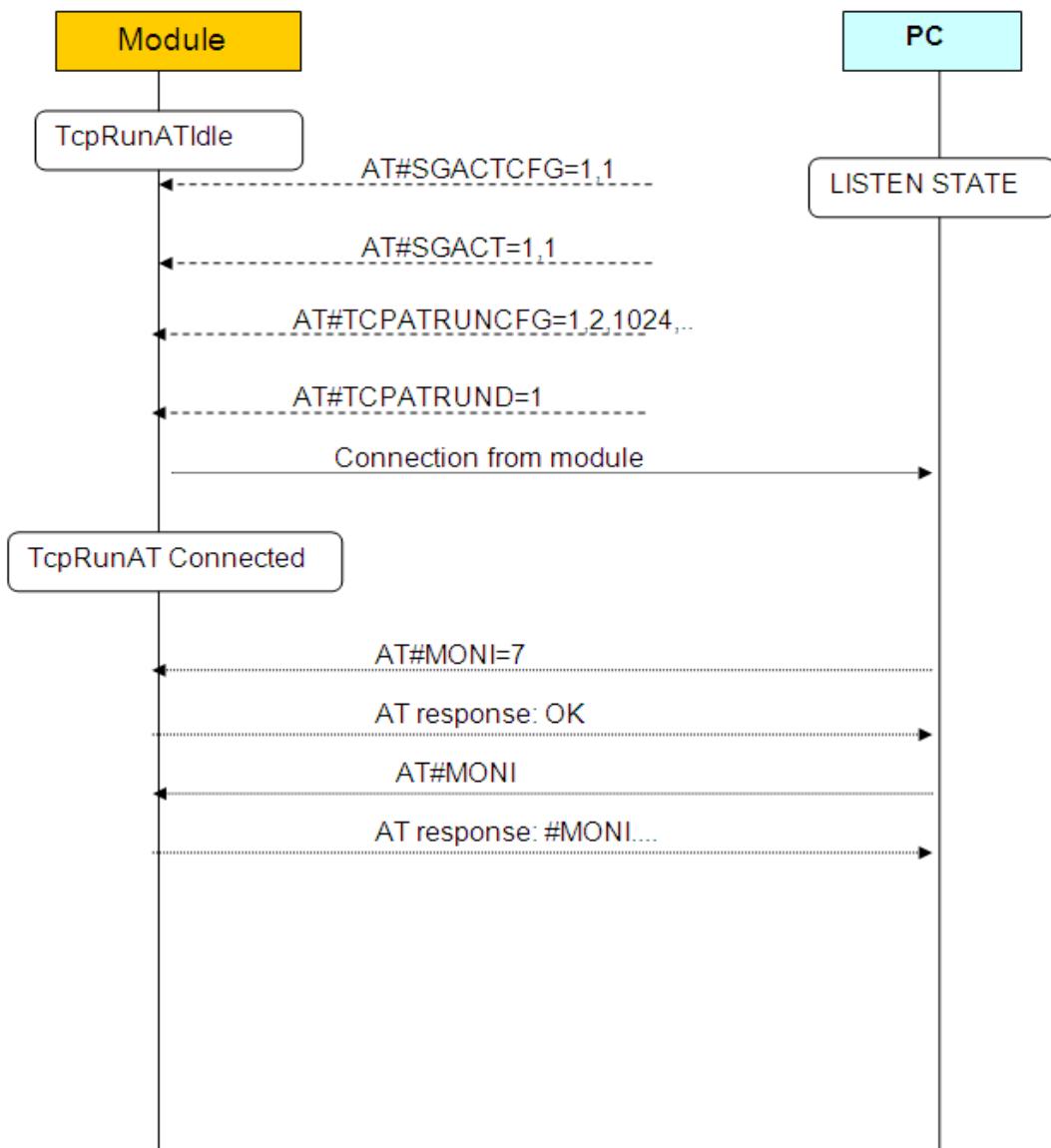


Fig. 7: TCPATRUN in Client Mode



3.4. AT commands in sequence

The standard usage of AT interface consists in waiting for response to an AT command before issuing another one. Each command ends with the delimiter character (<CR> is the default). Alternatively, AT commands can be issued separated by “;”, the string is treated like a single command. Consider the following AT commands string:

AT# SGACT=1,1;#SS

For example, the response to the string is:

#SGACT: 212.141.252.44

#SS: 1,1,212.141.121.05, 1024,212.141.121.148,1332

#SS: 2,0

#SS: 3,0

#SS: 4,0

#SS: 5,0

#SS: 6,0

OK

What happens if the remote application doesn't wait for an AT command response before issuing another one?

If AT commands are issued with the delimiter character but without waiting for response, some commands can be ignored, for example in the sequence

AT#SGACT=1,1

AT#SS

The AT#SS would be ignored.

In the case of AT commands managed by the TCPATRUN service, a new feature has been introduced that solves this problem. This feature is enabled through the command:

AT#TCPATCMDSEQ=1.



Consider the following AT commands sequence:

```
AT#SGACT=1,1
AT#SS
```

For example, the response to the sequence is:

```
#SGACT: 212.141.252.44
```

OK

```
#SS: 1,1,212.141.121.05, 1024,212.141.121.148,1332
```

```
#SS: 2,0
```

```
#SS: 3,0
```

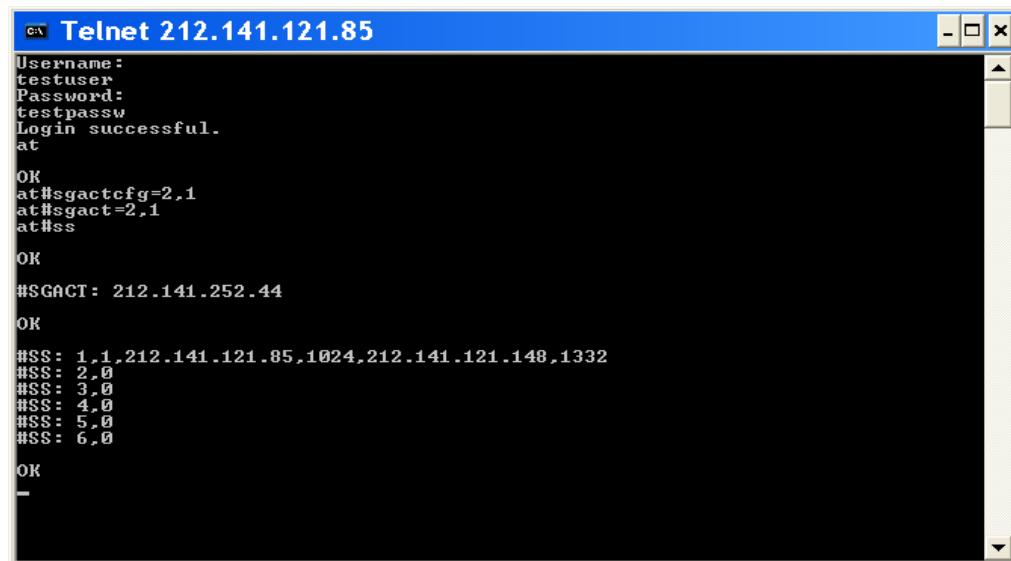
```
#SS: 4,0
```

```
#SS: 5,0
```

```
#SS: 6,0
```

OK

The AT#TCPATCMDSEQ command can be set from any instance, but it just affects AT commands managed by the ATRUN service.



The screenshot shows a Telnet session window titled "Telnet 212.141.121.85". The session starts with a login prompt:

```
Username: testuser
Password: testpassw
Login successful.
at
```

Following the login, the user enters the AT command sequence:

```
OK
at#sgactcfg=2,1
at#sgact=2,1
at#ss
```

The device responds with "OK" and then displays the current SS configuration:

```
#SGACT: 212.141.252.44
OK
```

Next, the user sends the AT#SS command again:

```
#SS: 1,1,212.141.121.05, 1024,212.141.121.148,1332
#SS: 2,0
#SS: 3,0
#SS: 4,0
#SS: 5,0
#SS: 6,0
```

The device responds with "OK" and a hyphen character at the end of the session.



3.5. Transparency

Data running on the specified serial port can be transferred, in transparent mode, directly to the TCP ATRUN. That is to say that any data input on the serial port can be sent to the PC via TCP/IP protocol (and vice-versa). This connection is opened by means of the AT#TCPATCONSER=<port>,<rate> command using the TCPATRUN instance.

After this command has been issued, and if no error has occurred, a "CONNECT" message is returned from the module to advise the user that the TCPATRUN instance is in *online mode* and connected to the specified port.

To exit from *online mode* and close the connection, the escape sequence (the default is "++") has to be sent on the TCPATRUN instance, taking into account that the escape sequence needs to be sent in one single packet. The use of Telnet for Windows sending every single byte in a TCP packet is not appropriate to perform this connection.



3.6. Alerts

- The TCPATRUN service uses a socket identifier (id) provided by the Easy IP service, the id is configured via AT#TCPATRUNCFG command. Please, refer to the Telit IP Easy User Guide to have more information about the use of AT commands concerning TCP/IP sockets configuration.
- The AT#SCFG command allows the user to read and change: the packet size, Transmission Timeout, and Idle Activity Timeout:
 - **Transmission Timeout:** if the bytes to be sent from the module are less than the packet size, then they will be sent just at Transmission Timeout expiry. So in the case of TCPATRUN service, it is suggested to set the minimum Transmission Timeout to avoid waiting for responses. It is strongly recommended to avoid having it set to infinite. It is also recommended not to set a Packet Size smaller than the default one.
 - **Idle Activity Timeout:** if there is no data exchange within the Idle Activity Timeout period, the connection is closed. This is useful in the following situation: if the TCP/IP socket is closed by the PC in an “unilateral” way, without signal exchange with the other side of the connection, on the module side the connection is still up, but cannot be used anymore. In this situation, if the Idle Activity Timeout has been set locally on the module, it will automatically close the connection, otherwise the only way to close it is using the AT#TCPATRUNCLOSE command locally.
In particular, if the PC closes the connection in the “unilateral” way during a “transparent mode”, see chapter 3.5, there is no way to close locally the TCP ATRUN. So if the user wants to use the “transparent mode”, it is very important to set the Idle Activity Timeout.
After the connection local closure, the TCPATRUN status is still enabled, so the service re-starts automatically.
- If the TCPATRUN service, in server or client mode, is activated on the first instance (<muxInstance> = 1 in AT#TCPATRUNCFG), that instance is only dedicated to TCPATRUN service. Use CMUX (multiplexer) to access the second or third instance and deactivate the service to free the instance, refer to the document: Virtual Serial Device, 80000NT10045A.

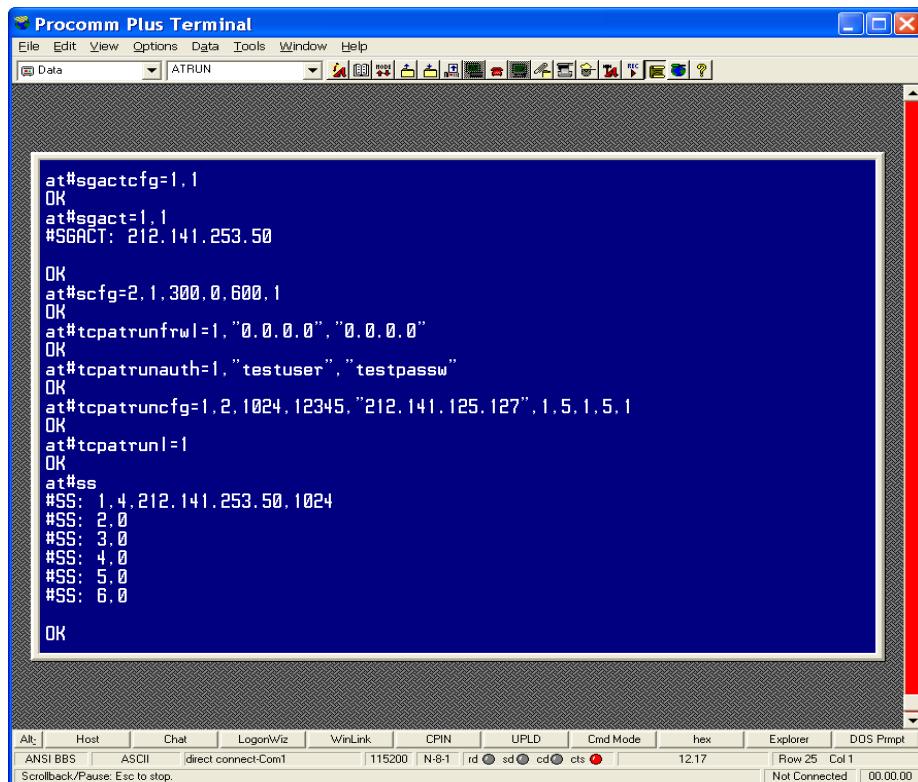
NOTICE: Using HE910 products could be not necessary to use the CMUX if two ports are available at the same time, refer to the document: HE910 Family Ports Arrangements, 1vv0300971.



3.7. Examples

3.7.1. TCPATRUN in Server Mode

The Fig. 8 shows the AT commands list used to configure the service in Server Mode.



```

Procomm Plus Terminal
File Edit View Options Data Tools Window Help
Data ATRUN
at#sgactcfg=1,1
OK
at#egact=1,1
#SGACT: 212.141.253.50

OK
at#scfg=2,1,300,0,600,1
OK
at#tcpatrunfrwl=1,"0.0.0.0","0.0.0.0"
OK
at#tcpatrunauth=1,"testuser","testpassw"
OK
at#tcpatruncfg=1,2,1024,12345,"212.141.125.127",1,5,1,5,1
OK
at#tcpatrunl=1
OK
at#ss
#SS: 1,4,212.141.253.50,1024
#SS: 2,0
#SS: 3,0
#SS: 4,0
#SS: 5,0
#SS: 6,0

OK

```

Alt Host Chat LogonWiz WinLink CPIN UPLD Cmd Mode hex Explorer DDS Pmpk
ANSI BBS | ASCII direct connect-Com1 | 115200 N-8-1 rd sd cd cts | 12.17 | Row 25 Col 1
Scrollbar/Pause: Esc to stop. Not Connected 00.00.00

Fig. 8: Module in Server Mode



Running AT Commands Remotely

80000NT10029a Rev.9 – 2015-06-04

The Fig. 9 shows the sending of the AT#MONI=7 command to the server (module) and the reception of the command results sent back by the server after the execution of the command.

```

Telnet 212.141.112.216
Username: testuser
Password: testpassw
Login successful.
at+creg?
+CREG: 0,1
OK
at#moni=7
OK
at#moni
#MONI: Cell BSIC LAC CellId ARFCN Power C1 C2 TA RxQual PLMN
#MONI: S 75 55FA 1297 983 -72dbm 32 32 1 0 I WIND
#MONI: N1 20 D5BD 5265 59 -70dbm 38 0
#MONI: N2 20 55FA 0000 979 -76dbm 28 23
#MONI: N3 72 55FA 1281 975 -81dbm 23 17
#MONI: N4 20 55FA 12EA 985 -85dbm 19 13
#MONI: N5 72 55FA 1D0D 751 -88dbm 14 22
#MONI: N6 73 55FA 1D23 754 -92dbm 10 19
OK

```

Fig. 9: PC (Client) using Telnet



3.8. Delay and Notes

3.8.1. Delays

Some AT commands can fail if executed by means of the TCPATRUN service due to an interaction of the AT command running with the GPRS service. The problem can be avoided by inserting a delay before AT command execution. The delay is set through the following command:

AT#ATRUNDELAY=0,10 sets for TCPATRUN commands a delay of 10 seconds

Once set, the delay is valid till a new AT#ATRUNDELAY is issued. The following table shows the suggested delays (in seconds).

AT+COPS=?	10
-----------	----

NOTICE: the delay is needed in GSM technology.

3.8.2. Notes

1) AT#SKTD, AT#SKTL, AT#SKTSV:

Initial setting for TCPATRUN service (through AT#TCPATRUNCFG) has to be done:
<connId> different from <muxInstance>.

2) AT#QDNS:

a) Initial setting for TCPATRUN service (through AT#TCPATRUNCFG) has to be done: <muxInstance> = 1.
b) Initial setting by the command AT#SCFG has to be done:
<connId> set for TPCATRUN service(through AT#TCPATRUNCFG) has to be associated to a <cid> different from 1.

3) AT#GPRS: same as 2b)

4) AT#SKTOP: same as 2) and furthermore:

Initial setting for TCPATRUN service (through AT#TCPATRUNCFG) has to be done:
<connId> different from 1.

NOTICE: HE910 products don't support AT#SKTOP command.

5) If initial setting for TCPATRUN service (through AT#TCPATRUNCFG) has been done:
<connId> = <muxInstance>,



Running AT Commands Remotely 80000NT10029a Rev.9 – 2015-06-04

then parameters set by AT#PKTSZ, AT#SKTTO, AT#DSTO affect the same <connId> socket on which TCPATRUN service is running.

- 6) AT+CGATT=0, AT+CGACT=0, AT+CGCLASS="CC", AT#SHDN, AT#BND=2, AT#BND=3, ATQ1, ATQ2, AT#SIMDET=0, AT+CFUN=2 and AT+CFUN=4 commands cause an irreversible block of the TCP AT RUN mechanism;
- 7) The commands of AT#CSURV family can have the response truncated if too long: these commands in fact send their responses during the network search, but while they are executed, the GPRS service is suspended and so the TCPATRUN cannot transmit anything. It stores these responses in a buffer till the network search completes. Anyway, the use of this command is deprecated.

NOTICE: this problem arises in GSM technology.

- 8) The ATZ command returns ERROR if a Data Call is up and this Data Call has been set up on a "muxInstance" different from the TCPATRUN "muxInstance". The ATZ returns ERROR also in the case the TCPATRUN works on an IPEGSM context (opened through AT#SGACT=0,1), because the IPEGSM context is based on a Data Call.



3.9. TCPATRUN and Python

TCPATRUN service allows downloading, upgrading, and executing a Python script into Telit modules via a remote TCP/IP connection (e.g. Telnet). TCPATRUN service is configured in server mode. Below is a generic example illustrating this feature.

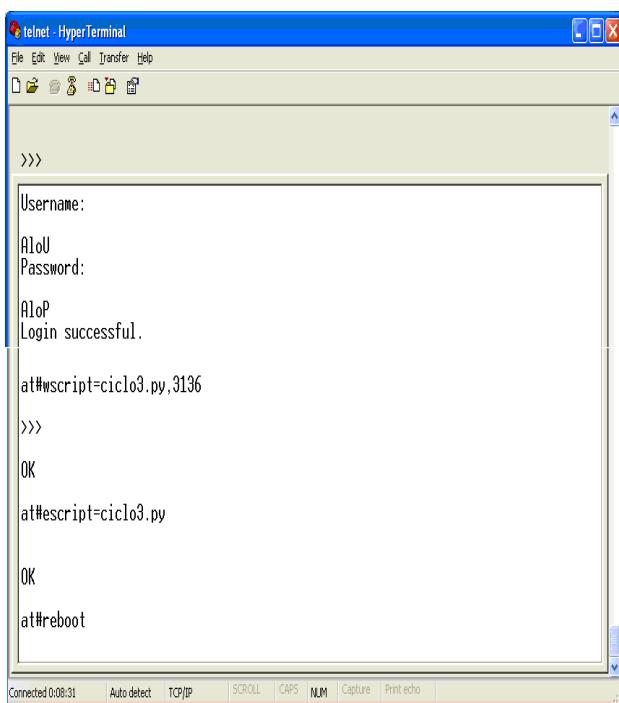
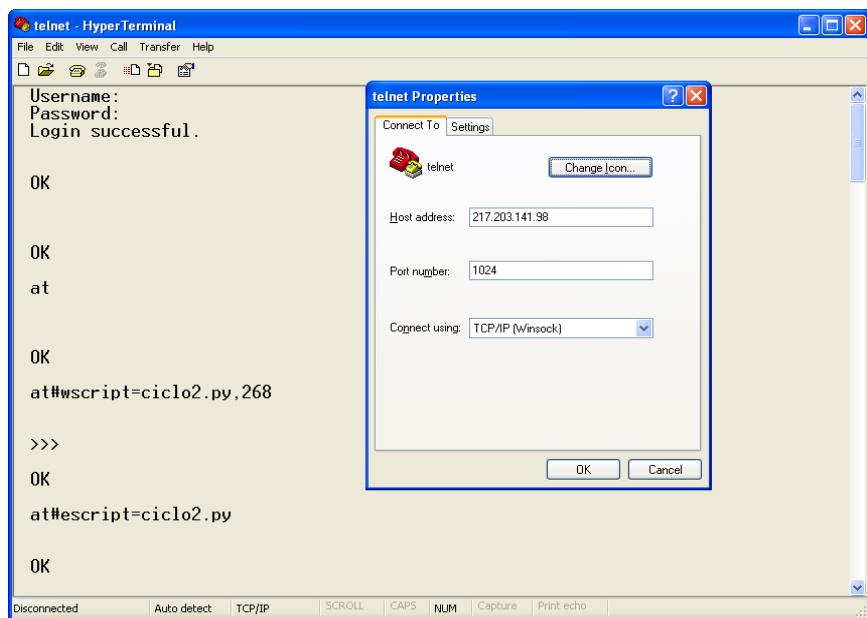
```
AT+CGDCONT=1,"IP","myAPN"  
  
AT#SCFG=1,1,1500,0,600,50      // set Inactivity Timeout to 0  
  
AT#TCPATRUNCFG=1,3,1024,...   // listening on port 1024  
  
AT#TCPATRUNFRWL=2;#TCPATRUNFRWL=1,"217.0.0.0","255.0.0.0"  
  
AT#TCPATRUNAUTH=1,"MyUser","MyPwd"  
  
AT#SGACT=1,1                  // activate the context  
  
AT#TCPATRUNL=1                // start TCPATRUN service
```

A remote computer works as an IP client and Telnet is used to connect to the module.

To start a Telnet session on the remote computer, it is possible to use Hyper Terminal or any other terminal emulator via TCP/IP, setting the right port (1024 in this case) and the IP address of the module (returned by AT#SGACT). After the login phase (login requires username and password), the Python script can be downloaded on the module, entering on Telnet, AT#WSHRIPT command using the same settings described in the Telit Easy Script Python User Guide, see the following screenshot.



Running AT Commands Remotely 80000NT10029a Rev.9 – 2015-06-04



Refer to the screenshot on the left side: we recommend the following procedure to upgrade a running Python script. The procedure can be easily executed using remote Telnet.

- 1) Download a new PY script via the AT#WSCRIPT command,
- 2) Enable the new PY script via AT#ESCRIPT=ciclo3.py”,
- 3) Reboot the module with AT#REBOOT to start the execution of the new PY script (ciclo3.py).



3.10. TCPATRUN and AppZone

TCPATRUN service allows also upgrading an App into Telit modules via TCP/IP connection (e.g. Telnet). AppZone feature is supported only by xE910 product family. TCPATRUN service is configured in serve mode. Below is reported a generic example illustrating this feature.

```

AT+CGDCONT=1,"IP","myAPN"

AT#SCFG=1,1,1500,0,600,50                                // set Inactivity Timeout to 0

AT#TCPATRUNCFG=1,3,1024,...                            // listening on port 1024

AT#TCPATRUNFRWL=2;#TCPATRUNFRWL=1,"217.0.0.0","255.0.0.0"

AT#TCPATRUNAUTH=1,"user","pwd"

AT#SGACT=1,1                                         // activate the context

AT#TCPATRUNL=1                                       // start TCPATRUN service

```

A remote computer works as an IP client and Telnet is used to connect to the module.

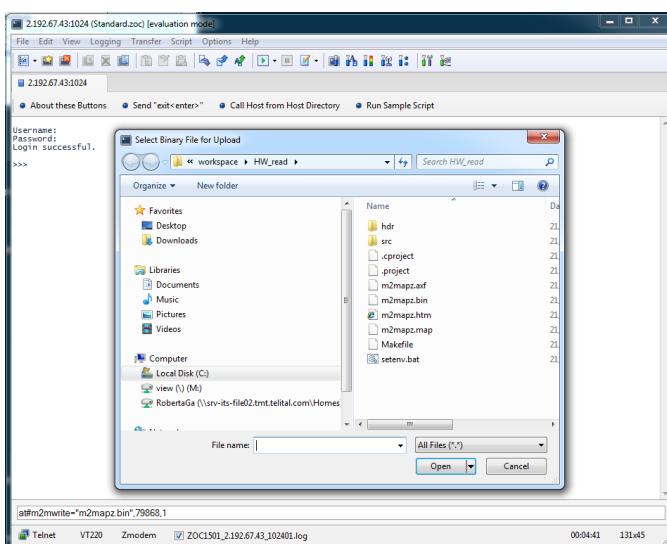
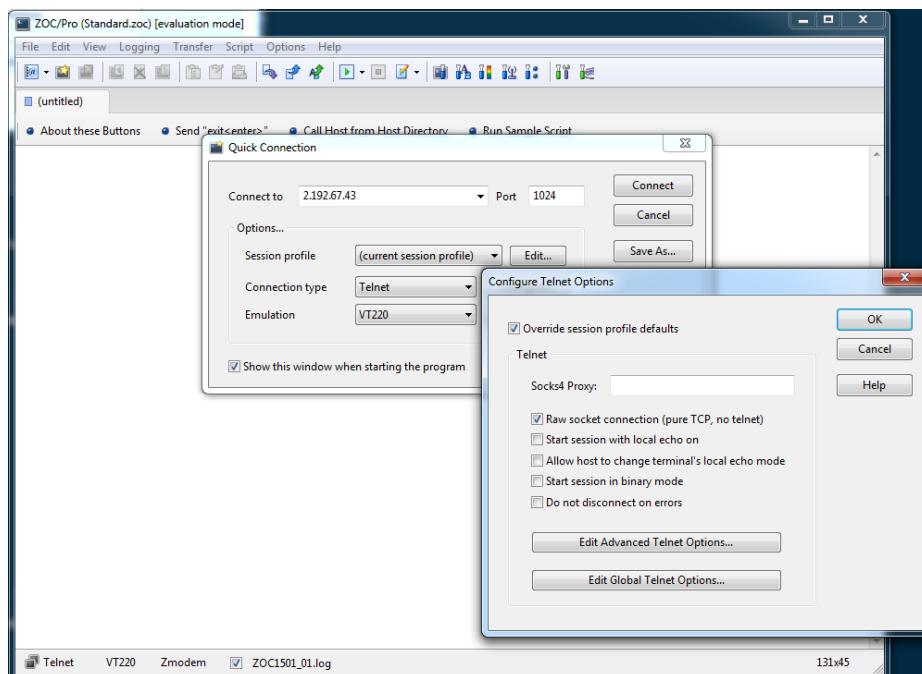
To start a Telnet session on the remote computer, it is possible to use any terminal emulator via TCP/IP that supports RAW ASCII, setting the right port (1024 in this case) and the IP address of the module (returned by AT#SGACT). After the login phase (login requires username and password), the AppZone application can be downloaded on the module, entering on Telnet, AT#M2MWWRITE command using the same settings described in the command description.

The procedure to upgrade the app is similar to that explained for python, an example of remote AppZone application upgrade using ZOC as terminal emulator on client side is reported:



Running AT Commands Remotely

80000NT10029a Rev.9 – 2015-06-04



1) Download the new App through AT#M2MWRITE command (be sure that file is sent using RAW ASCII file transfer and without any Telnet protocol)

2) Enable new App execution with AT#M2MRUN

3) Run the new App

