

Running AT Commands Remotely Application Note

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, SUPPORT5
 - 1.2. TEXT CONVENTIONS6
 - 1.3. RELATED DOCUMENTS6
 - 1.4. DOCUMENT HISTORY7
- 2. SMSATRUN8**
 - 2.1. SMS FORMATS8
 - 2.2. ALERTS.....10
 - 2.3. DESCRIPTION11
 - 2.4. EXAMPLE15
 - 2.5. DELAYS AND NOTES17
 - 2.5.1. *Delays*.....17
 - 2.5.2. *Notes*17
- 3. TCPATRUN19**
 - 3.1. CONFIGURATION20
 - 3.2. TCPATRUN IN SERVER MODE21
 - 3.2.1. *Description*.....21
 - 3.2.2. *Authentication mode*.....24
 - 3.3. TCPATRUN IN CLIENT MODE25
 - 3.3.1. *Description*.....25
 - 3.4. AT COMMANDS IN SEQUENCE27
 - 3.5. TRANSPARENCE29
 - 3.6. ALERTS.....30
 - 3.7. EXAMPLES.....31
 - 3.7.1. *TCPATRUN in Server Mode*.....31
 - 3.8. DELAY AND NOTES.....33
 - 3.8.1. *Delays*.....33
 - 3.8.2. *Notes*33
 - 3.9. TCPATRUN AND PYTHON.....35
 - 3.10. TCPATRUN AND APPZONE37



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
-



2.3. Description

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

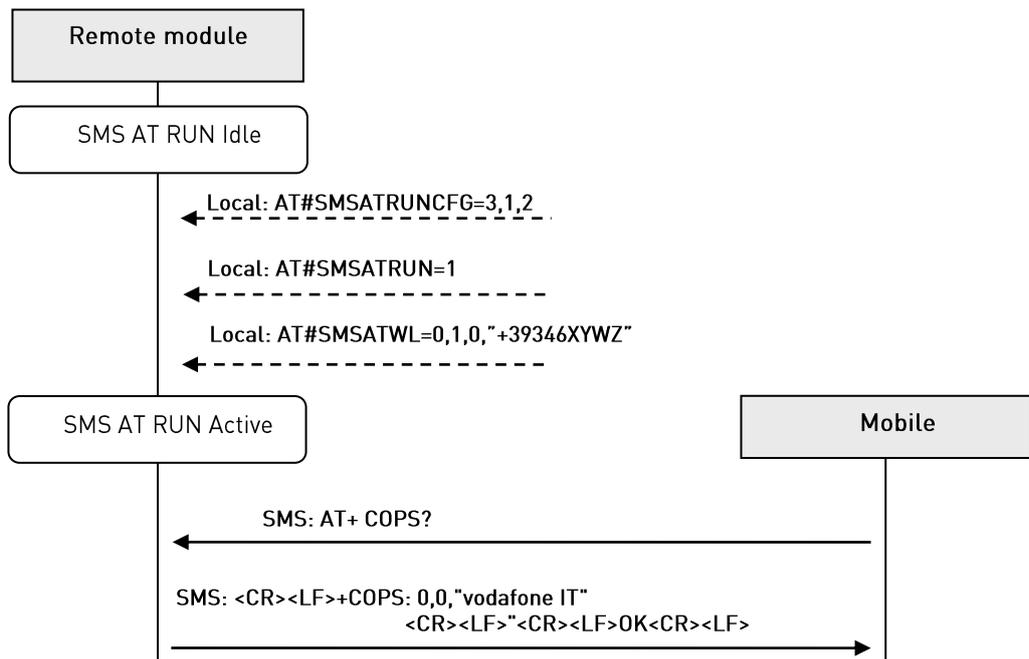


Fig. 2: Local Configuration and SMS messages Exchange

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

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:



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.

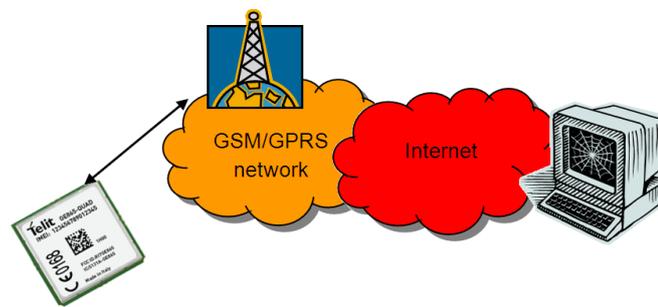


Fig. 5: TCPATRUN Scenario

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#TCPATRNL` 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#TCPATRNL =1`

See the examples showed in chapter 3.7.1.



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

Hereafter are some 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.



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.



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
```

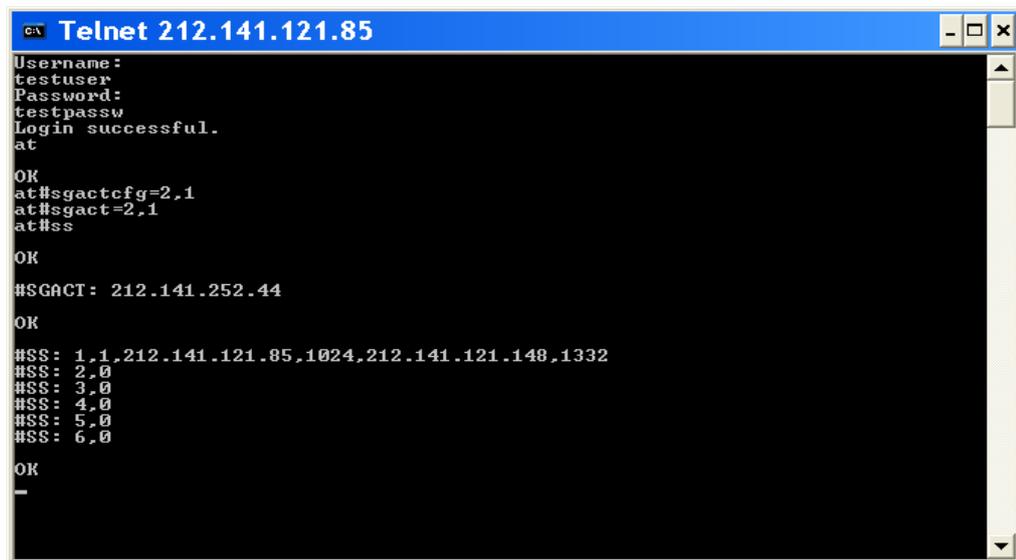
```
#SS: 4,0
```

```
#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.



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, 1v0300971.



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#TCPATRUNAATH=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#WSCRIPT command using the same settings described in the Telit Easy Script Python User Guide, see the following screenshot.



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#M2MWRITE 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:



