Telit SE867 & Python Application Note

80000nt10040a Rev.2 – 2010-10-18

Making machines talk.

APPLICABILITY TABLE

PRODUCT
GE865-QUAD
GE864-QUAD V2
GC864-QUAD V2
SE867-AGPS

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 document.

Contents

1. Introduction

1.1. Scope

Scope of this document is to provide some examples about using embedded Telit Python script interpreter in Telit GSM/GPRS modules in order to control the Telit SE867-AGPS stand alone module.

1.2. Audience

This document is intended for Telit customers aiming to develop an application suited to embed the Telit GE865-QUAD (or another module cited in Applicability Table) as GSM/GPRS modem and Telit SE867-AGPS as stand alone GPS module.

1.3. 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.

1.4. Document Organization

This document contains the following chapters:

"Chapter 1: "Introduction" provides a scope for this document, target audience, contact and support information, and text conventions.

"Chapter 2: "System overview" gives an overview of the system including GE865-QUAD and SE867 modules.

"Chapter 3: "Telit Python IDE" give a brief overview about Telit Python IDE.

"Chapter 4: "Pass-Through application" explains details about the Pass-through application.

"Chapter 5: "Remote Tracking application" explains details about the Remote Tracking application.

1.5. 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.

1.6. Related Documents

- [1] AT Commands Reference Guide, 80000ST10025a
- [2] Easy Script in Python, 80000ST10020a
- [3] Telit GE865 Product Description, 80309ST10054
- [4] GE/GC864-QUAD V2 Product Description, 80331ST10074A

- [5] Telit SE867-AGPS Product Description, 80311ST10073
- [6] Telit GE865 Hardware User Guide, 1vv0300799
- [7] GE864-QUAD V2 Hardware User Guide, 1vv0300841
- [8] Telit SE867 User Guide, 1VV0300860
- [9] SE867-AGPS Assisted GPS Application Note, 80000nt10036a r0

1.7. Document History

Revision	Date	Changes
ISSUE#0	2010-06-25	First issue
ISSUE#1	2010-09-06	Updated §2.1 Specified serial port ASC0 in §2.2, 4.0 and 4.4
ISSUE#2	2010-10-18	Add Assisted GPS Support and wake up events: - Updated §1.6 - Updated chapter 5

2. System overview

Using the Python interpreter inside the Telit GE865-QUAD, it is possible to control the GPS module Telit SE867-AGPS in order to easy create a position-tracking system.

2.1. Hardware architecture

GE865-QUAD could be connected to the SE867-AGPS through the ASC1 port, reserving the ASC0 port to the usage with CMUX (AT parser instances and RTD debug) or for the pass-through feature.

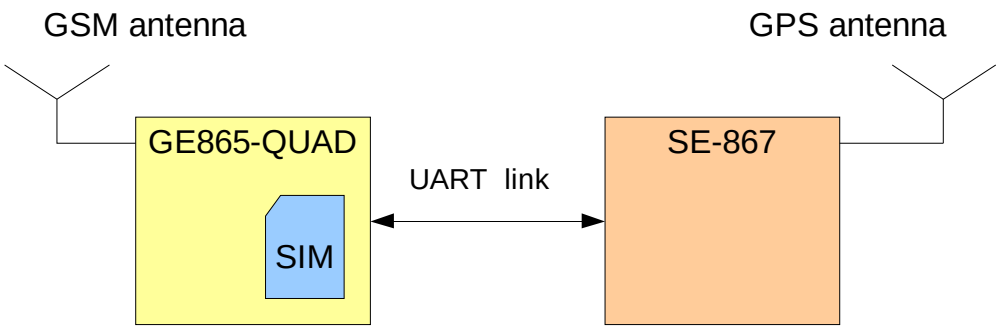

Figure 2.1

In order to complete the position-tracking system, SIM card, GPS and GSM antenna are required.

Figure 2.1 shows the connections between the main components of the system.

2.2. Software architecture

As showed in the Figure 2.2, a Python script running inside the GE865-QUAD is capable to manage the information coming from GPS module (NMEA sentences) and redirect them over GSM/GPRS network (Remote Tracking application feature) or directly toward the ASC0 port of Telit module (Pass-through application feature).

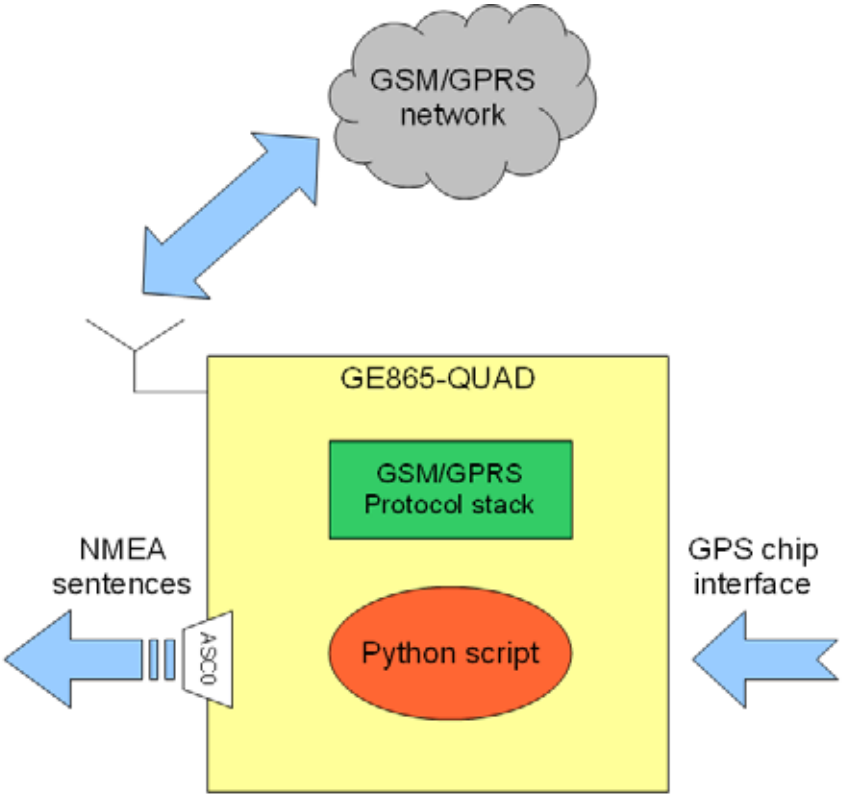

Figure 2.2

2.3. System setup

After the set up of hardware connections it is necessary to install the Python scripts inside the GE865-QUAD. These scripts will be provided by Telit as source code (*.py) in order to permit some customizations.

Two files are available:

- passThrough.py (Pass-through application)
- core.py (Remote tracking application)

Each one implements a different feature and is independent from the other one. The following paragraphs will introduce some notions about the Telit Python environment and will explain the characteristics of each script with all the necessary steps to activate them.

3. Telit Python IDE

In order to customize the source files provided by Telit it is necessary to install Telit Python development environment (file TelitPy1.5.2+_v4.1.exe).

3.1. Installation of Telit Python IDE

For information about installation of Telit Python IDE please refer to document [2].

3.2. Python script customization procedure

Once installed the IDE it is possible to customize the desired file following the steps listed below:

1) Open the file core.py using the menu File->Open as showed in the Figure 3.1.

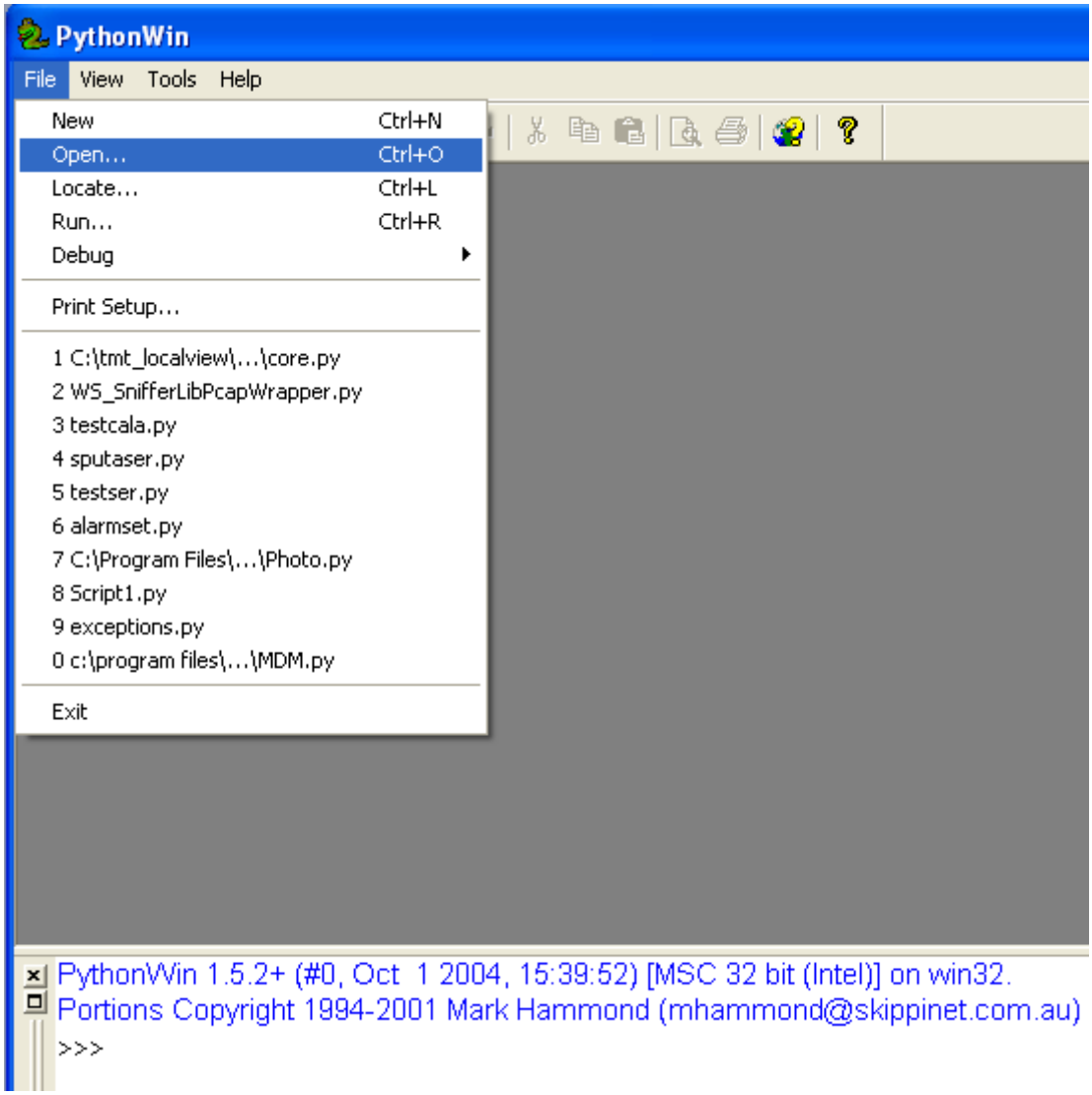

Figure 3.1

2) Edit the selected file using the functionalities offered by the IDE as showed in Figure 3.2 and apply the desired changes saving the file.

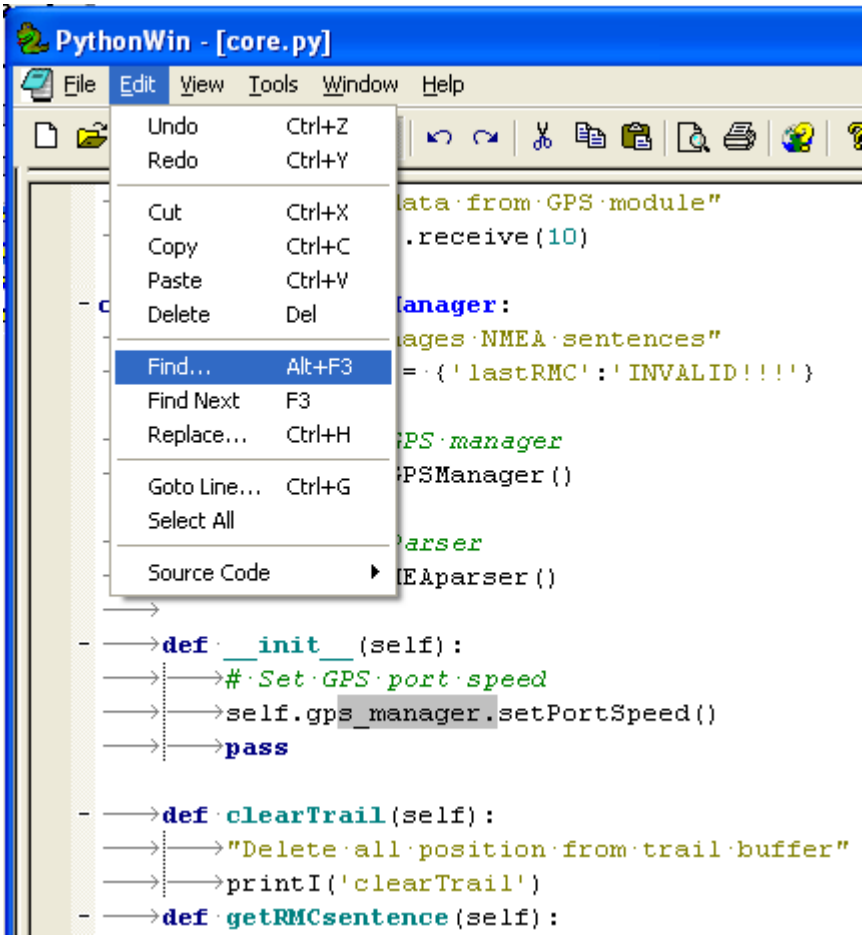

Figure 3.2

3) Once modified, the source file shall be cross-compiled.

Go to the folder where the modified file has been stored and use the right-button of the mouse clicking on it. It will appear a contextual menu similar to the one showed in Figure 3.3, select "Compile" in order to complete this step.

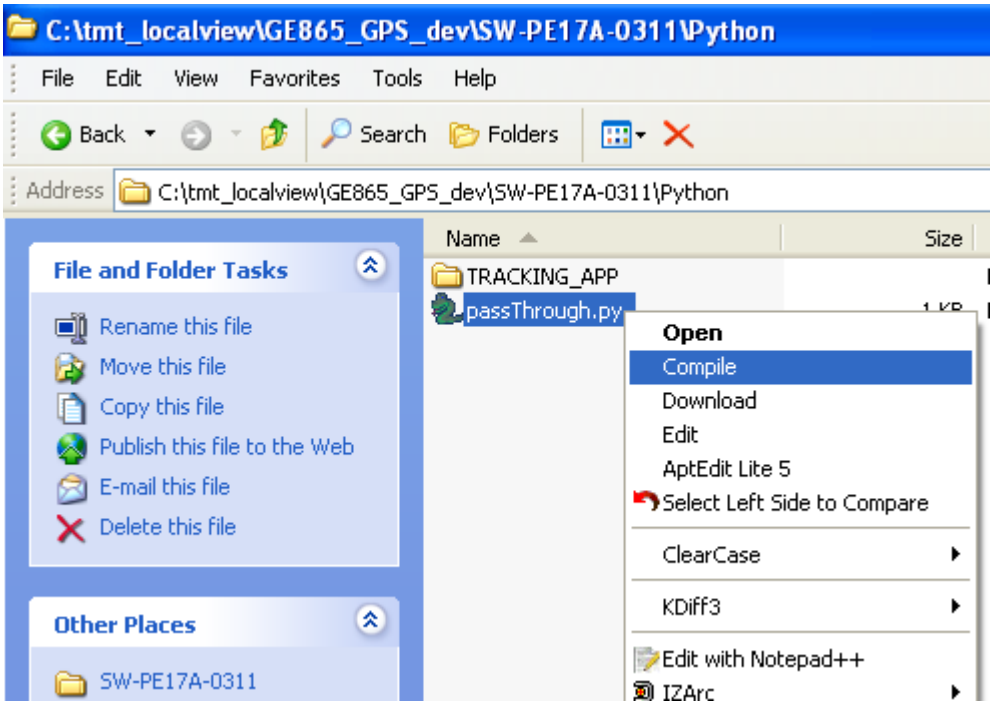

Figure 3.3

4) If there aren't any errors in the file syntax, the Python compiler will generate an associate xxx.pyo file. This file shall be downloaded in the GE865-QUAD module.

Use the right-button of the mouse clicking on it. It will appear a contextual menu similar to the one showed in Figure 3.4, select "Download" in order to complete this step.

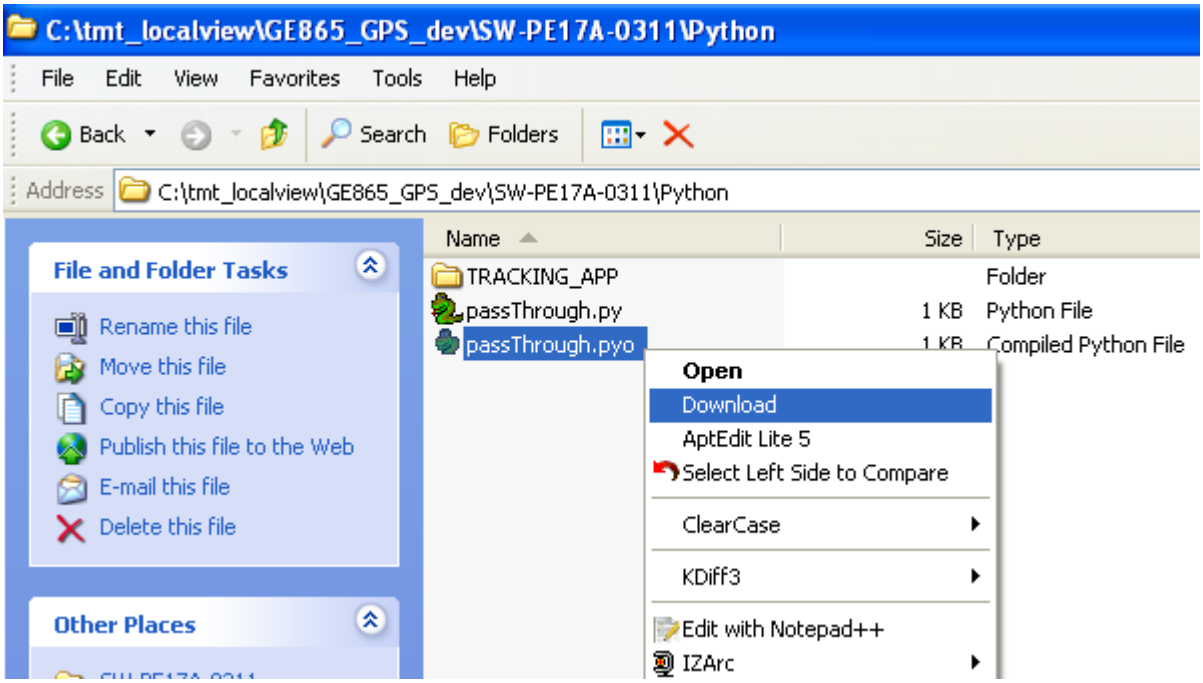

Figure 3.4

4. Pass-Through application

The Pass-Through application implements a simple pass-through of GPS data coming from GPS module toward the ASC0 serial port of the GE865-QUAD.

4.1. Script overview

The application is provided as a single file named "passThrough.py"; once installed on the GE865-QUAD, at the start-up, the script will initialize the GPS chip and then each byte received from the GPS chip interface will be redirected toward the ASC0 port. This communication channel is mono-directional; hence it is not possible to send data from ASC0 toward GPS chip.

4.2. Script installation

For details about installation on the Telit module please follow the steps 3) and 4) listed in paragraph 3.2.

4.3. Script start-up configuration

Once installed in the GE865-QUAD module it is necessary to enable the script in order to be executed at the Telit module power-on.

Follow the steps listed below in order to complete the procedure:

1) Connect, using a program like Hyper Terminal for Windows, to an instance of the AT Parser of Telit GE865-QUAD module.
2) Issue the command "AT#LSCRIPT" and check the response in order to be sure that the file xxx.pyo associated to the script has been installed properly on the Telit module.
3) Issue the command "AT#ESCRIPT=xxx.pyo", where xxx.pyo is the name of the desired file to be enabled, in order to make the file executable by the Python Virtual Machine.
4) Issue the command "AT#STARTMODESCR=0,10", in order to enable the execution of the script at the power-on of GE865-QUAD (the DTR line on the ASC0 serial port shall be low at the power-on).
5) If no error will be returned by the AT parser instance, the script will be executed at the next power-on of the GE865-QUAD.

For further details about AT Parser and AT Commands please refer to document [1].

4.4. Script running behaviour

After the first installation and the start-up configuration, the passThrough.pyo script will be executed automatically at the next restart and it could be stopped only with the power-off of the Telit module.

GPS data sent toward ASC0 could be received by an external host using a 9600bps 8N1 configuration for the serial link.

5. Remote Tracking application

The Remote Tracking application allows to get information about the GPS position of the system (GE865-QUAD + SE867-AGPS) using the GSM network.

5.1. Script overview

The application is provided as a single file named "core.py"; once installed on the GE865-QUAD, at the start-up, the script will initialize the GPS chip, by also performing Assisted GPS [9], and then it will be possible to get the GPS position of the system sending a specific command.

It is also possible to configure the application in order to receive the GPS position, periodically, without any request and putting the system in power-save mode between each "tracking-event".

Commands and responses are both channeled via SMS.

5.1.1. Main state machine

The behavior of the application could be schematized with the state machine diagram shown in Figure 5.1:

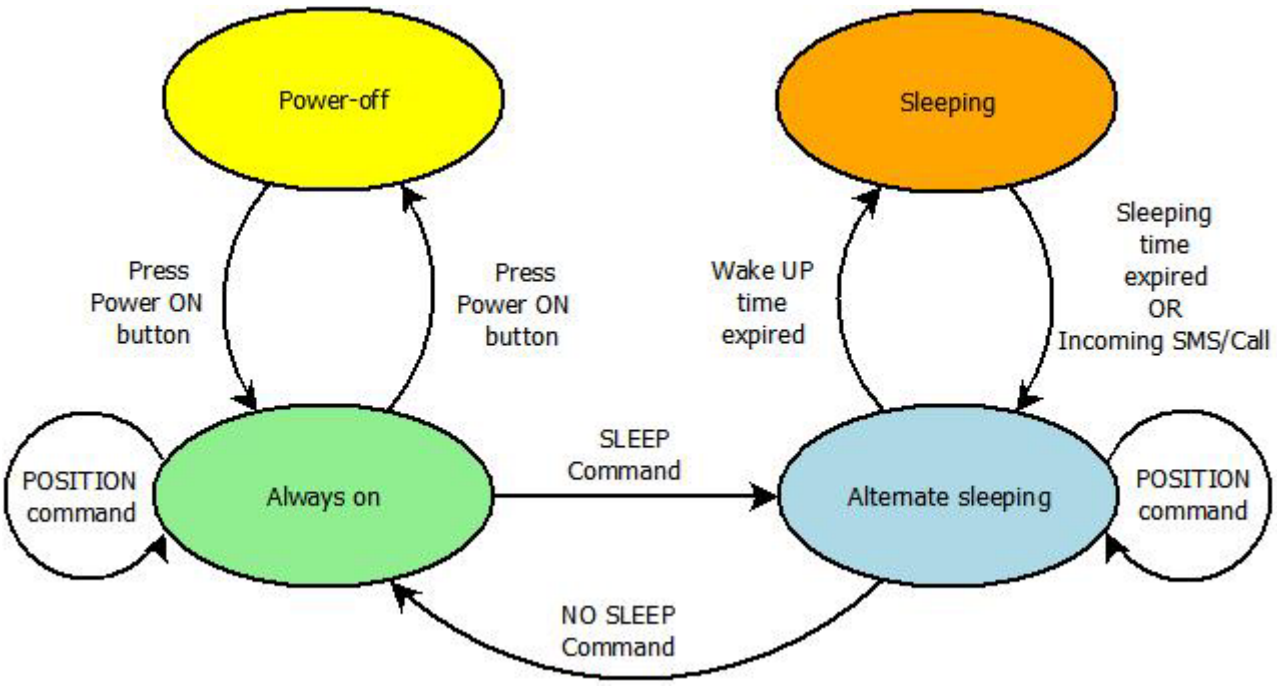

Figure 5.1

The meaning of each state is reported in Table 5.1

State	Description
Power-off	This is the default state of the system when the GE865-QUAD has been just connected to a power-supply but the power-on button hasn't been pressed yet.
Always on	Once pressed the power-on button, the GE865-QUAD will execute the script "core.pyo" and the system will be in the "Always-on" state. In this state the system is waiting for SMS-command from the GSM Network and it will return in this state after the execution of a SMS with a POSITION-command. If the power-on button is pressed during this state, the system will return in the Power-off state.
Alternate sleeping	When the system receives a SMS with a SLEEP-command, it will switch to the "Alternate sleeping" state. In this state the system will enter in power-saving mode (GSM in +CFUN=0 power saving mode [1] and GPS not active) for a certain amount of time (180 seconds). At the end of this "sleeping interval", or whenever a SMS/Call is received, the system will wake-up (GSM active and GPS active) and it will send the GPS position toward a "default phone number" via SMS. In order to manage additional incoming SMS-commands the system will wait up to 60 seconds before re-enter in power-saving mode and restart the cycle of alternate wake-up. If a SMS with NO SLEEP-command is received during this state, the system will switch to the "Always on" state. If a SMS with POSITION-command is received during this state, the system will execute the command and it will return to this state.
Sleeping	In this state the system is in power-saving mode (GSM in +CFUN=0 power saving mode [1] and GPS disabled) and current consumption is reduced to the minimum. In this state the system can be awoken by an incoming SMS/Call or by the expiration of the sleeping interval.

Table 5.1

5.1.2. SMS Commands

Table 5.2 lists the commands which can be sent via SMS in order to control the Remote Tracking Application.

Command	Description
POSITION	Retrieve the GPS position of the module. The SMS with the position will be sent toward the sender of the command.
SLEEP	Enable the tracking with sleeping mode (the system will switch to the "Alternate sleeping" state). In this mode the module will enter in a sleeping state for 180 seconds (default) and it will send the GPS position toward the default phone number at every wake up. In order to confirm the execution of the command a confirmation SMS will be sent toward the sender of the command. Important: After receiving the SLEEP-command the system will wait 60 seconds for additional incoming SMS-commands before entering in the sleeping state.
NO SLEEP	Exit from the tracking with sleeping mode (the system will switch to the "Always on" state). In order to confirm the execution of the command a confirmation SMS will be sent toward the sender of the command. An SMS containing a warning message will be sent to the sender if the system is already in the "Always on" state.

Table 5.2

Each SMS shall contain only one command (matching exactly the case listed above), without any additional character; otherwise it will be discarded automatically by the system and no response will be returned to the sender.

5.1.3. SMS Responses

The Table 5.3 lists the responses which could be received from the Remote Tracking Application via SMS.

Response	Description	Recipient
$GPRMC,012748.000,V,3828.8462,N,12242.4916,W,,,,,,N*68	It is the GPS position of the system. If it has been requested with a POSITION-command, the SMS with the position will be sent toward the sender of the POSITION-command (both in "Always on" and "Alternate sleeping" state). Otherwise, if the system has been switched in the "Alternate sleeping" state, the SMS with the position will be sent, automatically and periodically, toward the default phone number.	POSITION-command sender (if requested) or default phone number (if unsolicited)
SEE YOU LATER...	It is the confirmation SMS for the SLEEP-command. It will be sent if the command has been executed correctly.	SLEEP-command sender
OK, the alarm has been deactivated. NO-SLEEPING mode is now active!	It is the confirmation SMS for the NO SLEEP-command. It will be sent if the command has been executed correctly.	NO SLEEP-command sender
WARNING, the alarm was not active, stay in NO-SLEEPING mode!	It is the warning SMS for the NO SLEEP-command. It will be sent if the system is already in the "Always on" state and if the user sends an SMS with a NO SLEEP-command.	NO SLEEP-command sender

Table 5.3

The recipient of the SMS sent by the application is listed in the last column of the Table 5.3.

5.2. Script installation

For details about installation on the Telit module please follow the steps 3) and 4) listed in paragraph 3.2.

5.3. Script start-up configuration

Please refer to paragraph 4.3.

5.4. Script running behaviour

After the first installation and the start-up configuration, the core.pyo script will be executed automatically at the next restart of the Telit module.

In order to stop the script it is necessary to switch the system in the "Always on" state (using the NO SLEEP-command, if required) and then to power-off the Telit module.

5.4.1. Usage example

The diagram in Figure 5.2 shows a possible interaction between the user and the system using the SMS-commands.

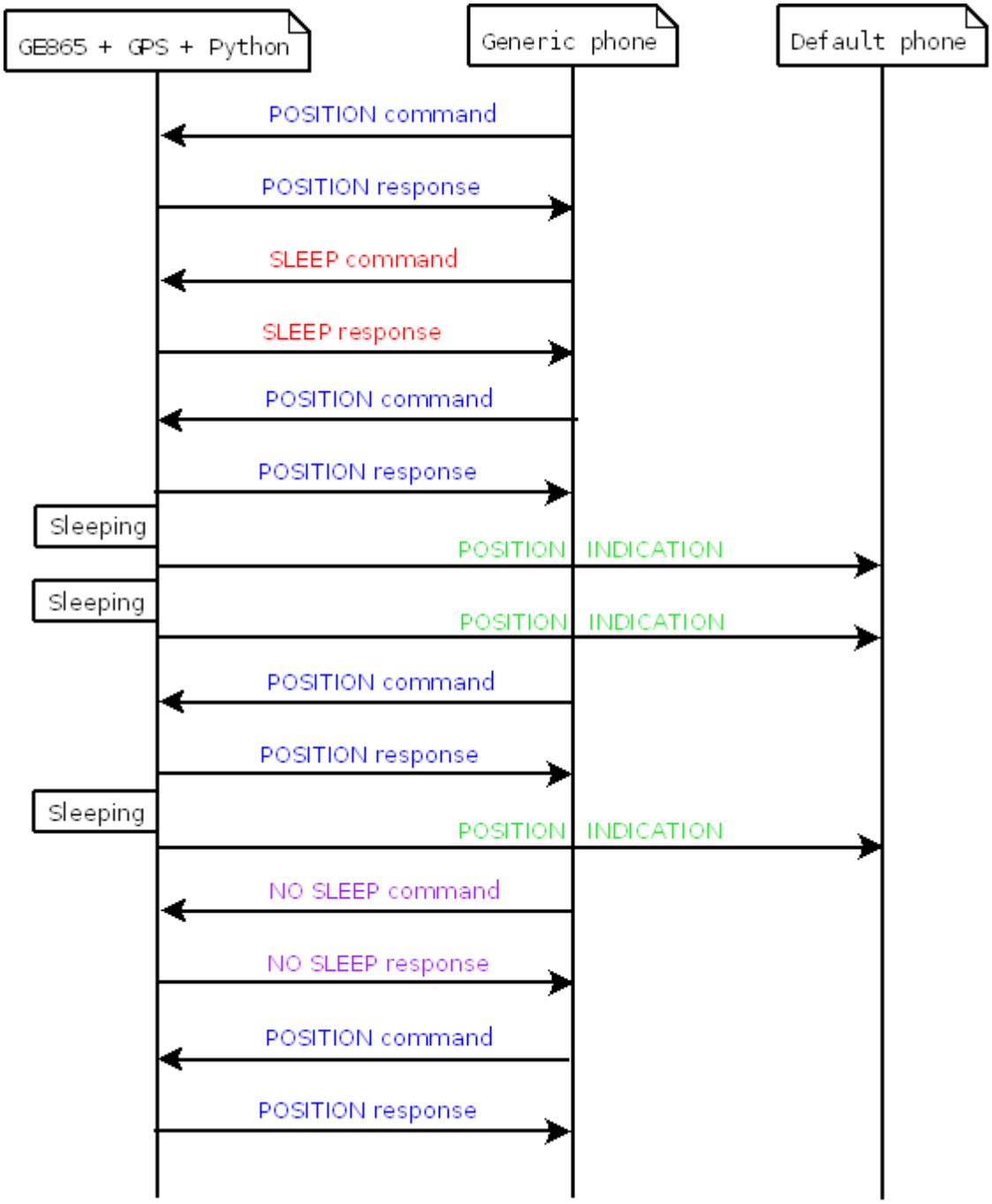

Figure 5.2

In this case, the user interacts with the system using two different phones:

1) Default phone, to receive the unsolicited SMS containing GPS position when the system is in the "Alternate sleeping" state.

2) Generic phone, to send SMS-commands and receive related response in order to control the system.

Using two different phones is not mandatory, in fact, through the script customization, it is possible to modify this aspect.

5.5. Script customization tips

Telit provides the source code of the Remote Tracking application in order to permit some useful customizations of the script behaviour.

This paragraph points out some parameters for the source code that permit a first level of customization.

Line	Parameter description	Range / Notes
7	The "GPS_PORT_SPEED" variable specifies the port speed used to communicate with the GPS module.	4800 for Orion version 3.1 GPS firmware / 9600 for Orion version 3.2 and higher GPS firmware
8	The "GPIO_BOOT_SEL" variable specifies the GPIO number used by the Python script to control the Boot Sel PIN of the GPS module.	Refer to hardware design
9	The "GPIO_RESET" variable specifies the GPIO number used by the Python script to control the Reset PIN of the GPS module.	Refer to hardware design
10	The "GPIO_POWER_ON" variable specifies the GPIO number used by the Python script to control the Power ON PIN of the GPS module.	Refer to hardware design
13	The "TIME_SERVER" variable specifies the Time Server (RFC-867) used by the Python script to retrieve UTC time for time assistance during Assisted GPS.	Refer to [9] for further information on Assisted GPS protocol and its usage Please contact Telit Technical Support Center (TTSC) for Assistance Server Access Credentials
14	The "TIME_SERVER_PORT" variable specifies the Time Server Port (RFC-867) used by the Python script to retrieve UTC time for time assistance during Assisted GPS.	
15	The "AGPS_SERVER" variable specifies the Assistance Server used by the Python script to retrieve the assistance file to be used during Assisted GPS.	
16	The "AGPS_USER" variable specifies the User Name used by the Python script to log on the Assistance Server.	
17	The "AGPS_PASSW" variable specifies the Password used by the Python script to log on the Assistance Server.	
18	The "APN" variable specifies the Access Point Name used by the Python script to set up the GPRS Context.	

20	The "SLEEP_INTERVAL" variable specifies the duration of the sleeping interval. It could be increased in order to improve power saving performance.	180 seconds suggested
24	The "def_recipient_num" key in the dictionary "all_values" contains the default phone number used to send the unsolicited SMS with GPS position.	It could be expressed in both national and international (with "+") form
24	The "wake_up_interval" key in the dictionary "all_values" contains the amount of seconds which the woken system waits before enter/re-enter in the sleeping mode. It could be reduced in order to improve power saving performance.	wake_up_interval > 60 (suggested)