Telit
GM862-PYTHON
GM862-PCS
GM862-GPRS
GM862-GSM
Product Description

Contents

1 Overview

Aim of this document is the description of features, functions and interfaces of the Telit GM862-PYTHON, Telit GM862-PCS, Telit GM862-GPRS, Telit GM862-GSM modules.

The Telit GM862-PYTHON, Telit GM862-PCS, Telit GM862-GPRS, Telit GM862-GSM modules are small, lightweight and low power consumption devices that allow digital communication services wherever a GSM 900, DCS 1800, PCS1900[1] network is present; they are the evolution of the previous Telit GM862 module, with an on board SIM Reader and therefore standalone Full Type Approval.

The Telit GM862-PYTHON module is the extended GPRS triband evolution of the previous wireless telecommunication module Telit GM862-PCS, integrating the Easy Script feature: a script interpreter Python engine allowing self controlled operations. It includes all the features of the previous PCS module such as GPRS, Voice, Circuit Switch Data transfer, Fax, Phonebook and SMS support, the third band PCS1900 support, hot removal sensing on board SIM Reader, therefore standalone FTA, and the higher upload speed of the GPRS Class 10 Packet Data transfer, battery charger circuitry. With the Easy script feature the Telit GM862-PYTHON module becomes a finite product, it just needs your script to be run.

It is specifically designed and developed by Telit for OEM usage and dedicated to portable data, voice and telematics applications needing the self controlled operations:

- **Fast Worldwide GPRS Telemetry and Telecontrol (SCADA applications)**
- **Worldwide Smart GPRS Security systems**
- **Worldwide Smart GPRS Vending machines**
- **Worldwide Smart Automotive and Fleet Management applications**

The Telit GM862-PCS module is the improved GPRS triband evolution of the previous wireless telecommunication module Telit GM862-GPRS, therefore allowing worldwide operations. It includes all the features of the previous GPRS module such as GPRS, Voice, Circuit Switch Data transfer, Fax, Phonebook and SMS support, furthermore it has been enhanced with the third band PCS1900 support, hot removal sensing on board SIM Reader, therefore standalone FTA, and the higher upload speed of the GPRS Class 10 Packet Data transfer. Furthermore a battery charger circuitry has been embodied inside the module.

It is specifically designed and developed by Telit for OEM usage and dedicated to portable data, voice and telematics applications needing the added triband and GPRS Class 10 improved speed features and the battery charger such as:

[1] PCS1900 network is supported only by the GM862-PCS and GM862-PYTHON modules.

- **Fast Worldwide GPRS Telemetry and Telecontrol (SCADA applications)**
- **Worldwide Smart GPRS Security systems**
- **Worldwide Smart GPRS Vending machines**
- **Fast Worldwide GPRS POS terminals**
- **Worldwide PDAs**
- **Worldwide Phones and Payphones**
- **Worldwide Smart Automotive and Fleet Management applications**
- **Battery powered applications needing a battery charger**

The Telit GM862-GPRS module is the improved GPRS evolution of the previous wireless telecommunication module Telit GM862. It includes all the features of the previous GSM module such as Voice, Circuit Switch Data transfer, Fax, Phonebook and SMS support, furthermore it has been enhanced with the on board SIM Reader, therefore standalone FTA, and the speed and flexibility of the Packet Data transfer and the always connected feature of the GPRS system.

It is specifically designed and developed by Telit for OEM usage and dedicated to voice and telematics applications needing the added GPRS features such as:

- **GPRS Telemetry and Telecontrol (SCADA applications)**
- **Smart GPRS Security systems**
- **Smart GPRS Vending machines**
- **GPRS POS terminals**
- **PDAs**
- **Phones and Payphones**
- **Smart Automotive and Fleet Management applications**

The Telit GM862-GSM module is the evolution of the previous wireless telecommunication module Telit GM862. It includes all the features of the previous module such as Voice, Circuit Switch Data transfer, Fax, Phonebook and SMS support; furthermore it has been improved with the on board SIM Reader and standalone FTA.

It is specifically designed and developed by Telit for OEM usage and dedicated to cost effective voice and telematics applications where the Packed Data features of the GPRS are not a constrain such as:

- **Telemetry and Telecontrol (SCADA applications)**
- **Security systems**
- **Cost Effective Vending machines**
- **Low cost POS terminals**
- **Phones and Payphones**
- **Automotive and Fleet Management applications**
- **Domestic Appliances with simple remote control**

The Telit GM862-PYTHON, GM862-PCS, GM862-GPRS, GM862-GSM modules will evolve in future in order to face the competitive telecommunication market requirements supporting the new incoming standards:

- ***Bluetooth support***
- ***GPS embedded functionality***

Furthermore the Telit GM862-PYTHON, GM862-PCS, GM862-GPRS, GM862-GSM modules have the full conformity assessment against R&TTE Directive and they don't require any further certification test effort and expense.

Telit GM862 Module Pin-to-Pin Upgrade Policy will enable you to include in your application the new Telit GM862-PYTHON, GM862-PCS, GM862-GPRS, GM862-GSM modules and future versions in place of the Telit GM862, allowing you to save your investments and to successfully penetrate new markets. [2]

In order to meet the competitive OEM and vertical market stringent requirements, Telit supports its customers with a dedicated GM862 Module Support Policy with:

- an Evaluation Kit to help you develop you application;
- a dedicated Website (www.GM862.com) with all updated information available;
- an Application Module for your Telit GM862-PYTHON, GM862-PCS, GM862-GPRS, GM862-GSM: for compatibility with the previous product Telit GM862-S1 the Telit GM862 module can be replaced in the Application Module with one of the new Telit GM862-PYTHON GM862-PCS, GM862-GPRS, GM862-GSM and therefore bring the new PCS, GPRS, GSM features to your product, without any further change.
- a Full Quality Certification Laboratory, to support you in case your application requires additional certification effort to comply with the European or International laws (see par.7, Conformity Assessment Issues
- a high level specialist technical support (see par.8) to assist you in your development;

For more updated information concerning product Roadmap and availability, technical characteristics, commercial and other issues please check the Telit GM862 modules dedicated Website www.GM862.com or mail to:

ts-gm862@telit.net , for any **Technical** information or support you may need

ci-gm862@telit.net , for any **Commercial**, sales or marketing related info

af-gm862@telit.net , for **After Sales** issues

[2] The GM862 Pin-to-Pin Upgrade Policy applies to the hardware and software that the product features support.

For Example by substituting the GM862-GPRS with the GM862-GSM all the hardware features remain unaffected but only the GSM software functions will be compatible and supported, the GPRS will be lost.

NOTE: All the characteristics referred to the Telit GM862-GPRS module where not expressly stated apply also to the other products Telit GM862-PYTHON, GM862-PCS, Telit GM862-GSM.

NOTE: Some of the performances of the Telit GM862 family modules depend on SW version installed on the module itself. In particular on Chapter 6, dedicated to the supported AT commands, the minimum SW version required to support each specific AT command is reported.

The Telit GM862 family modules SW group is continuously working in order to add new features and improve the overall performances. From time to time that a new SW version is released, it will be freely distributed by the Telit GM862-GPRS module technical support.

The Telit GM862 family modules are easily upgradeable by the developer using the Telit GM862 family module Transfer Flash Image program.

For more info about this topic (and not only !) please check Telit GM862 family modules dedicated Website www.GM862.com. and www.telit.net

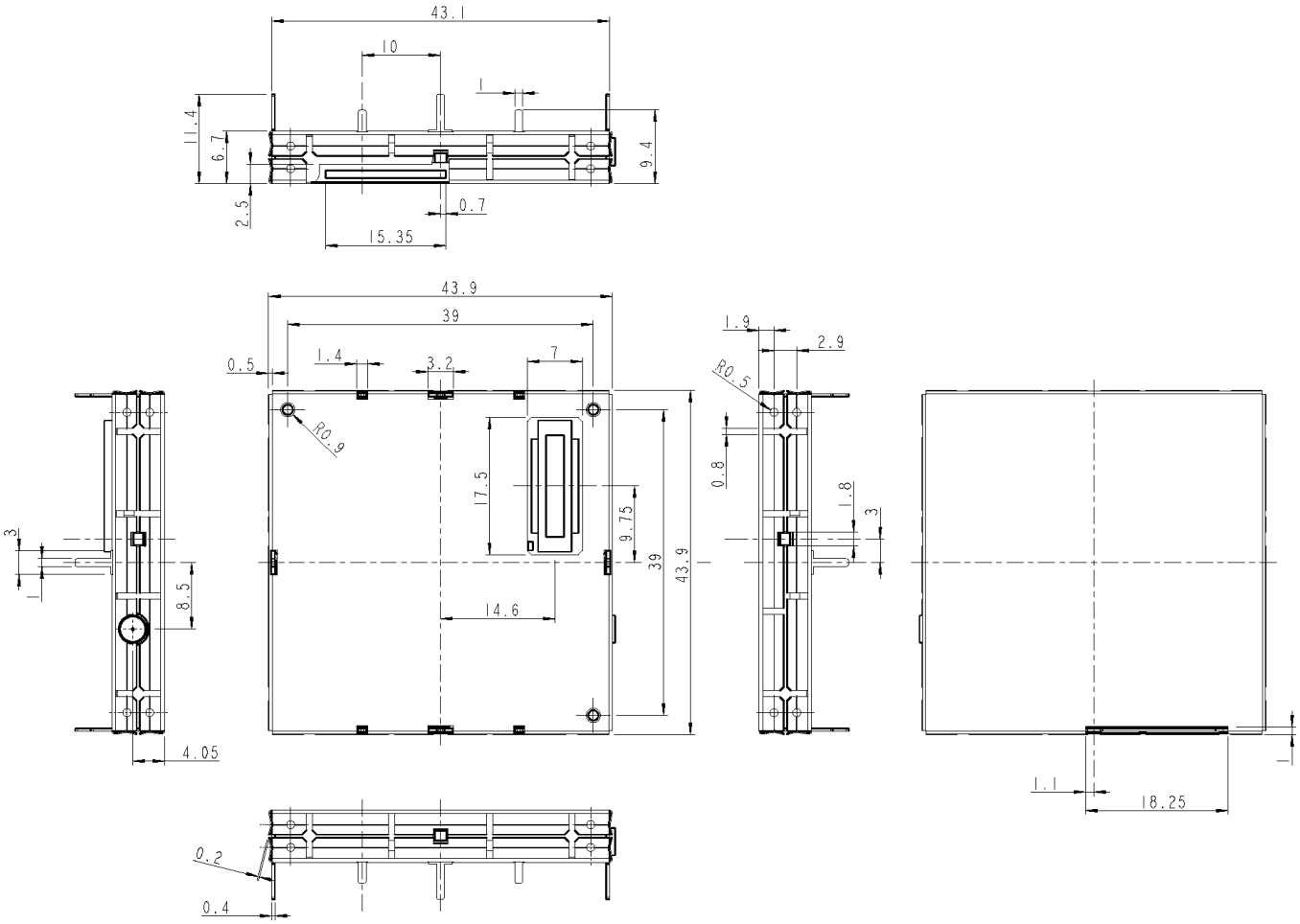
2 General Product Description

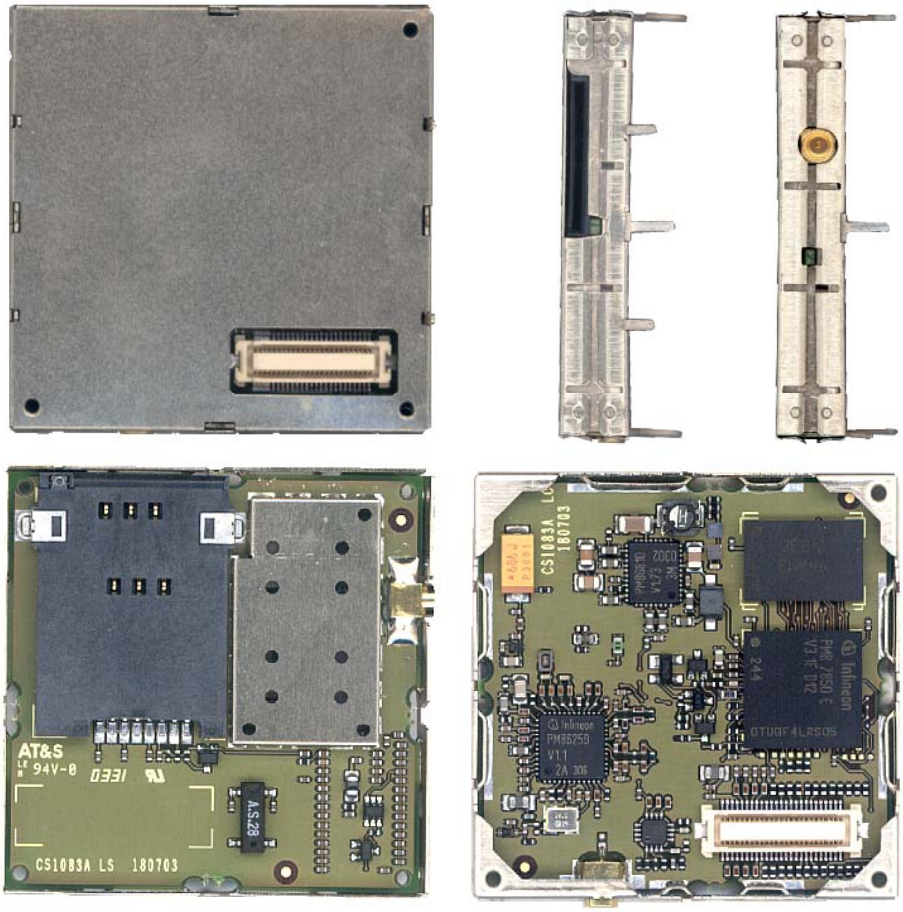
2.1 Dimensions

The Telit GM862 Family modules overall dimension are:

- **Lenght:** **43.9 mm**
- **Width:** **43.9 mm**
- **Thickness:** **6.9 mm**
- **Volume:** **≅ 13 cm^3**

The layout of Telit GM862 Family modules is shown in the following figure:





2.2 Weight

The Telit GM862 Family modules weight is **23 gr** with shield and **16 gr** without shield**.**

2.3 Environmental requirements

The Telit GM862 Family modules is compliant with the applicable ETSI reference documentation GSM 05.05 Release1999 ETSI EN300910 V8.4.1

2.3.1 Temperature range

- Temperature in normal functional conditions –10°C ÷ +55°C
- Temperature in extreme functional conditions* –20°C ÷ +70°C
- Temperature in storage conditions –30°C ÷ +85°C

*these temperature can affect the sensitivity and performance of the module

2.3.2Vibration Test (non functional)

- 10 ÷12Hz ASD = 1.92m 2 /s 3
- 12 ÷ 150Hz –3dB/oct

2.4 Operating Frequency

The operating frequencies in GSM, DCS, PCS modes are conform to the GSM specifications.

Mode	Freq. TX (MHz)	Freq. RX (MHz)	Channels (ARFC)	TX - RX offset
E-GSM-900	890.0 - 914.8	935.0 - 959.8	0 – 124	45 MHz
	880.2 - 889.8	925.2 - 934.8	975 - 1023	
DCS-1800	1710.2 - 1784.8	1805.2 - 1879.8	512 – 885	95 MHz
PCS-1900	1850.2 - 1909.8	1930.2 - 1989.8	512 - 810	80 MHz

NOTE: PCS 1900 is supported only in the Telit GM862- PCS/Python module

2.5 Transmitter output power

GSM–900

The GM862-GPRS transceiver module in GSM–900 operating mode are of **class 4** in accordance with the specification which determine the nominal 2W peak RF power (+33dBm) on 50 Ohm.

DCS–1800

The GM862-GPRS transceiver module in DCS–1800 operating mode are of **class 1** in accordance with the specifications which determine the nominal 1W peak RF power (+30dBm) on 50 Ohm.

PCS–1900

The GM862-PCS transceiver module in PCS–1900 operating mode are of **class 1** in accordance with the specifications which determine the nominal 1W peak RF power (+30dBm) on 50 Ohm.

2.6 Reference sensitivity

GSM–900

The sensitivity of the GM862-GPRS transceiver module according to the specifications for the class 4 GSM–900 portable terminals is better than **–102dBm** in all the operational conditions.

DCS–1800

The sensitivity of the GM862-GPRS transceiver module according to the specifications for the class 1 portable terminals GSM 1800 is better than **–102dBm** in normal operating conditions.

PCS–1900

The sensitivity of the GM862-PCS transceiver module according to the specifications for the class 1 portable terminals PCS 1900 is better than **–102dBm** in normal operating conditions.

2.7 Antenna

The antenna that the customer chooses to use, depending on his application, should fulfil the following requirements:

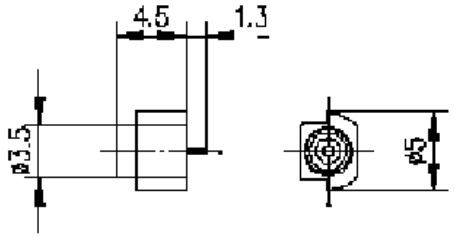
Frequency range	Standard Dual Band frequency range, Standard Tri Band frequency range for -PCS
Bandwith	80 MHz in GSM & 170 MHz in DCS & 140 MHz PCS band
Gain	> 0 dB (referenced to l/2 dipole)
Impedance	50 ohm
Input power	> 2 W peak power
VSWR absolute max	<= 10:1
VSWR recommended	<= 2:1

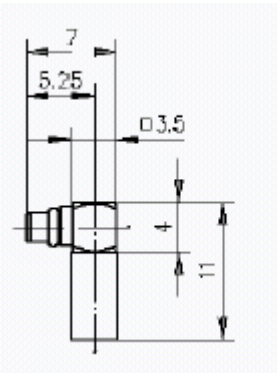
NOTE: PCS 1900 frequency coverage is required only for the Telit GM862- PCS/Python module

2.7.1 Antenna connector

The Telit GM862 Family modules includes a 50 Ohm MMCX coaxial female 2 PIN Angle Coax SMD J01341A0081 connector to allow the antenna connection.

On the user application side the following connector must be used:





a) Telegärtner MMCX angle plug crimp - Order n. J01340A0121

connector is locatable at:

Telegärtner - K. Gärtner GmbH
Lerchenstrasse 35, D-71144 Steinenbronn
Phone: (+49) 71 57 125 100

NOTE: be very careful when connecting the Telit GM862 Family modules RF connector. The RF connector can be damaged if not connected with the proper antenna RF connector. The minimum number of insertion cycles are recommended.

2.8 Supply voltage

The external power supply must be connected to VBATT signal (see paragraph 2.15,Interface connectors on GM862-GPRS) and must fulfil the following requirements:

- **N**ominal operating voltage 3.8 V
- Operating voltage range 3.4 V - 4.2 V

NOTE: Operating voltage range must never be exceeded, care must be taken in order to fulfill min/max voltage requirements and peak current supply.

2.9 Power consumption

The typical current consumption of the Telit GM862 Family modules are:

- Power off current (typical) 26 µA;
- Stand–by current < 17 mA_{rms} (< 4 mA_{rms} using command AT+CFUN)
- Operating current 250 mA_{rms} @ typical network conditions
- Operating current < 350 mA_{rms} 1.9 A_{peak} @ worst network conditions
- Operating current in GPRS class 10 < 700 mA_{rms} @ worst network conditions (only for the Telit GM862- PCS/Python module)

2.10 Embodied Battery charger

This feature is available only on the Telit GM862-PCS/Python.

The battery charger is suited for 3.7V Li-Ion rechargeable battery (suggested capacity 500-1000mAH). The Charger needs only a CURRENT LIMITED power source input and charges the battery directly through VBATT connector pins.

- Battery charger input pin CHARGE
- Battery pins VBATT, GND
- Battery charger input voltage min 5.0 V
- Battery charger input voltage typ 5.5 V
- Battery charger input voltage max 7.0 V
- Battery charger input current max 400mA
- Battery type Rechargeable Li-Ion

NOTE: If embodied battery charger is used, then a LOW ESR capacitor of at least 100µF must be mounted in parallel to VBATT GM862-PCS pins.

NOTE: when power is supplied to the CHARGE pin, a battery must always be connected to the VBATT pins of the GM862-PCS.

2.11 User Interface

The user interface is managed by AT commands specified on the GSM 07.07 and 07.05 specification and listed in the chapter 6, AT Command.

2.11.1 Speech Coding

The Telit GM862-PCS/Python, Telit GM862-GPRS and Telit GM862-GSM modules vocoder supports the following rates:

- Half Rate.
- Full rate,
- Enhanced Full Rate

2.11.2 Sim Reader

The Telit GM862 Family modules support phase 2 GSM11.14 - SIM 3 volts ONLY. For 5V SIM operation an external level translator can be added.

Furthermore the Telit GM862 Family modules have an internal built-in SIM card reader, so there's no need for an external SIM housing.

The Telit GM862-PCS/Python has an enhanced internal built-in SIM card reader that allows also hot removal of the SIM sensing, therefore the SIM can be extracted and reinserted while the module is still on, so there's no need for an external SIM housing.

NOTE: On the Telit GM862-PCS/Python the hot removal of the SIM sensing is not supported during power saving mode (+CFUN: 5).

2.11.3 SMS

The Telit GM862 Family modules supports the following SMS types:

Mobile Terminated (MT) class 0 – 2 with signalling of new incoming SMS, SIM full, SMS read

Mobile Originated class 0 – 3 with writing, memorise in SIM and sending

Cell Broadcast compatible with CB DRX with signalling of new incoming SMS.

2.11.4 Real Time Clock and Alarm

The Telit GM862 Family modules supports the Real Time Clock and Alarm functions through AT commands, furthermore an alarm output pin (GPIO6) can be configured to indicate the alarm with a hardware line output.

2.11.5 Data/fax transmission

The Telit GM862 Family modules supports:

- Packed Data transfer GPRS Class B, Multislot Class 8. (only for the Telit GM862-GPRS module)
- Packed Data transfer GPRS Class B, Multislot Class 10. (only for the Telit GM862-PCS/Python module)
- Data transmission according to the GSM 07.07, 07.05
- CSD up to 14.4 Kbps
- Fax service, Class 1 Group 3
- Fax service, Class 2 Group 3 (future SW release)

2.11.6 Local security management

With lock of Subscriber Identity module (SIM), and security code request at power–up.

2.11.7 Call control

Call cost control function.

2.11.8 Phonebook

Function available to store the telephone numbers in SIM memory.

Capability depends on SIM version/memory

2.11.9 Characters management

Availability of lowercase, uppercase and IRA characters. (international reference alphabet)

In SMS PDU mode all character set are supported.

2.11.10 SIM related functions

Activation/deactivation of the numbers stored in phone book FDN, ADN and PINs. Extension at the PIN2 for the PUK2 insertion capability for lock condition.

2.11.11 Call status indication

By AT commands.

2.11.12 Indication of network service availability

By AT commands and LED indication on dedicated output.

The STAT_LED is an Open Collector output where it is possible to directly connect a LED to show information on the network service availability and Call status.

STAT_LED indications

LED status	Device Status
permanently off	device off
fast blinking (period 1s, Ton 0,5s)	Net search / Not registered / turning off

	off
slow blinking (period 3s, Ton 0,3s)	Registered full service
permanently on	a call is active

2.11.13 Automatic answer (Voice, Data or FAX)

After n (depends of settings) rings automatically answers with beep (see S0 param).

2.11.14 Supplementary services (SS)

- Call Barring,
- Call Forwarding,
- Calling Line Identification Presentation (CLIP),
- Calling Line Identification Restriction (CLIR),
- Call Waiting, other party call Waiting Indication,
- Call Hold, other party Hold / Retrieved Indication,
- Closed User Group supplementary service (CUG),
- Advice of Charge,
- Unstructured SS Mobile Originated (MO)

2.11.15 Acoustic signaling

The acoustic signalling of Telit GM862 Family modules on the selected acoustic device are the following:

- Call waiting;
- Ringing tone;
- SMS received tone;
- Busy tone;
- Power on/off tone;
- Off Hook dial tone;
- Congestion tone;
- Connected tone;
- Call dropped;
- No service tone;
- Alarm tone.

2.11.16 DTMF tones

DTMF tones managed by specific AT commands.

These tones are generated with AT commands only during voice calls.

The minimum duration of a DTMF tone is 100 ms.

	Group high		
Group low	1209 Hz	1336 Hz	1477 Hz
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

2.11.17 Buzzer output

The General Purpose I/O pin GPIO7 can be configured to output the BUZZER output signal, with only an external Mosfet/transistor and a diode a Buzzer can be directly driven.

The ringing tone and the other signaling tones can be redirected to this Buzzer output with a specific AT command.

2.12 EMC

Compliant to EN301-489-1 and EN301-489-7 and all applicable GSM Specifications.

Compliant to Directive 1999/05/CE.

2.13 Logic level specifications

Where not specifically stated, all the interface circuits work at 2.8V CMOS logic levels.

The following table shows the logic level specifications used in the Telit GM862 Family modules interface circuits:

Absolute Maximum Ratings -Not Functional

Parameter	**Min**	**Max**
Input level on any digital pin when on	-0.3V	+3.75V
Input voltage on analog pins when on	-0.3V	+3.0 V

Operating Range - Interface levels (2.8V CMOS)

Level	Min	Max
Input high level	2.1V	3.3V
Input low level	0V	0.5V
Output high level	2.2V	3.0V
Output low level	0V	0.35V

For 2,0V signals:

Operating Range - Interface levels (2.0V CMOS)

Level	Min	Max
Input high level	1.6V	3.3V
Input low level	0V	0.4V
Output high level	1,65V	2.2V
Output low level	0V	0.35V

2.13.1 Reset signal

Signal	Function	I/O	Pin
RESET	Phone reset	I	23 (connector SO301)

RESET is used to reset the Telit GM862 Family modules. Whenever this signal is pulled low, the GM862-GPRS is reset. When the device is reset it stops any operation and after the release of the reset it shuts down, without doing any detach operation from the network where it is registered to. This behavior is not a proper shut down because any GSM device is requested to issue a detach request on turn off. For this reason the Reset signal must not be used to normally shutting down the device, but only as an emergency exit in the rare case the device remains stucked waiting for some network response.

The RESET is internally controlled on start-up to achieve always a proper power-on reset sequence, so there's no need to control this pin on start-up. It may only be used to reset a device already on that is not responding to any command.

NOTE: do not use this signal to power off the Telit GM862 modules. Use the ON/OFF signal (Pin 17 of SO301) to perform this function or the AT#SHDN command.

Reset Signal Operating levels:

Signal	Min	Max
RESET Input high	2.2V*	3.3V
RESET Input low	0V	0.2V

* this signal is internally pulled up so the pin can be left floating if not used.

If unused, this signal may be left unconnected. If used, then it **must always be connected with an open collector transistor**, to permit to the internal circuitry the power on reset and undervoltage lockout functions.

2.14 Audio levels specifications

The audio of the Telit GM862 Family modules is organised into two main paths:

- internal path (called also MT)
- external path (called also HF)

These two paths are meant respectively for handset and headset/handsfree use.

The Telit GM862 Family modules has a built in echo canceller and a noise suppressor, tuned separately for the two audio paths; for the internal path the echo canceller parameters are suited to cancel the echo generated by a handset, while for the external audio path they are suited for a handsfree use.

For more information on the audio refer to the Hardware User Guide.

The following table reports all the audio level specifications.

Microphone characteristics

	Internal audio mic. input	**External audio mic. input**
Line coupling	AC (100nF cond.)	AC (100nF cond.)
Line type	Balanced	Balanced
Differential input resistance	25kΩ	25kΩ
Line nominal sensitivity	$50mV_{rms}$	$3mV_{rms}$
Max input voltage	$360mV_{rms}$	$22mV_{rms}$
Microphone nominal sensitivity - Analog Gain suggested	$-45dB_{Vrms/Pa}$ / +24dB	$-45dB_{Vrms/Pa}$ / +10dB
Echo canceller type	handset	Car kit hands free

Speaker characteristics

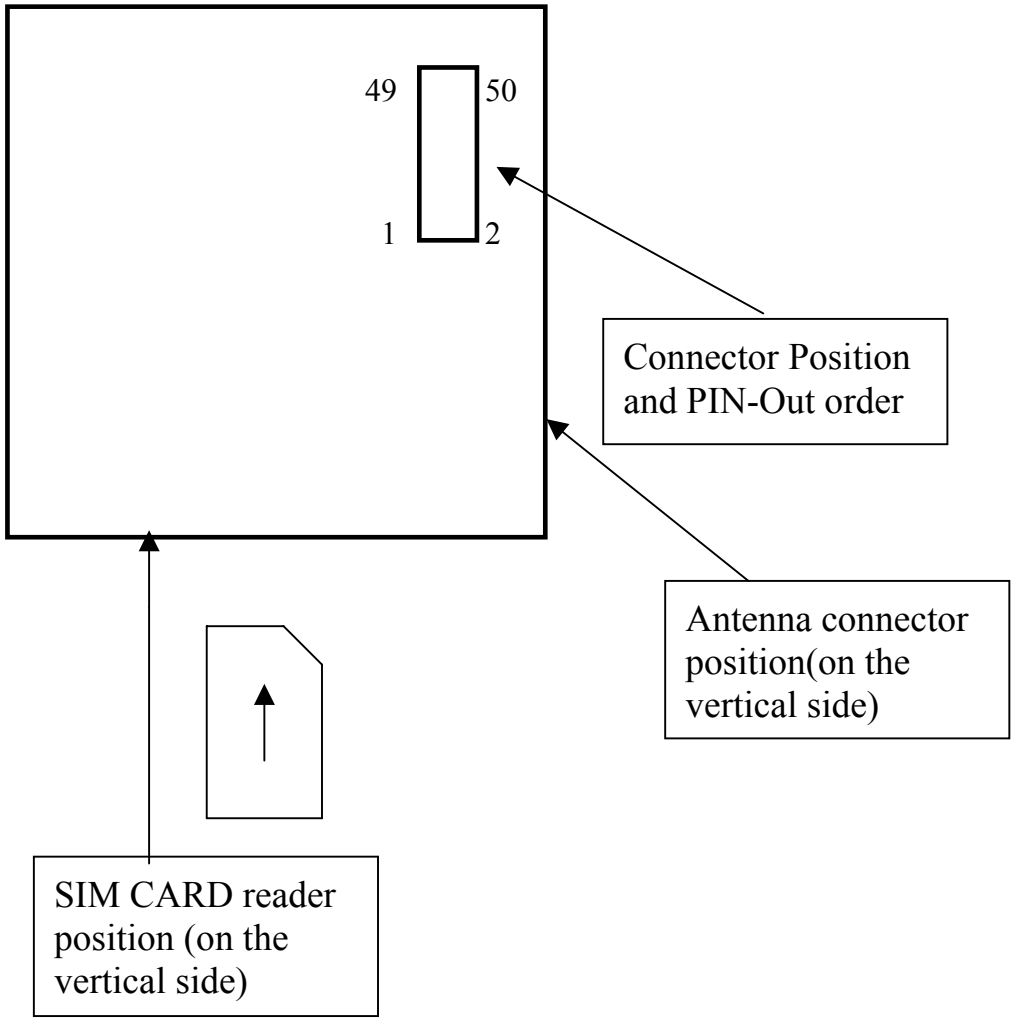
	Internal audio ear. output	**External audio ear. output**
Line coupling	DC	DC
Line type	Bridged	Bridged
Speaker impedance	≥ 16Ω ± 5% @ 1kHz	≥ 16Ω ± 5% @ 1kHz
Minimun load impedance	15Ω	15Ω
Signal bandwidth	150-8000 Hz @ -3dB	150-8000 Hz @ -3dB
Maximum output	$1700mV_{rms}$	$850mV_{rms}$
Maximum power output	30mW	7.5mW
Volume level steps (SW)	-2dB	-2dB
Number of volume steps (SW)	10	10

2.15 Interface connectors on GM862 modules

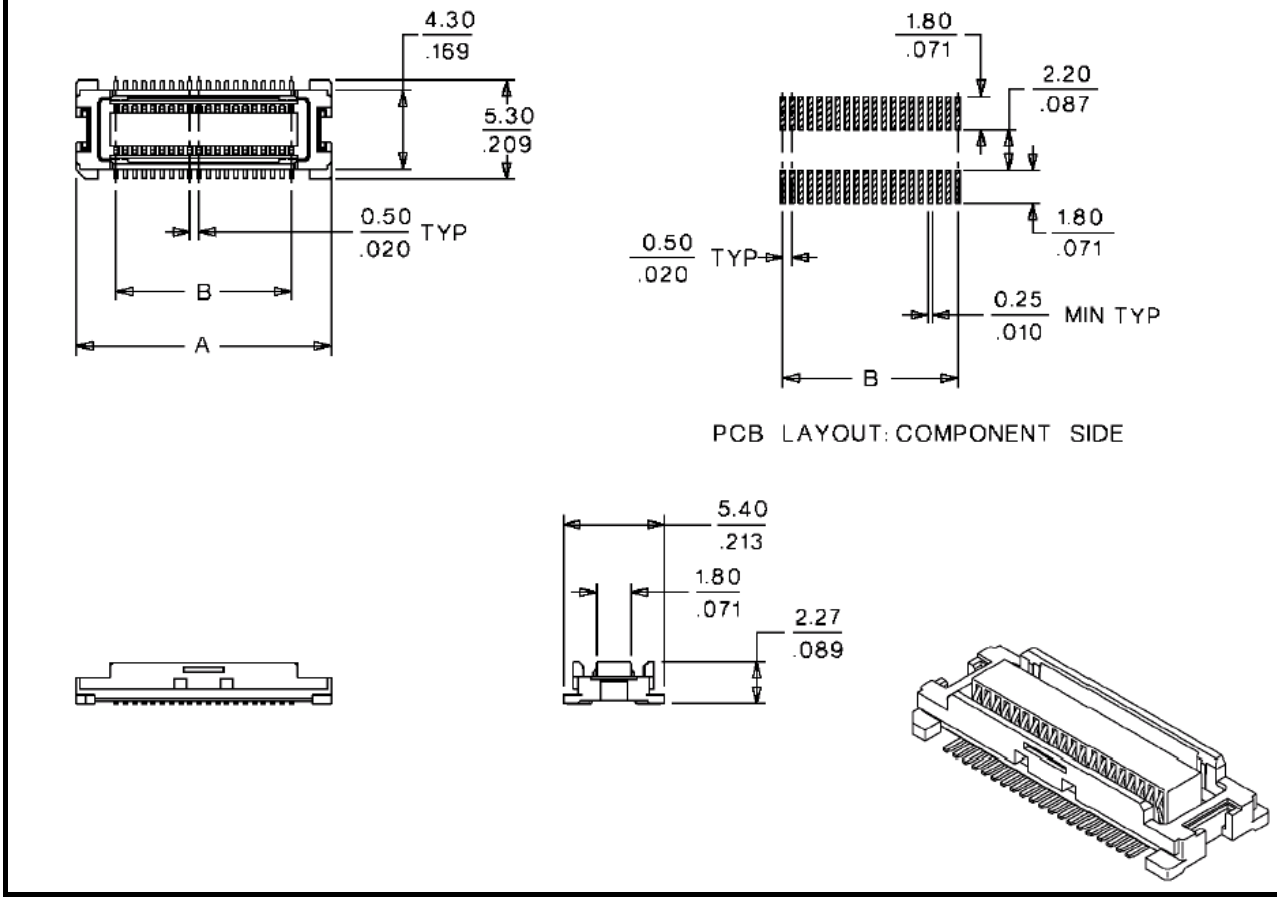
The Telit GM862 Family modules has two interface connectors and a SIM Card reader:

1) SO601: Antenna connector (see 2.7.1, Antenna connector)
2) SO301: Interface connector
3) SO402: SIM Card Reader

The GM862-GPRS transceiver module interface connector SO301 is a CSTP 50 pin vertical SMD Molex 52991–0508 (male). Its pin-out (down view) is:



Molex 52991-0508 (male) GM862 Connector LAY-OUT



Molex 52991-0508 (male) GM862 Connector PIN-OUT					
Pin	**Signal**	**I/O**	**Function**	**Internal Pull up**	**Type**
1	VBATT	-	Main power supply		Power
2	GND	-	Ground		Power
3	VBATT	-	Main power supply		Power
4	GND	-	Ground		Power
5	VBATT	-	Main power supply		Power
6	GND	-	Ground		Power
7	VBATT	-	Main power supply		Power
8	CHARGE	AI	Charger input (5)		Power
9	EAR_HF+	AO	Handsfree ear output, phase +		Audio
10	EAR_MT-	AO	Handset earphone signal output, phase -		Audio
11	EAR_HF-	AO	Handsfree ear output, phase -		Audio
12	EAR_MT+	AO	Handset earphone signal output, phase +		Audio
13	MIC_HF-	AI	Handsfree microphone input; phase -, nominal level 3mVrms		Audio
14	MIC_MT+	AI	Handset microphone signal input; phase+, nominal level 50mVrms		Audio
15	MIC_HF+	AI	Handsfree microphone input; phase +, nominal level 3mVrms		Audio
16	MIC_MT-	AI	Handset microphone signal input; phase-, nominal level 50mVrms		Audio
17	ON/OFF	I	Input command for switching power ON or OFF (toggle command). The pulse to be sent to the GM862 must be equal or greater than 1 second.	47KΩ	VBATT (6)
18	AXE	I	Handsfree switching	47KΩ	2.0V signal
19	SIMIO	I/O	External SIM signal - Data I/O		3V ONLY
20	C103/TXD	I	Serial data input (TXD) from DTE		CMOS 2.8V
21	PWRCTL/ CAM_SYNC	O	Module Status ON indication (Signal output for power on/off control of external devices) / Camera Interface (4)	820Ω - 10KΩ (7)	CMOS 2.8V
22	SIMVCC	-	External SIM signal – Power (3)		3V ONLY
23	RESET	I	Reset input	2KΩ	see par.2.13.1
24	SIMRST	O	External SIM signal – Reset		3V ONLY

25	TEST32kHz/ MON1_CAM		For test purpose (1) / Camera Interface (4)		CMOS 2.8V
26	SIMCLK	O	External SIM signal – Clock		3V ONLY
27	CCIN	I/O	External SIM signal - Presence (active low)	47KΩ	CMOS 2.8V
28	GPIO2/JDR	O	General purpose output (Open Collector) / Camera Interface (4)		Open Collector
29	C106/CTS	O	Output for Clear to send signal (CTS) to DTE		CMOS 2.8V
30	C125/RING	O	Output for Ring indicator signal (RI) to DTE		CMOS 2.8V
31	GPIO1	I	General purpose input		transistor base
32	CLKSXM / PD[0]		DAI - For test purpose (1) / Camera Interface (4)		CMOS 2.8V
33	C107/DSR	O	Output for Data set ready signal (DSR) to DTE		CMOS 2.8V
34	TXDD / PD[1]		DAI - For test purpose (1) / Camera Interface (4)		CMOS 2.8V
35	EMMI TX	O	TX Data output for debug monitor (1)		CMOS 2.8V
36	C109/DCD	O	Output for Data carrier detect signal (DCD) to DTE		CMOS 2.8V
37	C104/RXD	O	Serial data output to DTE		CMOS 2.8V
38	SCLK / PD[2]		DAI - For test purpose (1) / Camera Interface (4)		CMOS 2.8V
39	STAT_LED	O	Status indicator led		Open Collector
40	RXDD / PD[3]		DAI - For test purpose (1) / Camera Interface (4)		CMOS 2.8V
41	EMMI RX	I	RX Data input for debug monitor (1)		CMOS 2.8V
42	RFSD / PD[4]		DAI - For test purpose (1) / Camera Interface (4)		CMOS 2.8V
43	C108/DTR	I	Input for Data terminal ready signal (DTR) from DTE (4)		CMOS 2.8V
44	TFSD / PD[5]		DAI - For test purpose (1) / Camera Interface (4)		CMOS 2.8V
45	C105/RTS	I	Input for Request to send signal (RTS) from DTE		CMOS 2.8V
46	GPIO3 / PD[6]	I/O	Configurable general purpose I/O pin / Camera Interface (4)		CMOS 2.8V
47	GPIO4 / IICSDA	I/O	Configurable general purpose I/O pin / Camera Interface (4)		CMOS 2.8V

48	GPIO5 / CAM_DRDY	I/O	Configurable general purpose I/O pin / Camera Interface (4)		CMOS 2.8V
49	GPIO6 /ALARM / PD[7]	I/O	GP I/O pin or ALARM output pin / Camera Interface (4)		CMOS 2.8V
50	GPIO7 /BUZZER /CAM_PWR_ON	I/O	GP I/O pin or BUZZER output pin / Camera Interface (4)		CMOS 2.8V

(1) For the exclusive use of the Technical Support Service

(2) An earphone with a 150 ohm impedance can be directly connected to EAR+ and EAR–

(3) On this pin a maximum of 47nF bypass capacitor is allowed.

(4) When activating the Easy camera these pins will not be available for other use (Camera interface available only on GM862-PCS/PYTHON)

(5) charger is available only on GM862-PCS/PYTHON. For the other products the pin is GND

(6)This pin is pulled up internally to VBATT and has a transistor base input.

(7) pull up is 820Ω on GM862-PCS/PYTHON and 10KΩ on GM862-GPRS/GM862-GSM.

2.16Mounting the GM862 modules on your Board

In order to electrically connect your board to the Telit GM862 Family modules, use a **CSTP 2x25 pin vertical SMD SCH–SCH Molex 53748 - 0504 (female, low profile)** as a counterpart to the **CSTP 50 pin vertical SMD Molex 52991–0508 (male)** of your Telit GM862 Family modules.

When mounting the Telit GM862 Family modules on your board, take care of soldering the GM862-GPRS shielding reeds on a ground plane or signal.

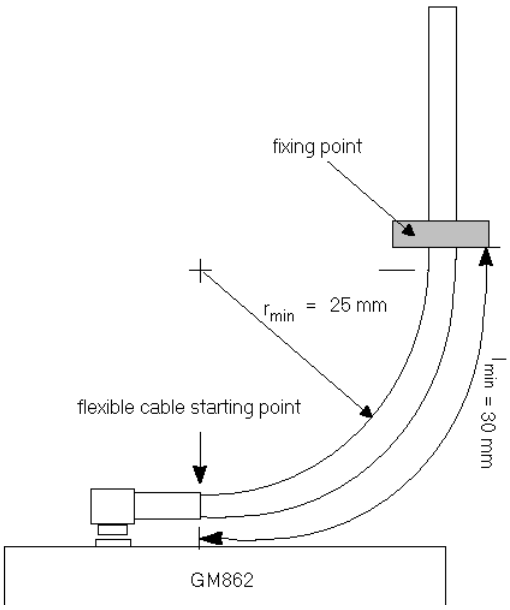
NOTE: be very careful when connecting the Telit GM862 module RF connector. The Telit GM862 module RF connector can be damaged if not connected with the proper antenna RF connector. The minimum number of insertion cycles is recommended.

2.16.1 Antenna Coaxial cable fixing

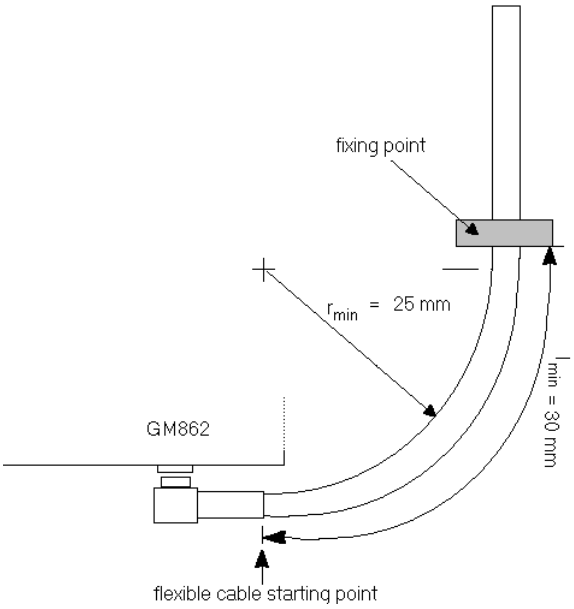
The following constraints must be respected in the Telit GM862 Family modules antenna cable connection:

- The fixing point of the coaxial cable must not be placed too close to the antenna connector of the GM862 module, leaving at least 30mm of flexible cable between the fixed point and the plug end.
- The fixing point must be at the same height of the GM862 module antenna connector, eventually using a wedge between the PCB and the cable if it is directly fixed to the PCB.
- The flexible cable must never be bent with a radius lower than 25mm (RG174 cable).
- The cable must be a RG174 type or more flexible ones.

The following pictures explain these constraints:



- Angle connector fixing example -



- Angle connector fixing example 2 -

NOTE: in the examples the cable is always bent, this is not a constrain. If the installation does not require it, then the cable can be kept straight, ensuring that the fixing is without sliding.

2.16.2 Precautions

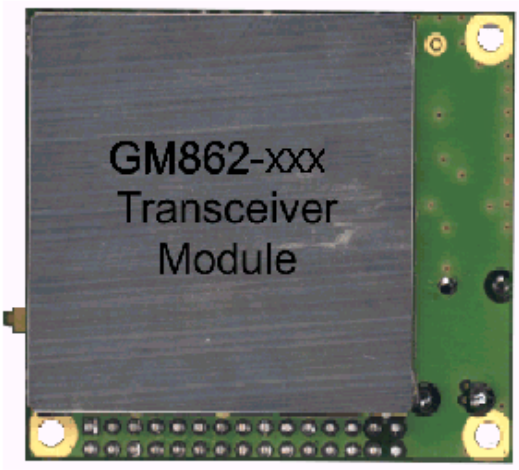
- The plug should be inserted in the connector only after the installation of the Telit GM862 Family modules in the board. This is to prevent accidental breaking of the antenna connection during the transport.
- The coaxial cable must be fixed at least in one point, without sliding possibilities.
- The plug insertion/removal must be done axially with the female connector of the GM862-GPRS, keeping lateral strains to a minimum.
- The insertion/crimping pull out force must be less than 15N.

3 Application Board: GM862-S1

In order to use the Telit GM862 Family modules as a stand-alone product, it is not anymore required any Application Board since the new Telit GM862 Family modules has the full ETSI Type Approval as a stand-alone product.

The Telit GM862 Family modules are fully backward compatible and therefore the upgrade of old projects that were designed using the Telit S1 Board to interface with the Telit GM862 module can be made by simply replacing the Telit GM862 module with a new Telit GM862 Family module in the Telit S1 Board.

Telit GM862-S1



3.1 Application Board Description

For a detailed description of the hardware interfaces, the recommended accessories and the connections to the Telit GM862-S1 please refer to the Telit GM862 Product Description document.

4 Evaluation Kit: GM862/S1 EVK

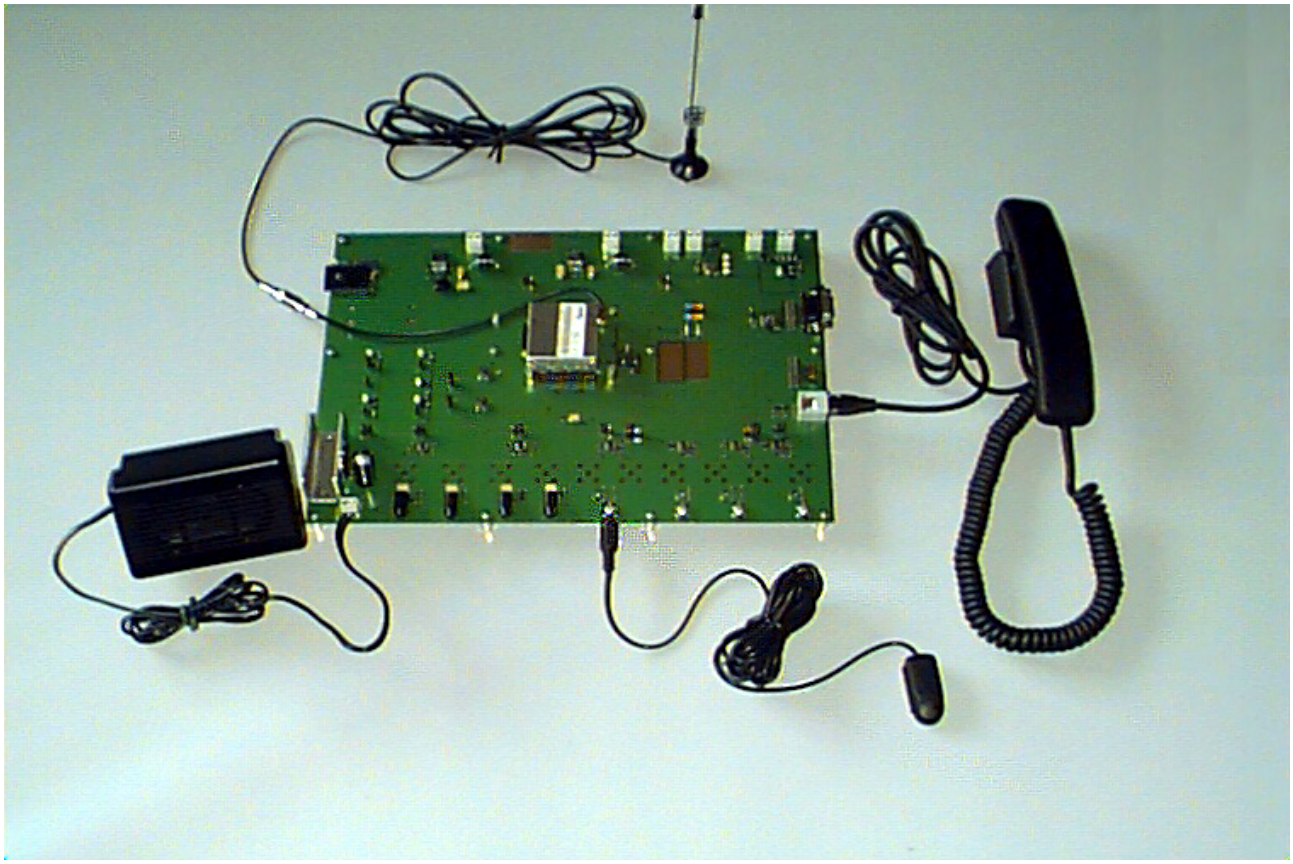
In order to assist you in the development of your Telit GM862 Family modules based application, Telit can supply an Evaluation Kit that interfaces the Telit GM862 Family modules directly and also to the previous **Application Board GM862-S1** with appropriate power supply, SIM card housing, RS 232 serial port level translator, direct UART connection, Handset, Headset and Hands-free (car kit) audio, antenna.

The EVK provides a fully functional solution for a complete data/phone application.

The standard serial RS232 9 pin connector placed on the Evaluation Kit allows the connection of the EVK system with a PC or other DTE.

The development of the applications utilizing the Telit GM862 Family modules must present a proper design of all the interfaces towards and from the module (e.g. power supply, audio paths, level translators), otherwise a decrease in the performances will be introduced or, in the worst case, a wrong design can even lead to an operating failure of the module.

In order to assist the hardware designer in his project phase, the EVK board presents a series of different solutions, which will cover the most common design requirements on the market, and which can be easily integrated in the OEM design as building blocks or can be taken as starting points to develop a specific one.



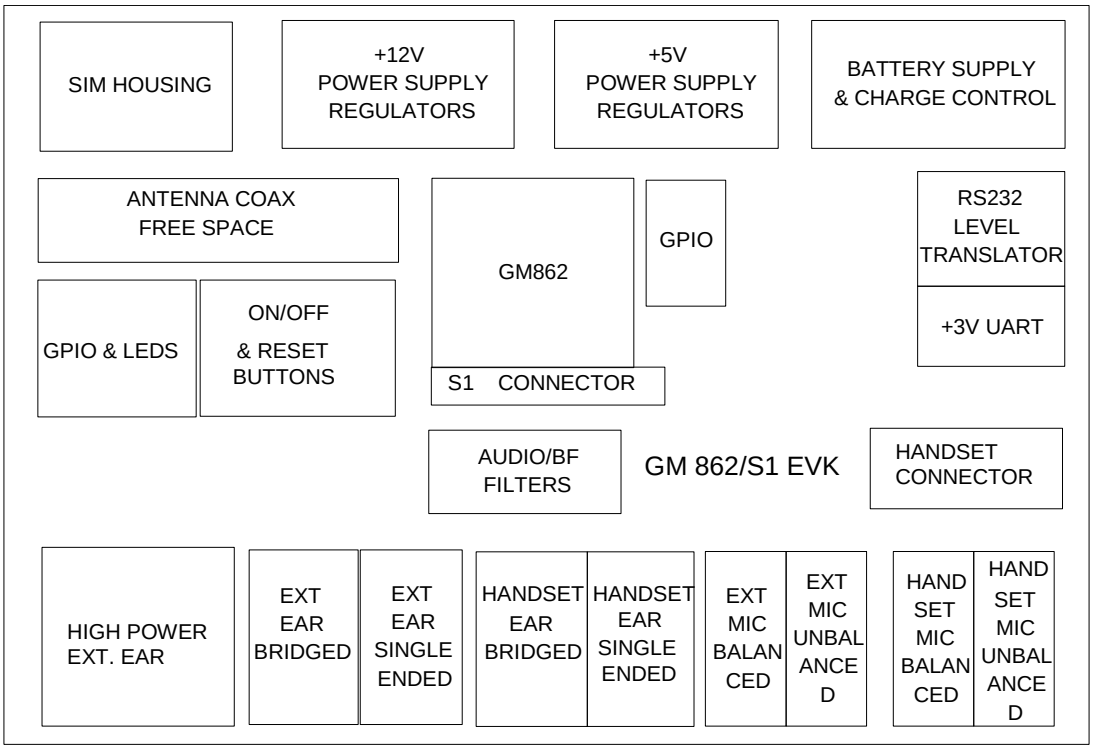
GM862-GPRS Evaluation Kit

4.1 Evaluation Kit description

For a detailed description of the Telit GM862/S1 Evaluation Kit refer to the documentation provided with the Telit GM862-GPRS Hardware User Guide.

The Telit GM862/S1 Evaluation Kit can be divided into different blocks depending on the function they implement.

A schematic showing the EVK board and its blocks is shown:



4.1.1 Power Supply

In the Board there are three different power sources, that embrace a wide range of applications, from the automotive +12V input to the stand alone battery powered device.

The power sources nominal input voltages are:

- +12V input (typically automotive)
- +5V input (typically embedded systems)
- +3.8V input (Li-Ion Battery)

Each one of these power sources can supply both the Telit GM862 Family modules and the whole circuitry embodied in the Evaluation Kit.

Only one of these power sources can be used at a time and it is selectable with two jumpers.

For the battery power supply a Li-Ion battery charger circuit is provided.

4.1.2 Serial interface

The communications between your application and the Telit GM862 Family modules must be done through a serial interface which can be a standard CMOS UART or a RS232 port.

All levels of the RS232 port are conform to RS232 and V.24 standard and a PC serial port can be directly connected to this connector.

Both these interfaces are supported.

4.1.3 Audio

The Evaluation Kit board provides two software/hardware selectable audio paths, the internal and the external one. All the handset, headset and hands free-car kit functions are supported.

Furthermore for each audio path two solution are presented: single ended and balanced.

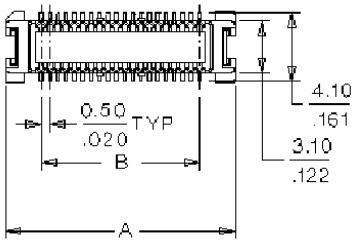
4.1.4 GPIO and Leds

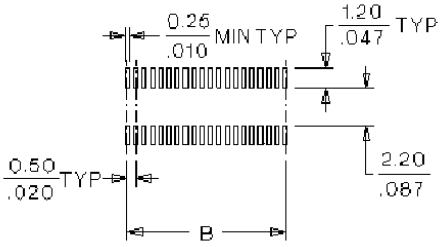
All the General Purpose Input Output ports of the Telit GM862 Family modules are supported in a dedicated connector and several leds are added as a debug aid.

4.1.4.1 Telit GM862 Family modules interface connector

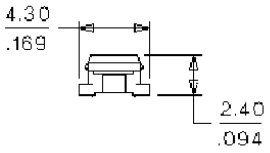
The PL401 transceiver interface connector placed on the Telit GM862/S1 Evaluation Kit is a female CSTP 2x25 pin vertical SMD SCH–SCH Molex 53748 - 0504 (low profile). See pin out on the SO301 connector on Telit GM862 Family modules.

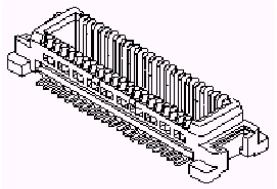
Molex 53748-0504 (female, low profile) Connector LAY-OUT





PCB LAYOUT: COMPONENT SIDE





5 Service and firmware update

The serial cable used for the communication with a PC can be also used to update the Telit GM862 Family modules firmware. The RS232 connector DB type 9 pin 90° is used to connect the Telit GM862/S1 Evaluation Kit. to any DTE .

All levels are conformed to RS232 and V.24 standard and a PC serial port can be directly connected to this connector.

The firmware update can be done with a specific software tool provided by Telit that runs on windows based PCs.

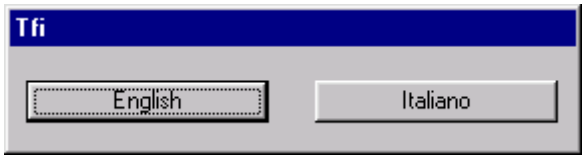
5.1.1.1 Step-by-Step upgrade procedure

I. Get the specific software for upgrading (Transfer Flash Image GM862_xxx.zip) requesting it at this e-mail address ts-gm862@telit.net

II. Unzip the file GM862_xxx.zip. A new exe file will be created.

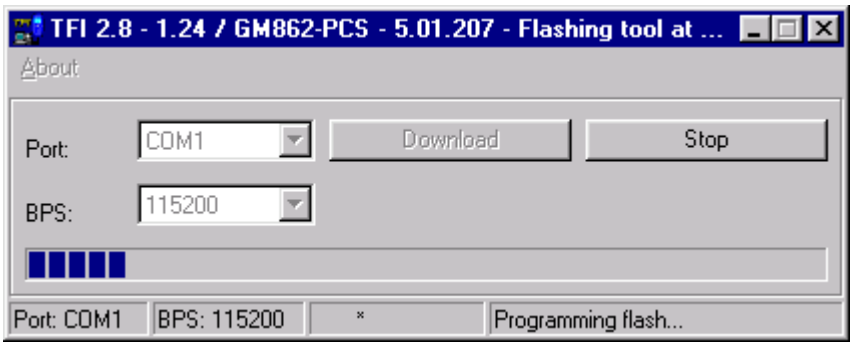
III. Mount the Telit GM862 module in the Evaluation Kit and supply power. Do not turn on the module for now.

IV. Run the file *TFI_GM862-xxx_xxxx.exe*. The following window should be displayed, Select the language preferred by pressing the correspondent button.

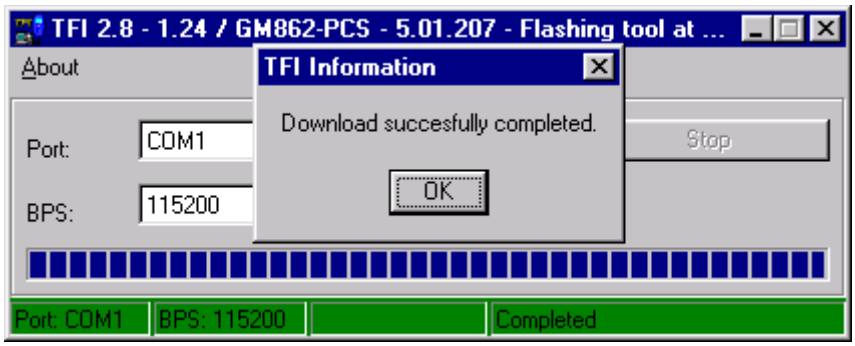


V. Press OK to the initial message.

VI. Select the right COM port and speed. Note that to go faster than 115200 you need a special hardware on the PC. Then Press the Download button and within 5 seconds press the ON/OFF button on the board (turn on the module). Keep pressed the ON/OFF button for 5 seconds and then leave it.



Wait for the end of programming green message OK



Your Telit GM862- module is now programmed with the new firmware.

6 AT Command

The Telit GM862 Family modules can be driven via the serial interface using the standard AT commands[3]. The Telit GM862 Family modules are compliant with:

1. Hayes standard AT command set, in order to maintain the compatibility with existing SW programs.
2. ETSI GSM 07.07 specific AT command and GPRS specific commands.
3. ETSI GSM 07.05 specific AT commands for SMS (Short Message Service) and CBS (Cell Broadcast Service)
4. FAX Class 1 compatible commands

Moreover the Telit GM862 Family modules supports also Telit proprietary AT commands for special purposes.

In the following the dedicated description of how to use the AT commands with the Telit GM862 Family modules.

6.1 Definitions

The following syntactical definitions apply:

<CR> Carriage return character, is the command line and result code terminator character, which value, in decimal ASCII between 0 and 255,is specified within parameter S3.

The default value is 13.

<LF> Linefeed character, is the character recognised as line feed character. Its value, in decimal ASCII between 0 and 255, is specified within parameter S4. The default value is 10.

The line feed character is output after carriage return character if verbose result codes are used (V1 option used) otherwise, if numeric format result codes are used (V0 option used) it will not appear in the result codes.

<...> Name enclosed in angle brackets is a syntactical element. They do not appear in the command line.

[...] Optional subparameter of a command or an optional part of TA information response is enclosed in square brackets. Brackets themselves do not appear in the command line. When subparameter is not given in AT commands which have a read command, new value equals to its previous value. In AT commands which do not store the values of any of their subparameters, and so have not a read command, which are called *action type* commands, action should be done on the basis of the recommended default setting of the subparameter.

[3] The AT is an ATTENTION command and is used as a prefix to other parameters in a string. The AT command combined with other parameters can be set up in the communications package or typed in manually as a command line instruction.

6.2 AT Command Syntax

GSM commands use syntax rules of *extended* commands.

Every extended command has a *test command* (trailing =?) to test the existence of the command and to give information about the type of its subparameters. There are two types of extended command:

- *Parameter type* commands which also have a *read command* (trailing ?) to check the current values of subparameters.
- *Action type* commands do not store the values of any of their possible subparameters, and therefore do not have a read command.

In general when a command is issued without any trailing character, the read command is executed as if the trailing ? character was present.

6.2.1Command lines

The basic structure of the command lines are:

- ATCMD1<CR> where AT is the command line prefix, CMD1 is a basic command (i.e. it have not + prefix) and CR is the command line terminator character
- ATCMD2=10<CR> where 10 is a subparameter
- AT+CMD1;+CMD2=, ,10<CR> These are two examples of extended commands (which have a + prefix). They are delimited with semicolon. In the second command the subparameter is omitted
- +CMD1?<CR> This is a read command for checking current subparameter values
- +CMD1=?<CR> This is a test command for checking possible subparameter values

These commands might be performed in a single command line as shown below:

ATCMD1 CMD2=10+CMD1;+CMD2=, ,10;+CMD1?;+CMD1=?<CR>

anyway it is always preferable to separate into different command lines the basic commands (i.e. without + prefix) and the extended commands (the ones with + prefix); furthermore it is suggested to avoid placing several action commands in the same command line, because if one of them fails, then an error message is received but it is not possible to argue which one of them has failed the execution.

If command V1 is enabled (verbose responses codes) and all commands in a command line has been performed successfully, result code <CR><LF>OK<CR><LF> is sent from the TA to the TE, if subparameter values of a command are not accepted by the TA or command itself is invalid, or command cannot be performed for some reason, result code <CR><LF>ERROR<CR><LF> is sent and no subsequent commands in the command line are processed.

If command V0 is enabled (numeric responses codes), and all commands in a command line has been performed successfully, result code 0<CR> is sent from the TA to the TE, if sub-parameter values of a command are not accepted by the TA or command itself is invalid, or command cannot be performed for some reason, result code 4<CR> and no subsequent commands in the command line are processed.

In case of errors depending on ME operation, ERROR (or 4) response may be replaced by `+CME ERROR: <err>` or `+CMS ERROR: <err>`.

6.2.2 Information responses and result codes

The TA response, in case of verbose response format enabled, for the previous examples command line could be as shown below:

information response to +CMD1? <CR><LF>+CMD1:2,1,10<CR><LF>

information response to +CMD1=? <CR><LF>+CMD1(0-2),(0,1),(0-15)<CR><LF>

final result code <CR><LF>OK<CR><LF>

Moreover there are other two types of result codes:

- *result code*s that inform about progress of TA operation (e.g. connection establishment `CONNECT`)
- *result code*s that indicate occurrence of an event not directly associated with issuance of a command from TE (e.g. ring indication `RING`).

Here the basic result codes according to ITU-T V25Ter recommendation

Result Codes	
Numeric form	Verbose form
0	OK
1	CONNECT
2	RING
3	NO CARRIER
4	ERROR
6	NO DIALTONE
7	BUSY
8	NO ANSWER
12	CONNECT 9600

6.2.3 Command Response Timeout

Every command issued to the Telit GM862 Family modules returns a result response if response codes are enabled (default). The time needed to process the given command and return the response varies from command to command and may depend also from the network on which the command may interact. As a result every command is provided with a proper timeout time, if this time elapses without any result from the operation, then an ERROR response can be reported as if the operation was not successful and the operation is anyway terminated.

The timeout time is quite short for commands that imply only internal set up commands, but may be very long for command that interact with the network (or even a set of Networks).

The default timeout is 100 ms for all the commands that have no interaction with the network or upper software layers.

In the table below are listed all the commands whose timeout differs from the default 100 ms and their effective timeout is reported:

Command	Timeout (Seconds)
+CBST	0.2
+CR	0.2
+CRC	0.2
+CRLP	0.2
+CSCS	0.2
+CEER	5
+CGMI	5
+CGMM	5
+CGMR	5
+CGSN	20
+CIMI	20
+CNUM	20
+CREG	5
+COPS	180
+CLCK	180
+CPWD	180
+CLIP	180
+CLIR	180
+CCFC	180
+CCWA	20
+CHLD	20
+CUSD	180
+CAOC	20
+CSSN	20
+CLCC	20
+CPAS	5
+CPIN	20
+CSQ	5
+CPBS	5
+CPBR	20
+CPBF	20

+CPBW	20
+CALM	5
+CRSL	5
+CLVL	5
+CMUT	5
+CACM	20
+CAMM	20
+CPUC	20
+CMEE	5
+VTS	20
+GMI	5
+GMM	5
+GMR	5
+GSN	20
I3	5
I4	5
I5	5
+CSMS	5
+CPMS	5
+CMGF	5
+CSCA	20
+CSMP	5
+CSDH	5
+CSAS	5
+CRES	5
+CNMI	5
+CMGS	180 / 5 for prompt">"
+CMSS	180
+CMGW	5 / 5 for prompt">"
+CMGD	5
+CMGR	5
+CMGL	5
#CAP	10
#SRS	10
#SRP	10
#STM	10
#PCT	10
#SHDN	10

#QTEMP	10
#SGPO	10
#GGPI	10
#MONI	10
#CGMI	5
#CGMM	5
#CGMR	5
#CGSN	20
#CIMI	5
+CGACT	180
+CGATT	180
+CGDATA	20
+CGDCONT	20
+CGPADDR	20
+CGQMIN	20
+CGQREQ	20

6.2.4 Command issuing timing

The chain Command -> Response shall always be respected and a new command must not be issued before the GM862-GPRS has terminated all the sending of its response result code (whatever it may be).

This applies especially to applications that "sense" the OK text and therefore may send the next command before the complete code <CR><LF>OK<CR><LF> is sent by the GM862-GPRS.

It is advisable anyway to wait for at least 20ms between the end of the reception of the response and the issue of the next AT command.

If the response codes are disabled and therefore the module does not report any response to the command, then at least the 20ms pause time shall be respected.

During command mode, due to hardware limitations, under severe CPU load the serial port can loose some characters if placed in autobauding at high speeds. Therefore if you encounter this problem fix the baud rate with AT+IPR command.

6.2.5 Factory Profile and parameters stored in the profiles

The GM862-GPRS stores several command configurations in the internal non volatile memory (see command &W , &Y, &P, &F, &F1), allowing to remember this status even after power off.

There are two user customisable profiles and one factory profile in the NVM of the device, by default the device will start with user profile 0 equal to factory profile.

For backward compatibility the profile is divided into two sections, one base section which was historically the one that was saved and restored in early releases of code, and the extended section which includes all the remaining applicable commands.

There are some other commands which are particular and are stored in NVM outside the profile, being independent from &W, &Y, &P, &F commands and saved usually without any command issuing (except from Easy-GPRS and Email management command sets) having only one saved value which is always restored at start-up.

The &W command is used to save the actual values of both sections of profiles into the NVM user profile. The profile loaded at start-up can be set with both &Y and &P commands. The &Y command instructs the device to load at start-up only the base section remaining backward compatible. The new &P command instructs the device to load the full profile base + extended sections at start-up. In the same manner the &F command resets to factory profile values only the command of the base section of profile, while the &F1 resets to factory profile values the full set of base + extended section commands.

The parameters that are part of the base section of the profile:

GSM DATA MODE	: +CBST
AUTOBAUD	: +IPR
COMMAND ECHO	: E
RESULT MESSAGES	: Q
VERBOSE MESSAGES	: V
EXTENDED MESSAGES	: X
FLOW CONTROL OPTIONS	: &K, +IFC
CTS (C106) OPTIONS	: &B
DSR (C107) OPTIONS	: &S
DTR (C108) OPTIONS	: &D
DCD (C109) OPTIONS	: &C
RI (C125) OPTIONS	: \R
POWER SAVING	: +CFUN
DEFAULT PROFILE	: &Y0
S REGISTERS	: S0;S1;S2;S3;S4;S5;S7;S10;S12;S25;S30;S38
CHARACTER FORMAT	: +ICF

The parameters that are part of the extended section of the profile:

+CRC,	+FCLASS,	+DR,
+ILRR,	+CR,+CSNS,	+CRLP,
+CNMI,	+CMEE,	+CSMP,
+CSDH,	+CSCB,	+CSSN,

+CUSD,	+CALM,	+CRSL,
+CMUT,	+CAOC,	+CREG,
+CLIP,	+CLIR,	+CMGF,
+CCWA,	+CLVL,	+COPS,
#QSS,	#ACAL,	#SMOV,
#CAP,	#HFMICG,	#HSMICG,
#SHFEC,	#SRS,	#STM,
#SHFSD,	#SRP	

The parameters that are always saved in NVM independently from the profile (unique values):

+CGDCONT,	+CGQREQ,	+CGQMIN,
#BND		

The parameters that are saved in NVM independently from the profile on request:

#USERID,	#PASSW,	#PKTSZ,
#DSTO,	#SKTTO,	#SKTSET
#SKTCT		

saved with #SKTSAV command and reset with #SKTRST command.

#ESMTP,	#EADDR,	#EUSER,
#EPASSW		

saved with #ESAV command and reset with #ERST command.

6.3 Command Availability Table

The available command, depending on the module are evidenced in the following table:

	PRODUCT			
COMMAND	**GM862-PYTHON**	**GM862-PCS**	**GM862-GPRS**	**GM862-GSM**
&F	•	•	•	•
&F1	•	•	•	•
Z	•	•	•	•
+FCLASS	•	•	•	•
&Y	•	•	•	•
&W	•	•	•	•
&Z	•	•	•	•
&N	•	•	•	•
+GMI	•	•	•	•
+GMM	•	•	•	•
+GMR	•	•	•	•
+GCAP	•	•	•	•
+GSN	•	•	•	•
&V	•	•	•	•
&V0	•	•	•	•
&V1	•	•	•	•
&V2	•	•	•	•
&V3	•	•	•	•
\V	•	•	•	•
%L	•	•	•	•
%Q	•	•	•	•
+GCI	•	•	•	•
L	•	•	•	•
M	•	•	•	•
E	•	•	•	•
Q	•	•	•	•
V	•	•	•	•
X	•	•	•	•

I	•	•	•	•
&C	•	•	•	•
&D	•	•	•	•
&K	•	•	•	•
&S	•	•	•	•
\R	•	•	•	•
+IPR	•	•	•	•
+IFC	•	•	•	•
+ILRR	•	•	•	•
+ICF	•	•	•	•
D	•	•	•	•
T	•	•	•	•
P	•	•	•	•
A	•	•	•	•
A/	•	•	•	•
H	•	•	•	•
O	•	•	•	•
&G	•	•	•	•
&P	•	•	•	•
&Q	•	•	•	•
+MS	•	•	•	•
%E	•	•	•	•
+DS	•	•	•	•
+DR	•	•	•	•
\B	•	•	•	•
\K	•	•	•	•
\N	•	•	•	•
S0	•	•	•	•
S1	•	•	•	•
S2	•	•	•	•
S3	•	•	•	•
S4	•	•	•	•
S5	•	•	•	•
S7	•	•	•	•

S10	•	•	•	•
S12	•	•	•	•
S25	•	•	•	•
S30	•	•	•	•
S38	•	•	•	•
ETSI GSM 07.07 AT COMMANDS				
+CGMI	•	•	•	•
+CGMM	•	•	•	•
+CGMR	•	•	•	•
+CGSN	•	•	•	•
+CSCS	•	•	•	•
+CIMI	•	•	•	•
+CBST	•	•	•	•
+CRLP	•	•	•	•
+CR	•	•	•	•
+CEER	•	•	•	•
+CRC	•	•	•	•
+CNUM	•	•	•	•
+COPN	•	•	•	•
+CREG	•	•	•	•
+COPS	•	•	•	•
+CLCK	•	•	•	•
+CPWD	•	•	•	•
+CLIP	•	•	•	•
+CLIR	•	•	•	•
+CCFC	•	•	•	•
+CCWA	•	•	•	•
+CHLD	•	•	•	•
+CUSD	•	•	•	•
+CAOC	•	•	•	•
+CLCC	•	•	•	•
+CSSN	•	•	•	•
+CCUG	•	•	•	•
+CPAS	•	•	•	•

+CFUN	•	•	•	•
+CPIN	•	•	•	•
+CSQ	•	•	•	•
+CPBS	•	•	•	•
+CPBR	•	•	•	•
+CPBF	•	•	•	•
+CPBW	•	•	•	•
+CCLK	•	•	•	•
+CALA	•	•	•	•
+CALM	•	•	•	•
+CRSL	•	•	•	•
+CLVL	•	•	•	•
+CMUT	•	•	•	•
+CACM	•	•	•	•
+CAMM	•	•	•	•
+CPUC	•	•	•	•
+CMEE	•	•	•	•
+VTS	•	•	•	•
+CGACT	•	•	•	
+CGATT	•	•	•	
+CGDATA	•	•	•	
+CGDCONT	•	•	•	
+CGPADDR	•	•	•	
+CGREG	•	•	•	
+CGQMIN	•	•	•	
+CGQREQ	•	•	•	
+CBC	•	•		
ETSI GSM 07.05 AT COMMANDS				
+CSMS	•	•	•	•
+CPMS	•	•	•	•
+CMGF	•	•	•	•
+CSMP	•	•	•	•
+CSDH	•	•	•	•
+CSAS	•	•	•	•

+CRES	•	•	•	•
+CSCB	•	•	•	•
+CSCA	•	•	•	•
+CNMI	•	•	•	•
+CMGL	•	•	•	•
+CMGR	•	•	•	•
+CMGS	•	•	•	•
+CMSS	•	•	•	•
+CMGW	•	•	•	•
+CMGD	•	•	•	•
CUSTOM AT COMMANDS				
#CGMI	•	•	•	•
#CGMM	•	•	•	•
#CGMR	•	•	•	•
#CIMI	•	•	•	•
#CGSN	•	•	•	•
#CAP	•	•	•	•
#SRS	•	•	•	•
#SRP	•	•	•	•
#STM	•	•	•	•
#PCT	•	•	•	•
#SHDN	•	•	•	•
#WAKE	•	•	•	•
#QTEMP	•	•	•	•
#SGPO	•	•	•	•
#GGPI	•	•	•	•
#GPIO	•	•	•	•
#MONI	•	•	•	•
#QSS	•	•	•	•
#ACAL	•	•	•	•
#SMOV	•	•	•	•
#SHFEC	•	•	•	•
#HFMICG	•	•	•	•
#HSMICG	•	•	•	•

#SHFSD	•	•	•	•
#/	•	•	•	•
#BND	•	•		
FAX CLASS 1 AT COMMANDS				
+FCLASS	•	•	•	•
+FMI	•	•	•	•
+FMM?	•	•	•	•
+FMR	•	•	•	•
+FTS	•	•	•	•
+FRS	•	•	•	•
+FTM	•	•	•	•
+FRM	•	•	•	•
+FTH	•	•	•	•
+FRH	•	•	•	•
+FLO	•	•	•	•
+FPR	•	•	•	•
+FDD	•	•	•	•
EASY GPRS EXTENSION AT COMMANDS				
#USERID	•	•	•	
#PASSW	•	•	•	
#PKTSZ	•	•	•	
#DSTO	•	•	•	
#SKTTO	•	•	•	
#SKTSET	•	•	•	
#SKTOP	•	•	•	
#QDNS	•	•	•	
#SKTCT	•	•	•	
#SKTSAV	•	•	•	
#SKTRST	•	•	•	
#GPRS	•	•	•	
#SKTD	•	•	•	
#SKTL	•	•	•	
#FRWL	•	•	•	
EASY CAMERA EXTENSION AT COMMANDS				

#CAMON	•	•		
#CAMOFF	•	•		
#TPHOTO	•	•		
#RPHOTO	•	•		
#OBJL	•	•		
#OBJR	•	•		
#CAMQUA	•	•		
#CMODE	•	•		
#ESMTP	•	•		
#EADDR	•	•		
#EUSER	•	•		
#EPASSW	•	•		
#ESAV	•	•		
#ERST	•	•		
#SEMAIL	•	•		
EASY SCAN EXTENSION AT COMMANDS				
*CSURV	•	•		
*CSURVC	•	•		
*CSURVU	•	•		
*CSURVUC	•	•		
#CSURV	•	•		
#CSURVC	•	•		
#CSURVU	•	•		
#CSURVUC	•	•		
#CSURVF	•	•		
JAMMING DETECT and REPORT				
#JDR	•	•	•	•
EASY SCRIPT EXTENSION COMMANDS				
#WSCRIPT	•			
#ESCRIPT	•			
#RSCRIPT	•			
#LSCRIPT	•			
#DSCRIPT	•			
#REBOOT	•			

Hayes Compliant AT Commands

6.3.1 Generic Modem Control

6.3.1.1 &F - reset base section factory profile configuration

&F – reset base section factory profile configuration	
Execute command **AT&F**	Calls the base section profile factory configuration, resetting the factory default values.
Read command	
Write command	
Test command	
Example	
Reference	V25ter.
SW release	Version A

6.3.1.2 &F1 - reset full factory profile configuration

&F1 – reset full factory profile configuration	
Execute command **AT&F1**	Calls the full factory profile configuration, resetting the full default profile values.
Read command	
Write command	
Test command	
Example	
Reference	Telit specifications
SW release	Version A

6.3.1.3 Z - soft reset

Z – soft reset	
Execute command **ATZ <n>**	Loads the configuration with the specified base section and the default extended section profile. Parameter: <n > = 0…1 - profile Note: Must be the last command on the line.
Read command	
Write command	
Test command	
Example	
Reference	V25ter.
SW release	Version A

6.3.1.4 +FCLASS - select active service class

+FCLASS - select active service class	
Execute command	
Read command **AT+FCLASS?**	Returns the current configuration value of the parameter <n>.
Write command **AT+FCLASS= <n>**	Set the GM862-GPRS in specified connection mode (data, fax, voice), hence all the calls done after, will be data or voice. Parameter: <n>: 0 = data <n>: 1 = fax class 1 <n>: 8 = voice
Test command **AT+FCLASS=?**	Returns all supported values of the parameters <n>.
Example	
Reference	GSM 07.07
SW release	Version A

6.3.1.5 &Y - designate a default reset basic profile

&Y – designate a default reset basic profile	
Execute command **AT&Y <n>**	The GM862-GPRS is able to store 2 complete configurations (see command &W). The command &Y defines which one of the 2 basic profiles will be the default profile to be loaded on startup. Parameter: <n> = 0…1 - profile Note: Differently from ATZ<n> command which simply loads the desired profile , with the command AT&Y the chosen profile becomes the one which will be loaded on every startup.
Read command	
Write command	
Test command	
Example	
Reference	
SW release	Version A

6.3.1.6 &P - designate a default reset full profile

&P – designate a default reset full profile	
Execute command **AT&P <n>**	The GM862-GPRS is able to store 2 complete configurations (see command &W). The command &P defines which one of the 2 full profiles will be the default profile to be loaded on startup. Parameter: <n> = 0…1 - profile Note: Differently from ATZ<n> command which simply loads the desired profile , with the command AT&P the chosen profile becomes the one which will be loaded on every startup.
Read command	
Write command	
Test command	
Example	
Reference	Telit Specifications
SW release	Version D

6.3.1.7 &W - store current configuration

&W – store current configuration	
Execute command **AT&W <n>**	Stores on profile n the complete configuration of the device. Parameter: <n> = 0…1 - profile
Read command	
Write command	
Test command	
Example	
Reference	
SW release	Version A

6.3.1.8 &Z - store telephone number in the Telit GM862 Family modules internal phonebook

&Z – store telephone number in the GM862 module internal phonebook	
Execute command **AT&Z <n> = <nr>**	The GM862 modules has a built in non volatile memory in which 10 telephone numbers of a maximum 24 digits can be stored. The command AT&Z <n> = <nr> stores in the record n the telephone number nr. The records cannot be overwritten, they must be cleared before rewriting. Parameter: <n>: phonebook record <nr>: telephone number Note1: To delete the record <n> the command AT&Z<n>= must be issued. Note2: The records in the GM862 memory can be viewed with the command AT&N, while the telephone number stored in the record n can be dialled by giving the command ATDS=<n>.
Read command	
Write command	
Test command	
Example	
Reference	
SW release	Version A

6.3.1.9 &N - display internal phonebook stored numbers

&N – display internal phonebook stored numbers	
Execute command **AT&N <n>**	Returns the telephone number stored in the internal memory at the record number <n>. Parameter: <n> - phonebook record Note: if parameter <n> is omitted then all the internal records are shown.
Read command	
Write command	
Test command	
Example	
Reference	
SW release	Version A

6.3.1.10 +GMI - request manufacturer identification

+GMI – request manufacturer identification	
Execute command **AT+GMI**	Returns the manufacturer identification.
Read command	
Write command	
Test command	
Example	
Reference	V.25ter
SW release	Version A

6.3.1.11 +GMM - request model identification

+GMM – request model identification	
Execute command **AT+GMM**	Returns the model identification.
Read command	
Write command	
Test command	
Example	
Reference	V.25ter
SW release	Version A

6.3.1.12 +GMR - request revision identification

+GMR – request revision identification	
Execute command **AT+GMR**	Returns the software revision identification.
Read command	
Write command	
Test command	
Example	
Reference	V.25ter
SW release	Version A

6.3.1.13 +GCAP - request capabilities list

+GCAP – request capabilities list	
Execute command **AT+GCAP**	Returns the equipment supported command set list. Where: +CGSM: GSM ETSI command set +FCLASS: Fax command set +DS: Data Service common modem command set +MS: Mobile Specific command set
Read command	
Write command	
Test command	
Example	
Reference	V.25ter
SW release	Version A

6.3.1.14 +GSN - request serial number

+GSN – request serial number	
Execute command **AT+GSN**	Returns the device board serial number. Note: The number returned is not the IMSI, it is only the board number
Read command	
Write command	
Test command	
Example	
Reference	V.25ter
SW release	Version A

6.3.1.15 &V - display current configuration & profile

&V – display current configuration & profile	
Execute command **AT&V**	Returns some of the base configuration parameters settings.
Read command	
Write command	
Test command	
Reference	V25ter
SW release	Version A

6.3.1.16 &V0 - display current configuration & profile

&V0 – display current configuration & profile	
Execute command **AT&V0**	Returns all the configuration parameters settings. Note: this command is the same as &V, it is included only for backwards compatibility.
Read command	
Write command	
Test command	
Example	
Reference	
SW release	Version A

6.3.1.17 &V1 - display S registers values

&V1 – display S registers values	
Execute command **AT&V1**	Returns the value of the S registers in decimal and hexadecimal value in the format: REG DEC HEX <reg> <dec> <hex> where <reg> - S register number (0 .. 38) <dec> - current value in decimal notation <hex> - current value in hexadecimal notation
Read command	
Write command	
Test command	
Example	
Reference	
SW release	Version A

6.3.1.18 &V2 - display last connection statistics

&V2 – display last connection statistics	
Execute command AT&V2	Returns the last connection statistics & connection failure reason.
Read command	
Write command	
Test command	
Example	
Reference	
SW release	Version A

6.3.1.19 &V3 - display S registers values

&V3 – display S registers values	
Execute command **AT&V3**	Returns the value of the S registers in decimal and hexadecimal value in the format: REG DEC HEX <reg> <dec> <hex> where <reg> - S register number (0 .. 38) <dec> - current value in decimal notation <hex> - current value in hexadecimal notation
Read command	
Write command	
Test command	
Example	
Reference	
Note	

6.3.1.20 \V - single line connect message

\V – Single line connect message	
Execute command **AT\V <n>**	Set single line connect message. Parameter: <n> = 0 - off <n> = 1 on
Read command	
Write command	
Test command	
Example	
Reference	
SW release	Version A

6.3.1.21 %L - report line signal level

%L – report line signal level	
Execute command **AT%L**	Reports the line signal level. Note: Since on mobile phone there's no line connection, command is included only for compatibility and returns only OK. To check signal strength (level) use the +CSQ command.
Read command	
Write command	
Test command	
Example	
Reference	
SW release	Version A

6.3.1.22 %Q - report line quality

%Q – report line quality	
Execute command **AT%Q**	Reports the line quality indicator. Note: Since on mobile phone there's no line connection, command is included only for compatibility and returns only OK. To check signal quality use the +CSQ command.
Read command	
Write command	
Test command	
Example	
Reference	
SW release	Version A

6.3.1.23 +GCI – select the country of installation

+GCI – select the country of installation	
Set command **AT+GCI=<code>**	Selects the code of the country of installation according to ITU-T.35 Annex A. Parameter: 59
Read command **AT+GCI?**	Reports the currently selected country code.
Test command **AT+GCI=?**	Reports the supported country codes (only 59 (Italy)).
Example	
Reference	V25ter.
Note	

6.3.1.24 L – monitor speaker loudness

L – monitor speaker loudness	
Execute command **ATL<n>**	It has no effect is included only for backward compatibility with landline modems. Parameter: <n> = 0 - 3
Read command	
Write command	
Test command	
Example	
Reference	V25ter.
Note	

6.3.1.25 M - monitor speaker mode

M – monitor speaker mode	
Execute command **ATM<n>**	It has no effect is included only for backward compatibility with landline modems. Parameter: <n> = 0 - 3
Read command	
Write command	
Test command	
Example	
Reference	V25ter.
Note	

6.3.2DTE - modem interface control

6.3.2.1 E - command echo

E – command echo	
Execute command	
Read command	
Write command **ATE <n>**	Controls the command echo response of the device, enabling or disabling the echo. Parameter: <n> = 0 - disables command echo, hence after this command the only characters received by DTE are the responses to commands sent to device. **<n> = 1 - enables command echo (default)** , hence command sent to the device are echoed back to the DTE before the response is given.
Test command	
Example	
Reference	V25ter
SW release	Version A

6.3.2.2 Q - quiet resut codes

Q – quiet result codes	
Execute command	
Read command	
Write command **ATQ <n>**	Enables or disables the result codes. Parameter: **<n> = 0 - enables result codes (default)** <n> = 1 - disables result codes Note: After disabling result codes with ATQ1, the device does not return a response to commands (usually OK), but command are executed anyway.
Test command	
Example	
Reference	V25ter
SW release	Version A

6.3.2.3 V- result code form

V – result code form	
Execute command	
Read command	
Write command **ATV <n>**	Sets the result code format. Parameter: <n> = 0 set the short format (terse) return codes are numbers 0-8. **<n> = 1 set the long form (verbose) return codes. (default)** Note: Line feed is not issued before a short format result code.
Test command	
Example	
Reference	V25ter
SW release	Version A

6.3.2.4 X - extended result codes

X – extended result codes	
Execute command	
Read command	
Write command **ATX <n>**	Selects the result code messages subset used by the modem to inform the DTE of the result of the commands. Parameter: <n> = 0 - send only OK, CONNECT, RING, NO CARRIER, ERROR, NO ANSWER results. Busy tones reporting is disabled. <n> = 1,2,3,4 - reports all messages (1 - default).
Test command	
Note	For complete control on CONNECT response message see also +DR command.
Reference	V25ter
SW release	Version A

6.3.2.5 I - Request identifier and software checksum

I – request identifier and software checksum	
Execute command	
Read command	
Write command **ATI <n>**	Returns the identifier and a software checksum. Parameter: <n> = 0 numerical identifier. <n> = 1 GM862 checksum <n> = 2 checksum check result <n> = 3 manufacturer <n> = 4 product name <n> = 5 DOB version
Test command	
Example	
Reference	V25ter
SW release	Version A

6.3.2.6 &C - data carrier detect (DCD) control

&C – Data carrier detect (DCD) control	
Execute command	
Read command	
Write command **AT&C <n>**	Controls the RS232 DCD output behaviour. Parameter: <n> = 0 DCD remains high always. <n> = 1 DCD follows the Carrier detect status: if carrier is detected DCD is high, otherwise DCD is low. (default)
Test command	
Example	
Reference	V25ter
SW release	Version A

6.3.2.7 &D - data terminal ready (DTR) control

&D – data terminal ready (DTR) control	
Execute command	
Read command	
Write command **AT&D <n>**	Controls the GM862 behaviour to the RS232 DTR transitions. Parameter: **<n> = 0 DTR transitions are ignored. (default)** <n> = 1 when the GM862 is connected, the high to low transition of DTR pin sets the device in command mode, the current connection is NOT closed. <n> = 2 when the GM862 is connected , the high to low transition of DTR pin sets the device in command mode and the current connection is closed. <n> = 3 C108/1 operation is enabled <n> = 4 C108/1 operation is disabled Note: if AT&D2 command is issued, the device does not answer to incoming calls if DTR is low, even if ATA command is given.
Test command	
Example	
Reference	V25ter
SW release	Version A

6.3.2.8 &K - flow control

&K – flow control	
Execute command	
Read command	
Write command **AT&K <n>**	Controls the RS232 flow control behaviour. Parameter. <n> = 0 - disabled <n> = 1 - only CTS active, Hardware mono-directional <n> = 2 - XON/XOFF software mono direction **<n> = 3 - RTS/CTS active, Hardware bi-directional (default)** <n> = 4 - XON/XOFF, Software bi-directional with filtering <n> = 5 - XON/XOFF, Software bi-directional without filtering (Pass Through) <n> = 6 - RTS/CTS active, Hardware bi-directional & software XON/XOFF (bi-directional) with filtering
Test command	
Example	
Reference	
SW release	Version A

6.3.2.9 &Q – sync/async mode

&Q – sync/async mode	
Execute command **AT&Q**	It is has no effect is included only for backward compatibility with landline modems.
Read command	
Write command	
Test command	
Example	
Reference	
Note	

6.3.2.10 &S - data set ready (DSR) control

&S – data set ready (DSR) control	
Execute command	
Read command	
Write command **AT&S <n>**	Controls the RS232 DSR pin behaviour: n = 0 - always ON n = 1 - follows the GSM traffic channel indication. n = 2 - ON when connected **n = 3 - ON when device is ready to receive commands (default)** Note: if option 1 is selected then DSR is tied up when the device receives from the network the GSM traffic channel indication.
Test command	
Example	
Reference	V25ter
SW release	Version A

6.3.2.11 \R - ring (RI) control

\R – ring (RI) control	
Execute command	
Read command	
Write command **AT\R <n>**	controls the RING output pin behaviour. Parameter: <n> = 0 - RING on during ringing and further connection **<n> = 1 – RING on during ringing (default)** <n> = 2 - RING follows the ring signal Note: to check the ring option status use the &V command.
Test command	
Example	
Reference	
SW release	Version A

6.3.2.12 +IPR - fixed DTE interface rate

+IPR – fixed DTE interface rate	
Execute command	
Read command **AT+IPR?**	Returns the current value of +IPR parameter.
Write command **AT+IPR = <rate>**	Specifies the DTE speed at which the device accepts commands during command mode operation, it may be used to fix the DTE-DCE interface speed. Parameter: <rate> = 0/300/1200/2400/4800/9600/19200/38400/57600/115200 If <rate> is unspecified or set to 0, then automatic speed detection is enabled and also character format (see +ICF) is set to auto-detect. (default) If <rate> is specified and not 0, DTE-DCE speed is fixed at that speed, hence no speed auto-detection (autobauding) is enabled.
Test command **AT+IPR=?**	Returns the supported serial port speed list.
Example	
Reference	V25ter
SW release	Version A

6.3.2.13 +IFC - DTE - DTA flow control

+IFC – DTE-DTA flow control	
Execute command	
Read command **AT+IFC?**	Returns active flow control settings.
Write command **AT+IFC = <by_te>, <by_ta>**	Response Selects the flow control behavior of the serial port in both directions: from DTE to DTA (<by_ta> option) and from DTA to DTE (<by_te>) Parameter: <by_te> - flow control option for the data received by DTE, <by_ta> - flow control option for the data sent by DTA <by_te> = 0 - flow control None <by_te> = 1 - XON/XOFF filtered **<by_te> = 2 - C105 (RTS) default** <by_te> = 3 - XON/XOFF not filtered <by_ta> = 0 - flow control None <by_ta> = 1 - XON/XOFF **<by_ta> = 2 - C106 (CTS) default** Note: This command is equivalent to &K command.
Test command **AT+IFC=?**	Response Returns all supported values of the parameters <by_te> and <by_ta>.
Example	
Reference	V25ter
SW release	Version A

6.3.2.14 +ILRR - DTE - modem rate reporting

+ILRR – DTE-modem rate reporting	
Execute command	
Read command **AT+ILRR?**	Returns active setting of port speed rate reporting information.
Write command **AT+ILRR = <n>**	Controls whether or not the +ILRR:<rate> information text is transmitted from the GM862 to the DTE. Parameter: **<n> = 0 - local port speed rate reporting disabled (default)** <n> = 1 - local port speed rate reporting enabled Note: this information if enabled is sent upon connection.
Test command **AT+ILRR=?**	Returns all supported values of the parameter <n>
Example	
Reference	V25ter
SW release	Version A

6.3.2.15 +ICF - DTE - modem character format

+ICF – DTE-modem character format	
Execute command	
Read command **AT+ICF?**	Returns current value of the character format. If autobauding is enabled the returned value is 0.
Write command **AT+ICF = <format>[,<parity>]**	Defines the asynchronous character format to be used when autobauding is disabled. Parameter: <format> = 0..5, < parity> = 0/1 AT+ICF = 0 - auto detect AT+ICF = 1 - 8N2 AT+ICF = 2,0 - 8O1 AT+ICF = 2,1 - 8E1 **AT+ICF = 3 - 8N1 (default)** AT+ICF = 5,0 - 7O1 AT+ICF = 5,1 - 7E1 Note: the character format is defined as: number of bit per char, parity bit and stop bit; where parity can be None, Odd and Even. E.g. 8E1 means 8 bit per char with the even parity bit and 1 stop bit.
Test command **AT+ICF=?**	Returns all supported values of the parameters <format> and <parity>
Example	
Reference	V25ter
SW release	Version A

6.3.3 Call Control

6.3.3.1 D - dial

D – dial	
Execute command **ATD <number>**	Starts a call to the phone number given as parameter. Parameter: <number> - phone number to be dialled Note: The call is a data call or voice call depending on `+FCLASS` setting. If `AT+FCLASS=8` command was issued before calling then the call will be done as a voice call, instead if `AT+FCLASS=0` command was issued the call will be a data one. The default value of `+FCLASS` is 0 hence calls are by default data ones. The numbers accepted are 0-9 and *,#,"A", "B", "C", "D","+". For backwards compatibility with landline modems modifiers "T", "P", "R", ",", "W", "!", "@" are accepted but have no effect.
ATD <number> ;	Issues a VOICE call to the number given regardless of the current value of the parameter +FCLASS, which remains unaffected. Parameter: <number> - phone number to be dialled.
ATD <n> [;]	Issues a call (VOICE if ";" modifier is added to the end of the command) to the number stored in the GM862 internal phonebook position number <n>. Parameter: <n> - internal phonebook position to be called
ATD> <n> [;]	Issues a call (VOICE if ";" modifier is added to the end of the command) to the number stored in the SIM phonebook in the record number <n>. Parameter: <n> - SIM phonebook position to be called
ATDL	Issues a call to the last number dialled.
ATDS=<nr>	Issues a call to the internally stored number at the position <nr>. (See commands &N and &Z)
ATD<n> I [;]	Issues a call [voice if ; is added] overwriting the CLIR setting in order to hide the CLI to the called party for the current call only.
ATD<n> i [;]	Issues a call [voice if ; is added] overwriting the CLIR setting in order to show the CLI to the called party for the current call only.
ATD<n> G[;] **ATD<n>g[;]**	Issues a call [voice if ; is added] checking the CUG supplementary service for the current call. Refer to +CCUG command.
ATD*<gprs_sc>[*<addr>][*[<L2P>][*[<cid>]]]]#	This command is specific of GPRS functionality and causes the MT to perform whatever actions are necessary to establish communication between the TE and the external PDN. Parameters: <GPRS_SC> - (GPRS Service Code) a digit string (value 99) which identifies a request to use the GPRS <addr> - string that identifies the called party in the address space applicable to the PDP. <L2P>: a string which indicates the layer 2 protocol to be used (see

D – dial	
	+CGDATA command). For communications software that does not support arbitrary characters in the dial string, the following numeric equivalents shall be used: 1 - PPP Other values are reserved and will result in an ERROR response to the set command. <cid>: a digit string which specifies a particular PDP context definition (see +CGDCONT command).
Read command	
Write command	
Test command	
Example	
Reference	V25ter.
SW release	Version A

6.3.3.2 T - set tone dial

T – set tone dial	
Execute command **ATT**	It has no effect is included only for backward compatibility with landline modems.
Read command	
Write command	
Test command	
Example	
Reference	V25ter.
SW release	Version A

6.3.3.3 P - set pulse dial

P – set pulse dial	
Execute command **ATP**	It has no effect is included only for backward compatibility with landline modems.
Read command	
Write command	
Test command	
Example	
Reference	V25ter.
SW release	Version A

6.3.3.4 A - answer

A – answer	
Execute command **ATA**	It is used to answer to an incoming call if automatic answer is disabled. Note: This command MUST be the last in the command line and must be followed immediately by a <CR> character.
Read command	
Write command	
Test command	
Example	
Reference	V25ter.
SW release	Version A

6.3.3.5 A/ - Last command automatic repetition

A/ – last command automatic repetition	
Execute command **A/**	It is used to execute again the last received command. This command works only at fixed IPR. **Note:** This command has been substituted with AT#/ command. Refer to #/ command reference.
Read command	
Write command	
Test command	
Example	
Reference	
SW release	Version A

6.3.3.6 H - disconnect

H - disconnect	
Execute command **ATH**	It is used to close the current conversation (voice, data or fax). **Note**: When a data conversation is active the device is in on-line mode commands are not sensed, instead characters are sent to the other interlocutor. To issue this command you must be in command mode operation, hence escape sequence (see register S2) may be required before issuing this command, otherwise if &D1 option is active, DTR pin should be tied low to return in the command mode. When a voice call is active, no escape sequence is needed.
Read command	
Write command	
Test command	
Example	
Reference	V25ter.
SW release	Version A

6.3.3.7 O - return to On Line Mode

O – return to on line mode	
Execute command **ATO**	It is used return to On-line mode from command mode. If there's no connection active returns ERROR. Note: After the issue of this command, if the device is in conversation, to send other commands to the device you must return to command mode by issuing the escape sequence or lowering DTR if &D1 option is active.
Read command	
Write command	
Test command	
Example	
Reference	V25ter.
SW release	Version A

6.3.3.8 &G - guard tone

&G – guard tone	
Execute command **AT&G**	It is has no effect is included only for backward compatibility with landline modems.
Read command	
Write command	
Test command	
Example	
Reference	
SW release	Version A

6.3.4 Modulation control

6.3.4.1 +MS - modulation control

+MS – modulation control	
Execute command	
Read command	
Write command **AT+MS = <modulation>, <automode>, <min_speed>, <max_speed>**	This command has no effect is included only for backward compatibility with landline modems. Parameter: <modulation> =V21 / V22 / V22B / V23C / V32 / V34 <automode> = 0/1 <min_speed>,<max_speed> 0 - unspecified 300 -14400 - speed in bps **Note**: to change modulation requested use +CBST command.
Test command	
Example	
Reference	
SW release	Version A

6.3.4.2 %E - enable/disable line quality monitor and auto retrain or fallback / fallforward

%E – enable/disable line quality monitor and auto retrain or fallback/fallforward	
Execute command **AT%<n>**	It has no effect is included only for backward compatibility with landline modems. <n> = 0,1,2
Read command	
Write command	
Test command	
Example	
Reference	
SW release	Version A

6.3.4.3 \N – operating mode

\N – operating mode	
Execute command	
Read command	
Write command **AT\N**	It has no effect is included only for backward compatibility with landline modems
Test command	
Example	
Reference	
Note	

6.3.5 Compression control

6.3.5.1 +DS - set data compression

+DS – set data compression	
Execute command	
Read command **AT+DS?**	Returns current value of the data compression parameter.
Write command **AT+DS = <n>**	Sets the V42 compression parameter. Parameter: <n> = 0 - no compression <n> = 1 - compression enabled Note: **The only value supported is 0 - no compression**
Test command **AT+DS=?**	Returns all supported values of the parameter n
Example	
Reference	V25ter
SW release	Version A

6.3.5.2 +DR - data compression reporting

+DR – data compression reporting	
Execute command	
Read command	
Write command **AT+DR = <n>**	Controls the data compression reporting upon connection. Parameter: <n> = 0 no data compression reporting is displayed at the connection <n> = 1 a data compression report message is sent by the device upon connection
Test command	
Example	
Reference	V25ter
SW release	Version A

6.3.6Break control

6.3.6.1 \B - transmit break to remote

\B – transmit break to remote	
Execute command	
Read command	
Write command **AT\B**	It has no effect is included only for backward compatibility with landline modems
Test command	
Example	
Reference	
SW release	Version A

6.3.6.2 \K - break handling

\K – break handling	
Execute command	
Read command	
Write command **AT\K<n>**	Response It has no effect is included only for backward compatibility with landline modems <n> = 1.. 5
Test command	
Example	
Reference	
SW release	Version A

6.3.7 S parameters

6.3.7.1 S0 - number of rings to auto answer

S0 – number of rings to auto answer	
Execute command	
Read command **ATS0?**	Returns the current value of S0 parameter.
Write command **ATS0 = <n>**	Sets the number of rings required before device automatically answers an incoming call. Parameter: <n> = 0-255 <n> = 0 auto answer disabled (default)
Test command	
Example	
Reference	V25ter
SW release	Version A

6.3.7.2 S1 - ring counter

S1 – ring counter	
Execute command	
Read command **ATS1?**	S1 is incremented each time the device detects the ring signal of an incoming call. S1is cleared if no ring occur over an eight seconds interval. This command returns the value of this parameter.
Write command	
Test command	
Example	
Reference	
SW release	Version A

6.3.7.3 S2 - escape character

S2 – escape character	
Execute command	
Read command **ATS2?**	Returns the current value of S2 parameter.
Write command **ATS2 = <char>**	S2 holds the decimal value of the ASCII character used as escape character. This command sets this character equal to <char>. Parameter: <char> - escape character 0-255 Note: The escape sequence consists of three escape characters preceded by n ms of idle and followed by m ms of idle. Note: Default 43 (+)
Test command	
Example	
Reference	
SW release	Version A

6.3.7.4 S3 - carriage return character

S3 – carriage return character	
Execute command	
Read command **ATS3?**	Returns the current value of S3 parameter.
Write command **ATS3 = <char>**	Sets the command line and result code terminator character in decimal ASCII. Parameter: <char> - carriage return character 0-127 Note: Default 13 (Carriage Return)
Test command	
Example	
Reference	V25ter
SW release	Version A

6.3.7.5 S4 - line feed character

S4 – line feed character	
Execute command	
Read command **ATS4?**	Returns the current value of S4 parameter.
Write command **ATS4 = <char>**	Sets the character recognized as line feed character. Parameter: <char> - line feed character 0-127 Note: Default 10 (Line Feed). The line feed character is output after carriage return character if verbose result codes are used (V1 option used).
Test command	
Example	
Reference	V25ter
SW release	Version A

6.3.7.6 S5 - backspace character

S5 – backspace character	
Execute command	
Read command **ATS5?**	Returns the current value of S5 parameter.
Write command **ATS5 = <char>**	Sets the decimal ASCII character that is recognized as backspace character. Parameter: <char> - backspace character 1-127 Note: Default 8 (backspace)
Test command	
Example	
Reference	V25ter
SW release	Version A

6.3.7.7 S7 - wait time for carrier, silence or dial tone

S7 – wait time for carrier, silence or dial tone	
Execute command	
Read command **ATS7?**	Returns the current value of S7 parameter.
Write command **ATS7 = <sec>**	S7 sets the length of time in seconds that the device will wait for carrier before hanging up the call. The timer is started when the device finishes dialing (originate), or 2 seconds after going off-hook. Parameter: <sec> - 1-255 Note: Default is 60 seconds
Test command	
Example	
Reference	V25ter
SW release	Version A

6.3.7.8 S10 - lost carrier to hang up delay

S10 – lost carrier to hang up delay	
Execute command	
Read command **ATS10?**	Returns the current value of S10 parameter.
Write command **ATS10 = <time>**	S10 defines the maximum time allowed to a carrier loss before disconnecting. Parameter: <time> - tenth of seconds 1-255 Note: default is 50 (5 seconds).
Test command	
Example	
Reference	V25ter
SW release	Version A

6.3.7.9 S12 - escape prompt delay

S12 – escape prompt delay	
Execute command	
Read command **ATS12?**	Returns the current value of S12 parameter.
Write command **ATS12 = <time>**	Defines the guard time of silence before and after the escape sequence in order to accept it as a valid one. Parameter: <time> - 20-255 fiftieth of seconds Note: default is 50 (1 second)
Test command	
Example	
Reference	
SW release	Version A

6.3.7.10 S25 - delay to DTR off

S25 –delay to DTR off	
Execute command	
Read command **ATS25?**	Returns the current value of S25 parameter.
Write command **ATS25 = <time>**	Defines how long the DTR must be off to consider the high to low transition as a valid one. Parameter: <time> - 1-255 hundredth of seconds Note: default is 5 (5 hundredth of second)
Test command	
Example	
Reference	
SW release	Version A

6.3.7.11 S30 - disconnect inactivity timer

S30 –disconnect inactivity timer	
Execute command	
Read command **ATS30?**	Returns the current value of S30 parameter.
Write command **ATS30 = <min>**	Defines the inactivity timeout timer value in minutes. The Device disconnects if no characters are exchanged for a time period of at least S30 minutes. Parameter: <min> - 0-127 minutes <min> = 0 - means inactivity timer disabled, hence no disconnect on inactivity feature is enabled. Note: Default is 0
Test command	
Example	
Reference	
SW release	Version A

6.3.7.12 S38 - delay before forced hang up

S38 –delay before forced hang up	
Execute command	
Read command **ATS38?**	Returns the current value of S38 parameter.
Write command **ATS38 = <sec>**	Defines the time in seconds that the device waits after a hang up command for transmit buffer to be emptied. 255 value means that disconnecting is done only after complete buffer upload. Parameter: <sec> - 0-255 s Note: Default is 20 s
Test command	
Example	
Reference	
SW release	Version A

6.4ETSI GSM 07.07 AT Commands

6.4.1General

6.4.1.1 +CGMI - request manufacturer identification

+CGMI – request manufacturer identification	
Execute command **AT+CGMI**	Returns the device manufacturer identification code without command echo.
Read command	
Write command	
Test command	
Example	
Reference	GSM 07.07
SW release	Version A

6.4.1.2 +CGMM - request model identification

+CGMM – request model identification	
Execute command **AT+CGMM**	Returns the device model identification code without command echo.
Read command	
Write command	
Test command	
Example	
Reference	GSM 07.07
SW release	Version A

6.4.1.3 +CGMR - request revision identification

+CGMR – request revision identification	
Execute command **AT+CGMR**	Returns device software revision number without command echo.
Read command	
Write command	
Test command	
Example	
Reference	GSM 07.07
SW release	Version A

6.4.1.4 +CGSN - request product serial number identification

+CGSN – request product serial number identification	
Execute command **AT+CGSN**	Returns the product serial number, identified as the IMEI of the mobile, without command echo.
Read command	
Write command	
Test command	
Example	
Reference	GSM 07.07
SW release	Version A

6.4.1.5 +CSCS - select TE character set

+CSCS – select TE character set	
Execute command	
Read command **AT+CSCS?**	Returns the current value of the active character set.
Write command **AT+CSCS <chset>**	Sets the current character set used by the device. Parameter: <chset> = "IRA" - char set Note: The only character set supported yet is the ITU–T.50 (IRA)
Test command **AT+CSCS=?**	Returns the supported values of the parameter chset. Yet only IRA is supported.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.1.6 +CIMI - request international mobile subscriber identity (IMSI)

+CIMI – request international mobile subscriber identify (IMSI)	
Execute command **AT+CIMI**	Returns the value of the Internal Mobile Subscriber Identity stored in the SIM without command echo. Note: A SIM card must be present in the SIM card housing, else the command returns ERROR.
Read command	
Write command	
Test command	
Example	
Reference	GSM 07.07
SW release	Version A

6.4.2 Call control

6.4.2.1 +CBST - select bearer service type

+CBST – select bearer service type	
Read command **AT+CBST?**	Returns current value of the parameters <speed>,<name> and <ce>
Write command **AT+CBST = <speed>, <name>, <ce>**	Selects the bearer service <name> with data rate <speed>, and the connection element <ce> to be used when data calls are originated. Values may also be used during mobile terminated data call setup, especially in case of single numbering scheme calls (refer +CSNS). Parameters: The default values of the subparameters are manufacturer specific since they depend on the purpose of the device and data services provided by it. Not all combinations of these subparameters are supported. The supported values are: <speed> 0 – autobauding (automatic selection of the speed) 1 - 300 bps (V.21) 2 - 1200 bps (V.22) 3 - 1200/75 bps (V.23) 4 - 2400 bps (V.22bis) 6 - 4800 bps (V.32) 7 - 9600 bps (V.32) 14 - 14400 bps (V.34) 65 - 300 bps (V.110) 66 - 1200 bps (V.110) 68 - 2400 bps (V.110 or X.31 flag stuffing) 70 - 4800 bps (V.110 or X.31 flag stuffing) 71 - 9600 bps (V.110 or X.31 flag stuffing) 75 - 14400 bps (V110 or X.31 flag stuffing) <name> 0 - data circuit asynchronous <ce> 0 – transparent 1 - non transparent Default is +CBST = 0,0,1 **NOTE**: The settings AT+CBST=14,0,0 AT+CBST=75,0,0 are not supported.
Test command **AT+CBST=?**	Returns the supported range of values of the parameters.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.2.2 +CRLP - radio link protocol

+CRLP – radio link protocol	
Execute command	
Read command **AT+CRLP?**	Returns the current value of the RLP protocol parameters.
Write command **AT+CRLP = <iws>, <mws>, <T1>, <N2>, <ver>,<T4>**	Radio link protocol (RLP) parameters used when non-transparent data calls are originated may be altered with this command. Parameters: <iws> = 1...61 - IWF window Dimension <mws> = 1...61 - MS window Dimension <T1> = 39…255 – acknowledge timer (10 ms units). <N2> = 1…255 – retransmission attempts <ver> = 0 - protocol Version <T4> = 0 – re sequencing period T4 The default values are: 61,61,78,6,0,0
Test command **AT+CRLP=?**	Returns supported range of values of the RLP protocol parameters.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.2.3 +CR - service reporting control

+CR – service reporting control	
Execute command	
Read command **AT+CR?**	Returns current intermediate report setting
Write command **AT+CR = <mode>**	Controls whether or not intermediate result code +CR: <serv> is returned from the TA to the TE. If enabled, the intermediate result code is transmitted at the point during connect negotiation at which the TA has determined which speed and quality of service will be used, before any error control or data compression reports are transmitted, and before the intermediate result code `CONNECT` is transmitted. Parameter: <mode> **<mode> = 0 - disables intermediate result code report (default)** <mode> = 1 - enables intermediate result code report This command replaces V.25ter [14] command Modulation Reporting Control +MR, which is not appropriate for use with a GSM terminal. The result code parameter <serv> has the meaning: ASYNC - asynchronous transparent SYNC - synchronous transparent REL ASYNC - asynchronous non-transparent REL SYNC - synchronous non-transparent.
Test command **AT+CR=?**	Returns the supported range of values of the parameter <mode>.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.2.4 +CEER - extended error report

+CEER – extended error report	
Execute command **AT+CEER**	Reports a information text regarding some error condition that may occur: - the failure in the last unsuccessful call setup (originating or answering) - the last call release - the last unsuccessful GPRS attach or unsuccessful PDP context activation, - the last GPRS detach or PDP context deactivation. Note: if none of this condition has occurred since power up then No Error condition is reported
Read command	
Write command	
Test command	
Example	
Reference	GSM 07.07
SW release	Version A

6.4.2.5 +CRC - cellular result codes

+CRC – cellular result codes	
Execute command	
Read command **AT+CRC?**	Returns current value of the parameter <mode>.
Write command **AT+CRC = <mode>**	Controls whether or not the extended format of incoming call indication is used. When enabled, an incoming call is indicated to the TE with unsolicited result code +CRING:<type> instead of the normal RING. Parameter: **<mode> = 0 - disables extended format reporting (default)** <mode> = 1 - enables extended format reporting Note: On +CRING indication <type> parameter is the call type: <type> - call type: ASYNC asynchronous transparent data SYNC - synchronous transparent data REL ASYNC - asynchronous non-transparent data REL SYNC - synchronous non-transparent data FAX - facsimile (TS 62) VOICE - normal voice (TS 11)
Test command **AT+CRC=?**	Returns supported values of the parameter <mode>.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.2.6 +CSNS - single numbering scheme

+CSNS – single numbering scheme	
Execute command	
Read command **AT+CSNS?**	Returns current value of the parameter <mode>.
Write command **AT+CSNS = <mode>**	Set command selects the bearer or teleservice to be used when mobile terminated single numbering scheme call is established. Parameter values set with +CBST command shall be used when <mode> equals to a data service. Parameter: **<mode> = 0 - voice (default)** <mode> = 2 - fax (TS 62) <mode> = 4 - data Note: if +CBST parameter is set to a value that is not applicable to single numbering calls, ME/TA shall map the value to the closest valid one. E.g. if user has set <speed>=71, <name>=0 and <ce>=1 (non-trasparent asynchronous 9600 bps V.110 ISDN connection) for mobile originated calls, ME/TA shall map the values into non-trasparent asynchronous 9600 bps V.32 modem connection when single numbering scheme call is answered.
Test command **AT+CSNS=?**	Returns supported values of the parameter <mode>.
Example	
Reference	GSM 07.07
SW release	Version D

6.4.3Network service handling

6.4.3.1 +CNUM - subscriber number

+CNUM – subscriber number	
Execute command **AT+CNUM**	Returns the subscriber number i.e. the phone number of the device that is stored in the SIM card. Note: the returned number format is: +CNUM: [<alpha>] , <number> , <type> <alpha> - optional alphanumeric string associated to <number> <number> - string containing the phone number in the format <type> <type>- type of number: <type> = 145 - international numbering scheme (contains the character "+") <type> = 129 - national numbering scheme
Read command	
Write command	
Test command	
Example	
Reference	GSM 07.07
SW release	Version A

6.4.3.2 +COPN – read operator names

+COPN – read operator names	
Set command	
Read command **AT+COPN**	Returns the list of operator names from the ME.
Test command	
Example	
Reference	GSM 07.07
Note	

6.4.3.3 +CREG - network registration report

+CREG – network registration report	
Execute command	
Read command	
Write command **AT+CREG = <mode>**	Enables/disables network registration reports depending on the parameter mode. Parameter: <mode> **0 – disable network registration unsolicited result code (default)** 1 - enable network registration unsolicited result code 2 - enable network registration unsolicited result code with network Cell identification data If mode=1, network registration result code reports: +CREG: <stat> <stat> = 0 - not registered, ME is not currently searching a new operator to register to <stat> = 1 – registered, home network <stat> = 2 - not registered, but ME is currently searching a new operator to register to <stat> = 3 - registration denied <stat> = 4 -unknown <stat> = 5 - registered, roaming If mode=2, network registration result code reports: +CREG: <stat>[,<Lac>,<Ci>] where: <Lac> is the Local Area Code for the currently registered on cell <Ci> is the Cell Id for the currently registered on cell **Note**: <Lac> and <Ci> are reported only if mode=2 and the mobile is registered on some network cell.
Test command **AT+CREG?**	Reports the <mode> and <stat> parameter values in the format: +CREG: <mode>,<stat>[,<Lac>,<Ci>] **Note**: <Lac> and <Ci> are reported only if mode=2 and the mobile is registered on some network cell.

+CREG – network registration report	
Example	AT OK at+creg? +CREG: 0,2 (the GM862-GPRS is in network searching state) OK at+creg? +CREG: 0,2 OK at+creg? +CREG: 0,2 OK at+creg? +CREG: 0,2 OK at+creg? +CREG: 0,1 (the GM862-GPRS is registered) OK at+creg? +CREG: 0,1 OK
Reference	GSM 07.07
SW release	Version A

6.4.3.4 +COPS - operator selection

+COPS – operator selection	
Execute command	
Read command **AT+COPS?**	Returns current value of <mode>,<format> and <oper>
Write command **AT+COPS =** **[<mode>[,<format>[,<oper>]]]**	Forces an attempt to register the GSM network operator, <mode> parameter defines whether the operator selection is done automatically or it is forced with <oper> parameter. Parameters: **<mode>** **0 - automatic choice (the parameter <oper> will be ignored) (default)** 1 - manual choice unlocked (network is kept as long as available, then it can be changed with some other suited networks to guarantee the service) 3 - set only <format> parameter (the parameter <oper> will be ignored) 4 - manual/automatic (<oper> field shall be present); if manual selection fails, automatic mode (<mode>=0) is entered 5 - manual choice locked (network is kept fixed, if the chosen network is not available, then the mobile has no service) <format> 0 - alphanumeric max length 16 digits 1 – alphanumeric short form 2 - Numeric 5 digits [country code (3) + network code (2)] <oper>: network operator in the <format> parameter defined format. Note: new value <mode>=5 has been added to guarantee backward compatibility and new functionality implementation.
Test command **AT+COPS=?**	Returns the list of network operators in the format: +COPS: (<stat> ,<oper (in format=0)>,"",<oper (in format=2)>) where the new parameter stat defines the status of the operator: <stat> = 0 – unknown <stat> = 1 – available <stat> = 2 – current <stat> = 3 – forbidden

	Note: since with this command a network scan is done, this command may require some seconds before the output is given.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.3.5 +CLCK - facility lock/ unlock

+CLCK – facility lock/unlock	
Execute command **AT+CLCK =** **<fac>,** **<mode>[,<passwd>** **[,<class>]]**	It is used to lock or unlock a ME o a network facility. Parameters: <fac>: "SC" - SIM (PIN request) (device asks SIM password at power-up and when this lock command issued) "AO"- BAOC (Barr All Outgoing Calls) "OI" - BOIC (Barr Outgoing International Calls) "OX"- BOIC-exHC (Barr Outgoing International Calls except to Home Country) "AI" - BAIC (Barr All Incoming Calls) "IR" - BIC-Roam (Barr Incoming Calls when Roaming outside the home country) "AB" - All Barring services (applicable only for <mode>=0) "AG" - All outGoing barring services (applicable only for <mode>=0) "AC" - All inComing barring services (applicable only for <mode>=0) "FD" - SIM fixed dialling memory feature (if PIN2 authentication has not been done during the current session, PIN2 is required as <passwd>) "PN" - network Personalisation "PU" - network subset Personalisation <mode>: defines the operation to be done on the facility <mode> = 0 - unlock facility <mode> = 1 - lock facility <mode> = 2 - query status (see below) <passwd>: shall be the same as password specified for the facility from the DTE user interface or with command Change Password +CPWD <class>: represents the class of information of the facility (sum of bits) <class> = 1- voice (telephony) <class> = 2 - data (refers to all bearer services) <class> = 4 - fax (facsimile services) ()
Read command **AT+CLCK=<fac>,2**	query the status of the facility <fac>; returns `+CLCK:<status>` where <status> defines the current status of the facility: 0 - not active (locked) 1 - active (unlocked) Note: For some facility the status is requested to the network, hence if no network is available the command returns the ERROR message.

+CLCK – facility lock/unlock	
Write command	
Test command **AT+CLCK=?**	Reports all the facility supported by the device.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.3.6 +CPWD - change facility password

+CPWD – change facility password	
Execute command **AT+CPWD = <fac>, <oldpwd>, <newpwd>**	Changes the password for the facility lock function of the facility fac. Parameter: <fac>, <oldpwd>, <newpwd> Note: parameter <oldpwd> is the old password while <newpwd> is the new one.
Read command	
Write command	
Test command	
Example	
Reference	GSM 07.07
SW release	Version A

6.4.3.7 +CLIP - calling line identification presentation

+CLIP – calling line identification presentation	
Execute command	
Read command **AT+CLIP?**	Returns the status of the supplementary service CLI in the format: +CLIP: <n>, <m> where: <n> = 0 - CLI presentation disabled <n> = 1 - CLI presentation enabled <m>: status if the CLIP service on the GSM network <m> = 0 - CLIP not provisioned <m> = 1 - CLIP provisioned <m> = 2 - unknown (e.g. no network is present) Note: This command issues a status request to the network, hence it may take a few seconds to give the answer due to the time needed to exchange data with it.
Write command **AT+CLIP = <n>**	Enables or disables the calling line identification supplementary service presentation depending on the value of the parameter <n>. Parameters: <n> **<n> = 0 - disables CLI indication (default)** <n> = 1 - enables CLI indication If enabled the device reports after each RING the response: +CLIP: <number>,<type>[,<subaddress>,<satype>[,<alpha>[,<CLI validity>]]] where: <number>: calling line number <type>: type of address octet in integer format 145 - international numbering scheme (contains the character "+") 129 - national numbering scheme <subaddress>: string type subaddress of format specified by <satype> <satype>: type of subaddress octet in integer format <alpha> : optional string type alphanumeric representation of <number> corresponding to the entry found in phonebook; used character set should be the one selected with command Select TE character set +CSCS <CLI validity>: 0 - CLI valid 1 - CLI has been withheld by the originator. 2 - CLI is not available due to interworking problems or limitation or originating network. Note: The command changes only the report behaviour of the device, it does not change CLI supplementary service setting on the network.

Test command **AT+CLIP=?**	Returns the supported values of the parameter <n>
Example	
Reference	GSM 07.07
SW release	Version A

6.4.3.8 +CLIR - calling line identification restriction

+CLIR – calling line identification restriction	
Execute command	
Read command **AT+CLIR?**	Reports the network & the device CLIR setting in the format: +CLIR:<n>,<m> where <n> is the facility status on the Mobile 0 – CLIR facility according to CLIR service network status 1 – CLIR facility active (CLI not sent) 2 – CLIR facility not active (CLI sent) <m> is the facility status on the Network 0 - CLIR service not provisioned 1 - CLIR service provisioned permanently 2 - unknown (e.g. no network present, etc.) 3 - CLI temporary mode presentation restricted 4 - CLI temporary mode presentation allowed
Write command **AT+CLIR = <n>**	Refers to CLIR – service that allows a calling subscriber to enable or disable the presentation of the CLI to the called party when originating a call according to the parameters <n> given Parameters: <n> <n> is the facility status on the Mobile 0 – CLIR facility according to CLIR service network status 1 – CLIR facility active (CLI not sent) 2 – CLIR facility not active (CLI sent) Note: This command set the default behaviour of the device in outgoing calls.
Test command **AT+CLIR=?**	Reports the supported values of the parameter <n>.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.3.9 +CCFC - call forwarding number and conditions

+CCFC – call forwarding number and condition	
Execute command **AT+CCFC =** **<reason>, <cmd>[,<number>[,<type>[,<class>][,,,<time>]]]**	Controls the call forwarding supplementary service. Registration, erasure, activation, deactivation, and status query are supported. Parameters: <reason> = 0 – unconditional <reason> = 1 - mobile busy <reason> = 2 - no reply <reason> = 3 - not reachable <reason> = 4 - all calls (not with query command) <reason> = 5 - all conditional calls (not with query command) <cmd> = 0 – disable <cmd> = 1 - enable <cmd> = 2 - query status (see below for further explanations) <cmd> = 3 – registration <cmd> = 4 – erasure <number>: phone number of forwarding address in format specified by <type> parameter <type>: type of address byte in integer format : 145 - international numbering scheme (contains the character "+") 129 - national numbering scheme <class>: is a sum of integers each representing a class of information which the command refers; default 7 (voice + data + fax) <class> = 1 - voice (telephony) <class> = 2 - data <class> = 4 - fax (facsimile services, from SW release C) <time>: is the time in seconds after which the call is diverted if "no reply" reason is chosen. Valid only for "no reply" reason.

+CCFC – call forwarding number and condition	
Read command **AT+CCFC = <reason>,2,[number>[,<type>[,<time>]]]**	Queries the network for forwarding service settings on a specific reason. The result is in the form: +CCFC:<status>,<class>[,<number>[,<type>[, <time>]]] where: <status> = 0 - service not active <status> = 1 - service active (calls will be forwarded) <time>: - 1...30 when "no reply" option for reason is enabled or queried, this gives the time in *seconds* to wait before call is forwarded. The other parameters are as seen before. Note: When querying the status of a network service (<cmd>=2) the response line for 'not active' case (<status>=0) should be returned only if service is not active for any <class>.
Write command	
Test command **AT+CCFC=?**	Reports supported values for the parameter <reason>.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.3.10 +CCWA - call waiting

+CCWA – call waiting	
Execute command **AT+CCWA = [<n>[,<cmd>[,<class>]]]**	Controls the call waiting indication supplementary service. Activation, deactivation, and status query are supported. Parameters: <n>: enables/disables call waiting indication reporting: <n> = 0 – disable <n> = 1 - enable <cmd> enables/disables or queries the service at network level: <cmd> = 0 - disable <cmd> = 1 – enable <cmd> = 2 - query status <class>: is a sum of integers each representing a class of information which the command refers; default 7 (voice + data + fax) <class> = 1 - voice (telephony) <class> = 2 - data <class> = 4 - fax (facsimile services) The response to the query command is in the form: +CCWA=<status>,<class> Where <status> represents the status of the service: <status> = 0 - inactive <status> = 1 - active <class> is the class of calls the service status refers to. Note: if parameter <cmd> is omitted then network is not interrogated. Note2: In the query command the class parameter must not be issued. Note3: The difference between call waiting report disabling (AT+CCWA = 0,1,7) and call waiting service disabling (AT+CCWA = 0,0,7) is that in the first case the call waiting indication is sent to the device by network but this last one does not report it to the DTE; instead in the second case the call waiting indication is not generated by the network. Hence the device results busy to the third party in the 2nd case while in the 1st case a ringing indication is sent to the third party. Note4: The command AT+CCWA=1,0 is a non sense and must not be issued.
Read command **AT+CCWA?**	Reports the current value of the parameter <n>.
Write command	
Test command	

+CCWA – call waiting	
AT+CCWA=?	Reports the supported values for the parameter <n>.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.3.11 +CHLD - call holding services

+CHLD – call holding services	
Execute command **AT+CHLD = <n>**	Controls the network call hold service. With this service it is possible to disconnect temporarily a call and keep it suspended while it is retained by the network, contemporary it is possible to connect another party or make a multiparty connection. The action of the command depends on the value of the parameter Parameter: <n> = 0 Releases all pending calls, or sets the UDUB (User Determined User Busy) indication for a waiting call. (Only from version D) <n> = 1 Releases all active calls (if present), and accepts or reconnects the pending or waiting call <n> = 1X Releases a specific active call X <n> = 2 Suspends (into pending status) the active call and accepts or reconnects the waiting or pending call. <n> = 2X Suspends (into pending status) all the active call except call X which will be or remains connected. (Only from version D) <n> = 3 Adds an held call to the conversation "X" is the numbering (starting with 1) of the call given by the sequence of setting up or receiving the calls (active, held or waiting) as seen by the served subscriber. Calls hold their number until they are released. New calls take the lowest available number. Where both a held and a waiting call exist, the above procedures apply to the waiting call (i.e. not to the held call) in conflicting situation. Note: ONLY for VOICE calls
Read command	
Write command	
Test command	
Example	
Reference	GSM 07.07
SW release	Version D

6.4.3.12 +CUSD - unstructured supplementary service data

+CUSD – unstructured supplementary service data	
Execute command **AT+CUSD = <n>,[<str>]**	Enables /disables the unstructured service data reporting. Parameters: <n>: is used to disable/enable the presentation of an unsolicited result code: +CUSD: <m>[,<str>,<dcs>] to the TE. <n>: <n> = 0 - disable the result code presentation in the DTA <n> = 1 - enable the result code presentation in the DTA <dcs> indicates which alphabet is used (Data Coding Scheme),refer to command Select TE Character Set +CSCS <str> - USSD-string (when <str> parameter is not given, network is not interrogated): - if TE character set other than "HEX": ME/TA converts GSM alphabet into current TE character set according to rules of GSM 07.05 - if TE character set is "HEX": ME/TA converts each 7-bit character of GSM alphabet into two IRA character long hexadecimal number (e.g. character Π (GSM 23) is presented as 17 (IRA 49 and 55)) -if<dcs> indicates that 8-bit data coding scheme is used: ME/TA converts each 8-bit octet into two IRA character long hexadecimal number (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65)) <m>: <m> = 0 - no further user action required, hence network initiated USSD-Notify, or no further information needed after mobile initiated operation. <m> = 1 - further user action required, hence network initiated USSD-Request, or further information needed after mobile initiated operation <m> = 2 - USSD terminated by the network <m> = 3 - other local client has responded <m> = 4 - operation not supported <m> = 5 - network time out Note: only mobile initiated operations are supported. Warning: In case of successful mobile initiated operation, DTA waits the USSD response from the network and sends it to the DTE before the final result code. This will block the AT command interface for the period of the operation.

+CUSD – unstructured supplementary service data	
Read command **AT+CUSD?**	Reports the current value of the parameter <n>
Write command	
Test command **AT+CUSD=?**	Reports the supported values for the parameter <n>: <n> = 0 - result code presentation in the TA disabled <n> = 1 - result code presentation in the TA enabled
Example	
Reference	GSM 07.07
SW release	Version A

6.4.3.13 +CAOC - advice of charge

+CAOC – advice of charge	
Execute command **AT+CAOC = <mode>**	Controls the advise of charge service; the command also includes the possibility to enable an unsolicited event reporting of the CCM information. If unsolicited reporting is enabled DTE sends the result code +CCCM: <ccm> when the CCM value changes, but not more than every 10 seconds. <ccm> - call meter value hexadecimal representation (3 byte) Parameter: <mode> = 0 - CCM (Call Meter) read request <mode> = 1 - disables unsolicited CCM reporting <mode> = 2 - enables unsolicited CCM reporting Note: +CAOC command uses the CCM of the device internal memory, not the CCM stored in the SIM. The difference is that the internal memory CCM is reset at power up, while the SIM CCM is reset only on user request. Advice of Charge values stored in the SIM (ACM, ACMmax, PUCT) can be accessed with commands +CACM, +CAMM and +CPUC.
Read command **AT+CAOC?**	Reports the value of parameter <mode> in the format: +CAOC: <mode>
Write command	
Test command **AT+CAOC=?**	Reports the supported values for <mode> parameter.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.3.14 +CLCC - list current calls

+CLCC – list current calls	
Execute command **AT+CLCC**	Reports the list of current calls active and their characteristics by sending the report +CLCC:<id>,<dir>,<stat>,<mode>,<mpty>[,<number>,<type>] for each active call, where: <id> - call identification number <dir> - call direction 0 - mobile originated call 1 - mobile terminated call <stat> - state of the call 0 - active 1 - held 2 - dialing (MO call) 3 - alerting (MO call) 4 - incoming (MT call) 5 - waiting (MT call) <mode> - call type 0 – voice 1 – data 2 – fax 9 - unknown <mpty> - multiparty call flag 0 - call is not one of multiparty (conference) call parties 1 - NOT SUPPORTED <number>: phone number in format specified by <type> <type>: type of phone number byte in integer format 145 - international numbering scheme (contains the character "+") 129 - national numbering scheme Note: If no call is active then only OK message is sent. This command is useful in conjunction with command +CHLD to know the various call status for call holding.
Read command	
Write command	
Test command	
Example	
Reference	GSM 07.07
SW release	Version A

6.4.3.15 +CSSN – SS Notification

+CSSN – SS notification	
Read command **AT+CSSN?**	Reports the current value of the parameters in the format: +CSSN: <n>,<m>
Set command **AT+CSSN = <n>, <m>**	Enable/disable the supplementary service related network initiated notifications, during the origin of outgoing calls (+CSSI) and incoming calls (+CSSU) Parameters: <n>: sets the +CSSI result code presentation status 0 disable 1 enable <m>: sets the +CSSU result code presentation status 0 disable 1 enable When <n>=1 and a supplementary service notification is received after a mobile originated call setup, an unsolicited code +CSSI: <code1> is sent to TE before any other MO call setup result codes. When <m>=1 and a supplementary service notification is received during a mobile terminated call setup or during a call unsolicited result code +CSSU: <code2> is sent to TE. <code1>: 1 some of the conditional call forwardings are active 2 call has been forwarded 3 call is waiting 5 outgoing calls are barred 6 incoming calls are barred <code2>: 0 this is a forwarded call (MT call setup) 2 call has been put on hold (during a voice call) 3 call has been retrieved (during a voice call)
Test command **AT+CSSN=?**	Reports the supported range of values for the parameters <n>,<m>
Example	
Reference	GSM 07.07
Note	

6.4.3.16 +CCUG – Closed User Group supplementary service control

+CCUG – Closed User Group supplementary service control	
Execute command	
Read command **AT+CCUG?**	Reports the current value of the parameters in the format: +CCUG: <n>,<index>,<info>
Write command **AT+CCUG=<n>[,<index>[,<info>]]**	Enable/disable the CUG supplementary service Parameters: <n> 0 - disables the temporary CUG settings for all the successive calls 1 - enables the temporary CUG settings for all the successive calls <index> 0...9 - CUG index 10 - no index (preferential CUG taken from subscriber data) <info> 0 - no information 1 - suppress Outgoing Access (OA) 2 - suppress preferential CUG 3 - suppress OA and preferential CUG
Test command **AT+CCUG=?**	Reports the supported range of values for the parameters <n>,<index>, <info>
Example	
Reference	GSM 07.07
SW release	Version A

6.4.4 Mobile Equipment control

6.4.4.1 +CPAS - phone activity status

+CPAS – phone activity status	
Execute command **AT+CPAS**	reports the device status in the form +CPAS: <pas> where <pas> - phone activity status 0 - ready (Device allows commands from TA/TE) 1 - unavailable (Device does not allow commands from TA/TE) 2 - unknown (Device is not guaranteed to respond to instructions) 3 - ringing (Device is ready for commands from TA/TE, but the ringer is active) 4 - call in progress (Device is ready for commands from TA/TE, but a call is in progress)
Read command **AT+CPAS?**	same as Execute command.
Write command	
Test command **AT+CPAS=?**	Reports the supported range of values of <pas>.
Example	ATD03282131321; OK AT+CPAS? +CPAS: 3 (the called phone is ringing) OK AT+CPAS? +CPAS: 3 OK AT+CPAS? +CPAS: 4 (the called phone has answered to your call) OK ATH OK
Reference	GSM 07.07
SW release	Version A

6.4.4.2 +CFUN Set phone functionality (Power Saving Management)

+CFUN – power saving management	
Execute command	
Read command **AT+CFUN?**	reports the power saving status in the form: +CFUN: <fun> where <fun> - power saving status **1 - power saving disabled (default)** 5 - power saving enabled **Note**: The phone functionality remains always FULL.
Write command **AT+CFUN=<fun>**	Enables/disables the power saving function. If enabled it reduces the power consumption during the idle time, thus allowing a longer standby time with a given battery capacity. Parameters: <fun>: is the power saving function mode 1 - Mobile full functionality with power saving disabled (default) 5 - Mobile full functionality with power saving enabled **Note**: To place the telephone in power saving mode, set the <fun> parameter at value = 5 and the line DTR (RS232) must be set to OFF. Once in power saving, the CTS line switch to the OFF status to signal that the telephone is really in power saving condition. During the power saving condition, before sending any AT command on the serial line, the DTR must be enabled and it must be waited for the CTS (RS232) line to go in ON status. Until the DTR line is ON, the telephone will not return back in the power saving condition. **Note2**: The power saving function does not affect the network behavior of the GM862, even during the power save condition the module remains registered on the network and reachable for incoming calls or SMS. If a call incomes during the power save, then the module will wake up and proceed normally with the unsolicited incoming call code
Test command	
Example	
Reference	GSM 07.07
SW release	Version A

6.4.4.3 +CPIN - enter PIN

+CPIN – enter PIN	
Execute command	
Read command **AT+CPIN?**	Reports the PIN/PUK/PUK2 request status of the device in the form: +CPIN:<code> where <code> is the PIN/PUK/PUK2 request status code: READY - device is not pending for any password SIM PIN - device is waiting SIM PIN to be given SIM PUK - device is waiting SIM PUK to be given SIM PIN2 - device is waiting SIM PIN2 to be given; this <code> is returned only when the last executed command resulted in PIN2 authentication failure (i.e. +CME ERROR: 17) SIM PUK2 - device is waiting SIM PUK2 to be given; this <code> is returned only when the last executed command resulted in PUK2 authentication failure (i.e. +CME ERROR: 18) PH-NET PIN - device is waiting network personalization password to be given PH-NETSUB PIN - device is waiting network subset personalization password to be given Note: Pin pending status at startup depends on PIN facility setting, to change or query the default power up setting use the AT+CLCK=SC,<mode>, <pin> command.
Write command **AT+CPIN = <pin>[,<newpin>]**	Sends to the device a password (SIM PIN, SIM PUK, PH-SIM PIN, etc.) which may be required to further operate the device. If the PIN required is SIM PUK or SIM PUK2, the <newpin> is required. This second pin, <newpin>,will replace the old pin in the SIM. The command may be used to change the SIM PIN by sending it with both parameters <pin> and <newpin> when if PIN request is pending; if no PIN request is pending the command will return an error code and to change the PIN the command +CPWD must be used instead. Parameters: <pin>, <newpin> Note: the only commands which are accepted when device is pending SIM PIN, SIM PUK, or PH-SIM are: +CGMI, +CGMM, +CGMR, +CGSN, D112; (emergency call), +CPAS, +CPIN. To check the status of the PIN request use the command AT+CPIN?
Test command	

+CPIN – enter PIN	
Example	AT+CMEE=1 OK AT+CPIN? +CME ERROR: 10 (error because you have to insert the SIM) AT+CPIN? +CPIN: READY (this response is after you have inserted the SIM) OK
Reference	GSM 07.07
SW release	Version A

6.4.4.4 +CSQ- signal quality

+CSQ – signal quality	
Execute command **AT+CSQ**	Reports received signal quality indicators in the form +CSQ:<rssi>,<ber> where <rssi> - received signal strength indication 0 - 113 dBm or less 1 - 111 dBm 2...30 - 109dBm ... -53 dBm / 2 dBm per step 31 - 51 dBm or greater 99 - not known or not detectable <ber> - bit error rate % 0 - less than 0.2 % 1 - 0.2% a 0.4% 2 - 0.4% a 0.8% 3 - 0.8% a 1.6% 4 - 1.6% a 3.2% 5 - 3.2% a 6.4% 6 - 6.4% a 12.8% 7 - more than 12.8% 99 - not known or not detectable Note: this command should be used instead of the AT%Q and AT%L commands, since GSM relevant parameters are the radio link ones and no line is present, hence %Q %L and have no meaning.
Read command	
Write command	
Test command **AT+CSQ=?**	Returns the supported range of values of the parameters <rssi>and <ber>.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.4.5 +CPBS - select phonebook memory storage

+CPBS – select phonebook memory storage	
Execute command	
Read command **AT+CPBS?**	Returns the actual value of the parameter <storage>.
Write command **AT+CPBS = <storage>**	Selects phonebook memory storage <storage>, which will be used by other phonebook commands. Parameter: <storage> "SM" - SIM phonebook "FD" - SIM fixed dialling-phonebook (only phase 2/2+ SIM) "LD" - SIM last-dialling-phonebook (+CPBW and +CPBF are not applicable for this storage) "MC" - device missed (unanswered received) calls list (+CPBW and +CPBF are not applicable for this storage) "RC" - ME received calls list (+CPBW and +CPBF are not applicable for this storage)
Test command **AT+CPBS=?**	Returns the supported range of values of the parameters <storage>.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.4.6 +CPBR - read phonebook entries

+CPBR – read phonebook entries	
Execute command	
Read command	
Write command **AT+CPBR = <index1>[,<index2>]**	returns phonebook entries in locations number from <index1> to <index2> from the current phonebook memory storage selected with +CPBS. If<index2> is omitted, only location <index1> is returned. Parameters: <index>, <index2> The response format is: +CPBR: <index>,<number>,<type>,<name> where <index> - the current position number of the PB index (to see the range of values use +CPBR=?) <number> - the phone number stored in the format <type> <type> - type of phone number byte in integer format 145 - international numbering scheme (contains the character "+") 129 - national numbering scheme <name> - the alphanumeric text associated to the number (e.g. name of address) Note: If all queried locations are empty (but available), no information text lines will be returned, while if listing fails in an ME error, +CME ERROR: <err> is returned.
Test command **AT+CPBR=?**	Returns the supported range of values of the parameters in the form +CPBR: (<minIndex> - <maxIndex>),<nlength>,<tlength> where <minIndex> - the minimum *index* number <maxIndex>- the maximum *index* number <nlength> - maximum *number* field length <tlength> - maximum *name* field length Note: remember to select the PB storage with +CPBS command before issuing PB commands.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.4.7 +CPBF - find phonebook entries

+CPBF – find phonebook entries	
Execute command	
Read command	
Write command **AT+CPBF = <text>**	Issues a search for the phonebook records that have the *tex*t sub-string at the start of the *name* field and returns a report in the form +CPBF: <index>,<number>,<type>,<name> where index, number, type and name have the same meaning than in the command +CPBR report. If no PB records satisfy the search criteria then an ERROR message is reported. Parameter: <text> Note: text is NOT case sensitive and may or not be included in double brackets.
Test command **AT+CPBF=?**	Reports the maximum lengths of fields <number> and <name> in the PB entry in the form: +CPBF: <max_number_length>,<max_name_length>
Example	
Reference	GSM 07.07
SW release	Version A

6.4.4.8 +CPBW - write phonebook entry

+CPBW – write phonebook entry	
Execute command	
Read command	
Write command **AT+CPBW = [<index>][,<number>[, <type>[,<name>]]]**	Stores at the position <index> a Phonebook record defined by <number>,<type> and <name> parameters Parameters: <index>, <number>, <type>, <name> <number> - the phone number in the format <type> <type> - the type of number 145 - international numbering scheme (contains the character "+") 129 - national numbering scheme <name> - the text associated to the number. Note: If record number <index> already exists, it will be overwritten. If only <index> is given, the record number <index> is deleted. If <index> is omitted, the number <number> is stored in the first free phonebook location. Text in the <name> field and number in the <number> field may or may not be included in double brackets.
Test command **AT+CPBW=?**	Returns the supported range of values of the parameters in the form +CPBW: (<minIndex> - <maxIndex>),<nlength>,<tlength> where <minIndex> - the minimum *index* number <maxIndex>- the maximum *index* number <nlength> - maximum *number* field length <tlength> - maximum *name* field length Note: remember to select the PB storage with +CPBS command before issuing PB commands.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.4.9 +CCLK - Clock Management

+CCLK – Clock Management	
Execute command	
Read command **AT+CCLK?**	Reports the current time stored in the internal Real Time Clock in the format: +CCLK: <time>
Write command **AT+CCLK = <time>**	Stores in the internal Real Time Clock the current time defined by the parameter <time> Parameter: <time> - current time in the format : "yy/MM/dd,hh:mm:ss±zz" where: yy - year MM - month (in digits) dd - day hh - hour mm - minute ±zz - time zone in quarter of an hour (difference between the GMT and the local time) NOTE: time zone is not supported and its value is not relevant, clock will always be set to LOCAL hour.
Test command	
Example	**AT+CCLK="02/09/07,22:30:00+00"** **OK** **AT+CCLK?** **+CCLK: 02/09/07,22:30:25** **OK**
Reference	GSM 07.07
SW release	Version A

6.4.4.10 +CALA - Alarm Management

+CALA – Alarm Management	
Execute command	
Read command **AT+CALA?**	Reports the current alarm time stored in the internal Real Time Clock, if present, in the format: +CALA: <time>,<n>,<type>[,<text>]
Write command **AT+CALA = [<time> [,<n>[,<type>[,<text>]]]]**	Stores in the internal Real Time Clock the current alarm time and settings defined by the parameters <time>, <n>,<type>,<text>. When the RTC time reaches the alarm time then the alarm starts, the behaviour of the GM862 depends upon the setting <type> and if the device was already ON at the moment when the alarm time had come. Parameter: <time> - current alarm time in the format : "yy/MM/dd,hh:mm:ss±zz" as defined for +CCLK command. <type> - alarm behaviour type 0 - reserved for other equipment use. 1 - the GM862 simply wakes up fully operative as if the ON/OFF pin had been pressed. If the device is already ON at the alarm time, then it does nothing. 2 - the GM862 wakes up in "alarm mode" if at the alarm time it was off, otherwise it remains fully operative. In both cases the GM862-GPRS issues an unsolicited code every 3s: +ALARM: <text> Where <text> is the other +CALA parameter previously set. The device keeps on sending the unsolicited code every 3s until a #WAKE or #SHDN command is received or a 90s timeout occurs. If the device is in "alarm mode" and it does not receive the #WAKE command within 90s then it shuts down. (default) 3 - the GM862 wakes up in "alarm mode" if at the alarm time it was off, otherwise it remains fully operative. In both cases the GM862 starts playing the alarm tone on the selected path for the ringer (see command #SRP) The device keeps on playing the alarm tone until a #WAKE or #SHDN command is received or a 90s timeout occurs. If the device is in "alarm mode" and it does not receive the #WAKE command within 90s then it shuts down. 4 - the GM862 wakes up in "alarm mode" if at the alarm time it was off, otherwise it remains fully operative. In both cases the GM862

	brings the pin GPIO6 high, provided its <direction> has been set to alarm output, and keeps it in this state until a #WAKE or #SHDN command is received or a 90s timeout occurs. If the device is in "alarm mode" and it does not receive the #WAKE command within 90s then it shuts down. 5 - the GM862 will make both the actions as for type=2 and type=3. 6 - the GM862 will make both the actions as for type=2 and type=4. 7 - the GM862 will make both the actions as for type=3 and type=4. <n> = 0 - index of the alarm The only value supported for <n> is 0. <text> - unsolicited alarm code text string In the case that <type> is equal to 2 or 5 or 6, then the unsolicited text parameter <text> can be set by this parameter. **Note:** The "alarm mode" is indicated by hardware pin CTS to the ON status and DSR to the OFF status, while the "power saving" status is indicated by a CTS - OFF and DSR - OFF status. The normal operating status is indicated by DSR - ON. During the "alarm mode" the device will not make any network scan and will not register to any network and therefore is not able to dial or receive any call or SMS, the only commands that can be issued to the GM862 in this state are the #WAKE and #SHDN, every other command must not be issued during this state. NOTE: time zone is not supported and its value is not relevant, alarm will always be referred to LOCAL hour.
Test command **AT+CALA=?**	Reports the supported range of values for the parameters <time>,<n>, <type> and <text> maximum length
Example	**AT+CALA="02/09/07,23:30:00+00"** **OK**
Reference	GSM 07.07
SW release	Version A

6.4.4.11 +CALM - alert sound mode

+CALM – alert sound mode	
Execute command	
Read command	
Write command **AT+CALM = <mode>**	It is used to select the general alert sound mode of the device. Parameter: <mode> = 0 - normal mode <mode> = 1 - silent mode (no sound will be generated by the device) Note: if silent mode (1) is selected then incoming calls will not produce alerting sounds but only the message RING or +CRING.
Test command **AT+CALM=?**	Returns the supported values for the parameter <mode>.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.4.12 +CRSL - ringer sound level

+CRSL – ringer sound level	
Execute command	
Read command **AT+CRSL?**	Reports the current <level> setting of the call ringer in the format: +CRSL: <level>
Write command **AT+CRSL = <level>**	It used to select the incoming call ringer sound level of the device. Parameter: <level> - ringer sound level <level> = 0 - Off <level> = 1 - low <level> = 2 - middle <level> = 3 - high <level> = 4 – progressive
Test command **AT+CRSL=?**	Reports <level> supported values range in the format: +CRSL: (0-4)
Example	
Reference	GSM 07.07
SW release	Version A

6.4.4.13 +CLVL - loudspeaker volume level

+CLVL – loudspeaker volume level	
Execute command	
Read command **AT+CLVL?**	Reports the current <level> setting of the loudspeaker volume in the format: +CLVL: <level>
Write command **AT+CLVL = <level>**	It used to select the volume of the internal loudspeaker audio output of the device. Parameter: <level> - loudspeaker volume varies between 0 (minimum) and a maximum level that can be read by issuing the test command AT+CLVL=?
Test command **AT+CLVL=?**	Reports <level> supported values range in the format: +CLVL: (0-maximum value)
Example	
Reference	GSM 07.07
SW release	Version A

6.4.4.14 +CMUT - microphone mute control

+CMUT – microphone mute control	
Execute command	
Read command **AT+CMUT?**	Reports the selected microphone (internal/external) mute status in the format: +CMUT: <n>
Write command **AT+CMUT = <n>**	Enables/disables the muting of the microphone audio line depending on n parameter value Parameter: <n > = 0 - microphone active <n > = 1 - microphone muted Note: this command mutes/activates both microphone audio paths, internal mic and external mic.
Test command **AT+CMUT=?**	Reports the supported values for <n> parameter.
Example	
Reference	GSM 07.07
SW release	Version A

6.4.4.15 +CACM - accumulated call meter

+CACM – accumulated call meter	
Execute command	
Read command **AT+CACM?**	Reports the current value of the SIM ACM in the format +CACM: <n> Note: the value <n> is in units whose price and currency is defined with command +CPUC
Write command **AT+CACM = <pwd>**	Resets the SIM Accumulated Call Meter (ACM). Internal memory CCM remains unchanged. Parameter: <pwd> - PIN2 password Note: to access this command PIN2 password is required
Test command	
Example	
Reference	GSM 07.07
SW release	Version A

6.4.4.16 +CAMM - accumulated call meter maximum

+CAMM – accumulated call meter maximum	
Execute command	
Read command **AT+CAMM?**	Reports the maximum value of ACM stored in SIM in the format: +CAMM : <acmmax>
Write command **AT+CAMM = <acmmax>, <pwd>**	Sets the Advice of Charge related Accumulated Call Meter ACM maximum value in SIM (see also +CACM command). This value represents the maximum number of home units allowed to be consumed by the subscriber. When ACM reaches <acmmax> value further calls are prohibited. SIM PIN2 is required to set the value. Parameter: <acmmax>- maximum number of units allowed to be consumed <pwd>- PIN2 password Note: The <acmmax> = 0 value disables the feature.
Test command	
Example	
Reference	GSM 07.07
SW release	Version A

6.4.4.17 +CPUC - price per unit and currency table

+CPUC – price per unit and currency table	
Execute command	
Read command **AT+CPUC?**	Reports the current values of <currency> and <ppu> parameters in the format: +CACM : <currency>,<ppu>
Write command **AT+CPUC = <currency>, <ppu>,<pwd>**	Sets the values of Advice of Charge related price per unit and currency table in SIM. The price per Unit currency table information can be used to convert the home units (as used in commands +CAOC, +CACM and +CAMM) into currency units. Parameters: <currency> - three-character currency code string (e.g. LIT, USD, DEM etc..) <ppu> - price per unit string (dot is used as decimal separator) e.g. 1989.27 <pwd> - SIM PIN2 Note: SIM PIN2 is required to set the values.
Test command	
Example	
Reference	GSM 07.07
SW release	Version A

6.4.5 Mobile equipment errors

6.4.5.1 +CMEE - report mobile equipment error

+CMEE – report mobile equipment error	
Execute command	
Read command	
Write command **AT+CMEE = <n>**	Enables/disables the report of result code. +CME ERROR: <err> as an indication of an error relating to the +Cxxx commands issued. When enabled, device related errors cause the +CME ERROR: <err> final result code instead of the default ERROR final result code. ERROR is anyway returned normally when the error message is related to syntax, invalid parameters, or DTE functionality. Parameter: <n> - enable flag <n> = 0 - disable +CME ERROR:<err> reports, use only ERROR report. <n> = 1 - enable +CME ERROR:<err> reports, with <err> in numeric format <n> = 2 - enable +CME ERROR: <err> reports, with <err> in verbose format
Test command	
Example	
Reference	GSM 07.07
SW release	Version A

6.4.5.2 +CME ERROR: - ME error result code

This is NOT a command, it is the error response to +Cxxx GSM 07.07 commands.

Syntax: AT+CME ERROR:<err>

Parameter: <err> - error code can be either numeric or verbose.

<err> values:

Numeric Format	*Verbose Format*
0	phone failure
1	no connection to phone
2	phone-adaptor link reserved
3	operation not allowed
4	operation not supported
5	PH-SIM PIN required
6	NOT SUPPORTED
7	NOT SUPPORTED
10	SIM not inserted
11	SIM PIN required
12	SIM PUK required
13	SIM failure
14	SIM busy
15	SIM wrong
16	incorrect password
17	SIM PIN2 required
18	SIM PUK2 required
20	memory full
21	invalid index
22	not found
23	memory failure
24	text string too long
25	invalid characters in text string
26	dial string too long
27	invalid characters in dial string
30	no network service
31	network timeout
32	network not allowed - emergency calls only
40	network personalization PIN required
41	network personalization PUK required
42	network subset personalization PIN required
43	network subset personalization PUK required
44	service provider personalization PIN required
45	service provider personalization PUK required
46	corporate personalization PIN required
47	corporate personalization PUK required
100	unknown

GPRS related errors to a failure to perform an Attach:

Numeric Format	*Verbose Format*
103	Illegal MS (#3)
106	Illegal ME (#6)
107	GPRS service not allowed (#7)
111	PLMN not allowed (#11)
112	Location area not allowed (#12)
113	Roaming not allowed in this location area (#13)

GPRS related errors to a failure to Activate a Context:

Numeric Format	*Verbose Format*
132	service option not supported (#32)
133	requested service option not subscribed (#33)
134	service option temporarily out of order (#34)
149	PDP authentication failure

(values in parentheses are GSM 04.08 cause codes)

Easy-GPRS related errors:

Numeric Format	*Verbose Format*
400	generic undocumented error
401	wrong state
402	wrong mode
403	context already activated
404	stack already active
405	activation failed
406	context not opened
407	cannot setup socket
408	cannot resolve DN
409	timeout in opening socket
410	cannot open socket
411	remote disconnected or timeout
412	connection failed

Easy CAMERA related errors:

Numeric Format	*Verbose Format*
50	Camera not foud

6.4.6 Voice Control (TIA IS-101)

6.4.6.1 +VTS: - DTMF tones transmission

+VTS – DTMF tones transmission	
Execute command	
Read command	
Write command **AT+VTS = <DTMF>**	Allows the transmission of DTMF tones Parameter: <DTMF> - a single ASCII character in the set 0-9, #,*,A-D Note: this commands operates in voice mode only
Test command **AT+VTS=?**	Returns +VTS(),(),()
Example	
Reference	GSM 07.07 / TIA IS-101
SW release	Version A

6.4.7 Commands For GPRS

6.4.7.1 +CGACT - PDP context activate or deactivate

+ CGACT – PDP context activate or deactivate	
Execute command	
Read command **AT+CGACT?**	Returns the current activation states for all the defined PDP contexts.
Write command **AT+CGACT = <state>[, <cid>[, <cid>[,…]]]]**	The command is used to activate or deactivate the specified PDP context (s) Parameters: <state> - indicates the state of PDP context activation 0 - deactivated 1 - activated <cid> - a numeric parameter which specifies a particular PDP context definition (see +CGDCONT command)
Test command **AT+CGACT=?**	Reports information on the supported PDP context activation states parameters in the format: +CGACT: (0,1)
Example	AT+CGACT? +CGACT: 1,1 OK AT+CGACT=1,1 OK
Reference	GSM 07.07
SW release	Version A

6.4.7.2 +CGATT - GPRS attach or detach

+ CGATT – GPRS attach or detach	
Execute command **AT+CGATT = <state>**	The command is used to attach the terminal to, or detach the terminal from, the GPRS service depending on the parameter <state>. Parameter: <state> - indicates the state of GPRS attachment 0 - detached 1 - attached
Read command **AT+CGATT?**	Returns the current GPRS service state.
Write command	
Test command **AT+CGATT=?**	Requests information on the supported GPRS service states.
Example	AT+CGATT? +CGATT: 0 OK AT+CGATT=? +CGATT: (0,1) OK AT+CGATT=1 OK
Reference	GSM 07.07
SW release	Version A

6.4.7.3 +CGDATA - Enter data state

+ CGDATA – Enter data state	
Execute command **AT+CGDATA = [<L2P> ,[<cid> [,<cid> [,…]]]]**	The command causes to perform whatever actions are necessary to establish communication with the network using one or more GPRS PDP types. Parameters: <L2P> - string parameter that indicates the layer 2 protocol to be used "PPP" - PPP Point-to-point protocol If the value is omitted, the layer 2 protocol is unspecified <cid> - numeric parameter which specifies a particular PDP context definition (see +CGDCONT command).
Read command	
Write command	
Test command **AT+CGDATA=?**	Requests information on the supported layer 2 protocols. Reports the supported <L2P> parameters values
Example	AT+CGDATA=? +CGDATA: (”PPP”) OK AT+CGDATA=”PPP”,1 OK
Reference	GSM 07.07
SW release	Version A

6.4.7.4 +CGDCONT - define PDP context

+ CGDCONT – define PDP context	
Execute command	
Read command **AT+CGDCONT?**	Returns the current settings for each defined context in the format: +CGDCONT : <cid>,<PDP_type>,<APN>,<PDP_addr>, <d_comp>,<h_comp>,<pd1> [,…[,pdN]]]]]]]]]
Write command **AT+CGDCONT = [<cid>[,<PDP_type>[,<APN>[,<PDP_addr>,<d_comp>[,<h_comp>[,<pd1>[,…[,pdN]]]]]]]]**	The command specifies PDP context parameter values for a PDP context identified by the (local) context identification parameter, <cid> Parameters: <cid> (PDP Context Identifier) : a numeric parameter which specifies a particular PDP context definition. The range of permitted values (minimum value = 1) is returned by the test form of the command. <PDP_type> (Packet Data Protocol type) : a string parameter which specifies the type of packet data protocol "IP" - Internet Protocol "PPP" - Point to Point Protocol <APN> (Access Point Name) : a string parameter which is a logical name that is used to select the GGSN or the external packet data network. If the value is null or omitted, then the subscription value will be requested. <PDP_address>: a string parameter that identifies the terminal in the address space applicable to the PDP. The allocated address may be read using the +CGPADDR command. <d_comp>: a numeric parameter that controls PDP data compression 0 - off (default if value is omitted) 1 - on <h_comp>: a numeric parameter that controls PDP header compression 0 - off (default if value is omitted) 1 - on <pd1>, … <pdN>: zero to N string parameters whose meanings are specific to the <PDP_type>
Test command **AT+CGDCONT=?**	Returns values supported as a compound value
Example	AT+CGDCONT? +CGDCONT: 1,"IP","uni.tim.it","10.10.10.10",0,0 OK

	AT+CGDCONT=? +CGDCONT: (1-3),"IP",,,(0-1),(0-1) OK AT+CGDCONT=1, "IP", "uni.tim.it", "10.10.10.10",0,0 OK
Reference	GSM 07.07
SW release	Version A

6.4.7.5 +CGPADDR - show PDP address

+ CGPADDR – show PDP address	
Execute command	
Read command	
Write command **AT+CGPADDR = [<cid> [,<cid> [,…]]]**	Returns a list of PDP addresses for the specified context identifiers in the format: +CGPADDR: <cid>,<PDP_addr> [<CR><LF>+CGPADDR: <cid>,<PDP_addr>] Parameters: <cid> - a numeric parameter which specifies a particular PDP context definition (see +CGDCONT command). If no <cid> is specified, the addresses for all defined contexts are returned. <PDP_address> – a string that identifies the terminal in the address space applicable to the PDP. The address may be static or dynamic. For a static address, it will be the one set by the +CGDCONT command when the context was defined. For a dynamic address it will be the one assigned during the last PDP context activation that used the context definition referred to by <cid>,<PDP_addr> is omitted if none is available
Test command **AT+CGPADDR=?**	Returns a list of defined <cid>s.
Example	AT+CGPADDR=1,2 +CGPADDR = 1,"10.10.15.15" +CGPADDR = 2,"10.10.20.10" OK AT+CGPADDR=? +CGPADDR: (1-3) OK
Reference	GSM 07.07
SW release	Version A

6.4.7.6 +CGREG - GPRS network registration status

+ CGREG – GPRS network registration status	
Execute command	
Read command **AT+CGREG?**	Returns the status of result code presentation mode <n> and the integer <stat> which shows whether the network has currently indicated the registration of the terminal in the format: +CGREG:<n>,<stat>.
Write command **AT+CGREG = <n>**	The command controls the presentation of an unsolicited result code +CGREG: <stat> when <n>=1 and there is a change in the terminal GPRS network registration status, or an unsolicited code +CGREG: <stat>[,<lac>,<ci>] when <n>=2 and there is a change of the network cell. Parameter: <n> - result code presentation mode 0 - disable network registration unsolicited result code 1 - enable network registration unsolicited result code +CGREG: <stat> 2 - enable network registration and location information unsolicited result code +CGREG:<stat>[,<lac>,<ci>] where: <lac> - location area code in hexadecimal format (e.g. "00C3" equals 195 in decimal) <ci> - cell ID in hexadecimal format <stat> - registration status 0 - not registered, terminal is not currently searching a new operator to register to 1 - registered, home network 2 - not registered, but terminal is currently searching a new operator to register to 3 - registration denied 4 - unknown 5 - registered, roaming
Test command **AT+CGREG=?**	Returns supported values for parameter <n>
Example	
Reference	GSM 07.07
SW release	Future Software Release

6.4.7.7 +CGQMIN - quality of service profile (minimum acceptable)

+ CGQMIN – quality of service profile (minimum acceptable)	
Execute command	
Read command **AT+CGQMIN?**	Returns the current settings for each defined context in the format: +CGQMIN: <cid>,<precedence>,<delay>,<reliability>,<peak>,<mean>
Example	
Write command **AT+CGQMIN = [<cid> [,<precedence > [,<delay> [,<reliability> [,<peak> [,<mean>]]]]]]**	This command allows to specify a minimum acceptable profile which is checked by the terminal against the negotiated profile returned in the Activate PDP Context Accept message Parameters: <cid> - PDP context identification (see +CGDCONT command). The following parameters are defined according to GSM 03.60: <precedence> - precedence class <delay> - delay class <reliability> - reliability class <peak> - peak throughput class <mean> - mean throughput class If a value is omitted for a particular class then this class is not checked.
Test command **AT+CGQMIN=?**	Returns values supported as a compound value.
Example	AT+CGQMIN=1,0,0,3,0,0 OK AT+CGQMIN? +CGQMIN: 1,0,0,5,0,0 OK AT+CGQMIN=? +CGQMIN: (0-3),(0-3),(0-5),(0-9),(0-19,31) OK
Reference	GSM 07.07
SW release	Version A

6.4.7.8 +CGQREQ - quality of service profile (requested)

+ CGQREQ – quality of service profile (requested)	
Execute command	
Read command **AT+CGQREQ?**	Returns the current settings for each defined context in the format: +CGQREQ: <cid>,<precedence>,<delay>,<reliability>,<peak>,<mean>
Write command **AT+CGQREQ = [<cid> [,<precedence > [,<delay> [,<reliability> [,<peak> [,<mean>]]]]]]**	The command allows to specify a Quality of Service Profile that is used when the terminal sends an Activate PDP Context Request message to the network. It specifies a profile for the context identified by the (local) context identification parameter, <cid>. Parameters: <cid> - PDP context identification (see +CGDCONT command). The following parameters are defined according to GSM 03.60: <precedence> - precedence class <delay> - delay class <reliability> - reliability class <peak> - peak throughput class <mean> - mean throughput class If a value is omitted for a particular class then this class is not checked.
Test command **AT+CGQREQ=?**	Returns parameter values supported as a compound value.
Example	AT+CGQREQ? +CGQREQ: 1,0,0,3,0,0 OK AT+CGQREQ=1,0,0,3,0,0 OK AT+CGQREQ=? +CGQREQ: (0-3),(0-3),(0-5),(0-9),(0-19,31) OK
Reference	GSM 07.07
SW release	Version A

6.4.8 Commands For Battery Charger

6.4.8.1 +CBC - Battery Charge

+ CBC – Battery Charge	
Execute command	
Read command **AT+CBC**	Returns the current Battery Charge status in the format: +CBC: <bcs>,<bcl> Parameter: <bcs> - battery status 0 - Me is powered by the battery 1 - ME has a battery connected, and charger pin is being powered 2 - ME does not have a battery connected 3 - Recognized power fault, calls inhibited <bcl> - battery charge level 0 - battery is exhausted, or ME does not have a battery connected 25 - battery charge remained is estimated to be 25% 50 - battery charge remained is estimated to be 50% 75 - battery charge remained is estimated to be 75% 100 - battery is fully charged. NOTE: <bcs>=1 indicates that the battery charger supply is inserted and the battery is being recharged if necessary with it. Supply for ME operations is taken anyway from VBATT pins. NOTE: without battery/power connected on VBATT pins or during a power fault the unit is not working, therefore values 3 and 4 shold never appear.
Write command	
Test command **AT+CBC=?**	Returns parameter values supported as a compound value: +CBC: (0-3),(0-100)
Example	AT+CBC +CBC: 0,75 OK
NOTE	The ME does not make differences between being powered by a battery or by a power supply on the VBATT pins, so it is not possible to distinguish between these two cases.
Reference	GSM 07.07
SW release	Version A

6.5 ETSI GSM 07.05 AT Commands for SMS and CB services

6.5.1 General configuration

6.5.1.1 +CSMS - select message service

+CSMS – select message service	
Execute command	
Read command **AT+CSMS?**	Reports the value of the current parameters in the format: +CSMS: <service>,<mt>,<mo>,<cb> where: <service> - message syntax 0 - SMS syntax as defined in GSM-03.40 and GSM-03.41. AT commands are compatible with GSM 07.05 Phase 2. 1 - SMS syntax as defined in GSM-03.40 and GSM-03.41. AT commands are compatible with GSM 07.05 Phase 2+ <mt> - Mobile Terminated Messages 0 - not supported 1 - supported <mo> - Mobile Originated Messages 0 - not supported 1 - supported <cb> - Cell Broadcast Messages 0 - not supported 1 - supported
Write command **AT+CSMS = <service>**	Selects the SMS & CB services command syntax. Parameter: <service> = 0 - SMS syntax as defined in GSM-03.40 and GSM-03.41. AT commands are compatible with GSM 07.05 Phase 2. <service> = 1 - SMS syntax as defined in GSM-03.40 and GSM-03.41. AT commands are compatible with GSM 07.05 Phase 2+
Test command **AT+CSMS=?**	Reports the supported value of the parameter <service> = 0.
Example	
Reference	GSM 07.05
SW release	Version A

6.5.1.2 +CPMS - preferred message storage

+CPMS – preferred message storage	
Execute command	
Read command **AT+CPMS?**	Reports the message storage status in the format: +CPMS:<memr>,<usedr>,<totalr>,<memw>,<usedw>,<totalw>,<mems>,<useds>,<totals> where <memr> , <memw> and <mems> are the selected storage memories for reading, writing and storing respectively.
Write command **AT+CPMS = <memr>,<memw>,<mems>**	Selects memory storages <memr>, <memw> and <mems> to be used for reading, writing, sending and storing received SMS. Parameter: <memr> - memory storage for read and delete SMS commands "SM" - SIM SMS memory storage "ME" - Mobile Equipment internal storage (read only, no delete) <memw> - memory storage for write and send SMS commands "SM" - SIM SMS memory storage <mems> - memory storage for received SMS storing "SM" - SIM SMS memory storage The command returns the memory storage status in the format: +CPMS:<usedr>,<totalr>,<usedw>,<totalw>,<useds>,<totals> <usedr> - number of SMS stored into <memr> <totalr> - max number of SMS that <memr> can contain <usedw> - number of SMS stored into <memw> <totalw> max number of SMS that <memw> can contain <useds> - number of SMS stored into <mems> <totals> max number of SMS that <mems> can contain Note: The only memory storage for writing and sending supported is the SIM internal memory "SM", so <memw> = <mems> = "SM". Note: the received class 0 SMS are stored in the "ME" memory regardless the <mems> setting and they are automatically deleted at power off.
Test command **AT+CPMS=?**	Reports the supported values of the SMS storage memories in the format: +CPMS: ("ME","SM"), ("SM"),("SM")
Example	AT+CPMS? +CPMS: "SM",5,10,"SM",5,10,"SM",5,10 OK (you have 5 SMS SIM positions occupied of 10)
Reference	GSM 07.05
SW release	Version A

6.5.1.3 +CMGF - message format

+CMGF –message format	
Execute command	
Read command **AT+CMGF?**	Reports the current value of the parameter <mode> = 0.
Write command **AT+CMGF = <mode>**	Selects the SMS format to be used in reading and writing messages. Parameter: <mode> = 0 - PDU as defined in GSM 3.40 and GSM 3.41 <mode> = 1 - text
Test command **AT+CMGF=?**	Reports the supported value of <mode> parameter.
Example	
Reference	GSM 07.05
SW release	Version A

6.5.1.4 +CSMP – Set parameters in text mode

+CSMP –set parameters in text mode	
Execute command	
Read command **AT+CSMP?**	Reports the current setting in the format: +CSMP: < fo>,<vp>,<pid>,<dcs>
Write command **AT+CSMP = <fo>,<vp>,<pid>, <dcs>**	Set the additional parameters for storing and sending SMS when the text mode is used (+CMGF=1) Parameter: <fo>: message format, like defined for the first octet of message according to GSM 3.40 <vp>: Message validity period - numerical if in relative format or string if in absolute format according to GMS 3.40 <pid>: Protocol Identifier – defined by GSM 3.40 – in numerical format <dcs>: Data coding Scheme - defined by GSM 3.40 – in numerical format
Test command **AT+CSMP=?**	Reports the supported range of values for <fo>,<vp>,<pid>,<dcs> parameters.
Example	Set the parameters for an outgoing message with 24 hours of validity period and default properties: AT+CSMP=17,167,0,0 OK
Reference	GSM 07.05
SW release	Version A

6.5.1.5 +CSDH – Show parameters in text mode

+CSDH –show parameters in text mode	
Execute command	
Read command **AT+CSDH?**	Report the current setting in the format: +CSDH: <show>
Write command **AT+CSDH= <show>**	Show the additional parameters of received or stored SMS when the text mode is used (+CMFG=1). Parameter: <show> = 0 - no additional parameters are shown <show> = 1 - parameters set by the +CSCA and + CSMP commands are shown and the length, sender/addressee address of SMS when +CMT, +CMGL, +CMGR commands are utilised
Test command **AT+CSDH=?**	Reports the supported range of values for the parameter <show>
Example	
Reference	GSM 07.05
SW release	Version A

6.5.1.6 +CSAS – Save setting text mode

+CSAS –set saving text mode	
Execute command	
Read command	
Write command **AT+CSAS=<profile>**	Save setting which have been made by the +CSCA and +CSMP commands in local volatile memory (or in the SIM if it has a dedicated storage for this purpose). Parameter: <profile>: 0…3 - number of profile in which to save the setting. The number 0 corresponds to the non volatile memory, the others correspond to the SIM storage (if available).
Test command **AT+CSAS=?**	Return the possible range of values for the parameter <profile>.
Example	
Reference	GSM 07.05
SW release	Version A

6.5.1.7 +CRES – Restore text mode settings

+CRES –restore text mode settings	
Execute command	
Read command	
Write command **AT+CRES = <profile>**	Restore any setting saved by +CSAS command for +CSCA and +CSMP commands in local non volatile memory (or in the SIM if it has a dedicated store for this purpose). Parameter: <profile>: 0…3 – number of profile from which to recall the setting. The number 0 corresponds to the non volatile memory, the others correspond to the SIM storage (if available).
Test command **AT+CRES=?**	Return the possible range of values for the parameter <profile>.
Example	
Reference	GSM 07.05
SW release	Version A

6.5.1.8 +CSCB – Select Cell Broadcast Message types

+CSCB –select Cell Broadcast Message types	
Execute command	
Read command **AT+CSCB?**	Reports the current value of the parameters <mode>,<misd> and <dcss> in the format: +CSCB: <mode>,<mids>,<dcss>
Write command **AT+CSCB=<mode>[,<mids>[,<dcss>]]**	Selects the Cell Broadcast message types to be received by the device. Parameter: <mode> 0 - the message types defined by <mids> and <dcss> are accepted 1 - the message types defined by <mids> and <dcss> are rejected <mids> : all the possible combinations of the CBM identifiers <dcss> : all the possible combinations of CBM data coding schemes.
Test command **AT+CSCB=?**	Return the possible range of values for the parameter <mode>.
Example	AT+CSCB? +CSCB: 1,"","" *(all CBMs are accepted, none is rejected)* OK AT+CSCB=0,"0,1,300-315,450","0-3" OK
Reference	GSM 07.05
SW release	Version A

6.5.1.9 +CMS ERROR - message service failure result code

This is NOT a command, it is the error response to +Cxxx GSM 07.05 commands

Syntax: AT+CMS ERROR:<err>

Parameter: <err> - error code can be either numeric or verbose.

<err> values:

Numeric Format	*Verbose Format*
0...127	GSM 04.11 Annex E-2 values
128...255	GSM 03.40 sub clause 9.2.3.22 values
300	ME failure
301	SMS service of ME reserved
302	operation not allowed
303	operation not supported
304	invalid PDU mode parameter
305	invalid text mode parameter
310	SIM not inserted
311	SIM PIN required
312	PH-SIM PIN required
313	SIM failure
314	SIM busy
315	SIM wrong
316	SIM PUK required
317	SIM PIN2 required
318	SIM PUK2 required
320	memory failure
321	invalid memory index
322	memory full
330	SMSC address unknown
331	no network service
332	network timeout
340	no +CNMA acknowledgement expected
500	unknown error

6.5.2 Message configuration

6.5.2.1 +CSCA - service center address

+CSCA –service center address	
Execute command	
Read command **AT+CSCA?**	Reports the current value of the default SCA in the format: +CSCA: <number>,<type> Note: if SCA is not present the device reports an error message.
Write command **AT+CSCA=<number>, <type>**	Sets the Service center Address to be used for mobile originated SMS transmissions. Parameter: <number> - SC phone number in the format defined by <type> <type> - the type of number <type> = 145 - international numbering scheme (contains the character "+") <type> = 129 - national numbering scheme Note: to use the SMS service, is mandatory to set a Service Center Address at which service requests will be directed. SCA depends on the operator, hence contact your operator to set the right SCA. When this address is input, it is stored on the SIM if possible, else will be stored on the device internal memory so, once set, the input of this parameter is not anymore needed. In PDU mode, this setting is used, but only when the length of the SMSC address coded into the <pdu> parameter equals zero; else SCA defined in the PDU will be used instead.
Test command	
Example	
Reference	GSM 07.05
SW release	Version A

6.5.3 Message receiving and reading

6.5.3.1 +CNMI - new message indications to Terminal Equipment

+CNMI – new message indications to terminal equipment	
Read command **AT+CNMI?**	Returns the current parameter settings for +CNMI command in the form: `AT+CNMI:<mode>,<mt>,<bm>,<ds>,<bfr>`
Write command **AT+CNMI=<mode>[,<mt>[,<bm>[,<ds>[,<bfr>]]]]**	Selects the behavior of the device on how the receiving of new messages from the network is indicated to the DTE. Parameter: <mode>- unsolicited result codes buffering option 0 - Buffer unsolicited result codes in the TA. If TA result code buffer is full, indications can be buffered in some other place or the oldest indications may be discarded and replaced with the new received indications. 1 - Discard indication and reject new received message unsolicited result codes when TA-TE link is reserved, otherwise forward them to the TE. 2 - Buffer unsolicited result codes in the TA in case the DTE is busy and flush them to the TE after reservation. Otherwise forward them directly to the TE. <mt> result code indication reporting for SMS-DELIVER 0 - No SMS-DELIVER indications are reported to the TE. 1 - If SMS-DELIVER is stored into ME/TA, indication of the memory location is reported to the TE using unsolicited result code: +CMTI: <memr>,<index> where: <memr> - memory storage where the new message is stored ("SM" or "ME") <index> - location on the memory where SMS is stored 2 - SMS-DELIVERs (except class 2 messages and messages in the message waiting indication group) are reported to the TE using unsolicited result code: +CMT: [<alpha>],<length><CR><LF><pdu>(PDU mode) where: <alpha> - alphanumeric representation of originator/destination number corresponding to the entry found in MT phonebook. <length> - PDU length <pdu> - PDU message

+CNMI – new message indications to terminal equipment	
	or +CMT:<oa>,[<alpha>],<scts>*[,<tooa>,<fo>,<pid>,<dsc>,<sca>,<tosca>,<length>]*<CR><LF><data> (text mode enabled; about parameters in italic, refer command +CSDH) where: <oa> - originator address number <alpha> - alphanumeric representation of <oa> or <da> <scts> - arrival time of the message to the SC <tooa>, <tosca> - type of number <oa> or <sca> 145 - number in international format (contains the "+") 129 - number in national format <fo> - first octet of GSM 03.40 <pid> - Protocol Identifier <dsc> - Data Coding Scheme <sca> - Service Centre number <length> - text length <data>- text mode Class 2 messages and messages in the message waiting indication group (stored message) result in indication as defined in <mt> = 1. 3 - Class 3 SMS-DELIVERs are reported directly to TE using unsolicited result codes defined in <mt> = 2. Messages of other data coding schemes result in indication as defined in <mt> = 1. <bm> - broadcast reporting option 0 - Cell Broadcast Messages are not sent to the DTE 2 - New Cell Broadcast Messages are sent to the DTE with the unsolicited result code: +CBM: <length><CR><LF><PDU> (in PDU mode) or +CBM:<sn>,<mid>,<dcs>,<pag>,<pags><CR><LF><text> (in text mode) where: <length> - PDU length <PDU> - message PDU <sn> - message serial number <mid> - message ID <dcs> - Data Coding Scheme <pag> - page number <pags> - total number of pages of the message <text> - message text

+CNMI – new message indications to terminal equipment	
	<ds> - SMS-STATUS-REPORTs reporting option 0 - status report receiving is not reported to the DTE 1 - the status report is sent to the DTE with the unsolicited result code: +CDS: <length><CR><LF><PDU> (PDU mode) or +CDS: <fo>,<mr>,,,<scts>,<dt>,<st> (text mode) 2 - if a status report is stored, then unsolicited result code is sent: +CDSI: <memr>,<index> where <memr> - memory storage where the new message is stored ("SM") <index> - location on the memory where SMS is stored <length> - PDU length <PDU> - message PDU <fo> - first octet of the message PDU <mr> - message reference number <scts> - arrival time of the message to the SC <dt> - sending time of the message <st> - message status as coded in the PDU <bfr> - buffered result codes handling method 0 - TA buffer of unsolicited result codes defined within this command is flushed to the TE when <mode> 1…3 is entered (OK response shall be given before flushing the codes) 1 - TA buffer of unsolicited result codes defined within this command is cleared when <mode> 1…3 is entered. Note: DTR signal is ignored, hence the indication is sent even if the DTE is inactive (DTR signal is Low). In this case the unsolicited result code may be lost so if GM862-GPRS remains active while DTE is not, at DTE startup is suggested to check whether new messages have reached the device meanwhile with command AT+CMGL=0 that lists the new messages received.
Test command **AT+CNMI=?**	Reports the supported range of values for the +CNMI command parameters.
Reference	GSM 07.05
SW release	Version A

6.5.3.2 +CMGL - list messages

+CMGL – list messages	
Execute command	
Read command	
Write command **AT+CMGL = <stat>**	Reports the list of all the messages stored into <memr> (see command +CPMS) memory storage having the status equal to <stat> parameter. Parameter **(PDU Mode)**: <stat> 0 - new message 1 - read message 2 - stored message not yet sent 3 - store message already sent 4 - all messages (applies only to +CMGL command) A report is sent for each message that has to be listed in the format: +CMGL: <index>,<stat>,<length><CR><LF><pdu> where <index> - message position in the memory storage list. <stat> - status of the message <length> - length of the PDU in bytes <pdu> - message in PDU format according to GSM 3.40 Note: OK message is sent only at the end of the listing. Parameter **(Text Mode)**: <stat> "REC UNREAD" - new message "REC READ" - read message "STO UNSENT" - stored message not yet sent "STO SENT" - store message already sent "ALL" - all messages (applies only to +CMGL command) A report is sent for each message that has to be listed in the format: +CMGL: <index>,<stat>,<oa/da>[,,,<tooa/toda>,<length>]<CR><LF><text> where <index> - message position in the storage <stat> - message status <oa/da> - originator/destination number < tooa/toda > - type of number <oa/da> 145 - number in international format (contains the "+") 129 - number in national format

+CMGL – list messages	
	<length> - text length <text> - message text For each message delivery confirm a result code is reported in the format**:** +CMGL: <index>,<stat>,<fo>,<mr>,,,<scts>,<dt>,<st> Where <index> - message position in the storage <stat> - message status <fo> - first octet of the message PDU <mr> - message reference number <scts> - arrival time of the message to the SC <dt> - sending time of the message <st> - message status as coded in the PDU
Test command	
Example	
Reference	GSM 07.05
SW release	Version A

6.5.3.3 +CMGR - read message

+CMGR – read message	
Execute command **AT+CMGR = <index>**	Reports the message with location value <index> from preferred message storage <memr> (see +CPMS) in the format: **(PDU Mode)** +CMGR: <stat>,<length><CR><LF><pdu> where <stat> - status of the message 0 - new message 1 - read message 2 - stored message not yet sent 3 - store message already sent <length> - length of the PDU in bytes. <pdu> - message in PDU format according to GSM 3.40. **(Text Mode)** For the sent messages: +CMGR: <stat>,<oa>,,<scts> [,<tooa>,<fo>,<pid>,<dcs>,<sca>, <tosca>,<length>]<CR><LF><text> For the received messages: +CMGR: <stat>,<da>[,,<toda>,<fo>,<pid>,<dcs>,, <sca>,<tosca>,<length>]<CR><LF><testo> For the message delivery confirm: +CMGR: <stat>,<fo>,<mr>,,,<scts>,<dt>,<st> Where: <stat> - status of the message "REC UNREAD" - new received message unread "REC READ" - received message read "STO UNSENT" - message stored not yet sent "STO SENT" - message stored already sent <fo> - first octet of the message PDU <mr> - message reference number <scts> - arrival time of the message to the SC <dt> - sending time of the message

	<st> - message status as coded in the PDU <pid> - Protocol Identifier <dcs> - Data Coding Scheme <oa> - Originator address number <da> - Destination address number <sca> - Service Centre number < tooa>,<toda >,<tosca> - type of number <oa>,<da>,<sca> 145 - number in international format (contains the "+") 129 - number in national format <length> - text length <text> - message text The status of the message and entire message data unit <pdu> is returned. If status of the message is 'received unread', status in the storage changes to 'received read'. Parameter: <index> Note: if record number <index> on message storage memory is empty, then an error message will be returned.
Read command	
Write command	
Test command	
Example	
Reference	GSM 07.05
SW release	Version A

6.5.4 Message sending and writing

6.5.4.1 +CMGS - send message

+CMGS – send message	
Execute command *(PDU Mode)* **AT+CMGS = <length>**	**(PDU Mode)** Sends to the network a SMS message input as a PDU. Parameter: <length> = 8…176 - represents the length of the PDU to be sent in bytes. The device responds to the command with the prompt '>' and awaits for the number of bytes of PDU specified. To complete the operation send Ctrl-Z char (0x1A hexadecimal) , to exit without sending the message send ESC char (0x1B hexadecimal). If message is successfully sent to the network, then the result is sent in the format: +CMGS: <mr> where <mr> is the message reference number. If message sending fails for some reason, an error code is reported.
(Text Mode) **AT+CMGS=<da>**	**(Text Mode)** Sends to the network a SMS message input as a text message. Parameter: <da> = destination address number The device responds to the command with the prompt '>' and awaits for message text (max 160 characters). To complete the operation send Ctrl-Z char (0x1A hexadecimal) , to exit without sending the message send ESC char (0x1B hexadecimal). If message is successfully sent to the network, then the result is sent in the format: +CMGS: <mr> where <mr> is the message reference number. If message sending fails for some reason, an error code is reported. Note: Care must be taken to ensure that during the command execution, which might take several seconds, no other SIM interacting commands are issued. To avoid malfunctions is suggested to wait for the +CMGS:<mr> or +CMS ERROR:<err> response before issuing further commands.
Read command	

+CMGS – send message	
Write command	
Test command	
Example	
Reference	GSM 07.05
SW release	Version A

6.5.4.2 +CMSS - send message from storage

+CMSS – send message from storage	
Execute command **AT+CMSS = <index>**	Sends to the network, the message which is already stored in the <memw> storage (see +CPMS) at the location <index>. Parameter: <index> If message is successfully sent to the network, Sends to the network, then the result is sent in the format: +CMSS: <mr> where <mr> is the message reference number. If message sending fails for some reason, an error code is reported: +CMS ERROR:<err> Note: to store a message in the <memw> storage see command +CMGW. Care must be taken to ensure that during the command execution, which might take several seconds, no other SIM interacting commands are issued. To avoid malfunctions is suggested to wait for the +CMGS:<mr> or +CMS ERROR:<err> response before issuing further commands.
Read command	
Write command	
Test command	
Example	
Reference	GSM 07.05
SW release	Version A

6.5.4.3 +CMGW - write message to memory

+CMGW – write message to memory	
Execute command *(PDU Mode)* **AT+CMGW = <length>**	**(PDU Mode)** Writes in the <memw> memory storage a new SMS message input as a PDU. Parameter: <length> - represents the length of the PDU to be written in bytes. The device responds to the command with the prompt '>' and awaits for the number of bytes of PDU specified. To complete the operation send Ctrl-Z char (0x1A hexadecimal) , to exit without writing the message send ESC char (0x1B hexadecimal). If message is successfully written in the memory, then the result is sent in the format: +CMGW: <index> where <index> is the message location index in the memory <memw>("SM"). If message storing fails for some reason, an error code is reported
(Text Mode) **AT+CMGW = [<da>]**	**(Text Mode)** Writes in the <memw> memory storage a new SMS message input as Text. Parameter: <da> - destination address number The device responds to the command with the prompt '>' and awaits for the message text (max 160 characters). To complete the operation send Ctrl-Z char (0x1A hexadecimal) , to exit without writing the message send ESC char (0x1B hexadecimal). If message is successfully written in the memory, then the result is sent in the format: +CMGW: <index> where <index> is the message location index in the memory <memw>("SM"). If message storing fails for some reason, an error code is reported Note: Care must be taken to ensure that during the command execution, no other SIM interacting commands are issued. To avoid malfunctions is suggested to wait for the +CMGS:<mr> or +CMS ERROR:<err> response before issuing further commands.

+CMGW – write message to memory	
Read command	
Write command	
Test command	
Example	
Reference	GSM 07.05
SW release	Version A

6.5.4.4 +CMGD - delete message

+CMGD – delete message	
Execute command **AT+CMGD = <index>[,<delflag>]**	Deletes from memory the message/messages Note: if the location to be deleted is empty, an error message is reported. Parameter: <index> - message position index in the selected storage <memr> <delflag> - delete mode selection flag 0 (or not present) - delete message at position <index> 1 - delete all received read messages 2 - delete all received read and all sent messages 3 - delete all received read and all written sent/unsent messages 4 - delete all messages.
Read command	
Write command	
Test command	
Example	
Reference	GSM 07.05
SW release	Version A

6.6 Custom AT Commands

6.6.1 General configuration

6.6.1.1 #CGMI - request manufacturer identification

#CGMI – request manufacturer identification	
Execute command **AT#CGMI**	Returns the device manufacturer identification code with command echo.
Read command	
Write command	
Test command	
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version D

6.6.1.2 #CGMM - request model identification

#CGMM – request model identification	
Execute command **AT#CGMM**	Returns the device model identification code with command echo.
Read command	
Write command	
Test command	
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version D

6.6.1.3 #CGMR - request revision identification

#CGMR – request revision identification	
Execute command **AT#CGMR**	Returns device software revision number with command echo.
Read command	
Write command	
Test command	
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version D

6.6.1.4 #CGSN - request product serial number identification

#CGSN – request product serial number identification	
Execute command **AT#CGSN**	Returns the product serial number, identified as the IMEI of the mobile, with command echo.
Read command	
Write command	
Test command	
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version D

6.6.1.5 #CIMI - request international mobile subscriber identity (IMSI)

#CGSN – request international mobile subscriber identity (IMSI)	
Execute command **AT#CIMI**	Returns the international mobile subscriber identity, identified as the IMSI number, with command echo.
Read command	
Write command	
Test command	
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version D

6.6.1.6 #CAP - Change Audio Path

#CAP – change audio path	
Execute command **AT#CAP = <n>**	Switches the active audio path depending on parameter <n> Parameter: <n> audio path 0 - audio path follows the Axe input (default at start up): Axe = low - handsfree enabled Axe = high - internal path enabled 1 - enables handsfree external mic/ear audio path 2 - enables internal mic/ear audio path Note: The audio path are mutually exclusive, enabling one disables the other. When changing the audio path, the volume level is set at the previously stored value for that audio path. (see AT+CLVL).
Read command **AT#CAP?**	Reports the active audio path in the format: #CAP: <n>.
Write command	
Test command **AT#CAP=?**	Reports the supported values for the parameter <n>.
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.7 #SRS - Select ringer sound

#SRS – select ringer sound	
Execute command **AT#SRS = <n>[,<timeout>]**	Set the ringer sound. The parameters are: <n> - ringing tone 0 - current ringing tone 1 .. 12 - ringing tone number <timeout> ringing tone playing timeout in seconds. 0 - no ringing tone playing, only tone setting 1 .. 60 - <n> tone playing for <timeout> seconds and successive tone setting. Note: When the command is issued with <timeout> >0, the <n> ringing tone is played for <timeout> seconds and then stored as current tone. If command is issued with <timeout> = 0, the playing of the ringing is stopped (if present) and <n> ringing tone is set as default. If command is issued with <n> = 0 and <timeout> >0 then the default ringing tone is played. If both <n> and <timeout> are 0 then currently playing tone is set as default and ringing is stopped.
Read command **AT#SRS?**	Reports current selected ringing and its status in the form: #SRS: <n>,<status> where: <n> ringing tone number 1 .. 12 <status> ringing status 0 - selected but not playing 1 - currently playing
Write command	
Test command **AT#SRS=?**	Reports the supported values for the parameters <n> and <timeout>
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.8 #SRP –Select Ringer Path

#SRP – select ringer path	
Execute command **AT#SRP=<n>**	Selects the ringer path towards whom sending ringer sounds and all signaling tones. Parameter: <n> - Ringer path number 0 - sound output towards current selected audio path (see command #CAP) 1 - sound output towards handsfree 2 - sound output towards headset 3 - sound output towards Buzzer Output pin GPIO7 Note: In order to use the Buzzer Output an external circuitry must be added to drive it properly from the GPIO7 pin, furthermore the GPIO7 pin direction must be set to Buzzer output (Alternate function) see command #GPIO.
Read command **AT#SRP?**	Reports the current ringer path setting in the format: #SRP:<n>.
Write command	
Test command **AT#SRP=?**	Reports the supported values for the parameter <n>.
Example	**AT#SRP=?** **#SRP=(0-3)** **OK** **AT#SRP=3** **OK**
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.9 #STM - Signalling Tones Mode

#STM – signalling tones mode	
Execute command	
Read command **AT#STM?**	Reports the current signaling tones status in the format:#STM: <mode>
Write command **AT#STM = <mode>**	Enables/disables the signaling tones output on the audio path selected with #SRP command Parameter : <mode> - signaling tones status <mode> = 0 - signaling tones disabled <mode> = 1 - signaling tones enabled
Test command **AT#STM=?**	Reports supported range of values for <mode> parameter.
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version B

6.6.1.10 #PCT – display PIN Counter

#PCT – display PIN counter	
Execute command	
Read command **AT#PCT?**	Reports the PIN/PUK or PIN2/PUK2 input remaining attempts, depending on +CPIN requested password in the format: #PCT:<n> Note: If +CPIN requires the PIN then max 3 attempts are allowed, after that the PUK code is requested; if even PUK is input wrong for other three time then #PCT:0 and SIM remains blocked.
Write command	
Test command	
Example	
Reference	Telit GM862 AT-GPRS Command Specification
SW release	Version A

6.6.1.11 #SHDN – Software Shut Down

#SHDN – software shutdown	
Execute command **AT#SHDN**	After the issuing of this command device detaches from the network and shuts down. Before definitive shut down an OK response is returned. Note: after the issuing of this command any previous activity is terminated and the device will not respond to any further command. To turn it on again Hardware pin ON/OFF must be tied low.
Read command	
Write command	
Test command	
Example	
Reference	Telit GM862 AT-GPRS Command Specification
SW release	Version A

6.6.1.12 #WAKE – Wake from Alarm mode

#WAKE – Wake from Alarm Mode	
Execute command **AT#WAKE=0**	After the issuing of this command device exits the "Alarm mode" and enters the normal operating mode. After entering in normal operating mode an OK response is returned. With the execution of is command all Alarm activity (for example alarm tone playing) is immediately terminated. If the command is issued during the normal operating mode then the only action is to stop Alarm activity, if present, and return OK response.
Read command **AT#WAKE?**	This command returns the operating mode status of the device in the format: #WAKE: <status> where: <status> - operating mode 0 - normal operating mode 1 - alarm mode **Note:** The "alarm mode" is indicated by hardware pin CTS to the ON status and DSR to the OFF status, while the "power saving" status is indicated by a CTS - OFF and DSR - OFF status. The normal operating status is indicated by DSR - ON. During the "alarm mode" the device will not make any network scan and will not register to any network and therefore is not able to dial or receive any call or SMS, the only commands that can be issued to the GM862-GPRS in this state are the #WAKE and #SHDN, every other command must not be issued during this state.
Write command	
Test command	
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.13 #QTEMP –Query Temperature overflow

#QTEMP – query temperature overflow	
Execute command	
Read command **AT#QTEMP?**	Queries the internal temperature sensor of the device for over temperature. The result is reported in the format: #QTEMP:<temp> where <temp> - over temperature indicator <temp> = 0 - device temperature is in the working range <temp> = 1- device temperature is out of the working range, may be too high or too low. Note: The device should not be operated out of its working temperature range; if temperature is out of range proper functioning of the device is not ensured.
Write command **AT#QTEMP = <mode>**	Sets the type of indication. Parameter: <mode> - type of indication 0 - result reported only when querying with #QTEMP? 1 - reserved for future use 2 – reserved for future use
Test command **#QTEMP=?**	Reports supported range of values for <mode> parameter.
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.14 #SGPO –Set General Purpose Output

#SGPO – set general purpose output	
Execute command	
Read command **AT#SGPO?**	Reports the #SGPO command setting, hence the opposite status of the open collector pin in the format: #SGPO: <stat>.
Write command **AT#SGPO = <stat>**	Sets the value of the general purpose output pin GPIO2 according to <stat> parameter Parameter: <stat> 0 - output pin cleared to 0 (LOW) 1 - output pin set to 1 (HIGH) Note: The GPIO2 is an OPEN COLLECTOR output, the command sets the transistor base level, hence the open collector output is negated: AT#SGPO= 0 sets the open collector output HIGH AT#SGPO= 1 sets the open collector output LOW A pull up resistor is required on pin GPIO2.
Test command **AT#SGPO=?**	Reports the supported range of values of the command parameter <stat>.
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.15 #GGPI – Read General Purpose Input

#GGPI – read general purpose input	
Execute command	
Read command **AT#GGPI?**	Reports the read value for the input pin GPIO1in the format: #GGPI: <dir>,<stat> where <dir> direction setting (see #GGPI=<dir>) <stat> - logic value read from pin GPIO1 Note: Since the reading is done after the insulating transistor, the reported value is the opposite of the logic status of the GPIO1 input pin.
Write command **AT#GGPI = <dir>**	Set the General purpose input pin behavior depending on parameter <dir>. Parameter: <dir> - auxiliary input GPIO1 setting <dir> = 0 - the read command reports the logic input level read from GPIO1 pin. <dir> = 1…255 - reserved for future use Note: The device has an insulated input pin (the input goes the base of an internal decoupling transistor) which can be used as a logic general purpose input. This command sets the read behaviour for this pin, since only direct read report is supported, the issue of this command is not needed. In future uses the behavior of the read input may be more complex.
Test command	
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.16 #GPIO –General Purpose Input/Output pin control

#GPIO –General Purpose Input/Output pin control	
Execute command	
Read command **AT#GPIO=<pin>,2**	Reports the read value for the pin GPIO<pin> in the format: #GPIO: <dir>,<stat> where <dir> - current direction setting for the GPIO<pin> <stat> - logic value read from pin GPIO<pin> in the case the pin <dir> is set to input; - logic value present in output of the pin GPIO<pin> in the case the pin <dir> is currently set to output; - no meaning value for the pin GPIO<pin> in the case the pin <dir> is set to alternate function; Note for GPIO1: Since the reading is done after the insulating transistor, the reported value is the opposite of the logic status of the GPIO1 input pin.
Write command **AT#GPIO =<pin>, <mode>,<dir>**	Sets the value of the general purpose output pin GPIO<pin> according to <dir> and <mode> parameter. Not all configuration for the three parameters are valid. Parameter: <pin> - GPIO pin number supported range is from 1 to 7, but GPIO1 is input only and GPIO2 is output only. <dir> - GPIO pin direction 0 - pin direction in INPUT 1 - pin direction is OUTPUT 2 - pin direction is ALTERNATE FUNCTION *see Note <mode> - its meaning depends on <dir> setting: 0 - no meaning if <dir> = 0 - INPUT - output pin cleared to 0 (LOW) if <dir> = 1 - OUTPUT - no meaning if <dir> = 2 - ALTERNATE FUNCTION

	1 - no meaning if <dir> = 0 - INPUT - output pin set to 1 (HIGH) if <dir> = 1 - OUTPUT - no meaning if <dir> = 2 - ALTERNATE FUNCTION 2 - Reports the read value from the input pin (see Read command) if <dir> = 0 - INPUT - Reports the read value from the input pin (see Read command) if <dir> = 1 - OUTPUT - Reports a no meaning value (see Read command) if <dir> = 2 - ALTERNATE FUNCTION Note: "ALTERNATE FUNCTION" value is valid only for pins GPIO6 - alternate function is "Alarm Output" (see command +CALA) GPIO7 - alternate function is "Buzzer Output" (see command #SRP) While using the pins in the alternate function, the GPIO read/write access to that pin is not accessible and shall be avoided. Note: The GPIO2 is an OPEN COLLECTOR output, the command sets the transistor base level, hence the open collector output is negated
Test command **AT#GPIO=?**	Reports the supported range of values of the command parameters <pin>,<mode>,<dir>.
Example	AT#GPIO=3,0,1 OK AT#GPIO=3,2 #GPIO: 1,0 OK AT#GPIO=4,1,1 OK AT#GPIO=5,0,0 OK AT#GPIO=6,2 #GPIO: 0,1 OK AT#GPIO=7,0,0 OK
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.17 #MONI – Monitor Cells

#MONI – monitor cell	
Execute command **AT#MONI**	Reports the available data for cell and dedicated channel (if exists) in the format: #MONI: <netname> BSIC:<bsic> RxQual:<qual> LAC:<lac> Id:<id> ARFCN:<arfcn> PWR:<dBm> dBm TA: <timadv> or in the case the network name is not known: #MONI: Cc:<cc> Nc:<nc> BSIC:<bsic> RxQual:<qual> LAC:<lac> Id:<id> ARFCN:<arfcn> PWR:<dBm> dBm TA: <timadv> or in the case in which the data of the adjacent cell are under observation (number>0): #MONI: Adj Cell<n> [LAC:<lac> Id:<id>] ARFCN:<arfcn> PWR:<dBm> dBm where: netname = name of network operator cc = country code nc = network operator code n = progressive number of adjacent cell bsic = base station identification code qual = quality of reception (0-7) lac = localization area code id = cell identifier arfcn = assigned radio channel dBm = received signal strength in dBm timadv= timing advance Note: TA: <timadv> is reported only on GM862-PCS and for the serving cell.
Read command	
Write command **AT#MONI =<number>**	Set the neighbor cell to extract data of the cell number <number> Parameter: <number>
Test command **AT#MONI=?** *(Version B)*	Reports the available neighbour cells and current cell preset in the format: #MONI: <CellNo>,<CellSet> where: <CellNo> - neighbour cells number <CellSet> - cell preset = 0, …, n where 0 is the serving cell
Example	

Note	The refresh time of the measures is preset to 3 sec The timing advance value is meaningful only during calls or GPRS transfers active.
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.18 #QSS – Query SIM Status

#QSS – query SIM status	
Execute command	
Read command **AT#QSS?**	Reports the query SIM status in the format: #QSS: <mode>,<status> where <mode>: type of enabled notification <mode> = 0 - query only <mode> = 1 - send unsolicited indication (#QSS) every change of state <status>: current SIM status <status> = 0 - SIM NOT INSERTED <status> =1 - SIM INSERTED
Write command **AT#QSS = <mode>**	Set the type of notification. If enabled (mode = 1) at any status change is send: #QSS: <status> Parameter: <mode>
Test command **AT#QSS=?**	Returns the supported range of values of the parameter <mode>.
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.19 #ACAL – Set Automatic Call

#ACAL – set automatic call	
Execute command	
Read command **AT#ACAL?**	Returns the current status in the format: #ACAL: <mode>
Write command **AT#ACAL = <mode>**	Enable the automatic call. Parameter: <mode> = 0 – disable <mode> = 1 – enable If enabled, the transition OFF/ON of DTR causes an automatic data call to the number stored in position number 1 of the phone book. The &D command must be set to 2.
Test command **AT#ACAL=?**	Returns the supported range of values of the parameter <mode>.
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Future SW Version

6.6.1.20 #SMOV – SMS Overflow

#SMOV – SMS overflow	
Execute command	
Read command **AT#SMOV?**	Return the current status in the format: #SMOV: <mode>.
Write command **AT#SMOV= <mode>**	Enable the signalling of SMS overflow Parameter: <mode> = 0 - disable <mode> = 1 - enable If enable, when the maximum storage capacity has came, the #USMO:<memo> network initiated notification is send.
Test command **AT#SMOV=?**	Returns the supported range of values of the parameter <mode>.
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.21 #SHFEC – Set Handsfree echo canceller

#SHFEC – set handsfree echo canceller	
Execute command	
Read command **AT#SHFEC?**	Return the current status in the format: #SHFEC: <mode>.
Write command **AT#SHFEC= <mode>**	Set echo canceller on audio handsfree output. Parameter <mode>: 0 - disable echo canceller for headset mode (default) 1 - enable, setting for handsfree mode 2…255 reserved **Note**: This setting returns to default after power off.
Test command **AT#SHFEC=?**	Returns the supported range of values of the parameter <mode>.
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.22 #HFMICG – Handsfree Microphone Gain

#HFMICG – handsfree microphone gain	
Execute command	
Read command **AT#HFMICG?**	Return the current status of handsfree input gain in the format: #HFMICG: <level>
Write command **AT#HFMICG = <level>**	Set the microphone input gain Parameter: <level>: handsfree microphone input gain 0 - 7 handsfree microphone gain (+6dB/step) 8…255 reserved
Test command **AT#HFMICG=?**	Returns the supported range of values of the parameter <level>.
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.23 #HSMICG – Handset Microphone Gain

#HSMICG – handset microphone gain	
Execute command	
Read command **AT#HSMICG?**	Return the current status of handset input gain in the format: #HSMICG: <level>
Write command **AT#HSMICG = <level>**	Set the microphone input gain Parameter: <level>: handset microphone input gain 0 - 7 handset microphone gain (+6dB/step) 8…255 reserved
Test command **AT#HSMICG=?**	Returns the supported range of values of the parameter <level>.
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.24 #SHFSD – Set Handsfree side tone

#SHFSD – set handsfree side tone	
Execute command	
Read command **AT#SHFSD?**	Return the current status in the format: #SHFSD: <mode>
Write command **AT#SHFSD = <mode>**	Set the sidetone on handsfree audio output. Parameter <mode>: 0 - disabled - headset mode (default) 1 - enabled - handsfree mode 2…255 reserved **Note**: This setting returns to default after power off.
Test command **AT#SHFSD=?**	Returns the supported range of values of the parameter <mode>.
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.25 #/ – Repeat last command

#/ – Repeat last command	
Execute command **AT#/**	It is used to execute again the last received command. Note: This command replaces the A/ command
Read command	
Write command	
Test command	
Example	
Reference	Telit GM862-GPRS AT Command Specification
SW release	Version A

6.6.1.26 #BND – Select Band (DCS 1800 or PCS 1900)

#BND – Select Band (DCS1800 or PCS1900)	
Execute command	
Read command **AT#BND?**	Return the current band selected in the format: #BND: <band>
Write command **AT#BND = <band>**	Set the band parameter to the Band value. Parameter <band>: 0 - DCS 1800MHz 1 - PCS 1900MHz **Note**: This setting is maintained even after power off.
Test command **AT#BND=?**	Returns the supported range of values of the parameter <band>.
Example	
Reference	Telit GM862-PCS AT Command Specification
SW release	Version A

6.7FAX Class 1 Commands

6.7.1 General configuration

6.7.1.1 +FCLASS - select active service class

+FCLASS - select active service class	
Execute command	
Read command **AT+FCLASS?**	Returns the current configuration value of the parameter <n>.
Write command **AT+FCLASS= <n>**	Set the GM862-GPRS in specified connection mode (data, fax, voice), hence all the calls done after, will be data or voice. Parameter: <n>: 0 = data <n>: 1 = fax class 1 <n>: 8 = voice
Test command **AT+FCLASS=?**	Returns all supported values of the parameters <n>.
Example	
Reference	ITU T.31 and TIA/EIA-578-A specifications / GSM 07.07
SW release	Version A

6.7.1.2 +FMI – Report manufacturer ID

+FMI – Report manufacturer ID	
Execute command	
Read command **AT+FMI?**	Reports the manufacturer ID
Write command	
Test command	
Example	AT+FMI? Telit Mobile Terminals OK
Reference	ITU T.31 and TIA/EIA-578-A specifications
SW release	Version A

6.7.1.3 +FMM? – Report model ID

+FMM – Report model ID	
Execute command	
Read command **AT+FMM?**	Reports the model ID
Write command	
Test command	
Example	AT+FMM? GM862-GPRS - GSM900/1800 voice/data/fax module OK
Reference	ITU T.31 and TIA/EIA-578-A specifications
SW release	Version A

6.7.1.4 +FMR – Report revision ID

+FMR – Report revision ID	
Execute command	
Read command **AT+FMR?**	Reports the software revision ID
Write command	
Test command	
Example	AT+FMR? 1.01.000 CS988 OK
Reference	ITU T.31 and TIA/EIA-578-A specifications
SW release	Version A

6.7.2 Transmission/Reception control

6.7.2.1 +FTS – Stop Transmission and pause

+FTS – Stop transmission and pause	
Execute command	
Read command	
Write command **AT+FTS=<time>**	This command causes the modem to terminate a transmission and wait for <time> 10 ms intervals before responding with the OK result code. Parameter <time> 0-255 : length of time in 10ms intervals of the pause
Test command **AT+FTS=?**	Returns all supported values of the parameters <time>.
Example	
Reference	ITU T.31 and TIA/EIA-578-A specifications
SW release	Version A

6.7.2.2 +FRS – Wait for receive silence

+FRS – Wait for receive silence	
Execute command	
Read command	
Write command **AT+FRS=<time>**	this command causes the modem to listen and report an OK result code when silence has been detected for the specified period of time. This command when the required silence period is detected or when the DTE sends another character other than XON or XOFF. Parameter <time> 0-255 : length of time in 10ms intervals of the pause
Test command **AT+FTS=?**	Returns all supported values of the parameters <time>.
Example	
Reference	ITU T.31 and TIA/EIA-578-A specifications
SW release	Version A

6.7.2.3 +FTM – Transmit data modulation

+FTM – Transmit data	
Execute command	
Read command	
Write command **AT+FTM=<mod>**	this command causes the module to transmit facsimile data using the modulation defined by the parameter <mod>. parameter <mod> : carrier modulation 24 - V27ter/2400 bps 48 - V27ter/4800 bps 72 - V29/7200 bps 96 - V29/9600 bps
Test command **AT+FTM=?**	Returns all supported values of the parameters <mod>.
Example	
Reference	ITU T.31 and TIA/EIA-578-A specifications
SW release	Version A

6.7.2.4 +FRM – Receive data modulation

+FRM – Receive data modulation	
Execute command	
Read command	
Write command **AT+FRM=<mod>**	this command causes the module to receive facsimile data using the modulation defined by the parameter <mod>. parameter <mod> : carrier modulation 24 - V27ter/2400 bps 48 - V27ter/4800 bps 72 - V29/7200 bps 96 - V29/9600 bps
Test command **AT+FRM=?**	Returns all supported values of the parameters <mod>.
Example	
Reference	ITU T.31 and TIA/EIA-578-A specifications
SW release	Version A

6.7.2.5 +FTH – Transmit data with HDLC framing

+FTH – Transmit data with HDLC framing	
Execute command	
Read command	
Write command **AT+FTH=<mod>**	this command causes the module to transmit facsimile data using HDLC protocol and the modulation defined by the parameter <mod>. parameter <mod> : carrier modulation 3 - V21/300 bps
Test command **AT+FTH=?**	Returns all supported values of the parameters <mod>.
Example	
Reference	ITU T.31 and TIA/EIA-578-A specifications
SW release	Version A

6.7.2.6 +FRH – Receive data with HDLC framing

+FRH – Receive data data with HDLC framing	
Execute command	
Read command	
Write command **AT+FRH=<mod>**	this command causes the module to receive facsimile data using HDLC protocol and the modulation defined by the parameter <mod>. parameter <mod> : carrier modulation 3 - V21/300 bps
Test command **AT+FRH=?**	Returns all supported values of the parameters <mod>.
Example	
Reference	ITU T.31 and TIA/EIA-578-A specifications
SW release	Version A

6.7.3 Serial port control

6.7.3.1 +FLO – Select flow control specified by type

+FLO – Select flow control specified by type	
Execute command	
Read command	
Write command **AT+FLO=<type>**	Selects the flow control behavior of the serial port in both directions: from DTE to DTA and from DTA to DTE Parameter: <type> - flow control option for the data on the serial port 0 - flow control None 1 - flow control Software (XON-XOFF) 2 - flow control Hardware (CTS-RTS) **Note**: This command is a shortcut of the +IFC command.
Test command **AT+FLO=?**	Returns all supported values of the parameters <type>.
Example	
Reference	ITU T.31 and TIA/EIA-578-A specifications
SW release	Version A

6.7.3.2 +FPR – Select serial port rate

+FPR – Select serial port rate	
Execute command	
Read command	
Write command **AT+FPR=<rate>**	Selects the the serial port speed in both directions: from DTE to DTA and from DTA to DTE. When autobauding is selected, then the speed is detected automatically. Parameter: <rate> - serial port speed selection 0 - autobauding
Test command **AT+FPR=?**	Returns all supported values of the parameters <rate>.
Example	
Reference	ITU T.31 and TIA/EIA-578-A specifications
SW release	Version A

6.7.3.3 +FDD – Double escape character replacement control

+FDD – Double escape character replacement control	
Execute command	
Read command	
Write command **AT+FDD=<mode>**	This command sets the double escape character replacement behavior of the module depending on the parameter <mode>. Parameter <mode>= 0 DCE decode of <DLE><SUB>: <DLE><DLE> or discard DCE encode of <1/0><1/0>: <DLE><DLE><DLE><DLE>
Test command **AT+FDD=?**	Returns all supported values of the parameters <mode>.
Example	
Reference	ITU T.31 and TIA/EIA-578-A specifications
SW release	Version A

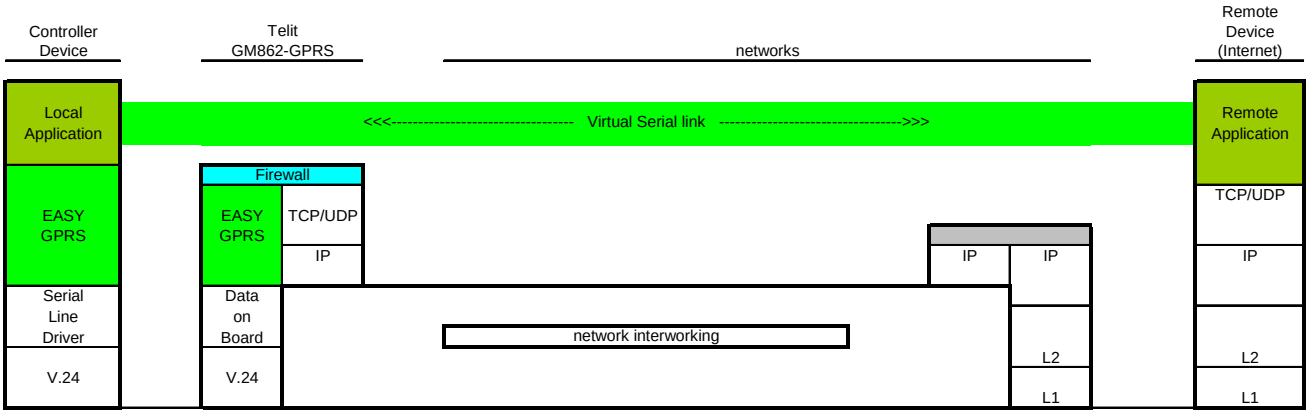
6.8 Enhanced Easy GPRS Extension

6.8.1Overview

The Easy GPRS feature allows a **Telit GM862-GPRS** user to contact a device in internet and establish with it a raw data flow over the GPRS and Internet networks.

This feature can be seen as a way to obtain a "virtual" serial connection between the Application Software on the Internet machine involved and the controller of the **Telit GM862-GPRS** module, regardless of all the software stacks underlying.

An example of the protocol stack involved in the devices is reported:



This particular implementation allows to the devices interfacing to the **Telit GM862-GPRS** module the use of the GPRS and Internet packet service without the need to have an internal TCP/IP stack since this function is embedded inside the module.

The new **Enhanced version** of the Easy GPRS overcomes some of the known limitations of the previous implementation and implements some new features such as:

- Keep the GPRS context active even after the closing of a socket, allowing the application to keep the same IP address;
- Also Mobile terminated (incoming) connections can be made, now it is possible to receive incoming TCP connection requests;
- A new internal firewall has been implemented in order to guarantee a certain level of security on internet applications.

6.8.2Easy GPRS definition

The Easy GPRS feature provides a way to replace the need of an Internet TCP/IP stack at the terminal equipment side. The steps that will be required to obtain a virtual serial connection (that is actually a socket) to the Internet peer are:

a) configuring the GPRS Access
b) configuring the embedded TCP/IP stack behaviour
c) defining the Internet Peer to be contacted
d) request the GPRS and socket connections to be opened (host is connected)
e) exchange raw data
f) close the socket and GPRS context

All these steps are achieved through AT commands.

As for common modem interface, two logical status are involved: command mode and data traffic mode.

- In Command Mode (CM), some AT commands are provided to configure the Data Module Internet stack and to start up the data traffic.
- In data traffic mode (Socket Mode, SKTM), the client can send/receive a raw data stream which will be encapsulated in the previously configured TCP / IP packets which will be sent to the other side of the network and viceversa. Control plane of ongoing socket connection is deployed internally to the module.

6.8.2.1 Configuring the GPRS access

The GPRS access configuration is done by setting:

- the GPRS context number 1 parameters (see +CGDCONT command)
- the Authentication parameters: User Name and Password (see commands #USERID, #PASSW)

6.8.2.2 Configuring the embedded TCP/IP stack

The TCP/IP stack behaviour must be configured by setting:

- the packetizer default packet size (see command #PKTSZ)
- the data sending timeout (see command #DSTO)
- the socket inactivity timeout (see command #SKTTO)

6.8.2.3 Defining the Internet peer to be contacted

As last setting definition, the host to be contacted and on which port/protocol must be set :

- the socket definition (see command #SKTSET)

This command permits also to specify the host name instead of its IP address, if a host name is given to the set command, then the module stores it as a host nick name. It is care of the module user to guarantee that the host nick name provided corresponds to an existing internet peer.

If an host nick name has been given then, while opening the connection in response to the AT#SKTOP command, the module will autonomously activate a GPRS connection and query its DNS to obtain the IP address relative to the host nick name provided. This process of context activation and DNS query may require a bit more time and requires that the GPRS network coverage is good enough to permit data transfers.

6.8.2.4 Open the connection with the internet host

With the AT#SKTOP all the process required to connect with the internet host starts:

- GM862-GPRS activates the first context
- GM862-GPRS proceeds to the authentication with the parameters specified on par. 6.8.6.1 and par. 6.8.6.2
- Eventually does the DNS query to resolve the IP address of the host name internet peer
- GM862-GPRS establishes a TCP/UDP (depending on the parameter request) connection with the given internet host
- Once the connection is up the module reports the code: CONNECT

From this moment the data incoming in the serial port is packet and sent to the Internet host, while the data received from the host is serialised and flushed to the Terminal Equipment.

6.8.2.5 Close the Socket and deactivate the context

The connection can be closed because of:

- remote host TCP connection close
- socket inactivity timeout
- Terminal Equipment by issuing the escape sequence "+++"
- Network deactivation

Note: if in the raw data to be sent there's an escape sequence, then the TE must work it out and sent it in a different fashion to guarantee that the connection is not closed.

The pause time is defined in the parameter S12.

On the reception of an escape sequence the GM862-GPRS closes the connection, deactivates the GPRS context returning to command mode and issuing the NO CARRIER code.

6.8.3Enhanced Easy GPRS Outgoing connection

The New Enhanced Easy GPRS feature provides a way to place outgoing TCP/UDP connections and keep the same IP address after a connection, leaving the GPRS context active.

The steps that will be required open a socket and close it without closing the GRPS context are:

g) configuring the GPRS Access
h) configuring the embedded TCP/IP stack behaviour
i) defining the Internet Peer to be contacted
j) request the GPRS context to be activated
k) request the socket connection to be opened
l) exchange data
m) close the TCP connection while keeping the GPRS active

All these steps are achieved through AT commands.

As for common modem interface, two logical status are involved: command mode and data traffic mode.

- In Command Mode (CM), some AT commands are provided to configure the Data Module Internet stack and to start up the data traffic.
- In data traffic mode (Socket Mode, SKTM), the client can send/receive a raw data stream which will be encapsulated in the previously configured TCP / IP packets which will be sent to the other side of the network and viceversa. Control plane of ongoing socket connection is deployed internally to the module.

6.8.3.1 Configuring the GPRS access

The GPRS access configuration is done by setting:

- the GPRS context number 1 parameters (see +CGDCONT command)
- the Authentication parameters: User Name and Password (see commands #USERID, #PASSW)

6.8.3.2 Configuring the embedded TCP/IP stack

The TCP/IP stack behaviour must be configured by setting:

- the packetizer default packet size (see command #PKTSZ)
- the data sending timeout (see command #DSTO)
- the socket inactivity timeout (see command #SKTTO)

6.8.3.3 Defining the Internet peer to be contacted

As last setting definition, the host to be contacted and on which port/protocol must be set :

- the socket definition (see command #SKTSET)

This command permits also to specify the host name instead of its IP address, if a host name is given to the set command, then the module stores it as a host nick name. It is care of the module user to guarantee that the host nick name provided corresponds to an existing internet peer.

If an host nick name has been given then, while opening the connection in response to the AT#SKTOP command, the module will autonomously activate a GPRS connection and query its DNS to obtain the IP address relative to the host nick name provided. This process of context

activation and DNS query may require a bit more time and requires that the GPRS network coverage is good enough to permit data transfers.

Note that this setting command is not needed if the new #SKTD command is used.

6.8.3.4 Request the GPRS context to be activated

With the new command #GPRS you can activate or deactivate a GPRS context INDEPENDENTLY from the TCP socket opening,

AT#GPRS=1 activates the context,

AT#GPRS=0 deactivates the context

Therefore with the AT#GPRS=1 command the module

- GM862-GPRS activates the context previously defined with AT+CGDCONT
- GM862-GPRS proceeds to the authentication with the parameters specified par. 6.8.6.1 and par. 6.8.6.2.

Note that activating a context implies getting an IP address from the network and this will be maintained throughout the session.

The response code to the AT#GPRS=1 command reports the IP address obtained from the network, allowing the user to report it to his server or application.

Deactivating the context implies freeing the network resources previously allocated to the device.

6.8.3.5 Open the connection with the internet host

With the new command #SKTD (socket Dial) the TCP/UDP request to connect with the internet host starts:

- Eventually does the DNS query to resolve the IP address of the host name internet peer
- GM862-GPRS establishes a TCP/UDP (depending on the parameter request) connection with the given internet host
- Once the connection is up the module reports the code: CONNECT

Note that the peer specifications of this socket Dial are within the command and not the one stored with #SKTSET command.

From this moment the data incoming in the serial port is packet and sent to the Internet host, while the data received from the host is serialised and flushed to the Terminal Equipment.

NOTE: this command differently from the AT#SKTOP DOES NOT automate all the process of activating the GPRS, if no GPRS is active the command reports ERROR; therefore before issuing this command the GPRS shall be activated with AT#GPRS=1 command.

In the same manner, when disconnecting the #SKTD command does not close the GPRS context, leaving it active for next connections until an AT#GPRS=0 command is issued or the network requests a context closing.

6.8.3.6 Close the Socket without deactivating the context

The connection can be closed because of:

- remote host TCP connection close
- socket inactivity timeout
- Terminal Equipment by issuing the escape sequence "+++"
- Network deactivation

Note: if in the raw data to be sent there's an escape sequence, then the TE must work it out and sent it in a different fashion to guarantee that the connection is not closed.

The pause time is defined in the parameter S12.

On the reception of an escape sequence if the socket was opened with the AT#SKTD command, the GM862-GPRS closes the connection, does not deactivate the GPRS context and returns to command mode issuing the NO CARRIER code.

6.8.4 Enhanced Easy GPRS Incoming Connection

The New Enhanced Easy GPRS feature provides a way to accept incoming TCP/UDP connections and keep the same IP address after a connection, leaving the GPRS context active.

The steps that will be required to open a socket in listen, waiting for connection requests from remote hosts and accept these request connections only from a selected set of hosts, then close it without closing the GRPS context are:

a) configuring the GPRS Access
b) configuring the embedded TCP/IP stack behaviour (see par. 6.8.3.2)
c) defining the Internet Peer that can contact this device (firewall settings) (see par. 6.8.4.1)
d) request the GPRS context to be activated (see par. 6.8.3.4)
e) request the socket connection to be opened in listen (see par. 6.8.4.2)
f) receive connection requests (see par. 6.8.4.3)
g) exchange data
h) close the TCP connection while keeping the GPRS active (see par. 6.8.3.6)

All these steps are achieved through AT commands.

As for common modem interface, two logical status are involved: command mode and data traffic mode.

- In Command Mode (CM), some AT commands are provided to configure the Data Module Internet stack and to start up the data traffic.
- In data traffic mode (Socket Mode, SKTM), the client can send/receive a raw data stream which will be encapsulated in the previously configured TCP / IP packets which will be sent to the other side of the network and viceversa. Control plane of ongoing socket connection is deployed internally to the module.

6.8.4.1 Defining the Internet Peer that can contact this device (firewall settings)

The GM862-GPRS has an internal Firewall that controls the behaviour of the incoming connections to the module.

The firewall applies for INCOMING (listening) connections, OUTGOING connections will be always done regardless of the firewall settings.

Firewall General policy is DROP, therefore all packets that are not included into an ACCEPT chain rule will be silently discarded.

When a packet incomes from the IP address <incoming IP>, the firewall chain rules will be scanned for matching with the following criteria:

<incoming IP> & <net mask> = <ip_address> ?

if the result is yes, then the packet is accepted and the rule scan is finished, otherwise the next chain is taken into account until the end of the rules when the packet is silently dropped if no matching was found.

For example, let assume we want to accept connections only from our devices which are on the IP addresses ranging from :

197.158.1.1 to 197.158.255.255

We need to add the following chain to the firewall:

AT#FRWL=1,"197.158.1.1","255.255.0.0"

6.8.4.2 Request the socket connection to be opened in listen

With the new command #SKTL (socket Listen) the TCP request to start listening for connection requests is executed:

- GM862-GPRS opens a listening socket on the port specified, waiting for incoming TCP connections (depending on the parameter request) with the internet hosts

The parameters that shall be specified are the local port where packets shall be received, the type of socket and the closing behaviour.

6.8.4.3 Receiving connection requests

Once the connection request is received, the module reports an indication of connection with an unsolicited code

+CONN FROM: <remote address>

- then connection is accepted and once it is up the module reports the code:
  CONNECT

From this moment the data incoming in the serial port is packet and sent to the Internet host, while the data received from the host is serialised and flushed to the Terminal Equipment.

Note that the connections request are FIRST screened in the firewall, then if they are accepted they pass to the listening socket; therefore only hosts that are in the ACCEPT chain rules of the firewall can induce a connection request, the other host requests will be silently discarded without any indication to the remote host (for security reasons).

Once the connection is received and closed, the socket is not anymore in listen. If the application needs again to be in listen, then it shall send again the socket listen #SKTL command.

NOTE: this command differently from the AT#SKTOP DOES NOT automate all the process of activating the GPRS, if no GPRS is active the command reports ERROR; therefore before issuing this command the GPRS shall be activated with AT#GPRS=1 command.

In the same manner, when disconnecting the #SKTL command does not close the GPRS context, leaving it active for next connections until an AT#GPRS=0 command is issued or the network requests a context closing.

6.8.5 Known limitations

The implementation of the EASY GPRS feature has the following known limitations:

- Only one socket can be opened at a time, no multiple socket connections can be made;
- Only one connection request can be accepted at a time, subsequent requests will be silently discarded.
- Only the first GPRS context is associated with this feature;
- It is taken for granted that external processor will be able to handle at least a limited v.24 implementation: RTS, CTS and, highly recommended, DCD lines; this because software flow control is not applicable to the feature;
- Due to the particularity of this feature, the flow control of both the directions uplink and downlink is interlocked

6.8.6 Enhanced Easy GPRS custom AT command Definition

6.8.6.1 #USERID – Authentication User ID control

#USERID – Authentication User ID control	
Execute command	
Read command **AT#USERID?**	Reports the current value of the parameter <user>.
Write command **AT#USERID="<user>"**	This command sets the user identification string to be used during the authentication step to be the string <user>. Parameter: <user> - authentication User ID - any string value up to max length reported in the Test command.
Test command **AT#USERID=?**	Returns the maximum allowed length of the string parameter <user>.
Example	AT#USERID = "myName" OK AT#USERID? #USERID: "myName" OK
Reference	Telit specifications
SW release	Version C

6.8.6.2 #PASSW – Authentication Password control

#PASSW – Authentication Password control	
Execute command	
Read command **AT#PASSW=?**	Reports the maximum password length.
Write command **AT#PASSW="<pwd>"**	This command sets the user password string to be used during the authentication step to be the string <pwd>. Parameter: <pwd> - authentication password - any string value up to max length reported in the Test command.
Test command	No test command allowed, for password security
Example	AT#PASSW = "myPassword" OK
Reference	Telit specifications
SW release	Version C

6.8.6.3 #PKTSZ –Packet Size control

#PKTSZ –Packet Size control	
Execute command	
Read command **AT#PKTSZ?**	Reports the current value of the parameter <size>.
Write command **AT#PKTSZ=<size>**	This command sets the default packet size to be used by the TCP/UDP/IP stack for data sending. Parameter: <size> - packet size in bytes 0 - automatically chosen by the device 1…512 - packet size in bytes
Test command **AT#PKTSZ=?**	Returns the allowed values for the parameter <size>.
Example	AT#PKTSZ = 100 OK AT#PKTSZ? #PKTSZ: 100 OK
Reference	Telit specifications
SW release	Version C

6.8.6.4 #DSTO – Data Sending TimeOut control

#DSTO –Data Sending TimeOut control	
Execute command	
Read command **AT#DSTO?**	Reports the current value of the parameter <time>.
Write command **AT#DSTO=<time>**	This timeout applies when the data to be sent is less that one full packet size and whose sending would have been delayed for a undefined time until new data to be sent is received and the full packet size is reached. This command sets the maximum time that the module awaits before sending anyway a packet whose size is less than the default one. Parameter: <time> - packet sending timeout in 100ms units 0 - no timeout, wait forever for packets to be completed before send. 1…255 hundreds of ms
Test command **AT#DSTO=?**	Returns the allowed values for the parameter <time>.
Example	AT#DSTO = 10 (1 sec. Timeout) OK AT#DSTO? #DSTO: 10 OK
NOTE	In order to avoid possible low performances issues, it is suggested to set the <time> parameter to a value greater than 5
Reference	Telit specifications
SW release	Version C

6.8.6.5 #SKTTO – Socket inactivity timeout control

#SKTTO – Socket inactivity timeout control	
Execute command	
Read command **AT#SKTTO?**	Reports the current value of the parameter <time>.
Write command **AT#SKTTO=<time>**	This timeout applies when the no data is exchanged in the socket for a long time and therefore the socket connection is automatically closed and the GPRS context deactivated. This command sets the maximum time that the module awaits without data exchange on the socket before closing the socket and deactivating the GPRS context. Parameter: <time> - socket inactivity timeout in seconds units 0 - no timeout. 1…65535 s
Test command **AT#SKTTO=?**	Returns the allowed values for the parameter <time>.
Example	AT#SKTTO = 30 (30 sec. Timeout) OK AT#SKTTO? #SKTTO: 30 OK
Reference	Telit specifications
SW release	Version C

6.8.6.6 #SKTSET – Socket definition control

#SKTSET – Socket definition control	
Execute command	
Read command **AT#SKTSET?**	Reports the current value of the parameters <socket type>,<remote port>, <remote address>.
Write command **AT#SKTSET=<socket type>, <remote port>, "<remote address>"** **,[<closure type>]** **,[<local port>]**	This command sets the socket parameters values. Parameter: <socket type> - socket protocol type 0 - TCP 1 - UDP <remote port> - remote host port to be opened 0…65535 - port number <remote address> - address of the remote host 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 in the format: <host name> <closure type> - socket closure behaviour for TCP (optional) **0 - local host closes immediately when remote host has closed (default)** 255 - local host closes after an escape sequence (+++) this parameter is valid only for TCP socket type, for UDP sockets shall be left unused. <local port> - local host port to be used on UDP socket 0…65535 - port number this parameter is valid only for UDP socket type, for TCP sockets shall be left unused.
Test command **AT#SKTSET=?**	Returns the allowed values for the parameters.
Example	AT#SKTSET = 0,1024,"123.255.020.001" OK Or AT#SKTSET=0,1024,"www.telit.net" OK

NOTE	The resolution of the host name is done when opening the socket, therefore if an invalid host name is given to the #SKTSET command, then no error message will be issued to the #SKTSET command. The DNS Query to be successful requests that: - the GPRS context 1 is correctly set with AT+CGDCONT - the authentication parameters are set (#USERID, #PASSW) - the GPRS coverage is enough to permit a connection
Reference	Telit specifications
SW release	Version C

6.8.6.7 #SKTOP – Socket Open command

#SKTOP – Socket Open Command	
Execute command **AT#SKTOP**	Activates the context number 1, proceeds with the authentication with the user ID and password previously set with #USERID #PASSW commands, and opens a socket connection with the host specified in the #SKTSET command. Eventually before opening the socket connection it issues automatically a DNS query to solve the IP address of the host name. If the connection succeeds a CONNECT indication is sent, otherwise a NO CARRIER indication is sent.
Read command	
Write command	
Test command	
Example	AT#SKTOP … GPRS context activation, authentication and socket open… CONNECT
Reference	Telit specifications
SW release	Version C

6.8.6.8 #QDNS – Query DNS

#QDNS – Query DNS	
Execute command **AT#QDNS="<host name>"**	This command activates a context, authenticates and proceeds to execute a DNS query to solve the host name into an IP address. If the DNS query is successful then the IP address will be reported in the result code: #QDNS:"<host name>",< IP address> the <IP address> is in the format: xxx.xxx.xxx.xxx
Read command	
Write command	
Test command	
NOTE	This command requires that the first context parameters, the authentication par. are correctly set and that the GPRS network is present. From Version D the command does the context deactivation if the context was not previously active, Hence if the context was already activated with #GPRS command, then the context after this command is left active.
Reference	Telit specifications
SW release	Version C

6.8.6.9 #SKTCT – Socket TCP Connection Timeout

#SKTCT – Socket TCP Connection Timeout	
Execute command	
Read command **AT#SKTCT?**	Reports the current value of the parameter <timeout>.
Write command **AT#SKTCT =<timeout>**	Sets the TCP first connection timeout for the initial connection answer from the other TCP peer. Parameter: <timeout> - TCP first connection answer timeout in 100ms units 10…1200 hundreds of ms (default value: 600) Note: This timeout applies only to TCP connection request, therefore only to the time that the TCP stack waits for the answer to its connection. The time for activate the GPRS and resolving the name with the DNS query (if the peer was specified by name and not by address) is not counted in this timeout.
Test command **AT#SKTCT=?**	Returns the allowed values for the parameter.
Example	AT#SKTCT=600 OK …socket first connection answer timeout has been set to 60 s.
Reference	Telit specifications
SW release	Version D

6.8.6.10 #SKTSAV – Socket Parameters Save Command

#SKTSAV – Socket Parameters Save Command	
Execute command **AT#SKTSAV**	Saves the actual Parameters of the Socket in the NVM of the device. The values stored are: - User Name - Password - Packet Size - Socket inactivity timeout - Data sending timeout - Socket type (UDP/TCP) - Remote port - Remote address - TCP first connection answer timeout
Read command	
Write command	
Test command	
Example	AT#SKTSAV OK …socket parameters have been saved in NVM
Note	If some parameters have not been previously specified then a default value will be taken.
Reference	Telit specifications
SW release	Version C

6.8.6.11 #SKTRST – Socket Parameters Reset Command

#SKTRST – Socket Parameters Reset Command	
Execute command **AT#SKTRST**	Resets the actual Parameters of the Socket in the NVM of the device to the default ones. The values reset are: - User Name (none) - Password (none) - Packet Size - Socket inactivity timeout - Data sending timeout - Socket type (UDP/TCP) - Remote port (none) - Remote address (none) - TCP first connection answer timeout
Read command	
Write command	
Test command	
Example	AT#SKTRST OK …socket parameters have been reset
Reference	Telit specifications
SW release	Version C

6.8.6.12 #GPRS – GPRS context activation control

#GPRS – GPRS context activation control	
Execute command	
Read command **AT#GPRS?**	Reports the current status of the GPRS context with the code: #GPRS: <mode>.
Write command **AT#GPRS=<mode>**	This command deactivates/activates the GPRS context, eventually proceeding with the Authentication with the parameters given with #PASSW and #USERID depending on the value of the parameter <mode>. Parameter: <mode> - GPRS mode **0 - GPRS context deactivated (disabled)** 1 - GPRS context activated (enabled) 2 - GPRS context operation pending (only for READ command answer) In the case that the GPRS context has been activated, the result code OK is preceded by the intermediate result code: +IP: <ip_address_obtained> reporting the local IP address obtained from the network.
Test command **AT#GPRS=?**	Returns the allowed values for the parameter <mode>.
Example	AT#GPRS = 1 +IP: 129.137.1.1 OK Now GPRS Context has been activated and our IP is 129.137.1.1 Or AT#GPRS=0 OK Now GPRS context has been deactivated, IP is lost.
NOTE	
Reference	Telit specifications
SW release	Version D

6.8.6.13 #SKTD – Socket Dial

#SKTD - Socket Dial	
Execute command **AT#SKTD=<socket type>, <remote port>, "<remote address>" ,[<closure type>] ,[<local port>]**	This command opens the socket towards the peer specified in the parameters. Parameter: <socket type> - socket protocol type 0 - TCP 1 - UDP <remote port> - remote host port to be opened 0…65535 - port number <remote address> - address of the remote host to be contacted 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 in the format: <host name> <closure type> - socket closure behaviour for TCP (optional) **0 - local host closes immediately when remote host has closed (default)** 255 - local host closes after an escape sequence (+++) this parameter is valid only for TCP socket type, for UDP sockets shall be left unused. <local port> - local host port to be used on UDP socket 0…65535 - port number this parameter is valid only for UDP socket type, for TCP sockets shall be left unused.
Read command	
Write command	
Test command **AT#SKTD=?**	Returns the allowed values for the parameters.
Example	AT#SKTD = 0,1024,"123.255.020.001",255 OK Or AT#SKTD = 1,1024,"123.255.020.001", ,1025 OK

	In this way my local port 1025 is opened to the remote port 1024 Or AT#SKTD=0,1024,"www.telit.net", 255 OK
NOTE	The resolution of the host name is done when opening the socket, therefore if an invalid host name is given to the #SKTD command, then an error message will be issued to the #SKTD command. The command to be successful requests that: - the GPRS context 1 is correctly set with AT+CGDCONT - the authentication parameters are set (#USERID, #PASSW) - the GPRS coverage is enough to permit a connection - the GPRS has been activated with AT#GPRS=1 NOTE: the main difference between this command and the AT#SKTOP is that this command does not interact with the GPRS context status, leaving it ON or OFF according to the #GPRS setting, therefore when the connection made with AT#SKTD is closed the context (and hence the local IP address) is maintained.
Reference	Telit specifications
SW release	Version D

6.8.6.14 #SKTL – Socket Listen

#SKTL - Socket Listen	
Execute command **AT#SKTL=<mode>,** **<socket type>,** **<input port>,** **[<closure type>]**	This command opens/closes the socket listening for connection requests on the port specified in the parameters depending from the parameter mode. Parameter: <mode> - socket mode 0 - socket inactive (close listen) 1 - socket listening (start listen) <socket type> - socket protocol type 0 - TCP 1 - UDP (not implemented) <input port> - local host input port to be listened 0…65535 - port number <closure type> - socket closure behaviour for TCP (optional) **0 - local host closes immediately when remote host has closed (default)** 255 - local host closes after an escape sequence (+++) this parameter is valid only for TCP socket type, for UDP sockets shall be left unused. Command returns the OK result code if successful, and when a connection request incomes on the input port, if the sender is not filtered by the internal firewall (see command #FRWL), an unsolicited code is reported: +CONN FROM: <remote address> where <remote address> is the host address of the remote machine that contacted the device. Then the connection is accepted and when established the CONNECT Indication is given and the modem goes into data transfer mode. On connection close the socket is closed an no listen is anymore active. The same applies when context is closed with #GPRS=0.

	If the context is closed by the network while in listening, forcing us to stop listening, then an unsolicited code is reported: #SKTL: ABORTED
Read command **AT#SKTL?**	If listen is active then it reports the listen socket parameters status in the format: #SKTL: <mode>,<socket type>,<input port>,<closure type> OK if no listen is active then it simply reports OK result code.
Write command	
Test command **AT#SKTL=?**	Returns the allowed values for the parameters.
Example	Activate GPRS: AT#GPRS=1 OK Start listening … AT#SKTL=1,1,1024 OK Or AT#SKTL = 1,0,1024, 255 OK Receive connection requests… +CONN FROM: 192.164.2.1 CONNECT … exchange data with the remote host… send escape sequence… +++ NO CARRIER Now listen is not anymore active, in order to listen again: AT#SKTL = 1,0,1024, 255 OK

	Now in order to stop listening .. AT#SKTL=0,0,1024, 255 OK
NOTE	The command to be successful requests that: - the GPRS context 1 is correctly set with AT+CGDCONT - the authentication parameters are set (#USERID, #PASSW) - the GPRS coverage is enough to permit a connection - the GPRS has been activated with AT#GPRS=1 NOTE: the main difference between this command and the AT#SKTD is that this command does not contact any peer, nor does any interaction with the GPRS context status, leaving it ON or OFF according to the #GPRS setting, therefore when the connection made with AT#SKTL is closed the context (and hence the local IP address) is maintained.
Reference	Telit specifications
SW release	Version D

6.8.6.15 #FRWL – Firewall setup

#FRWL - Firewall setup	
Execute command	
Read command **AT#FRWL?**	Reports the list of all ACCEPT chain rules registered in the Firewall settings in the format: #FRWL: <ip_address>,<net mask> #FRWL: <ip_address>,<net mask> …. OK
Write command **AT#FRWL=<action>, "<ip_address>", "<net mask>"**	This command controls the internal firewall settings. The firewall applies for INCOMING (listening) connections, OUTGOING connections will be always done regardless of the firewall settings. Firewall General policy is DROP, therefore all packets that are not included into an ACCEPT chain rule will be silently discarded. Parameter: <action> - command action 0 - remove selected chain 1 - add an ACCEPT chain 2 - reset all chains (DROP everything) <ip_address> - remote address to be added into the ACCEPT chain - any valid IP address in the format: xxx.xxx.xxx.xxx <net mask> - mask to be applied on the <ip_address> - any valid IP address mask in the format: xxx.xxx.xxx.xxx Command returns OK result code if successful. When a packet incomes from the IP address <incoming IP>, the firewall chain rules will be scanned for matching with the following criteria: <incoming IP> & <net mask> = <ip_address> ? if the result is yes, then the packet is accepted and the rule scan is finished, otherwise the next chain is taken into account until the end of the rules when the packet is silently dropped if no matching was found.

Test command **AT#FRWL=?**	Returns the allowed values for the parameters.
Example	Let assume we want to accept connections only from our devices which are on the IP addresses ranging from 197.158.1.1 to 197.158.255.255 We need to add the following chain to the firewall: AT#FRWL=1,"197.158.1.1","255.255.0.0"
NOTE	For outgoing connections made with AT#SKTOP and AT#SKTD the remote host is dynamically inserted into the ACCEPT chain for all the connection duration. Therefore the #FRWL command shall be used only for defining the AT#SKTL behavior, deciding which hosts are allowed to connect to the local device, since the packets that are dropped from the firewall are not sensed by the listening socket. Rules are not saved in NVM, at start-up the rules list will be empty.
Reference	Telit specifications
SW release	Version D

6.9 Easy Camera Extension

6.9.1 Camera management

6.9.1.1 #CAMON – Camera ON

#CAMON – Camera ON Command	
Execute command **AT#CAMON**	Turns ON the Camera.
Read command	
Write command	
Test command	
Example	AT#CAMON OK … camera is now powered up.
Reference	Telit specifications
SW release	Version D

6.9.1.2 #CAMOFF – Camera OFF

#CAMOFF – Camera off Command	
Execute command **AT#CAMOFF**	Turns OFF the Camera.
Read command	
Write command	
Test command	
Example	AT#CAMOFF OK … camera is now powered down.
Reference	Telit specifications
SW release	Version D

6.9.1.3 #TPHOTO – Camera Take Photo

#TPHOTO – Camera Take Photo Command	
Execute command **AT#TPHOTO**	Commands the CAMERA to take the photo and stores it in the GM862 memory.
Read command	
Write command	
Test command	
NOTE	The photo is kept in the GM862-PCS RAM memory, therefore after a power off it is lost. There's only 1 position for the photo, every photo will overwrite the previous. The photo is taken during IDLE time, if the mobile is busy on network operations, (e.g. during a call) the photo cannot be taken.
Example	AT#TPHOTO OK … the camera has taken the photo and it is now stored on the GM862 memory.
Reference	Telit specifications
SW release	Version D

6.9.1.4 #RPHOTO – Camera Read Photo

#RPHOTO – Camera Read Photo Command	
Execute command **AT#RPHOTO**	After this command the GM862 starts to flush the photo in its memory to the serial line, ending it with the Ok code..
Read command	
Write command	
Test command	
NOTE	The photo is flushed as hexadecimal characters in the format selected. The baudrate is fixed at 115200, using hardware flow control.
Example	AT#RPHOTO xxxxxxxxxxxx (binary digits of the JPEG image) <cr><lf>OK<cr><lf> … the photo has been flushed to the serial line.
Reference	Telit specifications
SW release	Version D

6.9.1.5 #OBJL – Object List

#OBJL– Object List Command	
Execute command	
Read command	
Write command **AT#OBJL="<obj>"**	Reports the list of the objects stored in the GM862 memory. Parameter: <obj> - a string parameter which specify the object listed. "IMG" - Image object (it is the only string supported). Returns the list of the stored objects: #OBJL: Snapshot,<size> where Snapshot - is the name of the object. It can't be changed. <size> is the size of the object reported in bytes.
Test command	
NOTE	
Example	AT#OBJL="IMG" #OBJL: Snapshot,47224 OK
Reference	Telit specifications
SW release	Version D

6.9.1.6 #OBJR – Object Read

#OBJR – Object Read Command	
Execute command	
Read command	
Write command **AT#OBJR="<obj>","Snapshot"**	After this command the GM862 starts to flush the photo in its memory to the serial line without ending it with OK. Parameter: <obj> - a string parameter which specify the object listed. "IMG" - Image object (it is the only string supported). "Snapshot" - is the name of the object.
Test command	
NOTE	The photo is flushed as hexadecimal characters in the format selected. The baudrate is fixed at 115200, using hardware flow control.

Example	AT#OBJR="IMG","Snapshot" xxxxxxxxxxxx (binary digits of the JPEG image) … the photo has been flushed to the serial line.
Reference	Telit specifications
SW release	Version D

6.9.1.7 #CAMQUA – Camera Select Quality of Photo

#CAMQUA - Camera Select Quality of Photo	
Execute command	
Read command **AT#CAMQUA?**	Reports the current value of the parameter <qual>
Write command **AT#CAMQUA=<qual>**	This command sets the quality of the photo stored on the memory of the GM862-PCS Parameter: <qual> - photo quality 0 - low (Jpeg compression high - low quality of picture) 1 - medium (Jpeg compression medium - med. quality of picture) **2 - high (Jpeg compression low - high quality of picture) - default**
Test command **AT#CAMQUA=?**	Returns the allowed values for the parameters.
NOTE	Increasing the photo quality increases its size.
Example	AT#CAMQUA=2 OK
Reference	Telit specifications
SW release	Version D

6.9.1.8 #CMODE – Camera Select Operating MODE

#CMODE - Camera Select Operating MODE	
Execute command	
Read command **AT#CMODE?**	Reports the current value of the parameter <mode>
Write command **AT#CMODE=<mode>**	This command sets the operating mode of the GM862-PCS camera Parameter: <mode> - camera operating mode **0 - daylight (short exposure) - default** 1 - nightlight (long exposure)
Test command **AT#CMODE=?**	Returns the allowed values for the parameters.
Example	AT#CMODE=0 OK
Reference	Telit specifications
SW release	Version D

6.9.2Email management

6.9.2.1 #ESMTP – Email SMTP server

6.10.2.1 #ESMTP – Email SMTP server	
Execute command	
Read command **AT#ESMTP?**	Reports the current value of the parameter <smtp>
Write command **AT#ESMTP="<smtp>"**	This command sets the SMTP server used for EMAIL sending. SMTP server can be specified as IP address or as nick name. Parameter: <smtp> - SMTP server address 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 in the format: <host name>
Test command **AT#ESMTP=?**	Returns the max character length for the parameter.
Example	AT#ESMTP="smtp.mydomain.com" OK
Reference	Telit specifications
Note	The SMTP server used shall be inside the APN space (the smtp server provided by the network operator) or it must allow the Relay, otherwise it will refuse to send the email.
SW release	Version D

6.9.2.2 #EADDR – Email sender address

6.10.2.2 #EADDR – Email sender address	
Execute command	
Read command **AT#EADDR?**	Reports the current value of the parameter <e-add>
Write command **AT#EADDR="<e-add>"**	This command sets the sender address string to be used for sending the Email. Parameter: <e-add> - sender address - any string value up to max length reported in the Test command. By default ""
Test command **AT#EADDR=?**	Returns the maximum allowed length of the string parameter <e-add>.
Example	AT#EADDR = "me@email.box.com" OK AT#EADDR? #EADDR: " me@email.box.com " OK
Reference	Telit specifications
SW release	Version D

6.9.2.3 #EUSER – Email authentication USER NAME

6.10.2.2 #EUSER – Email authentication USER NAME	
Execute command	
Read command **AT#EUSER?**	Reports the current value of the parameter <e-user>
Write command **AT#EUSER="<e-user>"**	This command sets the user identification string to be used during the authentication step of the SMTP to be the string <e-user>. Parameter: <e-user> - authentication User ID for email - any string value up to max length reported in the Test command. By default ""
Test command **AT#EUSER=?**	Returns the maximum allowed length of the string parameter <e-user>.
Example	AT#EUSER = "myE-Name" OK AT#EUSER? #EUSER: "myE-Name" OK
Reference	Telit specifications
Note	If no authentication is required then the e-user parameter shall be empty "". Note that it is a different user field than the one used for GPRS authentication (see AT#USERID).
SW release	Version D

6.9.2.4 #EPASSW – Email authentication PASSWORD

6.10.2.2 #EPASSW – Email authentication PASSWORD	
Execute command	
Read command	
Write command **AT#EPASSW="<e-pwd>"**	This command sets the user identification string to be used during the authentication step of the SMTP to be the string <user>. Parameter: <e-pwd> - authentication Password for Email - any string value up to max length reported in the Test command. By default ""
Test command **AT#EPASSW=?**	Returns the maximum allowed length of the string parameter <e-pwd>.
Example	AT#USERID = "myPassword" OK
Reference	Telit specifications
Note	If no authentication is required then the e-pwd parameter shall be empty "". Note that it is a different pwd field than the one used for GPRS authentication (see AT#PASSW).
SW release	Version D

6.9.2.5 #SEMAIL - Send Email

#SEMAIL - Send Email	
Execute command **AT#SEMAIL = "<da>","<subj>",<att>"**	Sends an Email message . Parameter: <da> - destination address <subj> - subject of the message <att> - attach image flag 0 - don't attach any image 1 - attach the snapshot taken (must be already been taken…) The device responds to the command with the prompt '>' and awaits for the message body text. To complete the operation send Ctrl-Z char (0x1A hexadecimal) , to exit without writing the message send ESC char (0x1B hexadecimal). If email message is successfully sent, then the response is OK. If message sending fails for some reason, an error code is reported Note: Care must be taken to ensure that during the command execution, no other commands are issued. To avoid malfunctions is suggested to wait for the OK or ERROR/ +CMS ERROR:<err> response before issuing further commands. Note: sending a mail with an image attachment can take quite a long time since it can be over 50Kb to send and can take over 1 minute.
Read command	
Write command	
Test command	
Example	AT#SEMAIL="me@myaddress.com","subject of the mail",1 ➢ message body… this is the text of the mail message… CTRL-Z ..wait.. OK Message has been sent.
Reference	Telit specifications
SW release	Version D

6.9.2.6 #ESAV – Email Parameters Save Command

#ESAV – Email Parameters Save Command	
Execute command **AT#ESAV**	Saves the actual Parameters of the Email sending commands in the NVM of the device. The values stored are: - Email User Name - Email Password - Email Sender Address - Email SMTP server
Read command	
Write command	
Test command	
Example	AT#ESAV OK …email parameters have been saved in NVM
Note	If some parameters have not been previously specified then a default value will be taken.
Reference	Telit specifications
SW release	Version D

6.9.2.7 #ERST – Email Parameters Reset Command

#ERST – Email Parameters Reset Command	
Execute command **AT#ERST**	Resets the actual Parameters of the Email sending commands in the NVM of the device to the default ones. The values reset are: - Email User Name (none) - Email Password (none) - Email Sender Address (none) - Email SMTP server (none)
Read command	
Write command	
Test command	
Example	AT#ERST OK …email parameters have been reset
Reference	Telit specifications
SW release	Version D

6.10 Easy Scan Extension

This section applies only to Telit GM862 PCS and Telit GM862-PYTHON modules.

6.10.1 Easy Scan custom AT commands

6.10.1.1 #CSURV - Network Survey of the complete 900/1800/1900 Network

#CSURV – network survey of the complete GSM900/1800/1900 Network	
Execute command **AT*CSURV** **AT#CSURV** (both syntax are possible)	The command allows to perform a quick survey through both GSM900 and DCS1800 or PCS1900 Bands (full band scan). After this the following information for every received BCCH is available. For BCCH-Carrier: arfcn: <value> bsic: <value> rxLev: <value> ber: <value> mcc: <value> mnc: <value> lac: <value> cellId: <value> cellStatus: <value> numArfcn: <value> arfcn: <value1>…<value64> numChannels: <value> array: <value1>…<value32> For non BCCH-Carrier: arfcn: <value> level: <value> where: arfcn = C0 carrier assigned radio channel (BCCH - Broadcast Control Channel) bsic = base station identification code rxLev = received level (in dBm) ber = bit error rate (in %) mcc = mobile country code mnc = mobile network code lac = localization area code cellId = cell identifier cellStatus = cell Status Cell Allocation NumArfcn = number of valid channels arfcn = arfcn-list BCCH Allocation NumChannels = number of valid channels array = arfcn-list The parameter CELLSTATUS indicates the following statuses: - CELL_SUITABLE indicates the C0 is a suitable cell. - CELL_LOW_PRIORITY indicates the cell is low priority based on the system information received. - CELL_FORBIDDEN indicates the cell is forbidden. - CELL_BARRED indicates the cell is barred based on the system

	information received. - CELL_LOW_LEVEL indicates the cell RXLEV is low. - CELL_OTHER indicates none of the above e.g. exclusion timer running, no BCCH available…etc. For every non BCCH the following information is available: arfcn = RF channel rxLev = received level (in dBm)
Read command	
Write command **AT#CSURV=<s>,<e>** **AT*CSURV=<s>,<e>** (both syntax are possible)	The command allows to perform a quick survey through a defined part of the GSM900/DCS1800/PCS1900 Bands. It scans the channel from <s> to <e>. The result format is similar to the full band scan. Parameters: <s> = starting channel <e> = ending channel
Test command	
Example	AT#CSURV Network survey started… arfcn: 48 bsic: 24 rxLev: -52 ber: 0.00 mcc: 610 mnc: 1 lac: 33281 cellId: 3648 cellStatus: CELL_SUITABLE numArfcn: 2 arfcn: 30 48 numChannels: 5 array: 14 19 22 48 82 arfcn: 14 level: 8 Network survey ended OK
Note	The command is executed within max. 2 minute.
Reference	Telit Specifications.
SW release	Version D

6.10.1.2 #CSURVC - Network Survey in computer friendly format

#CSURVC – network survey in computer friendly format	
Execute command **AT*CSURVC** **AT#CSURVC** (both syntax are possible)	The command allows to perform a quick survey through both GSM900 and DCS1800 or PCS1900 Bands (full band scan). The format is computer friendly. For BCCH-Carrier: <arfcn>,<bsic>,<rxLev>,<ber>,<mcc>,<mnc>,<lac>,<cellId>,<cellStatus>,<numArfcn>,<arfcn1>…<arfcn64>,<numChannels>,<array1>…<array32> For non BCCH-Carrier: <arfcn>,<level> The numeric format of the parameter CELLSTATUS is the following: 0 - CELL_SUITABLE 1 - CELL_LOW_PRIORITY 2 - CELL_FORBIDDEN 3 - CELL_BARRED 4 - CELL_LOW_LEVEL 5 - CELL_OTHER
Read command	
Write command **AT#CSURVC =<s>,<e>** **AT*CSURVC =<s>,<e>** (both syntax are possible)	The command allows to perform a quick survey through a defined part of the GSM900/DCS1800/PCS1900 Bands. It scans the channels from <s> to <e>. The result format is similar to the full band scan. Parameters: <s> = starting channel <e> = ending channel
Test command	
Example	AT#CSURVC Network survey started… 48,24,-52,0.00,610,1,33281,3648,0,2,30 48,5,14 19 22 48 82 14,8 Network survey ended OK
Note	The command is executed within max. 2 minute.
Reference	Telit Specifications.
SW release	Version D

6.10.1.3 #CSURVU - Network Survey of user defined 900/1800/1900 channels

#CSURVU – network survey of user defined GSM900/1800/1900 channels	
Execute command	
Read command	
Write command **AT*CSURVU =<ch1>,<ch2>,…,<chN>** **AT#CSURVU =<ch1>,<ch2>,…,<chN>** (both syntax are possible)	The command allows to perform a quick survey of user defined GSM900/DCS1800/PCS1900 channels. It scans the given channels. The result format is the same like the full band scan (AT*CSURV). Parameter: <chN> = channel number The max. number of scanned channels is 498 (124 for GSM, 374 for DCS or PCS). The <chN> must be in a increasing order.
Test command	
Example	AT#CSURVU=59,110 Network survey started… arfcn: 59 bsic: 16 rxLev: -76 ber: 0.00 mcc: 546 mnc: 1 lac: 54717 cellId: 21093 cellStatus: CELL_SUITABLE numArfcn 2 arfcn: 36 59 arfcn: 110 rxLev: -107 Network survey ended OK
Note	The command is executed within max. 2 minute.
Reference	Telit Specifications.
SW release	Version D

6.10.1.4 #CSURVUC - Network Survey in computer friendly format

#CSURVUC – network survey of user defined channels in computer friendly format	
Execute command	
Read command	
Write command **AT#CSURVUC =<ch1>,<ch2>,…,<chN>** **AT*CSURVUC =<ch1>,<ch2>,…,<chN>** (both syntax are possible)	The command allows to perform a quick survey of user defined GSM900/DCS1800/PCS1900 channels. It scans the given channels. The format is computer friendly. Parameter: <chN> = channel number The max. number of scanned channels is 498 (124 for GSM, 374 for DCS or PCS). The <chN> must be in a increasing order.
Test command	
Example	AT#CSURVUC = 59,110 Network survey started… 59,16,-76,0.00,546,1,54717,21093,0,2,36 59 110,-107 Network survey ended OK
Note	The command is executed within max. 2 minute.
Reference	Telit Specifications.
SW release	Version D

6.10.1.5 #CSURVF - Network Survey Format

#CSURVF – network survey format	
Execute command	
Read command **AT#CSURVF?**	Reports the current setting of the command, in the format: #CSURVF: <format>
Write command **AT#CSURVF=<format>**	The command controls the format of the numbers output by all the Easy Scan commands depending on the value of the parameter format Parameter: <format> - numbers format 0 - Decimal 1- Hexadecimal values, no text 2- Hexadecimal values with text
Test command **AT#CSURVF=?**	Reports the supported range of values for the parameter <format>
Example	
Note	
Reference	Telit Specifications.
SW release	Version D

6.11 Jammed Detect & Report Extension

6.11.1 Overview

The Jammed Detect & Report feature allows a **Telit GM862-PYTHON, GM862-PCS, GM862-GPRS, GM862-GSM** to detect the presence of a disturbing device such as a Communication Jammer and give indication to the user and/or send a report of that to the network.

This feature can be very important in alarm, security and safety applications that rely on the module for the communications. In these applications, the presence of a Jammer device can compromise the whole system reliability and functionality and therefore shall be recognized and reported either to the local system for countermeasure actions or to the network providing remote actions.

An example scenario could be an intrusion detection system that uses the module for sending the alarm indication for example with an SMS to the system owner, and a thief incomes using a Jammer to prevent any communication between the GSM module and the network.

In such a case, the module detects the Jammer presence even before the break in and can trigger an alarm siren, other communication devices (PSTN modem) or directly report this condition to the network that can provide further security services for example sending SMS to the owner or police. Obviously this last service depends also from network infrastructure support and it may not be supported by some networks.

6.11.2 Jammed Detect & Report custom AT command

6.11.2.1 #JDR - Jammed Detect & Report

#JDR - Jammed Detect & Report	
Execute command	
Read command **AT#JDR?**	Reports the current settings of the command, in the format: #JDR: <mode>,<MNPL>,<DCMN>
Write command **AT#JDR = <mode> [,<MNPL>,<DCMN>]**	This command allows to control the Jammed Detect & Report feature. The GM862-GPRS can Detect if a communication Jammer is active in its range and give indication to the user of this condition either on the serial line with an unsolicited code or on a dedicated GPIO by rising it. The GM862-GPRS can also report to the network the Jammed status condition, even if normal communications are inhibited by the Jammer, by using a unique message. Parameter: <mode> - behavior mode of the Jammed Detect & Report

	0 - Jammed Detect & Report DISABLED (Default) 1 - the Jammed Detect is active and the Jammed condition is reported on pin GPIO2/JDR GPIO2/JDR LOW - Normal Operating Condition GPIO2/JDR HIGH - Jammed Condition. 2 - the Jammed Detect is active and the Jammed condition is reported with a single unsolicited code on serial line whose format is: #JDR: JAMMED - Jammed condition detected #JDR: OPERATIVE - Normal Operating condition restored. Note: the #JDR: OPERATIVE code will show only after a jammed condition has occurred. 3 - the GM862-GPRS will make both the actions as for mode=1 and mode=2. 4 - the Jammed Detect is active and the Jammed condition is reported with an unsolicited code every 3s on serial line, whose format is: #JDR: JAMMED - Jammed condition detected #JDR: OPERATIVE - Normal Operating condition restored. Note: the #JDR: OPERATIVE code will show only once and after a jammed condition has occurred. 5 - the GM862-GPRS will make both the actions as for mode=1 and mode=4. 10 -The GM862-GPRS will Report the Jammed status to the network on Jammed Detection. 11 - the GM862-GPRS will make both the actions as for mode=10 and mode=1. 12 - the GM862-GPRS will make both the actions as for mode=10 and mode=2. 13 - the GM862-GPRS will make both the actions as for mode=10 and mode=3. 14 - the GM862-GPRS will make both the actions as for mode=10 and mode=4. 15 - the GM862-GPRS will make both the actions as for mode=10 and mode=5. <MNPL> - Maximum Noise power Level optional parameter 0 -127 <DCMN> - Disturbed Channel minimum number optional parameter 0 - 254

	Note: The values 10 - 15 of mode parameter will be available on future sw release. **Note1:** The parameters MNPL and DCMN are used for fine tuning the detection algorithms, their value is set to appropriate values in factory and there should be never the need for changing these parameters. If the device is installed in a particular environment and the default values are not satisfactory these two parameters permit to adapt the detection to all conditions.
Test command **AT#JDR=?**	Reports the supported range of values for the parameters <mode>,<MNPL>, <DCMN>
Example	**AT#JDR=2** **OK** **… jammer enters in the range…** **#JDR: JAMMED** **… jammer exits the range…** **#JDR: OPERATIVE**
Reference	Telit Specifications.
SW release	Version D

6.12 Easy Script Extension - Python interpreter

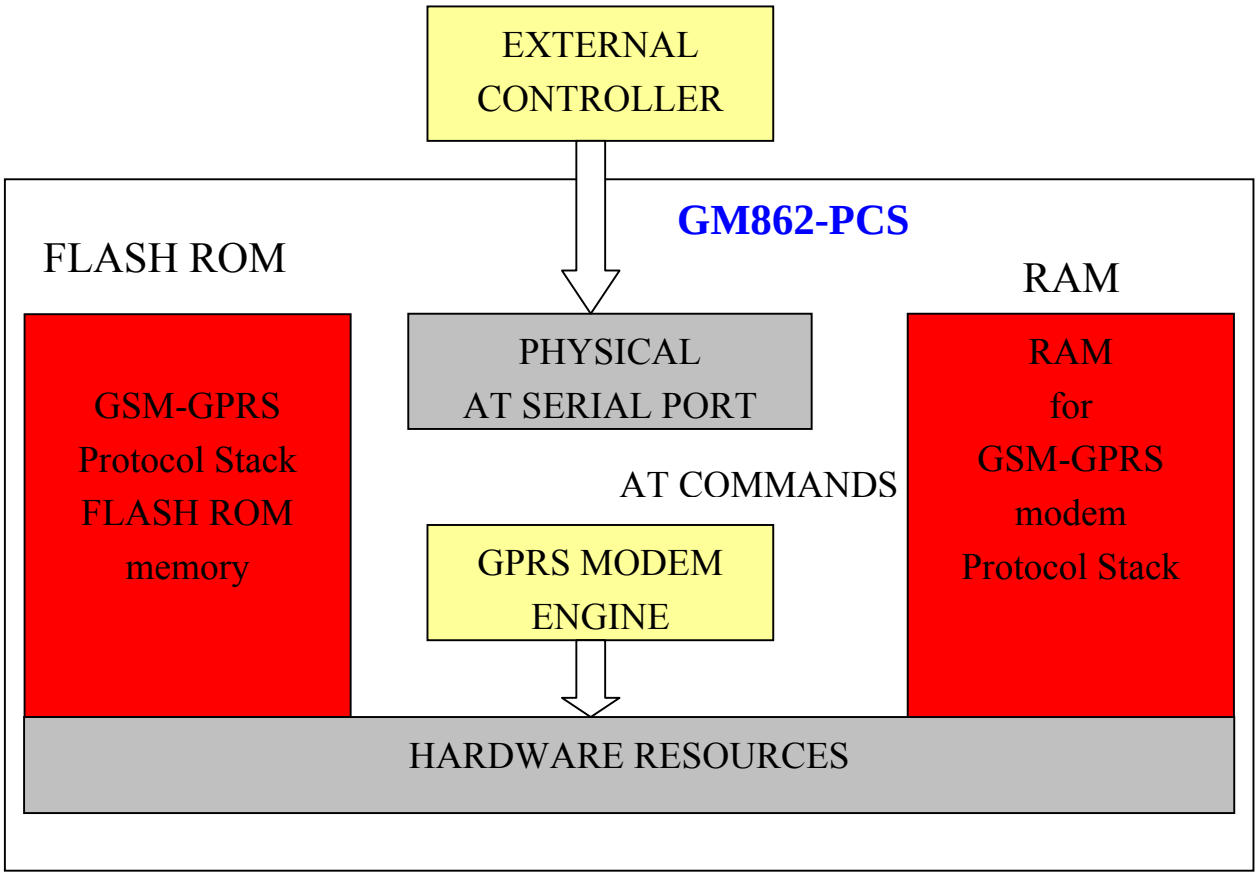
6.12.1 Overview

This feature is available only on the Telit GM862-PYTHON.

The Easy Script Extension is a feature that allows to drive the modem "internally" writing the controlling application directly in a nice high level language: Python.

The Easy Script Extension is aimed at low complexity applications where the application was usually done by a small microcontroller that managed some I/O pins and the GM862-PCS through the AT command interface.

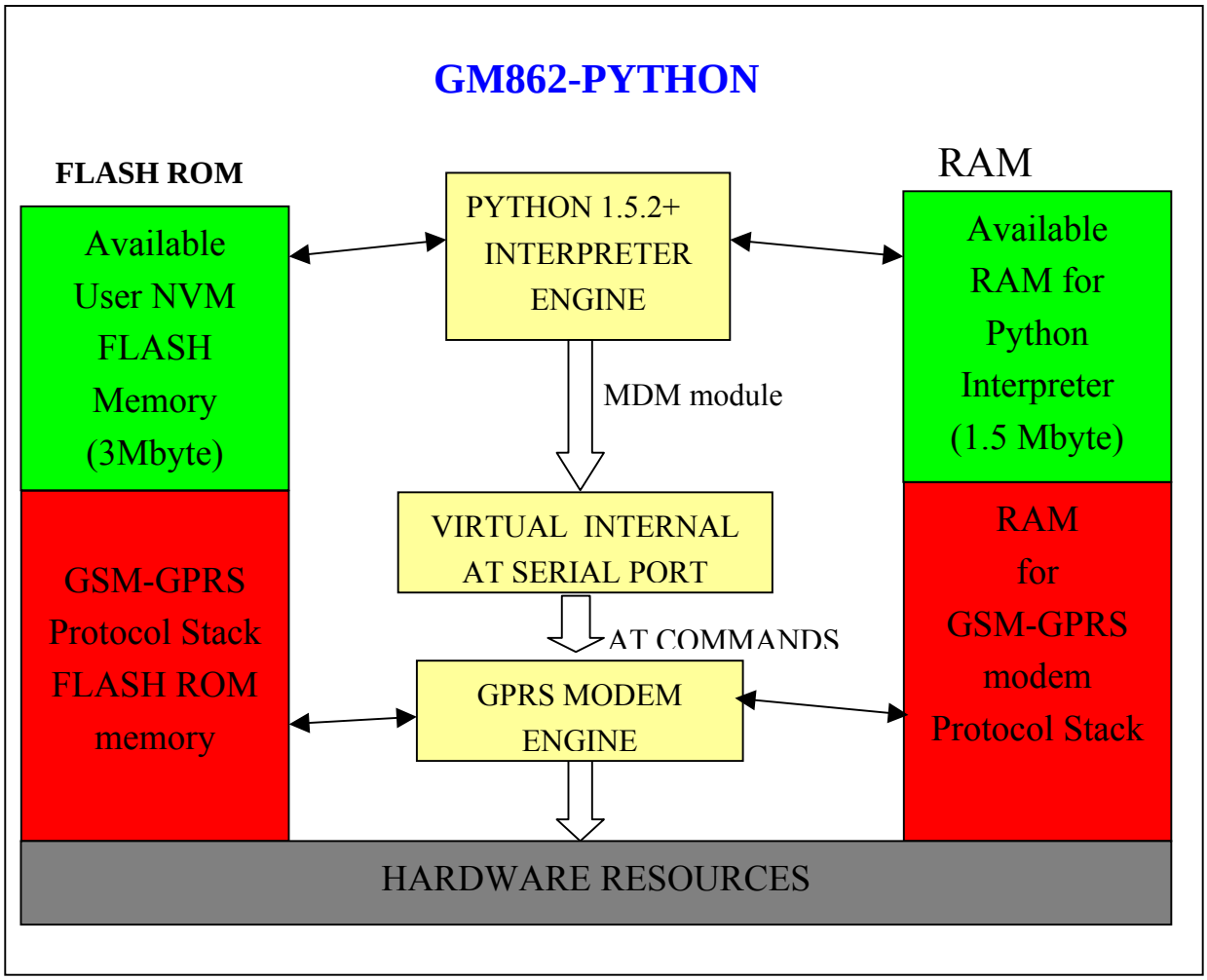
A schematic of such a configuration can be:



In order to eliminate this external controller, and further simplify the programming of the sequence of operations, inside the GM862-PYTHON it is included:

- ***Python script interpreter engine v. 1.5.2+***
- ***around 3MB of Non Volatile Memory room for the user scripts and data***
- ***1.5 MB RAM reserved for Python engine usage***

A schematic of this approach is:



6.12.2 Python 1.5.2+ Copyright Notice

The Python code implemented into the GM862-PYTHON is copyrighted by Stichting Mathematisch Centrum, this is the license:

Copyright © 1991-1995 by Stichting Mathematisch Centrum, Amsterdam, The Netherlands.

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Stichting Mathematisch Centrum or CWI or Corporation for National Research Initiatives or CNRI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

While CWI is the initial source for this software, a modified version is made available by the Corporation for National Research Initiatives (CNRI) at the Internet address ftp://ftp.python.org.

STICHTING MATHEMATISCH CENTRUM AND CNRI DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM OR CNRI BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

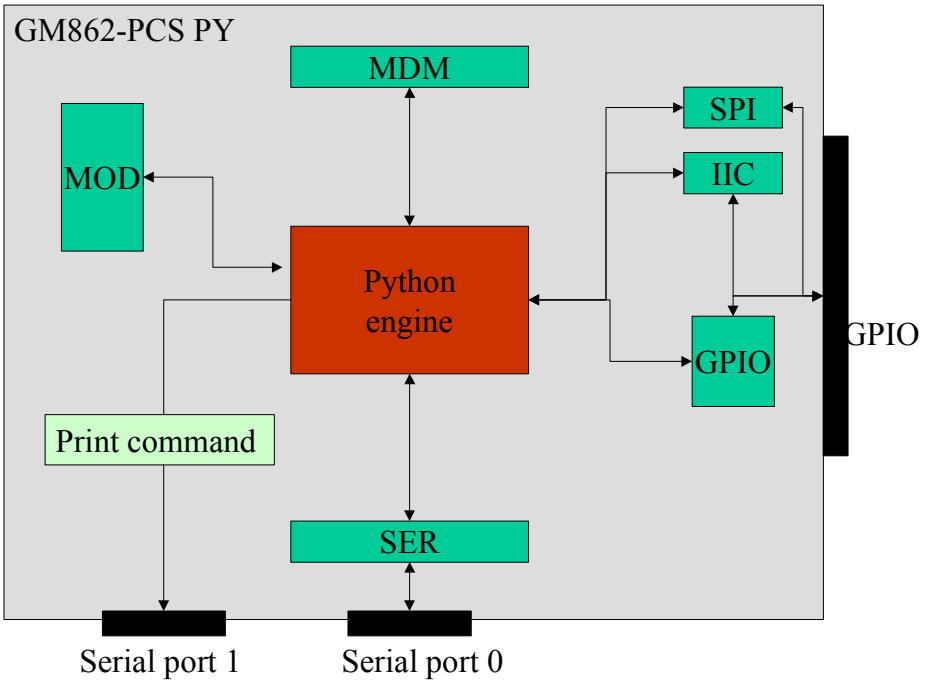
6.12.3 Python implementation description

Python scripts are text files, it is possible to run one Python script in the Telit GM862-PYTHON.

The Python script is stored in NVM inside the Telit GM862-PCS PY, there's a file system inside the GM862-PYTHON that allows to write and read files with different names on one single level (no subdirectories are supported).

The Python script is executed in a task inside the Telit GM862-PCS PY at the lowest priority, making sure this does not interfere with GPRS/GSM normal operations. This allows serial ports, protocol stack etc. to run independently from the Python script.

The Python script interacts with the Telit GM862-PCS Python functionality through four build-in interfaces.



The MDM interface is the most important one. It allows Python script to send AT commands, receive responses and unsolicited indications, send data to the network and receive data from the network during connections.

It is quite the same as the usual serial port interface in the Telit GM862-PCS. The difference is that this interface is not a real serial port but just an internal software bridge between Python and mobile internal AT command handling engine.

All AT commands working in the Telit GM862-PCS are working in this software interface as well. Some of them have no meaning on this interface, such as those regarding serial port settings.

The usual concept of flow control keeps its meaning over this interface, but it's managed internally.

The SER interface allows Python script to read from and write to the REAL, physical serial port where usually the AT command interface resides, for example to read NMEA information from a GPS device. When Python is running this serial port is free to be used by Python script because it is not used as AT command interface since the AT parser is mapped into the internal virtual serial port. No flow control is available from Python on this port.

The GPIO interface allows Python script to handle general purpose input output faster than through AT commands, skipping the command parser and going directly to control the pins.

The MOD interface is a collection of useful functions.

For the debug, the print command is directly forwarded on the EMMI TX pin (second serial port) at 9600 baud 8N1.

6.12.4 Python core supported features

The Python core version is 1.5.2+ (string methods added to 1.5.2).

You can use all Python statements and almost all Python built-in types and functions.

The following are not supported:

complex; ***float;*** ***long;*** ***docstring.***

Available modules are

marshal, ***imp,*** ***__main__,*** ***__builtin__,*** ***sys***
md5

All the others are not supported.

6.12.5 Python Build-in Custom Modules

Several build in custom modules have been included in the python core, specifically aimed at the hardware environment of the module.

The build in modules included are:

MDM: interface between Python and mobile internal AT command handling;

SER: interface between Python and mobile internal serial port ASC0 direct handling;

GPIO: interface between Python and mobile internal general purpose input output direct handling;

MOD: interface between Python and mobile miscellaneous functions.

IIC: custom software Inter IC bus that can be mapped on creation over almost any GPIO pin available.

SPI: custom software Serial Protocol Interface bus that can be mapped on creation over almost any GPIO pin available.

6.12.5.1 MDM built-in module

MDM built-in module is the interface between Python and the module AT command parser engine.

You need to use MDM built-in module if you want to send AT commands from Python script to the device and to receive responses from the device into your Python script.

Default start configuration is echo disabled (ATE0) and long form (verbose) return codes (ATV1),

If you want to use MDM built-in module you need to import it first:

import MDM

then you can use MDM built-in module methods like in the following example:

a = MDM.send('AT', 0)

b = MDM.sendbyte(0x0d, 0)

c = MDM.receive(10)

which sends 'AT' and receives 'OK'.

More details about MDM built-in module methods are in the following paragraphs.

6.12.5.1.1 MDM.send(string, timeout)

Sends a string to AT command interface.

First input parameter string is a Python string which is the string to send to AT command interface.

Second input parameter timeout is a Python integer which is the value in 1/10 s to wait for the string to be sent to AT command interface before timeout expires. Waiting time is caused by flow control.

Return value is a Python integer which is -1 if timeout expired otherwise is 1.

Example:

a = MDM.send('AT', 5)

sends string 'AT' to AT command handling, possibly waiting for 0.5 s, assigning return value to a.

6.12.5.1.2 MDM.receive(timeout)

Receives a string from AT command interface waiting for it until timeout is expired. Request to Send (RTS) is set to ON.

Input parameter timeout is a Python integer which is the value in 1/10 s to wait for a string from AT command interface before timeout expires.

Return value is a Python string which is an empty string if timeout expired without any data received otherwise is the string containing data received.

Example:

a = MDM.receive(15)

receives a string from AT command handling, possibly waiting for it for 1.5 s, assigning return value to a.

6.12.5.1.3 MDM.read()

Receives a string from AT command interface without waiting for it. Request to Send (RTS) is set to ON.

No input parameter.

Return value is a Python string which is an empty string if no data received otherwise is the string containing data received.

Example:

a = MDM.read()

receives a string from AT command handling, assigning return value to a.

6.12.5.1.4 MDM.sendbyte(byte, timeout)

Sends a byte to AT command interface.

First input parameter byte is a Python byte which is any byte value to send to AT command interface. It can be zero.

Second input parameter timeout is a Python integer which is the value in 1/10 s to wait for the byte to be sent to AT command interface before timeout expires. Waiting time is caused by flow control.

Return value is a Python integer which is -1 if timeout expired otherwise is 1.

Example:

b = MDM.sendbyte(0x0d, 0)

sends byte 0x0d, that is CR, to AT command handling, without waiting, assigning return value to b.

6.12.5.1.5 MDM.receivebyte(timeout)

Receives a byte from AT command interface waiting for it until timeout is expired. Request to Send (RTS) is set to ON.

Input parameter timeout is a Python integer which is the value in 1/10 s to wait for a byte from AT command interface before timeout expires.

Return value is a Python integer which is -1 if timeout expired without any data received otherwise is the byte value received. It can be zero.

Example:

b = MDM.receivebyte(20)

receives a byte from AT command handling, possibly waiting for it for 2.0 s, assigning return value to b.

6.12.5.1.6 MDM.readbyte()

Receives a byte from AT command interface without waiting for it. Request to Send (RTS) is set to ON.

No input parameter.

Return value is a Python integer which is -1 if no data received otherwise is the byte value received. It can be zero.

Example:

b = MDM.readbyte()

receives a byte from AT command handling, assigning return value to b.

6.12.5.1.7 MDM.getDCD()

Gets Carrier Detect (DCD) from AT command interface.

No input parameter.

Return value is a Python integer which is 0 if DCD is OFF or 1 if DCD is ON.

Example:

cd = MDM.getDCD()

gets DCD from AT command handling, assigning return value to cd.

6.12.5.1.8 MDM.getCTS()

Gets Clear to Send (CTS) from AT command interface.

No input parameter.

Return value is a Python integer which is 0 if CTS is OFF or 1 if CTS is ON.

Example:

cts = MDM.getCTS()

gets CTS from AT command handling, assigning return value to cts.

6.12.5.1.9 MDM.getDSR()

Gets Data Set Ready (DSR) from AT command interface.

No input parameter.

Return value is a Python integer which is 0 if DSR is OFF or 1 if DSR is ON.

Example:

dsr = MDM.getDSR()

gets DSR from AT command handling, assigning return value to dsr.

6.12.5.1.10 MDM.getRI()

Gets Ring Indicator (RI) from AT command interface.

No input parameter.

Return value is a Python integer which is 0 if RI is OFF or 1 if RI is ON.

Example:

ri = MDM.getRI()

gets RI from AT command handling, assigning return value to ri.

6.12.5.1.11 MDM.setRTS()

Sets Request to Send (RTS) in AT command interface.

Input parameter is a Python integer which is 0 if setting RTS to OFF or 1 if setting RTS to ON.

No return value.

Example:

MDM.setRTS(1)

sets RTS to ON in AT command handling.

6.12.5.1.12 MDM.setDTR()

Sets Data Terminal Ready (DTR) in AT command interface.

Input parameter is a Python integer which is 0 if setting DTR to OFF or 1 if setting DTR to ON.

No return value.

Example:

MDM.setDTR(0)

sets DTR to OFF in AT command handling.

6.12.5.2 SER built-in module

SER built-in module is the interface between Python core and the device serial port over the RXD/TXD pins direct handling.

You need to use SER built-in module if you want to send data from Python script to serial port and to receive data from serial port ASC0 to Python script.

This serial port handling module can be used for example to interface the module with an external device such as a GPS and read/send it's data (NMEA for example).

If you want to use SER built-in module you need to import it first:

import SER

then you can use SER built-in module methods like in the following example:

a = SER.SetSpeed('9600')

b = SER.send('test')

c = SER.sendbyte(0x0d)

d = SER.receive(10)

which sends 'test' followed by CR and receives data waiting for one second.

More details about SER built-in module methods are in the following paragraphs.

6.12.5.2.1 SER.send(string)

Sends a string to the serial port TXD/RXD.

Input parameter string is a Python string which is the string to send to serial port ASC0.

Return value is a Python integer which is -1 if an error occurred otherwise is 1.

Example:

a = SER.send('test')

sends string 'test' to serial port ASC0 handling, assigning return value to a.

6.12.5.2.2 SER.receive(timeout)

Receives a string from serial port TXD/RXD waiting for it until timeout is expired.

Input parameter timeout is a Python integer which is the value in 1/10 s to wait for a string from serial port before timeout expires.

Return value is a Python string which is an empty string if timeout expired without any data received otherwise is the string containing data received.

Example:

a = SER.receive(15)

receives a string from serial port handling, waiting for it for 1.5 s, assigning return value to a.

6.12.5.2.3 SER.read()

Receives a string from serial port TXD/RXD without waiting for it.

No input parameter.

Return value is a Python string which is an empty string if no data received otherwise is the string containing data received.

Example:

a = SER.read()

receives a string from serial port handling, assigning return value to a.

6.12.5.2.4 SER.sendbyte(byte)

Sends a byte to serial port TXD/RXD.

Input parameter byte is a Python byte which is any byte value to send to serial port. It can be zero.

Return value is a Python integer which is -1 if an error occurred otherwise is 1.

Example:

b = SER.sendbyte(0x0d)

sends byte 0x0d, that is CR, to serial port handling, assigning return value to b.

6.12.5.2.5 SER.receivebyte(timeout)

Receives a byte from serial port TXD/RXD waiting for it until timeout is expired.

Input parameter timeout is a Python integer which is the value in 1/10 s to wait for a byte from serial port before timeout expires.

Return value is a Python integer which is -1 if timeout expired without any data received otherwise is the byte value received. It can be zero.

Example:

b = SER.receivebyte(20)

receives a byte from serial port handling, waiting for it for 2.0 s, assigning return value to b.

6.12.5.2.6 SER.readbyte()

Receives a byte from serial port TXD/RXD without waiting for it.

No input parameter.

Return value is a Python integer which is -1 if no data received otherwise is the byte value received. It can be zero.

Example:

b = SER.readbyte()

receives a byte from serial port handling, assigning return value to b.

6.12.5.2.7 SER.SetSpeed(speed, <char format>)

Sets serial port TXD/RXD speed. Default serial port TXD/RXD speed is 9600.

Input parameter speed is a Python string which is the value of the serial port speed. It can be the same speeds as the +IPR command.

NOTE: sending the +IPR command to the device is not affecting the physical serial, when using Python engine you must use this function to set the speed of the port.

Optional Parameter <char format> is a Python string that represents the character format to be used:

first is the number of bits per char (7 or 8), then the parity setting (N - none, E- even, O- odd) and the number of stop bits (1 or 2). Default is "8N1"

Return value is a Python integer which is -1 if an error occurred otherwise is 1.

Example:

b = SER.SetSpeed('115200')

sets serial port speed to 115200, assigning return value to b.

6.12.5.3 GPIO built-in module

GPIO built-in module is the interface between Python core and module internal general purpose input output direct handling.

You need to use GPIO built-in module if you want to set GPIO values from Python script and to read GPIO values from Python script.

You can control GPIO pins also by sending internal 'AT#GPIO' commands using the MDM module, but using the GPIO module is faster because no command parsing is involved, therefore it's use is suggested.

Note that Python core does not verify if the pins are already used for other purposes (IIC module or SPI module) by other functions, it's the applicator responsibility to ensure that no conflict over pins occurs.

If you want to use GPIO built-in module you need to import it first:

import GPIO

then you can use GPIO built-in module methods like in the following example:

a = GPIO.getIOvalue(5)

b = GPIO.setIOvalue(4, 1)

which reads GPIO 5 value and sets GPIO 4 to output with value 1.

More details about GPIO built-in module methods are in the following paragraphs.

6.12.5.3.1 GPIO.setIOvalue(GPIOnumber, value)

Sets output value of a GPIO pin.

First input parameter GPIOnumber is a Python integer which is the number of the GPIO.

Second input parameter value is a Python integer which is the ouput value. It can be 0 or 1.

Return value is a Python integer which is -1 if an error occurred otherwise is 1.

Example:

b = GPIO.setIOvalue(4, 1)

sets GPIO 4 to output with value 1, assigning return value to b.

6.12.5.3.2 GPIO.getIOvalue(GPIOnumber)

Gets input or output value of a GPIO.

Input parameter GPIOnumber is a Python integer which is the number of the GPIO.

Return value is a Python integer which is -1 if an error occurred otherwise is input or output value. It is 0 or 1.

Example:

a = GPIO.getIOvalue(5)

gets GPIO 5 input or output value, assigning return value to b.

6.12.5.3.3 GPIO.setIOdir(GPIOnumber, value, direction)

Sets direction of a GPIO.

First input parameter GPIOnumber is a Python integer which is the number of the GPIO.

Second input parameter value is a Python integer which is the ouput value. It can be 0 or 1. It is only used if direction value is 1.

Third input parameter value is a Python integer which is the direction value. It can be 0 for input or 1 for output.

Return value is a Python integer which is -1 if an error occurred otherwise is 1.

Example:

c = GPIO.setIOdir(4, 0, 0)

sets GPIO 4 to input with value having no meaning, assigning return value to c.

6.12.5.3.4 GPIO.getIOdir(GPIOnumber)

Gets direction of a GPIO.

Input parameter GPIOnumber is a Python integer which is the number of the GPIO.

Return value is a Python integer which is -1 if an error occurred otherwise is direction value. It is 0 for input or 1 for output.

Example:

d = GPIO.getIOdir(7)

gets GPIO 7 direction, assigning return value to d.

6.12.5.4 MOD built-in module

MOD built-in module is the interface between Python and module miscellaneous functions.

You need to use MOD built-in module if you want to generate timers in Python script, to reactivate Python from Python script, etc.

If you want to use MOD built-in module you need to import it first:

import MOD

then you can use MOD built-in module methods like in the following example:

MOD.reactivatePython()

which reactivates Python after next exiting from Python script.

More details about MOD built-in module methods are in the following paragraphs.

6.12.5.4.1 MOD.secCounter()

Returns seconds elapsed since 1 January 1970.

This method is useful for timers generation in Python script.

No input parameter.

Return value is a Python integer which is the value of seconds elapsed since 1 January 1970.

Example:

a = MOD.secCounter()

returns seconds elapsed since 1 January 1970.

6.12.5.4.2 MOD.sleep(sleeptime)

Blocks Python script execution for a given time returning the resources to the system.

Input parameter timesleep is a Python integer which is the time in 1/10 s to block script execution.

No return value.

Example:

MOD.sleep(15)

blocks Python script for 1.5 s.

6.12.5.4.3 MOD.reactivatePython()

Reactivates Python script after exiting from actual Python script.

This method is useful for Python script restart.

The effect of this method is to restart the complete procedure of selecting the Python script to be executed and of executing it.

If you want this method to have the expected effect you need to exit actual Python script as soon as possible after calling it (for example braking while or for loops).

No input parameter.

No return value.

Example:

MOD.reactivatePython()

reactivates Python after next exiting from Python script.

6.12.5.5 IIC built-in module

IIC built-in module is an implementation on the Python core of the IIC bus Master (No Multi-Master) using the "bit-banging" technique.

You need to use IIC built-in module if you want to create one or more IIC bus on the available GPIO pins. This IIC bus handling module is mapped on creation on two GPIO pins that will become the Serial Data and Serial Clock pins of the bus. It can be multi-instantiated (you can create more than one IIC bus over different pins) and the pins used must not be used for other purposes.

Note that Python core does not verify if the pins are already used for other purposes (SPI module or GPIO module) by other functions, it's the applicator responsibility to ensure that no conflict over pins occurs.

If you want to use IIC built-in module you need to import it first:

import IIC

then you can create the new bus over the GPIO pins (for example over the pins GPIO3, GPIO4) and then use IIC built-in module methods like in the following example:

IICbus = IIC.new(3,4)

IICbus.init()

res = IICbus.send('test')

c = IICbus.sendbyte(0x0d)

d = IICbus.readbyte()

which sends 'test' followed by CR and receives data waiting for one second.

NOTE that you must provide external pull-up on SDA line since the line is working as open collector, SCLK instead is driven with a complete push pull.

More details about IIC built-in module object methods are in the following paragraphs.

6.12.5.5.1 IIC.new(SDA_pin, SCL_pin)

Creates a new IIC bus object on the GPIO pins number.

Input parameter SDA_pin, SCL_pin are Python bytes which are the GPIO pin number where the SDA (Serial DAta) and SCL (Serial CLock) lines are mapped.

Return value is the Python custom IIC bus object pointer which then shall be used to interface with the IIC bus created.

Example:

bus1 = IIC.new(3,4)

bus2 = IIC.new(5,6)

This creates two IIC bus, one over the GPIO3 and GPIO4 and one over the GPIO5 and GPIO6.

Available pins for the IIC bus are GPIO3 - GPIO13, while GPIO1 and GPIO2 are not available for IIC.

6.12.5.5.2 IIC object method: init()

Does the first pin initialisation on the IIC bus previously created.

Return value is a Python integer which is -1 if an error occurred otherwise is 1.

Example:

a = bus1.init()

6.12.5.5.3 IIC object method: sendbyte(byte)

Sends a byte to the IIC bus previously created.

Input parameter byte is a Python byte which is the byte to be sent to the IIC bus.

The start and stop condition on the bus are added by the function.

Return value is a Python integer which is -1 if an error occurred otherwise is 1 the byte has been acknowledged by the slave.

Example:

a = bus1.sendbyte(123)

sends byte 123 to the IIC bus , assigning return result value to a.

6.12.5.5.4 IIC object method: send(string)

Sends a string to the IIC bus previously created.

Input parameter string is a Python string which is the string to send to the IIC bus.

Return value is a Python integer which is -1 if an error occurred otherwise is 1 if all bytes of the string have been acknowledged by the slave.

Example:

a = bus1.send('test')

sends string 'test' to the IIC bus , assigning return result value to a.

6.12.5.5.5 IIC object method: dev_read(addr, len)

Receives a string of len bytes from IIC bus device at address addr.

Return value is a Python string which is containing data received.

Example:

a = bus1.read(114,10)

receives a string of 10 bytes from IIC bus device at address 114, assigning it to a.

6.12.5.5.6 IIC object method: dev_write(addr, string)

Sends a string to the IIC bus device at address addr.

Return value is a Python string which is 1 if data is acknowledged correctly, -1 otherwise.

Example:

a = bus1.dev_write(114,'123456789')

sends the string '123456789' to the IIC bus device at address 114, assigning the result to a.

6.12.5.5.7 IIC object method: dev_gen_read(addr, start, len)

Receives a string of len bytes from IIC bus device whose address is addr, starting from address start.

Return value is a Python string which is containing data received.

Example:

a = bus1.read(114,122, 10)

receives a string of 10 bytes from IIC bus device at address 114, starting from address 122 assigning it to a.

6.12.5.5.8 IIC object method: dev_gen_write(addr, start, string)

Sends a string to the IIC bus device whose address is addr, starting from address start.

Return value is a Python string which is 1 if data is acknowledged correctly, -1 otherwise.

Example:

a = bus1.dev_write(114,, 112, '123456789')

sends the string '123456789' to the IIC bus device at address 114, starting from address start, assigning the result to a.

6.12.5.6 SPI built-in module

SPI built-in module is an implementation on the Python core of the SPI bus Master using the "bit-banging" technique.

You need to use SPI built-in module if you want to create one or more SPI bus on the available GPIO pins. This SPI bus handling module is mapped on creation on three or more GPIO pins that

will become the Serial Data In/Out and Serial Clock pins of the bus, plus a number of optional chip select pins up to 8. It can be multi-instantiated (you can create more than one SPI bus over different pins) and the pins used must not be used for other purposes.

Note that Python core does not verify if the pins are already used for other purposes (IIC module or GPIO module) by other functions, it's the applicator responsibility to ensure that no conflict over pins occurs.

If you want to use SPI built-in module you need to import it first:

import SPI

then you can create the new bus over the GPIO pins (for example over the pins GPIO3, GPIO4, GPIO5) and then use SPI built-in module methods like in the following example:

```
SPIbus = SPI.new(3,4,5)

SPIbus.init(0,0)
res = SPIbus.send('test')
c = SPIbus.sendbyte(0x0d)
d = SPIbus.readbyte()
```

which sends 'test' followed by CR and receives data waiting for one second.

More details about SPI built-in module object methods are in the following paragraphs.

6.12.5.6.1 SPI.new(SCLK_pin, MOSI_pin, MISO_pin, <SS0>, <SS1>,…<SS7>)

Creates a new SPI bus object on the GPIO pins number corresponding.

Input parameter SCLK_pin, MOSI_pin and MISO_pin are Python bytes which are the GPIO pin number where the SCLK (Serial CLocK), MOSI (Master Output Slave Input), MISO (Master Input Slave Output) lines are mapped. The same is for the SS0 .. SS9 which are OPTIONAL Python bytes which are the GPIO pin number where the corresponding Slave Select line is mapped. Up to 8 slave select lines can be defined (also none if only 1 slave is used).

Return value is the Python custom SPI bus object pointer which then shall be used to interface with the SPI bus created.

Example:

bus3 = SPI.new(3,4,5)

bus4 = SPI.new(6,7,8,9,10)

This creates two SPI bus, one over the GPIO3, GPIO4, GPIO5 and one over the GPIO6, GPIO7, GPIO8, GPIO9, GPIO10 where the GPIO9 is the Slave 0 select and GPIO10 is the Slave 1 select pin.

Available pins for the SPI bus are GPIO3 - GPIO13, while GPIO1 and GPIO2 are not available for SPI.

6.12.5.6.2 SPI object method: init(CPOL, CPHA)

Does the first pin initialisation on the SPI bus previously created.

Bus clock polarity is controlled by CPOL value:

CPOL = 0 - clock polarity low

CPOL = 1 - clock polarity high

Bus clock phase transmission is controlled by CPHA value:

CPHA = 0 - data bit is clocked/latched on the first edge of the SCLK.

CPHA = 1 - data bit is clocked/latched on the second edge of the SCLK.

Return value is a Python integer which is -1 if an error occurred otherwise is 1.

Example:

a = bus3.init(0,0)

6.12.5.6.3 SPI object method: sendbyte(byte, <SS_number>)

Sends a byte to the SPI bus previously created addressed for the Slave number SS_number whose Slave Select signal is activated.

Input parameter byte is a Python byte which is the byte to be sent to the SPI bus.

Optional Parameter SS_number is a Python byte representing the Slave number to be activated, if not present no slave line is activated.

Return value is a Python integer which is -1 if an error occurred otherwise is 1 the byte has been sent.

Example:

a = bus3.sendbyte(123)

sends byte 123 to the SPI bus , assigning return result value to a.

b=bus4.sendbyte(111,1)

sends byte 111 to the SPI bus activating the Slave Select line of the SS1 device (in our example GPIO10)

6.12.5.6.4 SPI object method: send(string, <SS_number>)

Sends a string to the SPI bus previously created.

Input parameter string is a Python string which is the string to send to the SPI bus.

Optional Parameter SS_number is a Python byte representing the Slave number to be activated, if not present no slave line is activated.

Return value is a Python integer which is -1 if an error occurred otherwise is 1 if all bytes of the string have been sent.

Example:

a = bus3.send('test')

sends string 'test' to the SPI bus , assigning return result value to a.

6.12.5.6.5 SPI object method: read(len, <SS_number>)

Receives a string of len bytes from SPI bus device at Slave Select number SS_number.

Optional Parameter SS_number is a Python byte representing the Slave number to be activated, if not present no slave line is activated.

Return value is a Python string which is containing data received.

Example:

a = bus4.read(10,0)

receives a string of 10 bytes from SPI bus device on SS0 line, assigning it to a.

6.12.5.6.6 SPI object method: readwrite(string, len, <SS_number>)

Send the string "string" and contemporaneously receives a string of len bytes from SPI bus device at Slave Select number SS_number.

Optional Parameter SS_number is a Python byte representing the Slave number to be activated, if not present no slave line is activated.

Return value is a Python string which is containing data received.

Example:

a = bus4.readwrite("hello",10,0)

send the string "hello" and receives a string of 10 bytes from SPI bus device on SS0 line, assigning it to a.

6.12.6 Executing a Python script

The steps required to have a script running by the python engine of the module are:

- write the python script
- download the python script into the module NVM
- enable the python script
- execute it..

6.12.6.1 Write Python script

A Python script is a simple text file, it can be written with any text editor but for your convenience a complete Integrated Development Environment (IDE) is included in a software package that Telit provides called *Telit Python Package*.

Remembering the supported features described in 6.12.4, it is simple to write the script and test it directly from the IDE.

The following is the "Hello Word" short Python script that sends the simplest AT command to the AT command parser and waits for response, then ends.

```
import MDM
print 'Hello World!'
result = MDM.send('AT\r', 0)
print result
c = MDM.receive(10)
print c
```

6.12.6.2 Download Python script

The Script can be downloaded in the module using the #WSCRIPT command.

In order to guarantee your company know-how, you have the option to Hide the script text so that the #RSCRIPT command does not return the text of the script and keeps it "confidential", you can see only the name of the script with the #LSCRIPT command.

Remember that if you chose to hide the script text it's your responsibility to keep information on what is executing the module, for example by naming the script depending from the application and version of the script.

In order to download the script, first you have to choose a name for your script in the module taking care that:

- it must have extension .py;
- the maximum allowed length is 16 characters;
- script name is case sensitive.

Then you have to find out the exact size in bytes of the script (for example right clicking on the file and selecting “properties”)

The script download is done regardless the previous serial settings at 115200 baud 8-N-1 with hardware flow control active.

For example (script name and size are examples):

AT#WSCRIPT=”a.py”,110

wait for the prompt

>>>

and use “Send Text file” with ASCII Setup: Send line ends with line feeds in HyperTerminal enabled.

Wait for download result: OK.

6.12.6.3 Enable Python script

Select the Python script which will be executed (the enabled script) from the next start-up and on using the AT#ESCRIPT command.

First choose the script you want to enable between the ones you downloaded:

AT#LSCRIPT? can help you checking the names of the scripts;

for example:

AT#ESCRIPT=”a.py”

Wait for enable result: OK.

6.12.6.4 Execute Python script

The Python script you downloaded to module and enabled is executed at every module power on if the DTR line is sensed LOW (2.8V at the GM862-PYTHON DTR pin - RS232 signals are inverted -) at start-up, (then no AT command interface is connected to the modem port) and if the script name you enabled matches one of the script names of the scripts you downloaded.

The Python script is executed with –v -S -OO options.

In order to gain again the AT command interface on the modem physical port (for example to update locally a new script) the module shall be powered on with the DTR line HIGH (0V at the GM862-PYTHON DTR pin) so that the script is not executed and the Python engine is stopped.

The real execution of the Python script is delayed from the power on due to the time needed by Python to parse the script. The longer is the script, the longer is this delay.

Note that only the running script is compiled at run time, all the others that this script may include are compiled once and the compiled result is saved in the NVM as a file with extension .pyo.

This delay can be greatly reduced with a simple stratagem:

- type your script normally, and include the main loop in a function, for example "main()", save it to the NVM of the module with a known name, for example appl.py
- write a new script that includes the previous file object, for example "include appl", and this file should call only the main function of the appl.py script, for example appl.main().

In this way the first time the script is executed the imported files will be compiled and the result saved as compiled .pyo files (don't delete them during normal operations, but remember to delete them if you change the corresponding .py script otherwise your changes will not take effect). From the next start-up and on the imported files will not be anymore compiled and script execution delay is greatly reduced.

This stratagem is useful also for long complex scripts, that may run out of memory during compilation; splitting the script into several smaller scripts containing part of the functions/objects definitions will separate the compilation and allow for much bigger script usage.

6.12.6.5 Debug Python script

The debug of the active Python script can be done both on the emulated environment of the Telit Python Package (refer to its documentation) or directly on the target with the second serial port pin EMMI TX (actually a not translated RS232 serial port as the RXD pin).

Connect to the module serial port EMMI TX at 9600 8-N-1 with no flow control active.

Now you can see all Python outputs to stdout and stderr:

- Python information messages (for example the version);
- Python error information;
- results of all Python “print” statements.

6.12.7 Script Management commands

6.12.7.1 #WSCRIPT - Write script command

#WSCRIPT - Write script command	
Execute command	
Read command	
Write command **AT#WSCRIPT = "<script_name>",<size>, [,<hidden>]**	This command inserts a script text and save it with the name <script-name> in the Non Volatile Memory of the GM862-PYTHON. The script text should be sent using Raw Ascii file transfer. It is important to set properly the port settings. In particular: Baud rate: 115200 bps Flow control: hardware. Parameter: < script-name > - file name (max 16 chars - case sensitive) <size> - file size (number of bytes) <hidden> - file hidden attribute **0 - file content is readable with #RSCRIPT -default -** 1 - file content is hidden, #RSCRIPT command will report empty file. The device responds to the command with the prompt '>' and awaits for the script file text for <size> bytes. The operations completes when all the bytes are received. If script storing is successfully sent, then the response is OK. If script writing fails for some reason, an error code is reported **NOTE. The script name should be passed between quotes and ALL Executable Scripts files MUST have .py extension - Script names are Case sensitive.** NOTE2: When sending the script verify that the line terminator is <cr><lf> and that your terminal program does not change it. NOTE3: with the hidden attribute it is possible to protect your script from being viewed and copied, only the file name can be viewed, its content is hidden even if the file is still being run correctly. It's your care to maintain knowledge on what the script file contains.

#WSCRIPT - Write script command	
	NOTE4: This command can also be used to write any text file in the GM862-PYTHON memory (not script files), for example application data or settings files with a different extension than .py.
Test command	
Example	AT#WSCRIPT="First.py ",54,0 ➢ (here receive the prompt) import MDM MDM.send('AT\r',10) Ans=MDM.receive(20) OK Script has been stored.
Reference	Telit specifications
SW release	Version D

6.12.7.2 #ESCRIPT - Select Active script command

#ESCRIPT - Select Active script command	
Execute command	
Read command **AT#ESCRIPT?**	Reports the current value of the parameter <script_name>
Write command **AT#ESCRIPT = "<script_name>"**	This command selects the name of the script that will be executed by the Easy Script interpreter at the start-up. The script will be executed at start-up only if the DTR line is found LOW during initial start-up (that is: COM is not open on a PC), otherwise the Easy Script interpreter will not execute and the GM862-PYTHON will behave normally answering only to AT commands on the serial port. Parameter: < script-name > - file name (max 16 chars - case sensitive) NOTE. The script name should be passed between quotes and ALL Scripts files MUST have .py extension - Script names are Case sensitive. NOTE2: The <script_name> must match with a <script_name> written with the AT#WSCRIPT in order to have it run. The command does not check whether the script <script_name> does exist in the NVM of the GM862-PYTHON. If the file <script_name> is not present at the start-up then the Script Interpreter WILL NOT EXECUTE.
Test command	
Example	AT#ESCRIPT="First.py " OK Script First.py will be executed at the next start-ups if DTR is found LOW.
Reference	Telit specifications
SW release	Version D

6.12.7.3 #RSCRIPT - Read script command

#RSCRIPT - Read script command	
Execute command	
Read command	
Write command **AT#RSCRIPT = "<script_name>"**	This command reports the script file <script_name> content. Parameter: < script-name > - file name (max 16 chars - case sensitive) NOTE. The script name should be passed between quotes and ALL Executable Scripts files MUST have .py extension - Script names are Case sensitive. NOTE2: If the file <script_name> is not present an error code is reported. NOTE3: If the file <script_name> was saved with the hidden attribute, then an empty file is reported with the OK result code.
Test command	
Example	AT#RSCRIPT="First.py " import MDM MDM.send('AT\r',10) Ans=MDM.receive(20) OK
Reference	Telit specifications
SW release	Version D

6.12.7.4 #LSCRIPT - List script names command

#LSCRIPT - List script names command	
Execute command	
Read command **AT#LSCRIPT?**	This command reports the list of script files names currently saved into the NVM and the available free NVM memory in the format: #LSCRIPT: <script_name1>,<size> #LSCRIPT: <script_name2>,<size> … #LSCRIPT: <script_nameN>.,<size> free bytes: <free_NVM_size> Parameters: < script-name > - file name (max 16 chars - case sensitive) < size > - size of script in bytes < free_NVM_size > - size of available NVM memory in bytes
Write command	
Test command	
Example	AT#LSCRIPT? #LSCRIPT: First.py,51 #LSCRIPT: Second.py,178 #LSCRIPT: Third.py,95 free bytes: 20000 OK
Reference	Telit specifications
SW release	Version D

6.12.7.5 #DSCRIPT - Delete script command

#DSCRIPT - Delete script command	
Execute command	
Read command	
Write command **AT#DSCRIPT= <script_name>**	This command deletes the script file with the name <script_name> from NVM memory. Parameter: <script_name> (max 16 chars - case sensitive) NOTE. The script name should be passed between quotes - Script names are Case sensitive. NOTE2: If the file <script_name> is not present an error code is reported.
Test command	
Example	AT#DSCRIPT="Third.py" OK
Reference	Telit specifications
SW release	Version D

6.12.8 System management

6.12.8.1 #REBOOT - Reboot command

#REBOOT - Reboot command	
Execute command **AT#REBOOT**	This command reboots immediately the unit. It can be used to reboot the system after a remote update of the script in order to have the new one running.
Read command	
Write command	
Test command	
Note	This command does not return result codes.
Example	AT#REBOOT …. Module Reboots …
Reference	Telit specifications
SW release	Version D

7 Conformity Assessment Issues

The Telit GM862-PCS/-GPRS/-GSM modules are assessed to be conform to the R&TTE Directive as stand-alone products, so If the module is installed in conformance with Dai Telecom installation instructions require no further evaluation under Article 3.2 of the R&TTE Directive and do not require further involvement of a R&TTE Directive Notified Body for the final product.
In all other cases, or if the manufacturer of the final product is in doubt then the equipment integrating the radio module must be assessed against Article 3.2 of the R&TTE Directive**.**
In all cases assessment of the final product must be made against the Essential requirements of the R&TTE Directive Articles 3.1(a) and (b), safety and EMC respectively, and any relevant Article 3.3 requirements.

The Telit GM862-PCS/-GPRS/-GSM modules are conform with the following European Union Directives:

- ⊡ R&TTE Directive 1999/5/EC (Radio Equipment & Telecommunications Terminal Equipments)
- ⊡ Low Voltage Directive 73/23/EEC and product safety
- ⊡ Directive 89/336/EEC for conformity for EMC

In order to satisfy the essential requisite of the R&TTE 99/5/EC directive, the GM862PCS module is compliant with the following standards:

- GSM (Radio Spectrum). Standard: EN 301 511 and 3GPP 51.010-1
- EMC (Electromagnetic Compatibility). Standards: EN 301 489-1 and EN 301 489-7
- LVD (Low Voltage Directive) Standards: EN 60 950

In this document and the Hardware User Guide, Software User Guide all the information you may need for developing a product meeting the R&TTE Directive is included.

Furthermore the Telit GM862-PCS module is FCC Approved as module to be installed in other devices. This device is to be used only for fixed and mobile applications. If the final product after integration is intended for portable use, a new application and FCC is required.

The Telit GM862-PCS module is conform with the following US Directives:

- ⊡ Use of RF Spectrum. Standards: FCC 47 Part 24 (GSM 1900)
- ⊡ EMC (Electromagnetic Compatibility). Standards: FCC47 Part 15

To meet the FCC's RF exposure rules and regulations:

- The antenna(s) used for this transmitter must be installed to provide a separation distance of at least 20 cm from all the persons and must not be co-located or operating in conjunction with any other antenna or transmitter.
- The antenna(s) used for this module must not exceed 7 dBi for mobile and fixed or mobile operating configurations.
- Users and installers must be provided with antenna installation instructions and transmitter operating conditions for satisfying RF exposure compliance.

Manufacturers of mobile, fixed or portable devices incorporating this module are advised to clarify any regulatory questions and to have their complete product tested and approved for FCC compliance.

7.1GM862-GPRS: Conformity Assessment

DAI TELECOM S.p.A.
Dai Telecom Group

DECLARATION OF CONFORMITY

We,

Dai Telecom SpA

Of:

V.le Stazione di Prosecco 5/b
34010 Sgonico TS
Italy

declare under our sole responsibility that the product

GSM/GPRS 900/ 1800 Data Module type GM862GPRS

to which this declaration relates is in conformity with all the essential requirements of Directive 1999/05/EC

We hereby also declare that all essential [radio] test suites have been carried out and that the above named product is in conformity to all the essential requirements of Directive 1999/5/EC.

The conformity assessment procedure referred to in Article 10 and detailed in Annex IV of Directive 1999/5/EC has been followed with the involvement of the following Notified Body:

BABT, Claremont House, 34 Molesey Road, Walton-on-Thames, KT12 4RQ, UK

Identification mark: **0168**

The technical documentation relevant to the above equipment will be held at:

Dai Telecom SpA
V.le Stazione di Prosecco 5/b
34010 Sgonico TS
Italy

Trieste, ***28/10/2003***

Andrea Antonel
(Quality System Manager)

MOD.003 08/03 REV.5

Accredited by the Council for Accreditation

BABT

Certificate

This certificate is issued to

DAI Telecom S.p.A

of

Viale Stazione di Prosecco 5/B
Trieste
34010
Italy

to certify that the Equipment known as

GM862GPRS

conforms to the essential requirements of the Council Directive 1999/5/EC of the European Parliament and of the Council on the basis of the Technical Construction File number 19447_GM862GPRS_rev1 in relation to the essential requirements of Articles 3.1(a), 3.1(b) & 3.2 of the Directive.

Signed:

On Behalf of BABT

Issue Date: 17 December 2003

Number: NC/12061 Issue: 01

This certificate is issued by BABT.
The conditions for the validity of this certificate, if any, are listed in the Annex.
This certificate is not transferable and remains the property of BABT.

BABT, Claremont House, 34 Molesey Road, Walton-On-Thames, Surrey KT12 4RQ, U.K.
National Tel: 01932 251251 Fax: 01932 251252
International Tel: +44 1932 251251 Fax: +44 1932 251252
Web address: http://www.babt.com email: CustomerService@babt.com

7.2 GM862-GSM: Conformity Assessment

DAI TELECOM S.p.A.
Dai Telecom Group

DECLARATION OF CONFORMITY

We,
Dai Telecom SpA

Of:
V.le Stazione di Prosecco 5/b
34010 Sgonico TS
Italy

declare under our sole responsibility that the product

GSM/GPRS 900/ 1800 Data Module type GM862GSM

to which this declaration relates is in conformity with all the essential requirements of Directive 1999/05/EC

We hereby also declare that all essential [radio] test suites have been carried out and that the above named product is in conformity to all the essential requirements of Directive 1999/5/EC.

The conformity assessment procedure referred to in Article 10 and detailed in Annex IV of Directive 1999/5/EC has been followed with the involvement of the following Notified Body:

BABT, Claremont House, 34 Molesey Road, Walton-on-Thames, KT12 4RQ, UK

Identification mark: **0168**

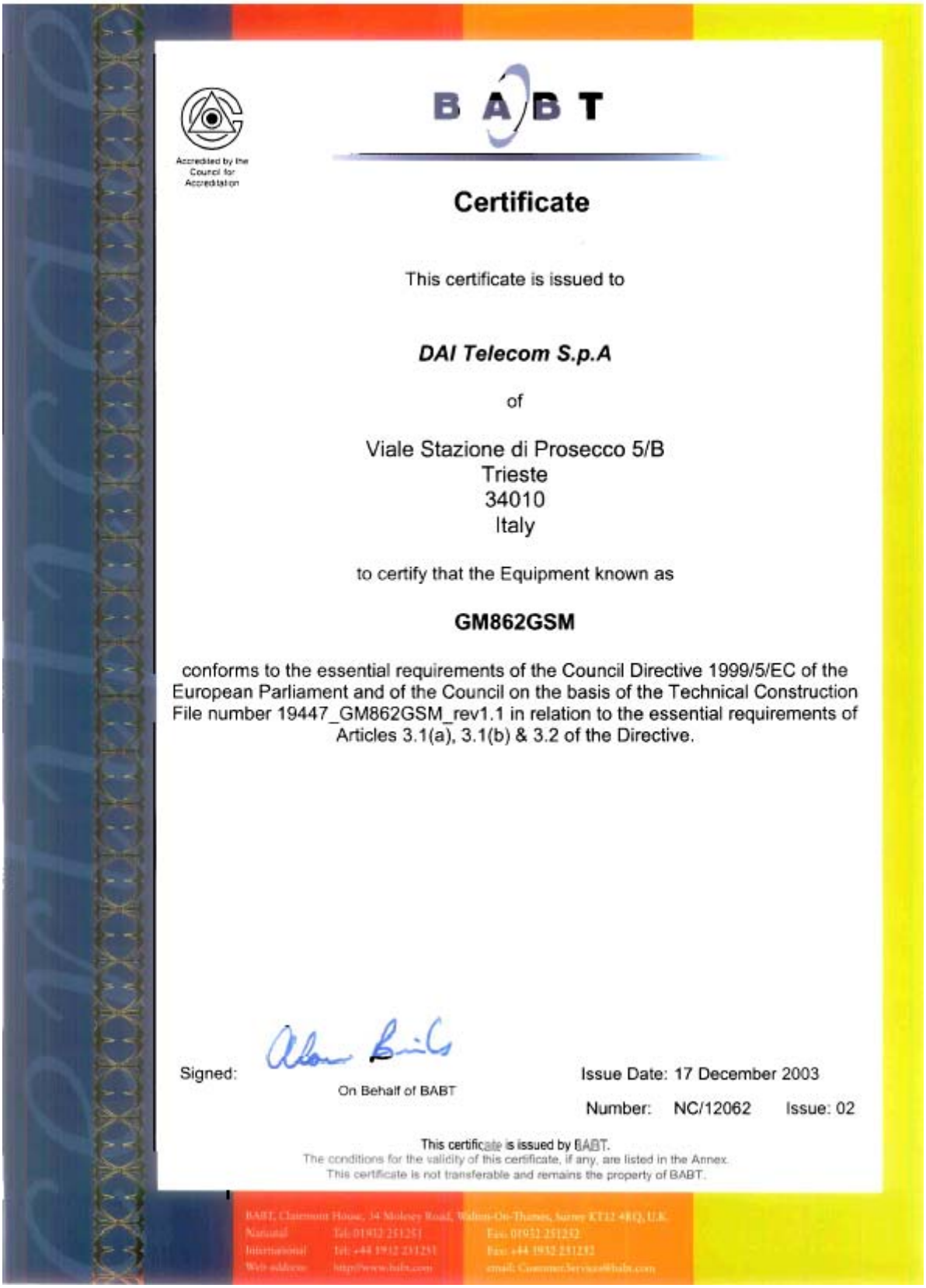
The technical documentation relevant to the above equipment will be held at:

Dai Telecom SpA
V.le Stazione di Prosecco 5/b
34010 Sgonico TS
Italy

Trieste, *28/10/2003*

Andrea Antonel
(Quality System Manager)

MOD 003 08/03 REV 5



Accredited by the Council for Accreditation

BABT

Certificate

This certificate is issued to

DAI Telecom S.p.A

of

Viale Stazione di Prosecco 5/B
Trieste
34010
Italy

to certify that the Equipment known as

GM862GSM

conforms to the essential requirements of the Council Directive 1999/5/EC of the European Parliament and of the Council on the basis of the Technical Construction File number 19447_GM862GSM_rev1.1 in relation to the essential requirements of Articles 3.1(a), 3.1(b) & 3.2 of the Directive.

Signed:

On Behalf of BABT

Issue Date: 17 December 2003

Number: NC/12062 Issue: 02

This certificate is issued by BABT.
The conditions for the validity of this certificate, if any, are listed in the Annex.
This certificate is not transferable and remains the property of BABT.

BABT, Claremont House, 34 Molesey Road, Walton-On-Thames, Surrey KT12 4RQ, U.K.
National Tel: 01932 251251 Fax: 01932 251252
International Tel: +44 1932 251251 Fax: +44 1932 251252
Web address: http://www.babt.com email: CustomerServices@babt.com

7.3GM862-PCS: Conformity Assessment

DAI TELECOM S.p.A.
Dai Telecom Group

DECLARATION OF CONFORMITY

We,
Dai Telecom SpA

Of:
V.le Stazione di Prosecco 5/b
34010 Sgonico TS
Italy

declare under our sole responsibility that the product

GSM/GPRS/PCS 900/ 1800/1900 Data Module type GM862PCS

to which this declaration relates is in conformity with all the essential requirements of Directive 1999/05/EC

We hereby also declare that all essential [radio] test suites have been carried out and that the above named product is in conformity to all the essential requirements of Directive 1999/5/EC.

The conformity assessment procedure referred to in Article 10 and detailed in Annex IV of Directive 1999/5/EC has been followed with the involvement of the following Notified Body:

BABT, Claremont House, 34 Molesey Road, Walton-on-Thames, KT12 4RQ, UK

Identification mark: **0168**

The technical documentation relevant to the above equipment will be held at:

Dai Telecom SpA
V.le Stazione di Prosecco 5/b
34010 Sgonico TS
Italy

Trieste, ***18/11/2003***

Andrea Antonel
(Quality System Manager)

MOD.003 08/03 REV.5

Accredited by the Council for Accreditation

BABT

Certificate

This certificate is issued to

DAI Telecom S.p.A

of

Viale Stazione di Prosecco 5/B
Trieste
34010
Italy

to certify that the Equipment known as

GM862PCS

conforms to the essential requirements of the Council Directive 1999/5/EC of the European Parliament and of the Council on the basis of the Technical Construction File number 19296_GM862PCS_rev1 in relation to the essential requirements of Articles 3.1(a), 3.1(b) & 3.2 of the Directive.

Signed: Alan Bibb
On Behalf of BABT

Issue Date: 11 December 2003

Number: NC/12058 Issue: 01

This certificate is issued by BABT.
The conditions for the validity of this certificate, if any, are listed in the Annex.
This certificate is not transferable and remains the property of BABT.

BABT, Claremont House, 34 Molesey Road, Walton-On-Thames, Surrey KT12 4RQ, U.K.
National Tel: 01932 251251 Fax: 01932 251252
International Tel: +44 1932 251251 Fax: +44 1932 251252
Web address: http://www.babt.com email: CustomerServices@babt.com

7.4GM862-Python: Conformity Assessment

UNDER APPROVAL

7.5 GM862-PCS FCC Equipment Authorization

TCB

GRANT OF EQUIPMENT AUTHORIZATION

TCB

Certification
Issued Under the Authority of the
Federal Communications Commission
By:

CETECOM ICT Services GmbH
Untertuerkheimer Strasse 6-10
D-66117 Saarbruecken,

Date of Grant: 12/15/2003

Application Dated: 12/15/2003

DAI Telecom S.p.A
Viale Stazione di Prosecco 5/b
Trieste, 34010
Italy

Attention: Andrea Fragiacomo , Ing.

NOT TRANSFERABLE

EQUIPMENT AUTHORIZATION is hereby issued to the named GRANTEE, and is VALID ONLY for the equipment identified hereon for use under the Commission's Rules and Regulations listed below.

FCC IDENTIFIER:	RI7GM862P
Name of Grantee:	DAI Telecom S.p.A
Equipment Class:	PCS Licensed Transmitter
Notes:	GSM Module

Grant Notes	FCC Rule Parts	Frequency Range (MHZ)	Output Watts	Frequency Tolerance	Emission Designator
	24E	1850.2 - 1909.8	0.845	0.0537 PM	300KGXW

Module Approval. Power Output listed is conducted. The antenna gain, including cable loss, must not exceed 7 dBi for fixed-mounted or mobile operating configurations, as defined in 2.1091 and 1.1307 of the rules for satisfying RF exposure compliance. Separate approval is required for portable operating configurations, as defined in 2.1093 of the rules. The final product operating with this transmitter must include operating instructions and applicable warnings, as described in this filing, for end-users and installers to satisfy RF exposure compliance requirements. OEM integrators must be informed of the specific requirements. This module contains 900 MHz and 1800 MHz GSM functions that are not operational in U.S. Territories. This filing is only applicable to GSM 1900 MHz operations.

7.6 GM862-PCS IC Equipment Certification

Industry Canada Industrie Canada

Submission No. ➢ 102134

TECHNICAL ACCEPTANCE CERTIFICATE

CERTIFICAT D'ACCEPTABILITÉ TECHNIQUE

CERTIFICATION No. / No. DE CERTIFICATION ➢ 5131A-GM862P

ISSUED TO / DÉLIVRÉ À ➢ DAI TELECOM S.p.A.

TYPE OF EQUIPMENT / GENRE DE MATÉRIEL ➢ Data terminal

TRADE NAME AND MODEL / MARQUE ET MODÈLE ➢ GM862PCS

FREQUENCY RANGE / BANDE DE FRÉQUENCES	EMISSION TYPE / GENRE D'ÉMISSION	R.F. POWER / PUISSANCE H.F.	SPECIFICATION / ISSUE / SPÉCIFICATION / ÉDITION
1.8502 to 1.9098 GHz	300KGXW	3.00 mW	RSS133 2

Certification of equipment means only that the equipment has met the requirements of the above noted specifications. License applications, were applicable to use certified equipment, are acted on accordingly by the issuing office and will depend on the existing radio environment, service and location of operation.

L'homologation de matériel terminal signifie seulement qu'il est conforme aux exigences du cahier des charges mentionné ci-dessus. Les demandes de licence, le cas échéant en vue de l'utilisation de matériel certifié seront traitées en conséquence par le bureau chargé de délivrer les dites licences, en tenant compte du milieu radioélectrique ambiant, du service radio existant et de l'emplacement de la station.

This certificate is issued on condition that the holder complies and will continue to comply with the requirements of the radio standards specifications and procedures issued by the Department.

Le présent certificat est délivré à condition que le détenteur se conforme et continue à se conformer aux cahiers des charges et procédures sur les normes radioélectriques publiées par le ministère.

ISSUED UNDER THE AUTHORITY OF MINISTER OF INDUSTRY
DELIVRÉ AVEC L'AUTORISATION DU MINISTRE DES INDUSTRIES

DATE **November 4 2004**

FOR

Bob Corey
DIRECTOR GENERAL
SPECTRUM ENGINEERING

DIRECTEUR GÉNÉRAL
GÉNIE DU SPECTRE

Canada

8 GM862 Family Technical Support

Telit technical support to Telit GM862-Family modules customer is included into the dedicated Website (www.GM862.com) and official Website (www.telit.net) which contains also all available technical documentation download, application examples, Telit engineering support accessible via selective E-Mail (ts-gm862@telit.net) service with 48 hr replies assured.

9 List of acronyms

ACM	Accumulated Call Meter
ASCII	American Standard Code for Information Interchange
AT	Attention commands
CB	Cell Broadcast
CBS	Cell Broadcasting Service
CCM	Call Control Meter
CLIP	Calling Line Identification Presentation
CLIR	Calling Line Identification Restriction
CMOS	Complementary Metal-Oxide Semiconductor
CR	Carriage Return
CSD	Circuit Switched Data
CTS	Clear To Send
DAI	Digital Audio Interface
DCD	Data Carrier Detected
DCE	Data Communications Equipment
DRX	Data Receive
DSR	Data Set Ready
DTA	Data Terminal Adaptor
DTE	Data Terminal Equipment
DTMF	Dual Tone Multi Frequency
DTR	Data Terminal Ready
EMC	Electromagnetic Compatibility
ETSI	European Telecommunications Equipment Institute
FTA	Full Type Approval (ETSI)
GPRS	General Radio Packet Service
GSM	Global System for Mobile communication
HF	Hands Free
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IRA	Internationale Reference Alphabet
ITU	International Telecommunications Union
IWF	Inter-Working Function

LCD	Liquid Crystal Display
LED	Light Emitting Diode
LF	Linefeed
ME	Mobile Equipment
MMI	Man Machine Interface
MO	Mobile Originated
MS	Mobile Station
MT	Mobile Terminated
OEM	Other Equipment Manufacturer
PB	Phone Book
PDU	Protocol Data Unit
PH	Packet Handler
PIN	Personal Identity Number
PLMN	Public Land Mobile Network
PUCT	Price per Unit Currency Table
PUK	PIN Unblocking Code
RACH	Random Access Channel
RLP	Radio Link Protocol
RMS	Root Mean Square
RTS	Ready To Send
RI	Ring Indicator
SCA	Service Center Address
SIM	Subscriber Identity Module
SMD	Surface Mounted Device
SMS	Short Message Service
SMSC	Short Message Service Center
SS	Supplementary Service
TIA	Telecommunications Industry Association
UDUB	User Determined User Busy
USSD	Unstructured Supplementary Service Data

10 Document Change Log

Revision	Date	Changes
ISSUE#1	11/07/02	initial release.
ISSUE#2	08/01/03	Added GM862-GPRS conformity assessment certificate.
ISSUE#3	29/01/03	Par. 2.10.2 Sim Reader updated.
ISSUE#4	11/02/03	Par. 6.9 Easy GPRS Extension added.
ISSUE#5	03/03/03	Par. 6.2.4 Command issuing timing added. Par. 6.9.2.3 Defining the Internet peer to be contacted updated. Par. 6.9.2.3 Defining the Internet peer to be contacted updated. Par. 6.9.2.4 Open the connection with the internet host updated. Par. 6.9.2.5 Close the Socket and deactivate the context updated. Par. 6.9.4.2 #PASSW – Authentication Password control updated. Par. 6.9.4.4 #DSTO – Data Sending Time Out control updated. Par. 6.9.4.6 #SKTSET – Socket definition control updated. Par. 6.9.4.7 #SKTOP – Socket Open command updated. Par. 6.9.4.9 Socket Parameters Save Command added. Par. 6.9.4.10 Socket Parameters Reset Command added.
ISSUE#6	06/03/03	Change all Finmek Telit references to DAI Telecom.
ISSUE#7	09/05/03	Par. 1 Overview updated. Par. 2.4 Operating Frequency updated. Par. 2.5 Transmitter output power updated. Par. 2.6 Reference sensitivity updated. Par. 2.7 Antenna updated. Par. 2.10.1 Speech Coding updated. Par. 2.10.2 Sim Reader updated. Par. 2.10.5 Data/fax transmission updated. Par. 2.14 Interface connectors on GM862 pull-up resistors updated. Par. 6.2.1 Command lines. Par. 6.2.3 Command Response Timeout updated. Par. 6.2.4 Command issuing timing updated. Par. 6.2.5 Factory Profile and parameters stored in the profile added. Par. 6.3 Command availability table updated. Par. 7 Conformity Assessment Issues updated. Par. 7.1 GM862-GPRS: Conformity Assessment updated. Par. 7.2 GM862-GSM: Conformity Assessment updated. Par. 8 Technical support updated.
ISSUE#8	03/06/03	Par. 1 Overview updated. Par. 2.10 Embodied Battery charger added.
ISSUE#9	14/07/03	Par. 6.3 Command availability table updated. Par. 6.7.1.21 #BND – Select Band (DCS 1800 or PCS 1900) command added. Par. 6.5.8 Commands for battery charger added. Par. 2.10 Embodied Battery charger updated.
ISSUE#10	25/11/03	Par. 6.3 Command availability table updated. Par. 6.10 Easy Camera Custom Commands added.
ISSUE#11	10/12/03	Par. 6.3 Command availability table updated. Par. 6.10 Easy Camera Custom Commands updated. Par. 5 Service and firmware update modified.
ISSUE#12	12/12/03	Par. 7.3 GM862-PCS Conformity Assessment updated.
ISSUE#13	17/12/03	Par. 7 Conformity Assessment issues updated. Par. 6.5.4.2 +CFUN Set phone functionality (Power Saving Management) updated.
ISSUE#14	22/01/04	Par. 6.10.1.5 Command change name from #CSQUAL to #CAMQUA. Par. 6.10.1.6 Command change name from #CSMODE to #CMODE.
ISSUE#15	24/05/04	Par. 2.15 ON/OFF pin21 on Molex 52991-0508 is pulled up with VBATT. Par. 6.11 Easy Scan Extension added. Par. 6.6.3.1 +CNMI - new message indication to Terminal Equipment updated. Par. 6.1.1 #CGMM, #CGMI, #CGMR, #CGSN, #CIMI added. Par. 6.9.1.5 #OBJL - Object list added. Par. 6.9.1.6 #OBJR - Object read added.
ISSUE#16	14/06/04	Par. 2.8 Supply Voltage, Note added Par. 2.9 Power consumption updated;

		Par. 6.2.5 Factory Profile and parameters stored in the profile changed; Par. 6.3 Command availability table updated; Par. 6.3.1.1 &F - restore factory configuration changed; New Par. 6.3.1.2 &F1 restore full factory profile configuration added and inserted; Par. 6.3.1.3 Z - Soft Reset updated; Par. 6.3.1.5 &Y - designate a default reset profile updated; Par. 6.3.1.6 &W - store current configuration updated; Par. 6.3.3.9 &P - pulse dial make/break ratio removed; Par. 6.4.2.6 +CSNS single numbering scheme added. Par. 6.4.3.3 +COPS - operator selection changed; Par. 6.4.3.8 +CCFC - call forwarding number and conditions updated; Par. 6.4.3.10 +CHLD - call holding services changed; Par. 6.4.5.1 +CMEE - report mobile equipment error updated; Par 6.4.5.2 Added new Easy GPRS and Easy Camera related errors (+CME ERROR). Par. 6.4.8.1 +CBC - Battery Charge changed; Par. 6.6.1.17 #MONI - Monitor Cells updated; Par. 6.9.2.6 #ESAV - Email Parameters Save Command added; Par. 6.9.2.7 #ERST - Email Parameters Reset Command added; Par. 6.10 Easy Scan Extension added;
ISSUE#17		Not released.
ISSUE#18	09/12/04	Par. 1 Overview changed - added GM862-PYTHON product Par. 2.15 Interface connectors on GM862-GPRS - updated with JDR. Par. 6.2 AT Command Syntax updated Par. &Z - store telephone number in the Telit GM862-GPRS module internal phonebook updated; Par. 6.3.4.1 +MS - modulation control updated; Par. 6.4.3.10 +CHLD - call holding services changed - call party added; Par. 6.4.3.12 +CAOC - advice of charge updated; Par. 6.4.4.8 +CPBW - write phonebook entry updated; Par. 6.4.3.8 +CCFC - call forwarding number and conditions - updated; Par. 6.4.4.9 +CCLK - Clock Management updated; Par. 6.4.4.10 +CALA - Alarm Management updated; Par. 6.8 Enhanced Easy GPRS - new enhanced version added; Par. 6.8.6.9 #SKTCT – Socket TCP Connection Timeout added; Par. 6.11 Jammed Detect & Report Extension added Par. 6.12 Easy Script Extension - Python interpreter added
ISSUE#19	31/03/05	Par. 1 Overview updated 2.12 EMC updated. Par.6.3.1.22/23 added L and M commands Monitor speaker Loudness and Control Par. 6.12.5.1.7 Ser.Setspeed updated. Updated GM862 Connector PIN-OUT Par. 7.5 GM862-PCS IC Equipment Certification added. Added par.6.3.1.19 &V3 Par. 6.3.4.3 \N Par. 6.3.2.9 &Q Par. 6.3.1.17 updated S register nr 0..38 Par. 6.3.4.2 updated <n> = 0,1,2 Par.6.4.2.2 updated <T4> = 0 – re sequencing period T4 and default value = 0 Par 6.3.2.7 updated <n> = 3 C108/1 operation is enabled <n> = 4 C108/1 operation is disabled Par 6.2.5 removed +CMOD command Added par.7.4 GM862-Python: Conformity Assessment Par. 8 changed technical service reply time. Par. 5 Removed any reference to Winzip Added par. 6.4.3.2 +COPN command and par. 6.3.1.23 +GCI command Par. 6.4.3.14 updated +CSSN command Par. 6.4.4.13 updated +CLVL command Par. 6.4.2.1 updated +CBST command Par. 6.3.2.15 updated +ICF command Par.6.6.1.8 updated #SRP command Par.6.3 Updated Command compatibility table

