Telit
EZ10 TERMINAL FAMILY
EZ10-GPRS
EZ10-PCS with PYG Option

GSM Wireless Modem
Product Description

Contents

DISCLAIMER

The information contained in this document is the proprietary information of Telit Communications S.p.A.

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

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

Telit Communications S.p.A. reserves the right to make modifications, additions and deletions to this document at any time and without notice.

1 Overview

Aim of this document is the description of features, functions and interfaces of the **Telit EZ10 TERMINAL FAMILY of wireless GSM/GPRS cellular modems.** Since these Terminals are ready for use as a subsystem for wireless connections, Safety Information and basic instructions for taking the EZ10 into operation are included, as well as guidance to other documentation and practical hints for the first steps.

All of the **Telit EZ10 Terminals** are GSM Mobile Stations (MS class B) and capable of using the GSM Circuit Switched Data (CSD), Fax and Short Message Services (SMS).

The frequency bands, GPRS class, availability for GSM Voice Services and the configuration of the RJ11 connector in relation to the names of the products are reported in the following table:

Frequency Bands	GPRS	GSM Voice	RJ11 AUX Connector	Product Name
EGSM900/1800 MHz Dual-Band	Class 8	Yes	AUDIO	**Telit EZ10-GPRS**
EGSM900/1800/ PCS1900 MHz Tri-Band	Class 10	NO	GPIO	**Telit EZ10-PCS** (with PYG Option)

For the use of GSM **Voice** Services, an external microphone, earphone or amplified loudspeaker needs to be connected to the RJ11 AUX=AUDIO connector of the **Audio** version of the EZ10 Terminal.

The **PYG Option** allows to perform stand-alone operations with the **EASY SCRIPT** feature, interpreting user-defined scripts in **PYTHON** language, e.g. to read external sensors connected to the General Purpose Inputs / Outputs on the RJ11 connector. Refer to the chapter about Software Features in this Product Description.

On the **AUDIO** version of the EZ10 Terminal Models, the 6-pin RJ11 **AUX** connector is configured with a microphone input and earphone output, whereas the **PYG** version have digital General Purpose Inputs/Outputs (**GPIO**) instead. The following table summarizes the possible interface configurations (for details refer to the dedicated chapters of this document):

	Telit EZ10-GPRS (AUDIO)	EZ10-PCS - PYG Option (GPIO & PYTHON)
RS232 Serial Interface for AT commands, data, fax, SMS	YES	YES
POWER 12V DC with digital General Purpose Input 3 for "power good" supervision	YES	YES
AUX RJ11 6-pin configuration	**AUDIO** (GSM Voice Service)**:** GND Remote ON/OFF Microphone in (balanced) Earphone out (unbalanced) Vmod (3,8 V DC) Remarks: • Backward compatible to EZ10-GSM and previous versions of EZ10-GPRS, -PCS and -PYTHON.	**GPIO: (no Audio)** GND GPIO 4, GPIO 6 / Alarm GPIO 7 / Buzzer GPIO 5 Vmod (3,8 V DC) Remarks: • Automatic POWER ON. • GPIOs can be configured as I2C or SPI bus by PYTHON script elements.

The EZ10 allows to be remotely controlled by AT commands (GSM 07.07 and 07.05) and the connection to the host controller (Data Terminal Equipment DTE) is established through one RS232 standard port. This port serves also for serial bi-directional Data and Fax transfer.

All the interfacing is done by 4 connectors placed on the front and rear panels.

The tri-band evolution **Telit EZ10-PCS** with PYG Option includes new features like sensing of hot removal of the SIM and the higher upload speed of the GPRS Class 10 Packet Data transfer.

EASY GPRS® Embedded

Telit EZ10-GPRS /-PCS with PYG Option embeds TCP/IP stack and DNS query protocol. Static and dynamic IP allocation, PPP and UDP as well as FTP Client are supported. Developers just need to add Telit Proprietary AT commands to establish a TCP/IP connection through the GPRS network.

EASY FREQUENCY SCAN® Function

Telit EZ10-PCS with PYG Option includes a dedicated set of AT commands to scan all the GSM channels, reporting all available parameters. With EASY FREQUENCY SCAN® the GSM network coverage at the location of the EZ10-PCS PYG Terminal can be examined even without inserted

SIM in order to select the provider with the best field strength, the most channels or base transceiver stations and to optimize the antenna position.

EASY SCRIPT® Function

Telit EZ10-PCS with PYG Option includes a dedicated set of AT commands to run scripts in **PYTHON**[1] language within the module. By using this EASY SCRIPT® feature, the script inside the module interacts with the interfaces, allowing other tasks to interrupt it. The script interpreter engine of **Telit EZ10-PCS wireless modem with PYG Option** is allowing self controlled operations.

With the **Telit EZ10-PCS with PYG Option** it is possible to configure the available GPIOs 4-7 as **I2C or SPI bus**.

JAMMING DETECTION and REPORT

The **Telit EZ10-GPRS /-PCS** with PYG Option offers the Jamming Detection and Report functionality, which allows the unit to detect attempts to discontinue GSM communication by interfering with the GSM radio signal. The "Jammed" status is reported as unsolicited message through the AT-Command interface on the RS232 connection.

The EZ10 is compliant with the relevant parts of ETSI GSM Phase 2+ specifications (Normal MS).

Furthermore, **Telit EZ10-GPRS /-PCS** are certified after the full conformity assessment against 1995/5 EC and R&TTE Directive and they don't require any further certification test effort and expense. The Terminals of the EZ10 family can be used in all EU countries and in other countries where a GSM 900/1800 MHz or PCS 1900 MHz network is available.

It is necessary to insert a **SIM** (Subscriber Identity Module, type 3 Volt Plug-in) on which the level of worldwide access to the services of the related GSM/GPRS network depends.

NOTE: The availability of the services of the GSM/GPRS networks, their quality and parameters depend mainly on the contractual conditions of the subscription to receive the SIM (Subscriber Identity Module) from the Mobile Network Operator or Service Provider.

The behavior of the networks may be different in certain details, thus it is always recommended that

- A complete and correct set-up of all parameters is included in the Application;
- The Application contains verification routines for the connections and transmissions,
- For critical communications, fall-back solutions to other services of the networks are implemented (e.g. SMS or Call, GPRS or CSD);
- The Application allows to update the Software of the contained GSM/GPRS engine;
- A validation of the Application in the relevant networks has been carried out.

Before importing the products in other than EU countries, make sure that eventually required national Certifications of Conformity and licenses are available. It is possible that conformity regulations contain conditions for the use of the product in the related countries. Please, refer to the CONFORMITY ASSESSMENT ISSUES chapter of this Product Description for more information, or contact the supplier.

[1] PYTHON is a registered trademark of the Python Software Foundation.

2 General Product Description

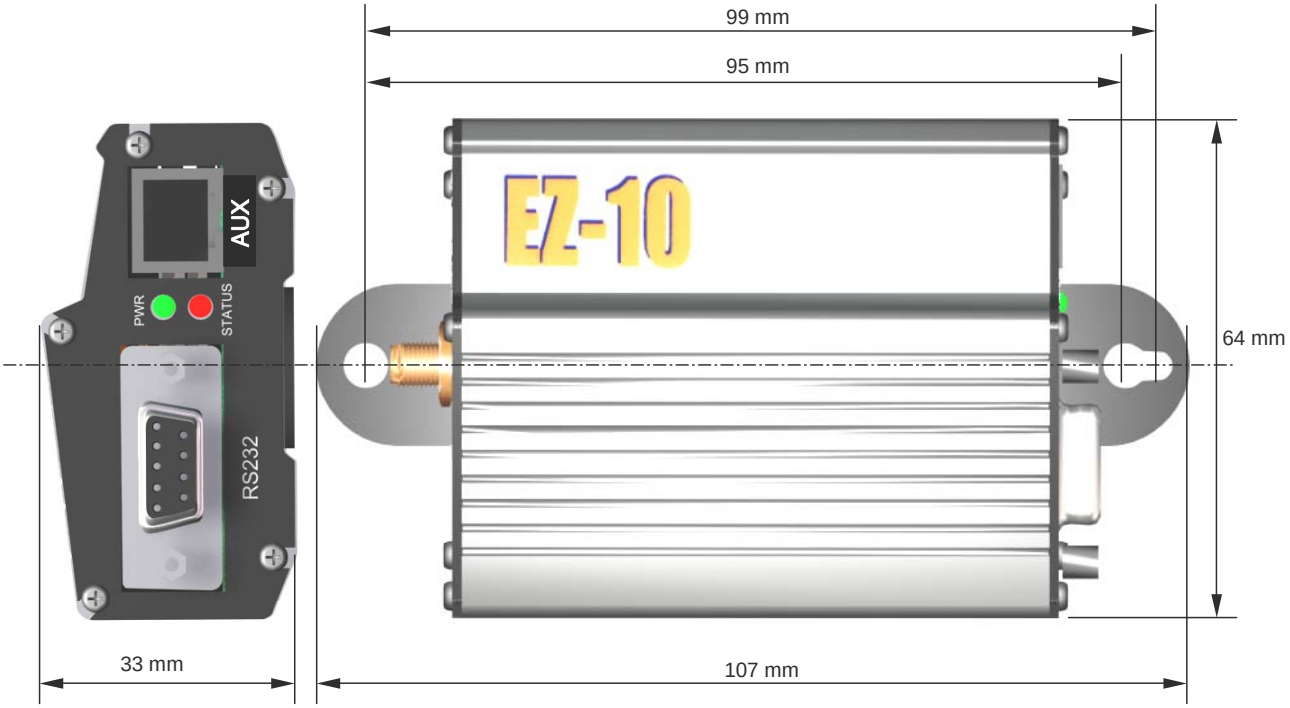
NOTE: The illustrations in this Product Description are only schematic and do not assure fidelity to construction or layout details, finishes, writings or colors.

2.1 Dimensions

The **Telit EZ10 wireless modem** dimensions are:

- **Housing Length:** **83 mm** (without connectors)
- **Overall Length:** **107 mm** (including fixtures)
- **Width:** **64 mm**
- **Thickness:** **33 mm**

See the following figure:



2.2 Weight

The net weight of the **Telit EZ10 wireless modem** is ca. **150 grams.**

2.3 Installation

The **Telit EZ10 wireless modem** can be fixed on a suitable surface by two screws through the holes (3,3 / 5,8 mm diam.) in the lids forming part of the front and rear panels. The figure shows the maximum and minimum inter-axis distance between the fixing holes.

In case of a permanent vertical installation in dusty environment, it is recommended to cover the SIM slot with a self-adhesive tape.

2.4 Environmental requirements

The **Telit EZ10 wireless modem** is compliant with the applicable ETSI reference documentation GSM 05.05 Release1999 ETSI EN300910 V8.4.1

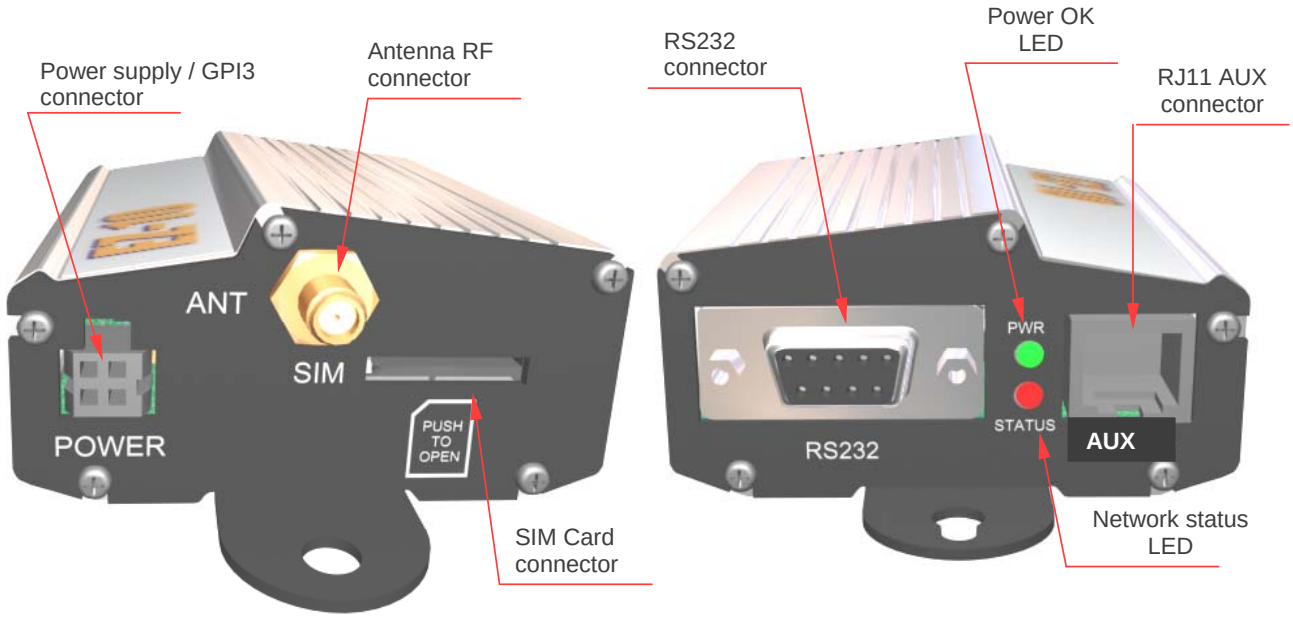
2.4.1 Temperature range

- 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

* Temperature exceeding the range of normal functional conditions can affect the sensitivity and performance of the modem.

2.5 EZ10 Interfaces

The interfaces of the EZ10 modem terminal are distributed on 4 connectors on the front and rear panels, see figure.

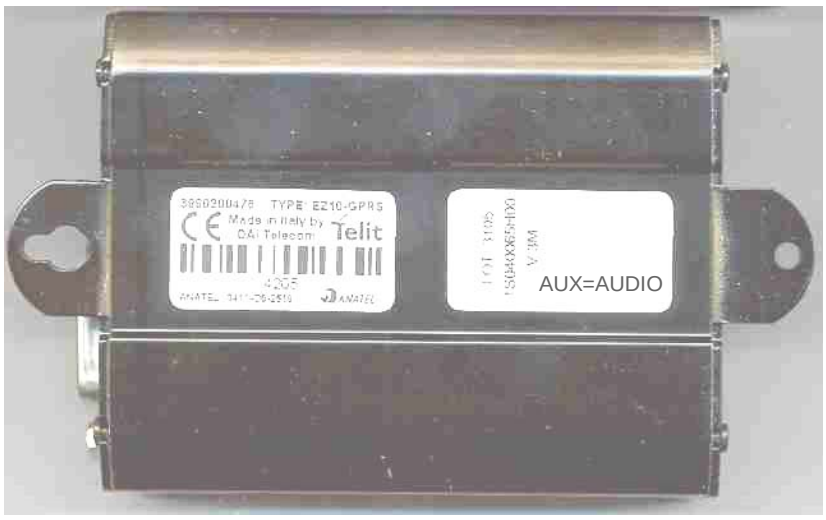


NOTE: Always verify the **Type Labels** on the under side of the **Telit EZ10 Terminals** in order to identify the model and especially the **version of the Interfaces at the RJ11 connector**. This connector can bear AUDIO lines or GPIO lines. It can be identified by the writing **AUDIO** in older production lots or by the writing **AUX=AUDIO or AUX=GPIO** in newer production lots.

2.5.1 Product Versions Identification - Documentation

2.5.1.1 Hardware Identification

The version of the product can be read from the labels attached on the under side of the Terminal.



Each Terminal has a **Serial Number** (e.g.4205) on the TYPE label, by which the **IMEI** of the **Telit GM862** GSM Engine inside can be retrieved from a production database at the manufacturer. In addition, the IMEI can be obtained in functional conditions with a Data Terminal Equipment (DTE) as response to a specific AT-command: AT+CGSN, AT#CGSN

Due to continuous improvements of the product, some parameter values or product properties in the following chapters of this Product Description depend on the **HW Version** of the EZ10 Terminal, which can be read from the right label (e.g. V.3M).

Model	**Function RJ11 AUX**	**Telit Part No.**	**HW Versions**	**Telit Order Code** (without accessories)
EZ10-GPRS	**AUDIO**	3990200476	2.2, 3, 3M, 4, 5	3990150451 EAN8025830004402
EZ10-PCS Samples, not for future orders	AUDIO	3990200477	2.2, 3, 3M, 4, 5	3990150452 EAN8025830004389
EZ10-PYTHON Samples, not for future orders	AUDIO	3990200484	2.2, 3, 3M, 4, 5	3990150459 EAN8025830004693
EZ10-PCS with PYG Option	**GPIO**	3990200498	5	3990150462 EAN8025830004884

NOTE: The Telit Part No. and Order Code numbers may depend on customer specific configurations and thus are only indicative. Distributors might use their specific order codes.

The following accessories for the EZ10 Terminals can be made available **separately**:

Article	Telit Part No.
GSM Tri-Band Antenna 2.5dBi gain, magnetic surface mount with ca. 2,5 m coaxial cable RG174 and SMA m connector;	1rr0100058ezr or 1rr0100056tlb
RS232 data cable, ca. 1,8m SubD9 m-f, straight	1ff1400065ezr
AC adapter: Input 100-240V AC, 50..60Hz, Euro Plug Output 9V DC 1,112 A, ca. 1,8m cable with 4-pin plug (Molex 43025-400), "Power good" bridged to GND at plug	1jj0400032ezr - not for future orders.
AC adapter: Input 100-240V AC 50..60Hz, Euro Plug Output12V DC 1,2 A, ca. 1,8m cable with 4-pin plug (Molex 43025-400), "Power good" bridged to GND at plug	1jj0400034ezr
AC adapter: Input 100-240V AC, 50..60Hz, UK Plug Output12V DC 1,2 A, ca. 1,8m cable with 4-pin plug (Molex 43025-400), "Power good" bridged to GND at plug	1jj0400035ezr
AC adapter: Input 100-240V AC, 50..60Hz, US Plug Output12V DC 1,2 A, ca. 1,8m cable with 4-pin plug (Molex 43025-400), "Power good" bridged to GND at plug	1jj0400036ezr

NOTE: Specifications and Part Numbers are subject to modifications.

2.5.1.2 Software identification

The version of the **Software (SW)** (firmware) implemented in production (on the Telit GM862 Cellular Engine inside) can be read from the SW label. In addition, the SW version can be obtained in functional conditions with a Data Terminal Equipment (DTE) as response to a specific AT-command: AT+CGMR. This is especially useful when the SW has been updated after delivery.

2.5.1.3 Documentation and User Guides

The availability of SW Features and of the related AT commands depends on the Model of the EZ10 Terminal. For products delivered a longer time prior to the issue date of this Product Description, a software update might be recommended to obtain all parts of the described features.

A table at the end of this Product Description gives an overview, which AT commands are implemented in each Model of the EZ 10 Terminal.

The **Software Features** are described in a dedicated chapter of this Product Description.

Some of the Functionalities described in the documentation of the GM862 Engines are not fully available in the related Models of the EZ10 Terminal Family, because they are depending on how certain interfaces of the Engine are made available on the outside of the EZ10 Terminal or used for internal functions, for example the Audio Interfaces, GPIO, Status LED.

Not supported functions in all EZ10 Terminal Models:

- EASY CAMERA®
- Battery charging function
- Python script debug serial port (only available on Evaluation Kit EVK)
- ADC, Vaux, (TRIZIUM and onwards)
- MIC_MT and EAR_MT audio path
- GPIO1, 2, 8 and upwards
- GPIO3 as output (only as input “power good”)

The description of the **AT Commands** is available on www.telit.com >Products >Modules >EZ10 Terminal Family as a separate document: “**GM862/TRIZIUM Family AT Commands Description** 80264ST10013a_r1” in parts referring to the GM862 GSM/GPRS/PCS/PYTHON Engine inside the related Model of the EZ10 Terminal.

Practical examples for the use of AT commands and some applications can be found in the **GM862 Software User Guide** available on the same location on www.telit.com

- General set-up
- Use of the basic GSM/GPRS network services (Voice, Data, SMS)
- Reading and setting of the GPIO ports
- Setting of signals in the audio paths

Technical information and reference designs for external circuitry partially relevant for the EZ10 Terminal Family can be found in the **GM862 Hardware User Guide** available on www.telit.com >Products >Modules >Terminal Family >related model

- for connecting to the GPIO ports,
- for connecting of microphones and headphones to the audio path (MIC_HF, EAR_HF).

2.5.2 Supply voltage requirements

The DC power supply must be connected to the POWER input:

- Input voltage range: HW version 2.2, 3, 3m, 4 12 - 24V DC
  HW version 5 9 - 24V DC

Note: these are the absolute maximum ratings – for use e.g. in trucks with 24V nominal voltage, an additional DC/DC converter has to be used.

- Power Supply current rating: min. 1,2A @12V
- Power Supply ripple: max. 120mV
- Input current in idle mode: 8mA @ 12V
- Input average current in communication mode: 110mA @ 12V

NOTE: In case that power supply equipment is to be ordered, its conformity needs to be verified with the mains supply voltage, frequency, connector type and other national requirements (e.g. certifications) in the countries of its use.

2.5.3 POWER / GPIO3 Connector

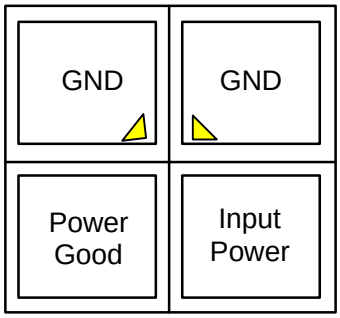
Connector type on the terminal:

- Molex 4-pin connector part no.:43045-0400 male

Pin description:

- "GND" = Ground reference (1x POWER, 1x Signal)
- "Input Power" = 9 - 24VDC (see versions & ratings above)
- "Power Good" = signal input/ connected to **GPIO3** through series resistor (version 2.2-4: 100 Ohm, version 5: 10k), considered for status sensing of attached special power supply units.
  Voltage range allowed on this pin: 0-3,6VDC vs. GND.

FRONT VIEW



2.5.3.1 "Power Good" Function

In certain applications, the EZ10 Terminal is supplied through a power supply circuitry including a backup battery. Therefore, repeated polling of the input signal on the "Power Good" (= high (3V), when mains power fails) can be used by the Software on the host controller (or by a PYTHON script) to initiate a certain behavior of the module or a notice through the GSM network services to the far end application.

NOTE: The Power Supply equipment available as separate accessories from Telit might have a bridge between "Power Good" and GND on the 4-pin connector (Molex 43025-400).

2.5.4 RJ11 (AUX) connector

Connector type on the terminal:

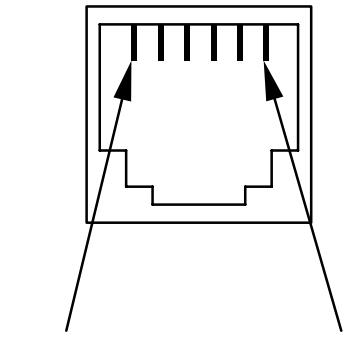
- Molex RJ11/6pin, part no.:95001-2661

The voltage range allowed on each pin is 0-3,6V DC.

The signals on this connector depend on the version of the EZ10 Terminal, which can be seen from the suffix on the **Type Label**:

- The models **Telit EZ10-GPRS wireless modems** (AUDIO) have analog audio input/output at the RJ11 connector. The pin 2 is connected to the ON/OFF signal input of the GM862 Engine, and the EAR_HF+ is referenced to GND.
- The model **EZ10-PCS with PYG Option** has 4 digital general purpose inputs/outputs (GPIO) at the RJ11 connector. AUTO POWER ON is enabled.

FRONT VIEW



NOTE: Always verify the **Type Labels** on the under side of the **Telit EZ10 Terminals** in order to identify the model and especially the **version of the Interfaces at the RJ11 connector**. This connector can bear AUDIO lines or GPIO lines. It can be identified by the writing **AUDIO** in older production lots or by the writing **AUX** in newer production lots.

The Telit EZ10-PCS with PYG Option has General Purpose Inputs/Outputs instead of the Audio/Remote Power ON/OFF lines on the RJ11 connector.

Before connecting any equipment and before taking the EZ10 into operation, make sure which configuration applies according to the type label of the EZ10 and that this configuration is compatible with the equipment and the Application (as DTE) which uses the EZ10. Production lots based on v5 or higher have an additional writing on the SW/HW version label: **AUX=AUDIO or AUX=GPIO.**

VMOD = direct connection to the pin of the power supply voltage input of the GM862 module (3.8V typical) and the output of the internal switching voltage regulator. The presence of this line at pin6 is to be considered as a **low power output (<30mA)** for microphone bias supplies or pull-up potential (requires external reduction of voltage to max. 3,6V DC!).
NOTE: Connections to VMOD shall be made only when familiar with designing circuits conform to EMC requirements.

2.5.4.1 Pin description, RJ11 with AUDIO configuration:

All audio input / output lines have a 100 pF bypass capacitor (v.5 and onwards) to ground and are connected to the related pins of the GM862 module over a 100 Ohms series resistor and 1uF series capacitor.

pin	Signal	I/O	Function	Internal Pull UP	Type
1	GND	-	Power Ground (negative)/ Signal Ground		POWER/GND
2	ON_OFF	I/O	Additional On/Off line for REMOTE ON/OFF	47KΩ	Pull Up to VBATT
3	MIC_HF-	I/O	audio in, negative, to EZ10 Terminal		AUDIO
4	MIC_HF+	I/O	audio in, positive, to EZ10 Terminal		AUDIO
5	EAR_HF+	I/O	audio out, positive, from EZ10 Terminal (referenced to GND)		AUDIO
6	VMOD	O	Low power supply output (typically +3.8V DC)		POWER OUT

2.5.4.2 Pin description, RJ11 with GPIO configuration (PYG Option):

All General Purpose input / output lines on the RJ11 GPIO interface have a 100 pF bypass capacitor to ground and are connected to the related pins of the GM862 module over a 100 Ohms series resistor.

pin	Signal	I/O	Function	Internal Pull UP	Type
1	GND	-	Power Ground (negative)/ Signal Ground		POWER/GND
2	GPIO4	I/O	Configurable general purpose I/O pin /		CMOS 2.8V
3	GPIO6 / ALARM	I/O	Configurable general purpose I/O pin / ALARM		CMOS 2.8V
4	GPIO7 / BUZZER	I/O	Configurable general purpose I/O pin / BUZZER		CMOS 2.8V
5	GPIO5	I/O	Configurable general purpose I/O pin		CMOS 2.8V
6	VMOD	O	Low power supply output (typically +3.8V DC)		POWER OUT

2.5.5 Audio Interface Specification

The audio input “MIC” and output “EAR” of the Telit EZ10 module are configured for headset/hands-free use, referenced in the related GM862 module descriptions as “HF” audio path.

The Telit EZ10 module has a built in echo canceller and a noise suppressor tuned for a hands-free use. See the GM862 HW User Guide for examples of microphone bias circuits and amplifiers as well as of audio output amplifiers.

All audio input / output lines have a 100 pF bypass capacitor (v.5 and onwards) to ground and are connected to the related pins of the GM862 module over a 100 Ohms series resistor and 1uF series capacitor.

2.5.5.1 Microphone input characteristics

	External audio MIC input
Line coupling	AC (1uF)
Line type	Balanced
Differential input resistance	25kΩ
Line nominal sensitivity	$3mV_{rms}$
Max input voltage	$22mV_{rms}$
Microphone nominal sensitivity - Analog Gain suggested	$-45dB_{Vrms/Pa}$ / +10dB
Echo canceller type	Car kit hands free

No bias voltage is provided on the MIC input. In case of need, external circuits have to be added, which could be supplied from the VMOD output.

2.5.5.2 Speaker output characteristics

	External audio EAR output
Line coupling	AC (1uF)
Line type	Bridged, unbalanced use
Speaker impedance	≥ 16Ω ± 5% @ 1kHz
Minimum load impedance	15Ω
Signal bandwidth	150-8000 Hz @ -3dB
Maximum output	$850mV_{rms}$
Maximum power output	7.5mW
Volume level steps (SW)	-2dB
Number of volume steps (SW)	10

2.5.6 GPIO Interface Specification

Where not specifically stated, all the interface circuits work at **2.62V CMOS** logic levels.

All General Purpose input / output lines on the RJ11 GPIO interface have a 100 pF bypass capacitor to ground and are connected to the related pins of the GM862 module over a 100 Ohms series resistor.

The following table shows the logic level specifications in the Telit EZ10 Modem terminal interface circuits:

LEVEL	MIN	MAX
Input high level	2.1 V	3.6V
Input low level	0 V	0.5 V
Output high level	2.2 V	3.0 V
Output low level	0 V	0.35 V

See the GM862 HW and SW User Guides for

- Examples how to use the GPIO, e.g. setting over the air, sensing (polling) over the air or locally by the DTE,
- Configuration as ALARM output or BUZZER,
- Example input circuit (e.g. to detect a 12-V signal or a passive contact),
- Example output circuit (e.g. to drive a Relay in 12-V-environment or to connect a LED).

2.5.7 RS232 standard interface connector

Connector type on the terminal:

- RS-232 through D9-pin female

Baud rate from 300 to 115.200 bit/s

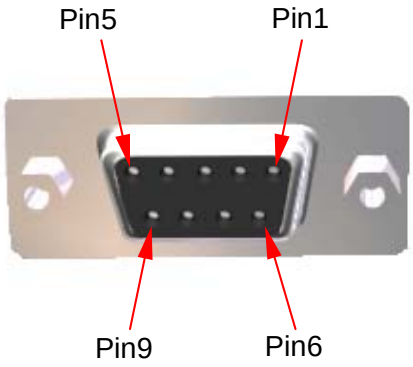
Autobauding (300 to 38.400 bit/s)

Short circuit (to Ground) protection on all outputs.

The EZ10 Terminal provides low-pass RF suppression circuits and level conversion to the GM862 Engine inside.

Input voltage range: -12V to +12V

Pin out (refers to DTE side):



Pin 1 = DCD Output
Pin 2 = RX Output
Pin 3 = TX Input
Pin 4 = DTR Input
Pin 5 = Ground
Pin 6 = DSR Output

Pin 7 = RTS Input

Pin 8 = CTS Output

Pin 9 = RI Output

To connect to a PC as DTE, a pin-to-pin, 9-wire cable with D9 type connectors on both sides is needed (1 male & 1 female). Shielding of this cable is recommended and its length shall not exceed 3m.

2.5.8 SIM Interface Characteristics

The **Telit wireless modems EZ10-GPRS, /-PCS (with PYG option)** support plug-in type Subscriber Identity Modules (SIM) according to GSM11.14 Phase 2 – with 3 Volts supply ONLY.

For the technical characteristics of the SIM, refer to the documentation supplied together with the SIM by the Mobile Network Operator or Service Provider.

The **Telit EZ10-PCS wireless modems (with PYG option)** have an enhanced SIM card reader with a sensor that allows to detect a “hot” removal of the SIM, therefore the SIM can be extracted and re-inserted while the module is still on.

NOTE: On the **Telit EZ10-PCS wireless modems (with PYG option)** the sensing of a hot removal of the SIM is not supported during power saving mode (AT+CFUN: 5).

2.5.9 Insertion and Removal of the SIM

The terminal has a **built-in toggle spring (Push-Push) SIM holder**, accessible through a slot in the panel at the antenna side of the housing.

Do not insert or remove the SIM when the product is in power saving mode.

To insert and remove the SIM, a plastic strip of the same width of the SIM and appropriate length of ca. 50-100 mm shall be prepared as a tool. The figure printed on the panel shows the position of the cut edge and the direction, how the SIM shall be inserted.

Insert the SIM and push it with the tool slightly inside until the spring snaps in. Removing the tool, the SIM shall remain inside the EZ10 Terminal. For removing the SIM, push the tool slightly inside until the spring is released so that it pushes the SIM outside when the tool is retracted.

2.5.10 Antenna Output

NOTE: BEFORE connecting the EZ10 Terminal to a Power Supply source, a suitable Antenna shall be connected and properly installed.

As accessory, a magnetic surface mount antenna with 2.5dBi gain, ca. 2,5 m of coaxial cable and SMA connector is available.

The antenna has to be installed with care in order to avoid any interference with other electronic devices and has to guarantee a minimum distance from persons (20 cm). In case this requirement cannot be satisfied, the system integrator has to assess the final product against the SAR regulation.

For a good efficiency of the antenna and minimum interference with other electronic systems, a space of min. 40 cm around the radiating part should be free, at least of electrically conducting materials (except the ground plane on which it is attached).

Less distance and less obstacles there are between the antenna connected to the EZ10 Terminal and the antenna of the GSM/GPRS network base station, the less power is radiated by the Terminal under normal conditions and the higher is the safety margin in case of disturbances.

A check of eventual interferences can be made when the EZ10 Terminal transmits at maximum power level to register to a GSM 900 network (see frequency channel numbers), immediately after being switched on.

Refer to the chapter “Safety Information” of this Product Description and the documentation of the antenna in use before activating the EZ10 Terminal.

2.5.10.1 Antenna Connector

The **Telit EZ10 wireless modem** includes a SMA bulkhead female, class 4 (2W) co-axial connector for the external antenna.

2.5.10.2 Antenna Specification

The antenna to be connected shall fulfill the following requirements:

Frequency range	EZ10-GPRS: Standard Dual Band frequency range, EZ10-PCS (PYG Option): Standard Tri Band frequency range
Bandwidth	80 MHz in GSM & 170 MHz in DCS & 140 MHz PCS band
Gain	1.5dBi ≤ Gain < 3dBi (referenced to isotropic radiator)
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 wireless modems EZ10-PCS (with PYG Option).**

2.6 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 frequency coverage is required only for the **Telit wireless modems EZ10-PCS (with PYG Option).**

2.7 Transmitter output power

GSM–900

The EZ10 wireless modem in GSM–900 operating mode are **class 4** in accordance with the specification which determine the nominal 2W peak RF power (+33dBm) on 50 Ohm.

DCS–1800

The EZ10 wireless modem 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 EZ10 wireless modem 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.

2.8 Reference sensitivity

GSM–900

The sensitivity of the EZ10 wireless modem 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 EZ10 wireless modem 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 EZ10 wireless modem according to the specifications for the class 1 portable terminals PCS 1900 is better than **–102dBm** in normal operating conditions.

2.9 User Interface

The user interface of the EZ10 Terminal is accessible from a Data Terminal Equipment DTE connected to the RS232 interface and it is managed by AT commands according to the GSM 07.07 and 07.05 specification and the supported commands of the different versions are listed in the **chapter 4, AT Command Interface**. For a detailed description of the AT Command Interface refer to the separate GM862 / TRIZIUM Software Description / User Guide documents.

2.9.1 Switching the EZ10 Terminals ON and OFF

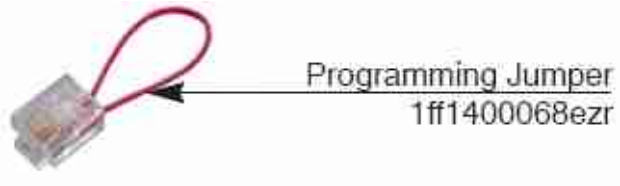
Depending on the versions of the **Telit EZ10 wireless modem**, there are generally the following ways to switch the EZ10 Terminals ON or OFF:

NOTE: Switching OFF the EZ10 by disconnecting the external power supply line requires to wait enough time (depending on the HW version!) that the load capacitor of the DC/DC converter inside the EZ10 discharges so that the GSM862 module inside goes to OFF condition.

2.9.1.1 ON/OFF on Telit EZ10-GPRS (models with AUDIO interface)

- The EZ10 Terminal switches on automatically each time the Power Supply is connected the first time or re-connected.
- The EZ10 Terminal can be turned OFF and ON by a signal on the REMOTE ON/OFF line at the AUDIO connector (RJ11). Refer to GM862 Hardware User Guide for details on circuitry and timing.
- The EZ10 Terminal can be turned OFF by the command AT#SHDN. For turning it ON again, the ON signal is required on the ON/OFF line (manually or by external device) or the Power Supply has to be disconnected and re-connected.

Use of the Programming Jumper: It connects pin2 to GND to create ON / OFF signal. It needs to be plugged **and unplugged** for that operation.



NOTE: The jumper might be plugged at delivery – remove it to switch the device ON

2.9.1.2 ON/OFF on Telit EZ10-PCS with PYG Option

- The EZ10 Terminal switches on automatically each time the Power Supply is connected the first time or re-connected.
- **The EZ10 Terminal can be turned OFF by the command AT#SHDN, but immediately thereafter it switches ON again automatically (AUTO POWER ON always enabled).**

NOTE: The jumper is not included with the terminal, because there is a GPIO at the place of the Remote ON/OFF line.

2.9.2 Functions of the RS232 Serial Interface

The RS232 standard interface serves to connect a PC, Data Terminal Equipment (DTE) or an application, which acts as host controller of the EZ10 Terminal with all its functions. Through the RS232 interface it can be used as GSM/GPRS modem for sending and receiving of SMS, Data and Fax calls. Establishment and control of voice calls is done by the DTE through this interface, whereas e.g. a headset to listen and talk has to be connected to the AUDIO connector.

The **Telit EZ10-PCS wireless modem with PYG Option**, on the basis of the **EASY SCRIPT®** feature and with a **PYTHON** script developed by the user, can allow self-controlled operations which put the RS232 interface in a different serial data transmission mode, e.g. to communicate with a sensor or actuator.

2.9.2.1 The PC as Data Terminal Equipment (DTE)

Normal Personal Computers (PC), which have a RS232 standard serial interface (COM-port), often include a software application for using it as Data Terminal Equipment (DTE).

Example (without success guarantee related to the DTE's role): Start-Programs-Accessories-Communications-**Hyper Terminal**. In the prompt window for New connection it is then possible to give a name and assign a symbol to the DTE. Connect using: the COM-port to which the EZ10 Terminal is connected. Of the default configurations, only the Bits per second should be set to 115200, the rest remains at "8-None-1-Hardware".

A cursor blinks in a window and the connection to the EZ10 Terminal is already active. In this moment, it is not possible to see what is typed to the keyboard, but the answer to a correct command will appear: e.g. type AT+CGMM<enter> and the type of the connected Mobile Station (MS) will appear on the screen. Type: ATE1<enter> to activate the command echo, which displays what is typed. AT+CGMR returns the Software version.

For more information about which AT Commands are implemented on the Models of EZ10 Terminals refer to the chapter "AT Command Interface" of this Product Description and the details can be found in the "GM862 Software User Guide" and "GM862/TRIZIUM FAMILY AT Commands Description", available on www.telit.com - >Products >Modules >EZ10 Terminal >Download.

NOTE: The character sets used by the terminal programs or equipments might vary and can cause malfunctions with special characters e.g. ß, @. Verify the related specifications.

2.9.2.2 GSM/GPRS Standard Modem

It is always worth a trial to install the EZ10 Terminal as modem on a PC in the following way, be aware that the following procedure is just an **example** without success guarantee:

First add it in the PC's Control Panel – Telephone and Modem. Select from the list the "Standard 33600 bps Modem" with connection to the related COM port, then make a new "Dial-up connection to private network" using this modem with the dial-up parameters of the GSM/GPRS Network in use. Sometimes the parameters indicated for the use of the WAP-Browser of mobile phones (over GSM CSD call or GPRS) will work.

Example (without liability) for a GPRS connection set-up:

Under "Properties-General Configure" the initialization has to be entered in the text box selecting "Run Script". The parameters are examples for subscribers of Italian GSM network operators.

Initialization: AT+CGDCONT=1,"IP","web.omnitel.it"

Dial number : *99***1#

User Name : none
Password : none

- -

Initialization: AT+CGDCONT=1,"IP","internet.wind.biz"

Dial number : *99***1#

User Name : none
Password : none

- -

Initialization: AT+CGDCONT=1,"IP","uni.tim.it"

Dial number : *99***1#

User Name : phone number without international prefix
Password : as defined at registration to the service uni tim (e.g. via SMS)

2.9.3 Functions of the RJ11 Interface

According to the Model of the EZ10 Terminal, the interface at the RJ11 6-pin connector has different functions and physical connections. Refer to the chapter “EZ10 Interfaces” for details.

2.9.3.1 AUDIO configuration (EZ10-GPRS):

- Connection of balanced microphone and unbalanced earphone for GSM voice calls to hands-free audio path (MIC_HF, EAR_HF)
- Signal input for switching ON/OFF
- VMOD: connection to the +3,8V DC internal supply voltage. Attention to low power rating and EMC requirements.

2.9.3.2 GPIO configuration (EZ10-PCS with PYG Option):

- Connection to General Purpose Inputs/Outputs 4, 5, 6, 7, which can be used to communicate between the EZ10 Terminal and external devices, like switches or relays. With a user defined PYTHON script element, it is possible within certain limits to configure these GPIOs as I2C or SPI bus. See Chapter on PYTHON of this Product Description.
- The GPIO6 can be set by AT commands as output for the ALARM (wake-up) signal, generated by the Real Time Clock RTC.
- The GPIO7 can be set by AT commands as output for the BUZZER (incoming call or SMS) signal.
- VMOD: connection to the +3,8V DC internal supply voltage. Attention to low power rating and EMC requirements.

Although no audio devices can be connected in this configuration, voice calls can be established and received e.g. for transferring of the Calling Line Identification (**CLI**, phone number).

2.9.4 Speech Coding

The **Telit EZ10-GPRS wireless modem** voice codec supports the following rates:

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

2.9.5 SMS

The **Telit wireless modem EZ10-GPRS / EZ10-PCS with PYG Option** supports the following SMS types:

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

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

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

2.9.6 Real Time Clock and Alarm

The **Telit wireless modem EZ10-GPRS / EZ10-PCS with PYG Option** supports the Real Time Clock and Alarm functions through the AT command interface.

Furthermore, on the **Telit wireless modem EZ10-PCS with PYG Option**, an alarm output pin (GPIO6) can be configured to indicate the alarm with a hardware line output.

The behavior of the RTC at power failures depends on the HW version of the EZ10 Terminal.

2.9.7 Data/fax transmission

The **Telit EZ10 wireless modems** support:

- Packet Data transfer GPRS Class B, Multi-slot Class 8. (**Telit EZ10-GPRS** only)
- Packet Data transfer GPRS Class B, Multi-slot Class 10. (**Telit EZ10-PCS (PYG Option)** only)
- Data transmission according to the GSM 07.07, 07.05
- CSD up to 14.4 Kbps
- Fax service, Class 1 Group 3

2.9.8 Local security management

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

2.9.9 Call control

Call cost control function.

2.9.10 Phonebook

Function available to store the telephone numbers in SIM memory.

Capability depends on SIM version/memory

2.9.11 Characters management

Availability of lowercase, uppercase and IRA characters. (International Reference Alphabet)

In SMS PDU mode all character sets are supported.

2.9.12 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.

Do not insert or remove the SIM when the product is in power saving mode.

NOTE: The SIM (Subscriber Identity Module) from the Network Operator or Service Provider can be configured in different ways, e.g. related to

- The presence of SIM Applications (not supported by the current Software of EZ10 Terminals),
- The possibility to de-activate the necessity to enter the PIN (Personal Identification Number) as condition for registering in the network;
- The possibility that the Terminal can be locked by factory SW settings to a certain SIM of a certain Network Operator.
- The presence and format of certain phone book directories (PB, FDN, ADN) or SMS memories.
- The presence, content and format of other registers in the SIM, e.g. network operator's name, list of preferred networks (names change from time to time), etc.
- The possibility to roam between several mobile networks even in the home country.

2.9.13 Call status indication

By AT commands and Status LED.

2.9.14 LED Indicators

The EZ10 Terminals have 2 LED indicators, one for Power, one for Status.

2.9.14.1 Power LED

When on, the green Power LED indicates that the supply voltage is arriving at the GSM Engine inside the EZ10 Terminal.

2.9.14.2 Status LED: Indication of network service availability

The red Status LED is connected internally to the STAT_LED output of the module by an inverting circuit. In addition to the status information obtainable via AT commands, this LED shows information on the network service availability and Call status.

Status LED indications

LED status	Device Status
permanently on	device off
fast interrupt sequence (period 0,5s, Ton 1s)	Net search / Not registered / turning off
slow interrupt sequence (period 0,3s, Ton 3s)	Registered full service
permanently off	a call is active

2.9.15 Automatic answer (Voice, Data or FAX)

After n (depends of settings) rings, the EZ10 automatically answers with beep (see S0 param.).

2.9.16 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)

2.9.17 Acoustic signaling

The acoustic signaling of **Telit wireless modems EZ10-GPRS / EZ10-PCS with PYG Option** on the selected acoustic device is 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.
  - The selected /selectable acoustic device on EZ10-GPRS model is the EAR_HF Audio path to be activated with the related AT commands.
  - The selected /selectable acoustic device on EZ10-PCS with PYG Option can be set to the GPIO7 as the BUZZER output.

2.9.18 DTMF tones

The sending of DTMF tones is managed by specific AT commands.

These tones are activated 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	#

NOTE: The GSM system architecture defines that the audio signal of the DTMF tones is inserted by the network switches on commands sent by the Mobile Station (MS). Thus, the default duration parameters may vary from network to network. In case that the devices to be controlled by DTMF are sensitive related to the duration of the tones and timing of the sequences, dedicated investigations on the parameter settings have to be made.

2.10 EMC

Compliant to & ETS 300–342–1 and all applicable GSM Specifications.

Compliant to Directive 1999/05/EC.

3 Software Features

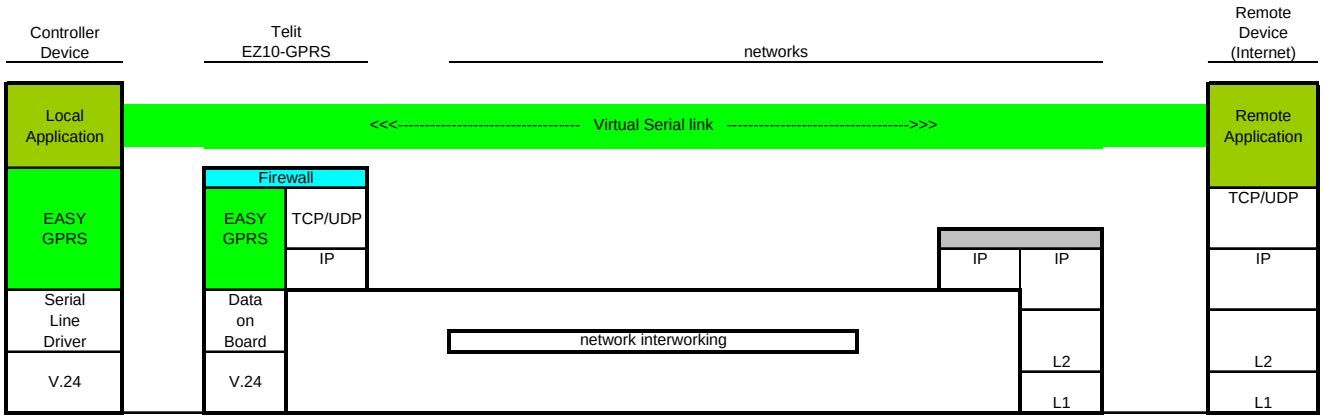
3.1 Enhanced EASY GPRS Extension

3.1.1 Overview

The EASY GPRS feature allows a user of the **Telit wireless modems EZ10-GPRS and EZ10-PCS with PYG Option** to contact a device in the 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 wireless modem EZ10-GPRS / EZ10-PCS with PYG Option** , 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 wireless modem EZ10-GPRS / EZ10-PCS with PYG Option** 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.

3.1.2 EASY 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.

3.1.2.1 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)

3.1.2.2 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)

3.1.2.3 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.

3.1.2.4 Open the connection with the internet host

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

- EZ10-GPRS activates the first context
- EZ10-GPRS proceeds to the authentication
- Eventually does the DNS query to resolve the IP address of the host name internet peer
- EZ10-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.

3.1.2.5 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 EZ10-GPRS closes the connection, deactivates the GPRS context returning to command mode and issuing the NO CARRIER code.

3.1.3 Enhanced 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.

3.1.3.1 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)

3.1.3.2 Configuring the embedded TCP/IP stack

The TCP/IP stack behavior 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)

3.1.3.3 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.

3.1.3.4 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

- EZ10-GPRS activates the context previously defined with AT+CGDCONT
- EZ10-GPRS proceeds to the authentication
- 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.

3.1.3.5 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
- EZ10-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.

3.1.3.6 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 EZ10-GPRS closes the connection, does not deactivate the GPRS context and returns to command mode issuing the NO CARRIER code.

3.1.4 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
c) defining the Internet Peer that can contact this device (firewall settings)
d) request the GPRS context to be activated
e) request the socket connection to be opened in listen
f) receive connection requests
g) exchange data
h) 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.

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

The EZ10-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"

3.1.4.2 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:

- EZ10-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.

3.1.4.3 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.

3.1.5 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

3.2 Jammed Detect & Report Extension

3.2.1 Overview

The Jammed Detect & Report feature allows a **Telit wireless modem EZ10-GPRS / EZ10-PCS with PYG Option** 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 the host controller 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.

3.3 EASY SCRIPT Extension - PYTHON Interpreter

NOTE: Embedding the original GM862-PYTHON GSM/GPRS Engine into the **Telit wireless modem EZ10-PCS with PYG Option**, the second serial interface is not accessible from outside. The only general purpose input output ports usable are the GPI3, GPIO4, 5, 6 and 7. This implicates restrictions on the creation of I2C and SPI busses. Of course, all scripts taking care of these restrictions can be run on **Telit wireless modem EZ10-PCS with PYG Option,** whereas for script development, the Python Kit or EVK are recommended because of their extra interfaces for debugging the script software.

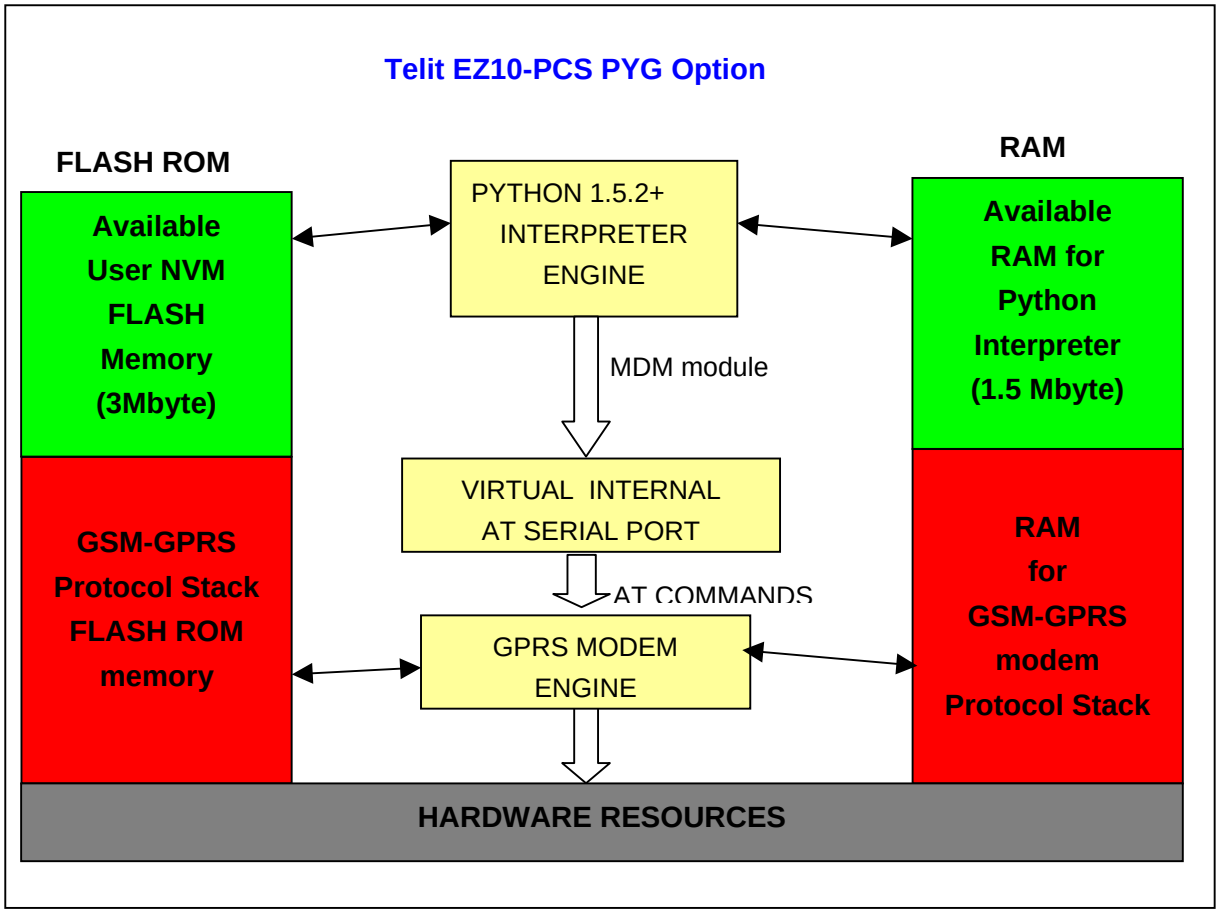
3.3.1 Overview

This feature is available only on the **Telit wireless modem EZ10-PCS with PYG Option.**

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 **Telit wireless modem EZ10-PCS with PYG Option** through the AT command interface.

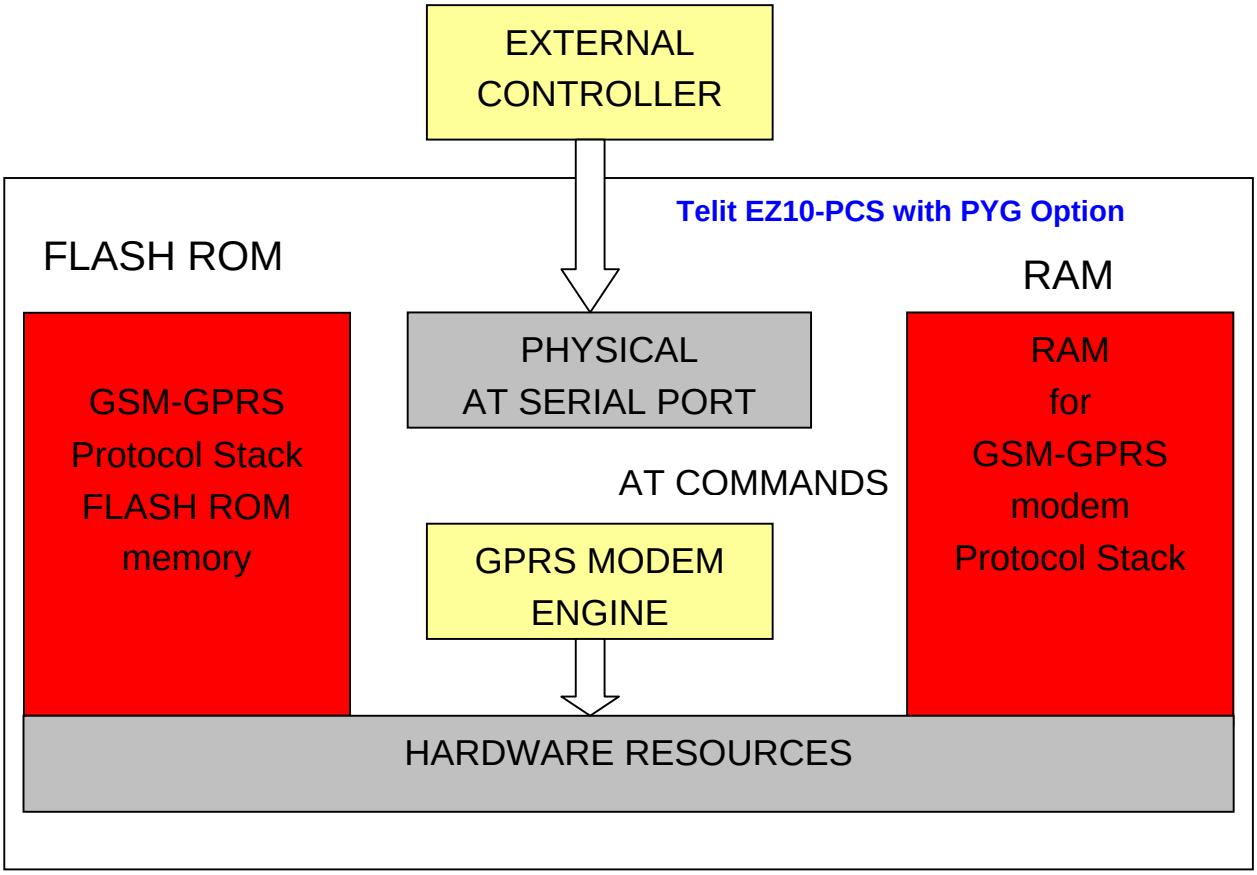
A schematic of such a configuration can be:



In order to eliminate this external controller, and further simplify the programming of the sequence of operations, inside the **Telit EZ10-PCS with PYG Option** 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:



3.3.2 Python 1.5.2+ Copyright Notice

The Python code implemented into the **Telit EZ10-PCS with PYG Option** 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.

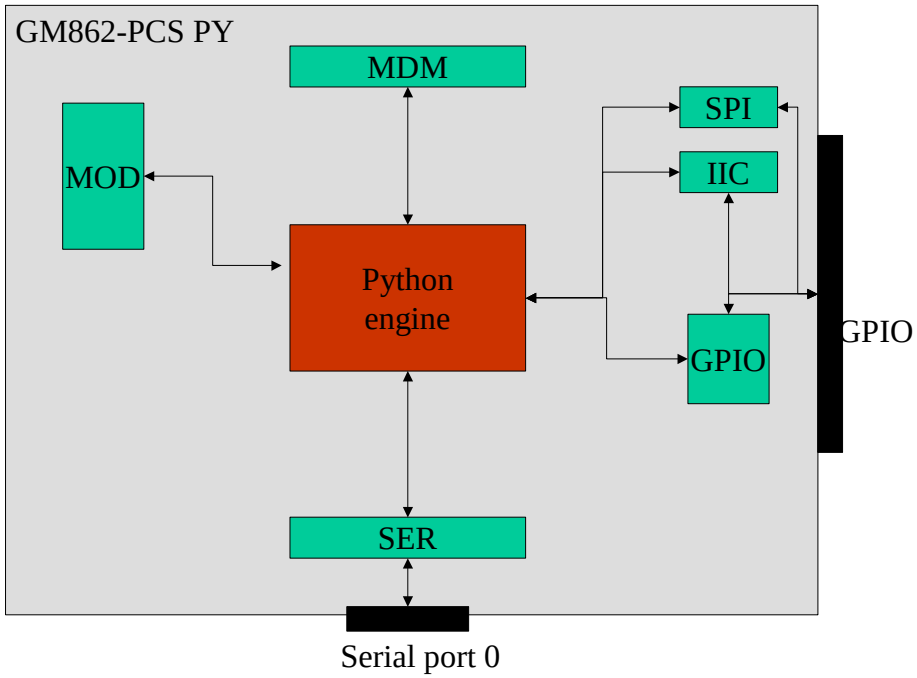
3.3.3 Python implementation description

Python scripts are text files, it is possible to run one Python script in the **Telit EZ10-PCS with PYG Option**.

The Python script is stored in NVM inside the **Telit EZ10-PCS with PYG Option**, there's a file system inside the **Telit EZ10-PCS with PYG Option** 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 **Telit EZ10-PCS with PYG Option** 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 EZ10-PCS with PYG Option** 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 EZ10-PCS with PYG Option**. 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 EZ10-PCS with PYG Option** 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 it's 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. 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.

3.3.4 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.

3.3.5 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.

3.3.5.1 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.

3.3.5.1.1 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.

3.3.5.1.2 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.

3.3.5.1.3 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.

3.3.5.1.4 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.

3.3.5.1.5 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.

3.3.5.1.6 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.

3.3.5.1.7 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.

3.3.5.1.8 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.

3.3.5.1.9 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.

3.3.5.1.10 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.

3.3.5.1.11 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.

3.3.5.1.12 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.

3.3.5.2 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.

3.3.5.2.1 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.

3.3.5.2.2 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.

3.3.5.2.3 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.

3.3.5.2.4 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.

3.3.5.2.5 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.

3.3.5.2.6 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.

3.3.5.2.7 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.

3.3.5.3 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.

3.3.5.3.1 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.

3.3.5.3.2 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.

3.3.5.3.3 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.

3.3.5.3.4 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.

3.3.5.4 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.

3.3.5.4.1 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.

3.3.5.4.2 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.

3.3.5.4.3 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.

3.3.5.5 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.

3.3.5.5.1 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.

3.3.5.5.2 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()

3.3.5.5.3 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.

3.3.5.5.4 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.

3.3.5.5.5 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.

3.3.5.5.6 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.

3.3.5.5.7 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.

3.3.5.5.8 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.

3.3.5.6 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.

3.3.5.6.1 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. 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)

3.3.5.6.2 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)

3.3.5.6.3 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.

3.3.5.6.4 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.

3.3.5.6.5 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.

3.3.6 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..

3.3.6.1 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 3.3.4, 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
```

3.3.6.2 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.

3.3.6.3 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.

3.3.6.4 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 EZ10-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 EZ10-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.

3.3.6.5 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 with an EVK that permits to work directly on the GM862 module and to have access to the second serial port.

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.

NOTE: The **Telit EZ10-PCS with PYG Option** does not have the second serial interface at the outside of the Terminal for receiving these Python script debug messages.

4 AT Command Interface

The **Telit EZ10 wireless modem** can be driven via the serial interface using the standard AT commands[2]. The **Telit EZ10 wireless modem** is 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 EZ10 wireless modem** supports also Telit proprietary AT commands for special purposes.

The following table lists all supported AT commands and related brief description.

The GM862 / TRIZIUM Family AT Command Description document, code 80264ST10013a, shows a dedicated detailed description of all supported AT commands and how to use the AT commands with the **Telit EZ10 wireless modem** through some example scripts.

[2] 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.

Hayes Compliant AT Commands		EZ10 Terminals Command availability	
Generic Modem Control		GPRS	PCS PYG
&F	Reset base section factory profile configuration	•	•
&F1	Reset full factory profile configuration	•	•
Z	Soft reset	•	•
+FCLASS	Select active service class	•	•
&Y	Designate a default reset basic profile	•	•
&P	Designate a default reset full profile	•	•
&W	Store current configuration	•	•
&Z	Store telephone number in the internal phonebook	•	•
&N	Display internal phonebook stored numbers	•	•
+GMI	Request manufacturer identification	•	•
+GMM	Request model identification	•	•
+GMR	Request revision identification	•	•
+GCAP	Request capabilities list	•	•
+GSN	Request serial number	•	•
&V	Display current configuration & profile	•	•
&V0	Display current configuration & profile	•	•
&V1	Display S registers values	•	•
&V2	Display last connection statistics	•	•
&V3	Display S registers values	•	•
\V	Single line connect message	•	•
%L	Report line signal level	•	•
%Q	Report line quality	•	•
+GCI	Select the country of installation	•	•
L	Monitor speaker loudness	•	•
M	Monitor speaker mode	•	•
DTE - modem interface control		GPRS	PCS PYG
E	Command echo	•	•
Q	Quiet resut codes	•	•
V	Result code form	•	•
X	Extended result codes	•	•
I	Request identifier and software checksum	•	•
&C	Data carrier detect (DCD) control	•	•
&D	Data terminal ready (DTR) control	•	•
&K	Flow control	•	•
&Q	Sync/async mode	•	•
&S	Data set ready (DSR) control	•	•
\R	Ring (RI) control	•	•
+IPR	Fixed DTE interface rate	•	•
+IFC	DTE - DTA flow control	•	•
+ILRR	DTE - modem rate reporting	•	•
+ICF	DTE - modem character format	•	•
Call Control		GPRS	PCS PYG
D	Dial	•	•
T	Set tone dial	•	•
P	Set pulse dial	•	•
A	Answer	•	•
A/	Last command automatic repetition	•	•
H	Disconnect	•	•
O	Return to On Line Mode	•	•
&G	Guard tone	•	•
Modulation control		GPRS	PCS PYG
+MS	Modulation control	•	•
%E	Enable/disable line quality monitor and auto retrain or fallback / fallforward	•	•
\N	Operating mode	•	•

Compression control		GPRS	PCS PYG
+DS	Set data compression	•	•
+DR	Data compression reporting	•	•
Break control		GPRS	PCS PYG
\B	Transmit break to remote	•	•
\K	Break handling	•	•
S parameters		GPRS	PCS PYG
S0	Number of rings to auto answer	•	•
S1	Ring counter	•	•
S2	Escape character	•	•
S3	Carriage return character	•	•
S4	Line feed character	•	•
S5	Backspace character	•	•
S7	Wait time for carrier, silence or dial tone	•	•
S10	Lost carrier to hang up delay	•	•
S12	Escape prompt delay	•	•
S25	Delay to DTR off	•	•
S30	Disconnect inactivity timer	•	•
S38	Delay before forced hang up	•	•
ETSI GSM 07.07 AT Commands		GPRS	PCS PYG
+CGMI	Request manufacturer identification	•	•
+CGMM	Request model identification	•	•
+CGMR	Request revision identification	•	•
+CGSN	Request product serial number identification	•	•
+CSCS	Select TE character set	•	•
+CIMI	Request international mobile subscriber identity (IMSI)	•	•
Call control		GPRS	PCS PYG
+CBST	Select bearer service type	•	•
+CRLP	Radio link protocol	•	•
+CR	Service reporting control	•	•
+CEER	Extended error report	•	•
+CRC	Cellular result codes	•	•
+CSNS	Single numbering scheme	•	•
Network service handling		GPRS	PCS PYG
+CNUM	Subscriber number	•	•
+COPN	Read operator names	•	•
+CREG	Network registration report	•	•
+COPS	Operator selection	•	•
+CLCK	Facility lock/ unlock	•	•
+CPWD	Change facility password	•	•
+CLIP	Calling line identification presentation	•	•
+CLIR	Calling line identification restriction	•	•
+CCFC	Call forwarding number and conditions	•	•
+CCWA	Call waiting	•	•
+CHLD	Call holding services	•	•
+CUSD	Unstructured supplementary service data	•	•
+CAOC	Advice of charge	•	•
+CLCC	List current calls	•	•
+CSSN	SS Notification	•	•
+CCUG	Closed User Group supplementary service control	•	•
Mobile Equipment control		GPRS	PCS PYG
+CPAS	Phone activity status	•	•
+CFUN	Set phone functionality (Power Saving Management)	•	•
+CPIN	Enter PIN	•	•
+CSQ	Signal quality	•	•
+CPBS	Select phonebook memory storage	•	•
+CPBR	Read phonebook entries	•	•
+CPBF	Find phonebook entries	•	•
+CPBW	Write phonebook entry	•	•

+CCLK	Clock Management	•	•
+CALA	Alarm Management	•	•
+CALM	Alert sound mode	•	•
+CRSL	Ringer sound level	•	•
+CLVL	Loudspeaker volume level	•	•
+CMUT	Microphone mute control	•	•
+CACM	Accumulated call meter	•	•
+CAMM	Accumulated call meter maximum	•	•
+CPUC	Price per unit and currency table	•	•
Mobile equipment errors		GPRS	PCS PYG
+CMEE	Report mobile equipment error	•	•
Voice Control (TIA IS-101)		GPRS	PCS PYG
+VTS:	DTMF tones transmission	•	•
Commands For GPRS		GPRS	PCS PYG
+CGACT	PDP context activate or deactivate	•	•
+CGATT	GPRS attach or detach	•	•
+CGDATA	Enter data state	•	•
+CGDCONT	Define PDP context	•	•
+CGPADDR	Show PDP address	•	•
+CGREG	GPRS network registration status	•	•
+CGQMIN	Quality of service profile (minimum acceptable)	•	•
+CGQREQ	Quality of service profile (requested)	•	•
Commands For Battery Charger		GPRS	PCS PYG
+CBC	Battery Charge	•	•
ETSI GSM 07.05 AT Commands for SMS and CB services		GPRS	PCS PYG
+CSMS	Select message service	•	•
+CPMS	Preferred message storage	•	•
+CMGF	Message format	•	•
+CSMP	Set parameters in text mode	•	•
+CSDH	Show parameters in text mode	•	•
+CSAS	Save setting text mode	•	•
+CRES	Restore text mode settings	•	•
+CSCB	Select Cell Broadcast Message types	•	•
Message configuration		GPRS	PCS PYG
+CSCA	Service center address	•	•
Message receiving and reading		GPRS	PCS PYG
+CNMI	New message indications to Terminal Equipment	•	•
+CMGL	List messages	•	•
+CMGR	Read message	•	•
Message sending and writing		GPRS	PCS PYG
+CMGS	Send message	•	•
+CMSS	Send message from storage	•	•
+CMGW	Write message to memory	•	•
+CMGD	Delete message	•	•
Custom AT Commands		GPRS	PCS PYG
#CGMI	Request manufacturer identification	•	•
#CGMM	Request model identification	•	•
#CGMR	Request revision identification	•	•
#CGSN	Request product serial number identification	•	•
#CIMI	Request international mobile subscriber identity (IMSI)	•	•
#CAP	Change Audio Path	•	•
#SRS	Select ringer sound	•	•
#SRP	Select Ringer Path	•	•
#STM	Signalling Tones Mode	•	•
#PCT	Display PIN Counter	•	•
#SHDN	Software Shut Down	•	•
#WAKE	Wake from Alarm mode	•	•
#QTEMP	Query Temperature overflow	•	•
#SGPO	Set General Purpose Output	•	•

#GGPI	Read General Purpose Input	•	•
#GPIO	General Purpose Input/Output pin control	•	•
#MONI	Monitor Cells	•	•
#QSS	Query SIM Status	•	•
#ACAL	Set Automatic Call	•	•
#SMOV	SMS Overflow	•	•
#SHFEC	Set Handsfree echo canceller	•	•
#HFMICG	Handsfree Microphone Gain	•	•
#HSMICG	Handset Microphone Gain	•	•
#SHFSD	Set Handsfree side tone	•	•
#/	Repeat last command	•	•
#BND	Select Band		•
FAX Class 1 Commands		GPRS	PCS PYG
+FCLASS	Select active service class	•	•
+FMI	Report manufacturer ID	•	•
+FMM?	Report model ID	•	•
+FMR	Report revision ID	•	•
Transmission/Reception control		GPRS	PCS PYG
+FTS	Stop Transmission and pause	•	•
+FRS	Wait for receive silence	•	•
+FTM	Transmit data modulation	•	•
+FRM	Receive data modulation	•	•
+FTH	Transmit data with HDLC framing	•	•
+FRH	Receive data with HDLC framing	•	•
Serial port control		GPRS	PCS PYG
+FLO	Select flow control specified by type	•	•
+FPR	Select serial port rate	•	•
+FDD	Double escape character replacement control	•	•
Enhanced Easy GPRS custom AT command Definition		GPRS	PCS PYG
#USERID	Authentication User ID control	•	•
#PASSW	Authentication Password control	•	•
#PKTSZ	Packet Size control	•	•
#DSTO	Data Sending TimeOut control	•	•
#SKTTO	Socket inactivity timeout control	•	•
#SKTSET	Socket definition control	•	•
#SKTOP	Socket Open command	•	•
#QDNS	Query DNS	•	•
#SKTCT	Socket TCP Connection Timeout	•	•
#SKTSAV	Socket Parameters Save Command	•	•
#SKTRST	Socket Parameters Reset Command	•	•
#GPRS	GPRS context activation control	•	•
#SKTD	Socket Dial	•	•
#SKTL	Socket Listen	•	•
#FRWL	Firewall setup	•	•
Easy Camera Extension - Camera Management		GPRS	PCS PYG
#CAMON	Camera ON		
#CAMOFF	Camera OFF		
#TPHOTO	Camera Take Photo		
#RPHOTO	Camera Read Photo		
#OBJL	Object List		
#OBJR	Object Read		
#CAMQUA	Camera Select Quality of Photo		
#CMODE	Camera Select Operating MODE		
Email management		GPRS	PCS PYG
#ESMTP	Email SMTP server		•
#EADDR	Email sender address		•
#EUSER	Email authentication USER NAME		•
#EPASSW	Email authentication PASSWORD		•
#SEMAIL	Send Email		•
#ESAV	Email Parameters Save Command		•

#ERST	Email Parameters Reset Command		•
Easy Scan Extension		GPRS	PCS PYG
#CSURV	Network Survey of the complete 900/1800/1900 Network		•
#CSURVC	Network Survey in computer friendly format		•
#CSURVU	Network Survey of user defined 900/1800/1900 channels		•
#CSURVUC	Network Survey in computer friendly format		•
#CSURVF	Network Survey Format		•
Jammed Detect & Report custom AT command		GPRS	PCS PYG
#JDR	Jammed Detect & Report	•	•
PYTHON Script Management commands		GPRS	PCS PYG
#WSCRIPT	Write script command		•
#ESCRIPT	Select Active script command		•
#RSCRIPT	Read script command		•
#LSCRIPT	List script names command		•
#DSCRIPT	Delete script command		•
#REBOOT	Reboot command		•

5 Service and SW / Firmware Update

The RS232 serial interface on the Terminal used for the communication with a PC can also be used to update the **Telit EZ10 Terminal Family** firmware (Software).

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

The update procedure requires to switch-on the Terminal at a prompt within the running update tool. Thus, the power supply line shall be disconnected before the launch of the update tool on the PC and prepared for re-connection.

NOTE: Switching OFF the EZ10 by disconnecting the external power supply line requires to wait enough time (depending on the HW version!) that the load capacitor of the DC/DC converter inside the EZ10 discharges so that the GSM862 module inside goes to OFF condition.

5.1 Step-by-Step upgrade procedure

I. Collect information about the Hardware and implemented version of Software by the command
- AT+CGMR<enter>, which returns the Software version information;
- AT+CGMM<enter>, which returns the Model Identification.

II. Verify the need for the update and check the availability of the software for the Model and version of the EZ10, requesting it at this e-mail address:

mailto:ts-gm862@telit.com

Receive or download the specific software for upgrading (Transfer Flash Image GM862_xxx.zip), ca. 5-6 MB at present.

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

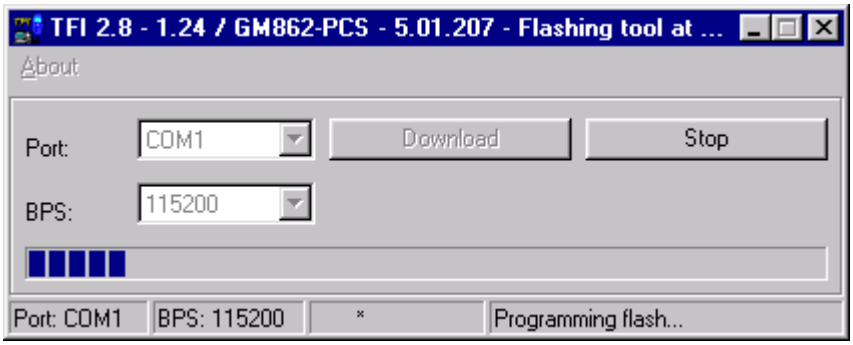
IV. Disconnect the EZ10 Terminal from its Power Supply and prepare for re-connection at the prompt of the running update tool.

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

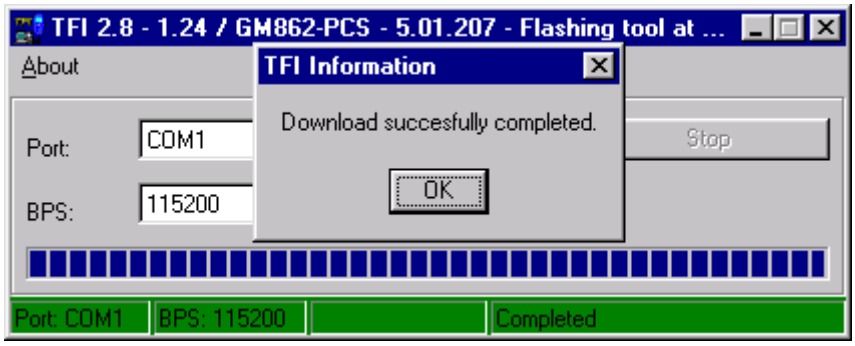
VI. Press OK to the initial message.

NOTE: In connection with the EZ10 Terminals, charged battery has to be understood that the power supply should not be disconnected or break down during the firmware update.

VII. 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 re-connect the EZ10 Terminal to its Power Supply.



Wait for the end of programming green message OK



The **Telit EZ10 Terminal** is now programmed with the new firmware.

6 Conformity Assessment Issues

The **Telit EZ10-PCS/-GPRS/ wireless modems** 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 EZ10 wireless modem has been tested in conformity with the following standards:

- EMC (Electromagnetic Compatibility). Standards: EN 301 489-1 and EN 301 489-7
- LVD (Low Voltage Directive) Standards: EN 60 950

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 for FCC approval is required.

The **Telit EZ10-PCS wireless modem** has been tested in conformity with the following US Directives:

- EMC (Electromagnetic Compatibility). Standards: FCC47 Part 15

Anatel (Brazil) Certifications are available for:

Nº Homologação	Número Uso Sitar	Modelo do Produto
0410052618	1131605WWW2618 1131705WWW2618 1153705WWW2618	TELIT TERMINAL MODEM EZ10 GM862-PCS TELIT MODULO GM862-PCS TELIT MODULO GM862-PCS-Python
0411052618	1131805WWW2618 1131905WWW2618	TELIT TERMINAL MODEM EZ10 GM862-GSM/GPRS TELIT MODULO GM862-GSM/GPRS

7 EZ10 Technical Support

Telit's technical support to Telit EZ10 wireless modem customers:

- All available technical documentation is included for download into the Website www.telit.com >Products >Modules >EZ10 Family > selected model.
- Telit's engineering support is accessible via a selective E-Mail service with 48 hr replies assured under normal conditions: mailto:ts-gm862@telit.com or mailto:ts-modules@telit.com from ca. October 2005 onwards.

In case of technical inquiries, the following information would be relevant to optimize Technical Support:

- Write Company, Project, Product Type, Trouble and Person reference in the "subject" field of the e-mail so that all mails can be easily retrieved also after several forwards
- The e-mail text should report:
  - Product Type
  - Delivery Date
  - Serial Number S/N of the EZ10 Terminal e.g. 4205
  - LOT N.
  - HW and SW code xSxxxxxxHxx
  - HW version of the EZ10 Terminal: v.xx
  - Installed SW version (AT+CGMR)
  - IMEI (AT+CGSN)
  - Description of the Application, reference to its version (SW, HW)
  - SIM Type (issued by which Mobile Network Operator of Home PLMN or Service Provider and SIM type & supplier)
  - Network Conditions: location, registered network, coverage (AT#CSURV)
  - Antenna type
  - Used Services (MO, MT, voice, SMS, data, fax, GPRS)

…and then the question or trouble:

- Command & Response sequences
- Listing of the relevant parts of a Python Script
- Signal / pin, timing, levels…

Thank you!

8 Safety Information

NOTE1: BEFORE connecting the Telit EZ10 Terminal to a Power Supply source, a suitable Antenna shall be connected and properly installed.

NOTE2: BEFORE connecting the Power Supply equipment to the mains socket, verify at least that the Input Voltage (V) and Frequency (Hz) range indicated on the equipment includes the Voltage and Frequency of the present mains (grid).

NOTE3: Always verify the **Type Labels** on the under side of the **Telit EZ10 Terminals** in order to identify the model and especially the **version of the Interfaces at the RJ11 connector**. This connector can bear AUDIO lines or GPIO lines. It can be identified by the writing **AUDIO** in older production lots or by the writing **AUX=AUDIO or AUX=GPIO** in newer production lots.

The Telit EZ10-PCS with PYG Option has General Purpose Inputs/Outputs instead of the Audio/Remote Power ON/OFF lines on the RJ11 connector.

Before connecting any equipment and before taking the EZ10 into operation, make sure which configuration applies according to the type label of the EZ10 and that this configuration is compatible with the equipment and the Application (as DTE) which uses the EZ10.

NOTE4: The product has to be handled with care, avoiding any contact with the connector pins, because electrostatic discharges may damage the product itself. The same cautions have to be taken for the SIM, checking carefully the instructions for its use. **Do not insert or remove the SIM when the product is in power saving mode.**

8.1 Precautions When Using Batteries

The **Telit EZ10 wireless modem** as such does not contain batteries, but in some applications external batteries might be used, for which the following precautions are valid:

- Never use any charger or battery that is damaged in any way.
- Use the battery only for its intended purpose.
- If you use the **Telit EZ10 wireless modem** near the network's base station, it uses less power; talk and standby times are greatly affected by the signal strength on the cellular network and the parameters set by the network operator.
- Battery charging time depends on the remaining battery charge and the type of battery and charger used. The battery can be charged and discharged hundreds of times, but it will gradually wear out. When the operation time (talk time and standby time) is noticeably shorter than normal, it is time to buy a new battery.
- If left unused, a fully charged battery will discharge itself over time.
- Use only approved chargers. When a charger is not in use, disconnect it from the power source. Do not leave the battery connected to a charger for more than a week, since overcharging may shorten its life..
- Extreme temperatures will affect the charging capacity of your battery: it may require cooling or warming first.

- Do not leave the battery in hot or cold places, such as in a car in summer or winter conditions, as you will reduce the capacity and life-time of the battery. Always try to keep the battery at room temperature. A **Telit EZ10 wireless modem** with a hot or cold battery may temporarily not work, even when the battery is fully charged. Li-ion batteries are particularly affected by temperatures below 0°(32°F).
- Do not short – circuit the battery. Accidental short-circuiting can occur when a metallic object (coin, clip or pen) causes a direct connection between the + and – terminals of the battery (metal strips on the battery), for example when you carry a spare battery in a pocket or bag.
- Short – circuiting the terminals may damage the battery or the object causing the short – circuit.
- Dispose of used batteries in accordance with local regulations. Always recycle. Do not dispose of batteries in a fire.

8.2 Road Safety

- Remember, road safety always comes first!
- Do not use a hand-held audio- or control device together with a **Telit EZ10 wireless modem** while driving a vehicle. Always park the vehicle before having a conversation.
- Make sure that the **Telit EZ10 wireless modem** is stored safely and will not fall or be broken in the event of a collision or emergency stop.
- Only qualified personnel should install or service the **Telit EZ10 wireless modem** in a vehicle. Faulty installation or service may be dangerous and may invalidate any warranty applicable to the unit.
- Electronic fuel injection, anti-skid braking, electronic cruise control or any other electronic systems may malfunction due to the lack of protection from radio signals. Check regularly that all cellular **Telit EZ10 wireless modem** equipment in your vehicle is mounted and operating correctly.

8.3 Operating Environment

- Remember to follow any special regulations in force in any area and always switch off your **Telit EZ10 wireless modem** whenever it is forbidden to use it, or when it may cause interference or danger (in a hospital for example).
- Operation of any radio-transmitting equipment, including cellular **Telit EZ10 wireless modem**, may interfere with inadequately protected medical devices. Consult a doctor or the manufacturer of the medical device if you have any questions. Other electronic equipment may also be subject to interference.
- As with other mobile radio-transmitting equipment, you are advised that for satisfactory operations and personal safety, the equipment should only be used in the normal operating position.
- Always switch off the **Telit EZ10 wireless modem** when at a refueling point (service station). You are reminded of the need to observe restrictions on the use of radio equipment in fuel depots (fuel storage and distribution areas), chemical plants or where blasting operations are in progress.
- Do not store or carry flammable liquids, gases or explosive materials in the same Compartment as the **Telit EZ10 wireless modem**, its parts or accessories.

- Switch off our cellular **Telit EZ10 wireless modem** when in aircraft. The use of cellular **Telit EZ10 wireless modems** in aircraft is illegal and may be dangerous to the operation of the aircraft or disrupt the cellular network. Failure to observe these instructions may lead to the suspension or denial of cellular telephone services to the offender, legal action or both.

8.4 Care and Maintenance

Your **Telit EZ10 wireless modem** is a product of superior design and craftsmanship and should be treated with care. The suggestions below will help you fulfill the warranty obligations and enjoy this product for many years.

- Keep the **Telit EZ10 wireless modem** and all its parts and accessories out of the reach of small children.
- Keep the **Telit EZ10 wireless modem** dry. Precipitation, humidity and liquids containing minerals will corrode the electronic circuits.
- Do not use or store the **Telit EZ10 wireless modem** in dusty, dirty areas as its components may be damages.
- Do not store the **Telit EZ10 wireless modem** in hot areas. High temperatures can shorten the life of electronic devices, damage batteries and warp or melt certain plastics.
- Do not store the **Telit EZ10 wireless modem** in cold areas. When the **Telit EZ10 wireless modem** warms up to its normal temperature, moisture can form inside the **Telit EZ10 wireless modem**, which may damage the electronic circuits.
- Do not attempt to open the **Telit EZ10 wireless modem**'s casing. Non-expert handling of the **Telit EZ10 wireless modem** may damage it.
- Do not drop or knock the **Telit EZ10 wireless modem**. Rough handling may damage the internal circuits.
- Do not use harsh chemicals, cleaning solvents or strong detergents to clean the **Telit EZ10 wireless modem**. Wipe it with a soft cloth slightly dampened in mild, soapy water.
- If the **Telit EZ10 wireless modem** or any of its accessories are not working properly, take them to your nearest qualified service center. The personnel there will assist you, and if necessary, arrange for the **Telit EZ10 wireless modem** to be repaired.

8.5 Emergency Calls

IMPORTANT! This **Telit EZ10 wireless modem**, like any cellular phone, uses radio signals, cellular and landline networks, as well as user-programmed functions that cannot guarantee connection in all conditions. Therefore, you should never rely solely on any cellular **Telit EZ10 wireless modem** for essential communications (medical emergencies for example).

Remember, to make or receive any calls, the **Telit EZ10 wireless modem** must be switched on and in a service area with adequate cellular signal strength.

Emergency calls may not be possible on all cellular phone networks or when certain network services and/or **Telit EZ10 wireless modem** features are in use. Check with local cellular service providers.

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
I2C, IIC	Inter IC Bus
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
SAR	Specific Absorption Rate (e.g. of the body of a person in an electromagnetic field)
SCA	Service Center Address
SIM	Subscriber Identity Module
SMD	Surface Mounted Device
SMS	Short Message Service
SMSC	Short Message Service Center
SPI	Serial Protocol Interface
SS	Supplementary Service
TIA	Telecommunications Industry Association
UDUB	User Determined User Busy
USSD	Unstructured Supplementary Service Data

10 Document Change Log

Revision	Date	Changes
1	22/07/05	Maximum voltage levels on several interfaces reduced Status LED timing inverted Replaced detailed description of AT commands with an availability table and reference to AT Commands Description Document. Added Jammed Detect & Report Extension Added Easy GPRS Extension General revision and introduction of ON/OFF functions Added Firmware (SW) Update chapter Added: Safety Information
2	06/09/05	Products on delivery defined as EZ10-GPRS with Audio and EZ10-PCS with PYG Option Added EASY SCRIPT / Python Interpreter with adaptation to EZ10 Added GPIO interface description Added Technical Support Trouble Description format proposal Added Anatel Brazil conformity issue Revision of Telit Part No. and Order Codes Revision of Accessory Specifications Revision of Safety Information I2C / IIC, SAR and SPI added to List of Acronyms
3	26/09/05	RJ11 AUX Vmod as power output GPIO voltage levels unified to input high = 3,6V net weight: ca. 150gr