GC864-QUAD-C2 / PY-C2 Software User Guide

GC864-QUAD-C2 and GC864-PY-C2 Software User Guide
1vv0300745 Rev. 1 - 26/10/06

Making machines talk.

This document is relating to the following products:

Model	P/N
GC864-QUAD-C2	3990250681
GC864-PY-C2	3990250686

Contents:

1 Overview

The purpose of this document is the description of some common AT command procedures that may be used with the **Telit GC864- C2 module**.
In this document, all the basic functions of a mobile phone will be taken into account and for each one of them, a proper command sequence will be suggested.
In the Advanced operation section the more useful services and features of the GSM network supported by the **Telit GC864-C2 module** is taken into account and some command sequence and usage are provided for each one of them.
This document and its suggested command sequences shall not be considered mandatory; instead, the information given shall be used as a guide for properly using the **Telit module**. For further commands and features that may not be explained in this document refer to the AT Commands Reference Guide[1] where all the supported AT commands are reported.

NOTE

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

[1] AT commands available for GC864-C2 are the same as for GE864-PY/QUAD

2 Basic Operations

2.1 Interface Style

The GC864-C2 module is using an AT interface that is defined in the document 80000ST10025a (AT Commands Reference Guide).
The specification defines 3 possible AT interfaces:

0 - AT command interface of the products, to the GM862-GSM and GM862-GPRS interface style
1 - AT interface of the products, to the GM862-PCS, PYTHON, QUAD-PY, TRIZIUM and GE863-QUAD, PY interface style
2 - switches the AT command interface style of the product, to the new products like GE864, GC864 and the GPS products

The default interface for the GC864-C2 product is 2.

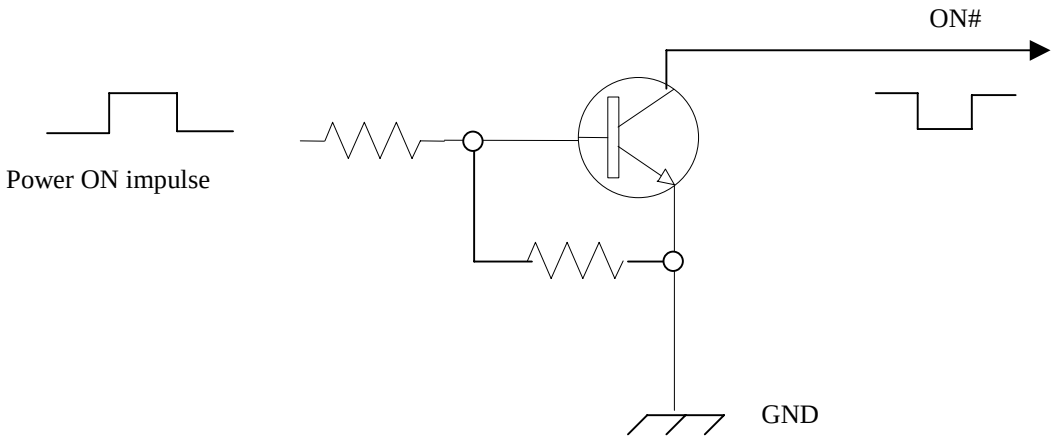
The switch between the different interfaces could also be performed with the ***#SELINT*** AT command. Refer to the AT Commands Reference Guide[1] for the full command description.

All the AT commands described in this specification is related to SELINT 2

[1] AT commands available for GC864-C2 are the same as for GE864-PY/QUAD

2.2 Turning ON the GC864-C2

To turn on the GC864-C2 the pin ON# must be tied low for at least 1 second and then released.
A simple circuit to do it is:



NOTE: don't use any pull up resistor on the ON# line. Using pull up resistor may bring to latch up problems on the GC864-C2 power regulator and improper power off of the module. The line ON# must be connected only in open collector configuration.

TIP: To check if power has raised it is possible to monitor the PWRMON line, when this line goes high the module is powered on, but before it remains on the device needs other 900 ms for software startup. Hence check the PWRMON line and 900 ms after its transition to high it is possible to release the ON# pin.

2.3 Turning OFF the GC864-C2

The turning off of the device can be done in two ways:

- by software command
- by hardware shutdown

When the device is shut down by software command or by hardware shutdown, it issues to the network a detach request that informs the network that the device will not be reachable any more.

2.3.1.1 Software shutdown

- Send command **AT#SHDN<cr>**
- wait for **OK** response

The device shuts down immediately after the issue of the OK response.

2.3.1.2 Hardware shutdown

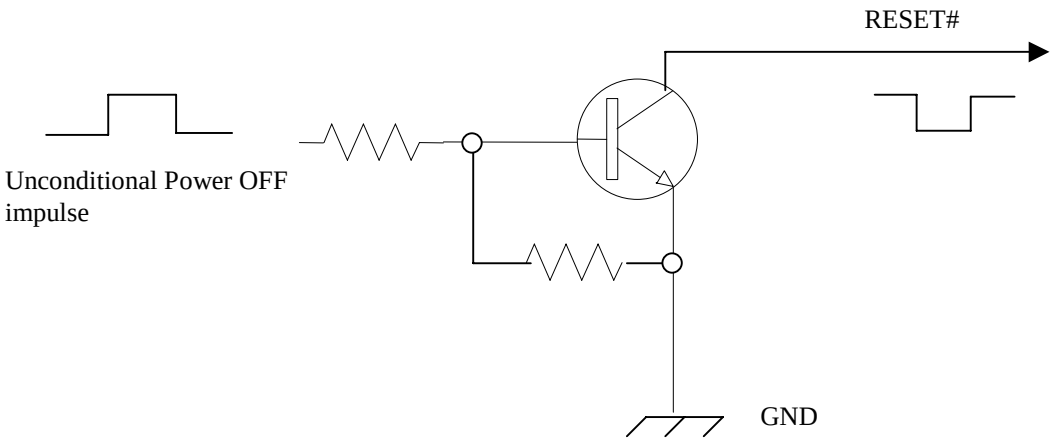
To turn OFF the GC864-C2 the pin **ON#** must be tied low for at least 2 second and then released.
The same circuitry for the power on can be used.
The device shuts down after the release of the **ON#** pin.

TIP: To check if the device has powered off, the hardware line PWRMON should be monitored. When it goes low, the device has powered off.

2.3.1.3 Hardware Unconditional RESTART

To unconditionally RESTART the GC864-C2 the pin **RESET#** must be tied low for at least 200 milliseconds and then released.

A simple circuit to do it is:



NOTE: don't use any pull up resistor on the RESET# line nor any totem pole digital output. Using pull up resistor may bring to latch up problems on the GC864-C2 power regulator and improper functioning of the module. The line RESET# must be connected only in open collector configuration.

TIP: The unconditional hardware RESTART should be always implemented on the boards and software should use it as an emergency exit procedure.

2.4 Checking GSM Device functionality

After a proper power on the device is ready to receive AT commands on the serial port.
Several things have to be checked in order to be sure that the device is ready to send and receive calls and SMS.
For more information refer to *GE864-QUAD/PY Software User Guide 1vv0300741.*

2.5 Call Functions

For more information about the voice call and CSD data call please refer to *GE864-QUAD/PY Software User Guide 1vv0300741*.

3 Advanced Operations

3.1 Accessing the phonebook

The GC864-C2 can access the phonebook storage of the SIM card inserted, by using specific AT commands it is possible to store and recall phone numbers and their associated name.
For more information refer to *GE864-QUAD/PY Software User Guide 1vv0300741*.

3.2 Distinguish Calls

The Telit GC864-C2 is able to identify the call type before answering; it is so possible to have different ring indications (unsolicited codes) depending on the call type.
For more information refer to *GE864-QUAD/PY Software User Guide 1vv0300741*.

3.3 GSM Power Saving function

The Telit GC864-C2 has a special function that reduces the power consumption during the idle time, thus allowing a longer standby time with a given battery capacity.

For more information refer to *GE864-QUAD/PY Software User Guide 1vv0300741*.

3.4 SMS handling

The Telit GC864-C2 supports the Short Message Service. It is possible to store, delete, write, send and receive a SMS, which is a short text message up to 160 characters long.

For more information refer to *GE864-QUAD/PY Software User Guide 1vv0300741*.

3.5 Using General Purpose Input/Output pins

The **Telit GC864-C2** provides various General Purpose Input/Output pins. These pins can be configured via AT commands as Inputs or Outputs , but two of them as "*alternate function*".
The "*alternate function*" is supported by pins *GPIO5 (*which can be configured to become a RF Transmission monitor output pin that reflects the RF transmission activation) and *GPIO7* (which can be configured to become a buzzer output pin).
With these pins your application can control external hardware directly using the **Telit GC864-C2** pins, with little or even no hardware added.

3.5.1 GPIO pin setup

Before using the GPIO pins, you must configure them to select their direction or alternate function.

3.5.1.1 Setting GPIO pin as OUTPUT

When you set a GPIO as output, you must specify also the value that the pin output must take:

- Send command **AT#GPIO=<pin>,<value>,1<cr>** where:

<pin> is the GPIO pin number at which the command applies:

1 – GPIO1
2 – GPIO2
3 – GPIO3
4 – GPIO4
5 – GPIO5
7 – GPIO7

<value> is the GPIO pin value that the pin will assume:

0 – LOW
1 – HIGH

- wait for response **OK**

NOTE: The #GPIO setting is not saved and will be lost on power off; so at start-up repeat pin initialization commands. At start-up the setting for GPIO7 instead is maintained even after a shutdown to permit buzzer feature to work always.

For example:
1- Let's assume you want to set GPIO3 pin as Output and you want it to be in LOW status:
command:
AT#GPIO=3,0,1<cr>
response:
OK

In this case, the GPIO3 pin was successfully put in output direction and its status has been set to LOW.

3.5.1.2 Setting GPIO pin as INPUT

When you set a GPIO as input, you must specify also a dummy value for the pin state:

- Send command **AT#GPIO=<pin>,<dummy_value>,0<cr>** where:

<pin> is the GPIO pin number at which the command applies:

1 – GPIO1
2 – GPIO2
3 – GPIO3
4 – GPIO4
5 – GPIO5
7 – GPIO7

<value> is a dummy value can be either:

0 – dummy value
1 – dummy value

- wait for response **OK**

NOTE: the #GPIO setting is not saved and will be lost on power off, so at start-up repeat pin initialization commands.
At start-up all the GPIOs are configured by default as INPUT, except GPIO7 whose setting instead is maintained even after a shutdown to permit buzzer feature to work always.

For example:
1- Let's assume you want to set GPIO4 pin as Input:
command :
AT#GPIO=4,0,0<cr>
response :
OK

In this case, the GPIO4 pin was successfully put in input direction.

3.5.2 GPIO pin use

After having set-up the GPIO pin direction you can query the input status of an INPUT pin or set the output status of an OUTPUT pin.

3.5.2.1 Querying GPIO pin status

To query for the pin status:

- Send command **AT#GPIO=<pin>,2<cr>** where :

<pin> is the GPIO pin number at which the command applies:

1 – GPIO1
2 – GPIO2
3 – GPIO3
4 – GPIO4
5 – GPIO5
7 – GPIO7

- wait for response in the format:

#GPIO: <dir>,<stat>
OK

where:

<dir> - GPIO<pin> direction setting
<stat> - status of the pin
0 - LOW
1 - HIGH

NOTE: In case the GPIO pin direction is set to ALTERNATE FUNCTION (2), then the reported <stat> has no meaning and shall not kept as valid, but shall be threaten as a dummy value.

TIP: The query reports depending on the pin direction:
- ***the read pin status in case the direction is input;***
- ***the previously set pin status in case the direction is output.***

In any case, you can know if the pin at the query moment is high or low and the pin direction.

For example:
1- Let's assume you want to query the GPIO3 pin for its status:
command:
AT#GPIO=3,2<cr>
response:
#GPIO: 0,1
OK
In this case, the GPIO3 pin was set in input direction and its status has been measured to be HIGH.

2- Let's assume you want to query the GPIO4 pin for its status:
command:
AT#GPIO=4,2<cr>
response:
#GPIO: 1,0
OK

In this case, the GPIO4 pin was set in output direction and its status is LOW.

3- Let's assume you want to query the GPIO7 pin for its status:
command:
AT#GPIO=7,2<cr>
response:
#GPIO: 2,0
OK
In this case, the GPIO7 pin was set in "*alternate function*" direction and therefore works as buzzer output. The reported status = LOW has no meaning.

3.5.2.2 Setting GPIO pin output status

To set the pin status (when pin is set as OUTPUT):

- Send command **AT#GPIO=<pin>,<value>,1<cr>** where:
-

<pin> is the GPIO pin number at which the command applies:

1 – GPIO1
2 – GPIO2
3 – GPIO3
4 – GPIO4
5 – GPIO5
7 – GPIO7

<value> is the pin value to be set and can be:

0 – LOW
1 – HIGH

- wait for response **OK**

For example:
1- Let's assume you want to set the GPIO3 pin HIGH:
command:
AT#GPIO=3,1,1<cr>
response:
OK
In this case, the GPIO3 pin was set in output direction and its status has been set to HIGH.

3.5.2.3 Using GPIO7 pin as BUZZER OUTPUT (alternate function)

When you set the GPIO7 pin as Buzzer output function, the pin will output a waveform suitable to drive a Buzzer, provided a simple external mosfet driver is developed and that the *#SRP* settings are adequate. To set the pin in alternate function you must specify also a dummy value for the pin state:

- Send command **AT#GPIO=7,<dummy_value>,2<cr>** where:

<value> is a dummy value can be either:
0 – dummy value
1 – dummy value

- wait for response **OK**

TIP: Remember that the alternate function places the GPIO7 pin always in OUTPUT direction and since the GPIO7 pin value is controlled by the internal software, the corresponding function (#SRP) must be setup properly.

NOTE: The #GPIO7 direction setting is saved and will be kept after a power off.

For example:
1- Let's assume you want to set GPIO7 pin as BUZZER OUTPUT:
command:
AT#GPIO=7,0,2<cr>
response:
OK

In this case, the GPIO7 pin was successfully put in buzzer output direction.

3.6 Clock/Alarm function

The **Telit GC864-C2** provides a Real Time Clock and Alarm embedded in the product; it is therefore possible to set-up the proper time, check the actual time, set-up an alarm time at which the alarm will be triggered with various behavior depending on the +CALA setting.
The only requirement is that the power input to the **Telit GC864-C2** has to be guaranteed without interruptions, the **Telit GC864-C2** has no backup battery; therefore it will lose the time setting if its power supply is interrupted.

On Alarm trigger the **Telit GC864-C2** can:

- automatically Wake-up fully operative from shutdown as if the ON/OFF

- automatically Wake-up from shutdown in a special status namely "alarm status" where it will not look for or try to register into any network, as if it would be off, except from the fact that it proceeds with the alarm action and it can receive commands to return completely operative or shutdown immediately.
- If already ON at alarm trigger time, simply proceed with the Alarm action

Once Woken-up the **Telit GC864-C2** proceeds with the chosen action that can be

- issue an unsolicited code "+ALARM: <user_text>" on the serial port until a 90s timeout expires or a special Wake-up command is received
- play an Alarm tone until a 90s timeout expires or a special Wake-up command is received
- any combination of these actions

With these features, the **Telit GC864-C2** for example can:

- Wake-up itself and its controlling hardware by using the GPIO6 pin at the desired time, so timely surveys can be programmed without the need to keep the any hardware on and therefore reducing power consumption to a minimum.
- Alert the controlling application that the alarm time has come with the unsolicited code "+ALARM:<user_text>"
- Alert the user with the alarm tone played

3.6.1 Clock date/time

Before using the Alarm feature, you must regulate the internal clock.

3.6.1.1 Regulate the Clock

- Send command **AT+CCLK="<time>"<cr>** where:

<time> - current time as quoted string in the format : "yy/MM/dd,hh:mm:ss±zz"
yy - year (two last digits are mandatory), range is 00..99
MM - month (two last digits are mandatory), range is 01..12
dd - day (two last digits are mandatory), range is 01..31 (if the month MM has less than 31 days, the clock will be set for the next month)
hh - hour (two last digits are mandatory), range is 00..23
mm - minute (two last digits are mandatory), range is 00..59
ss - seconds (two last digits are mandatory), range is 00..59
±zz - time zone (indicates the difference, expressed in quarter of an hour, between the local time and GMT; two last digits are mandatory), range is -47..+48

Note: If the parameter is omitted the behavior of *Set* command is the same as *Read* command.

- wait for response **OK**

TIP: Remember that the string time has to be encapsulated in double brackets.

NOTE: The time will start immediately after the time setting command.

For example:
1- Let's assume you want to regulate your clock to 7th November 2002 at 12h 24m 30s for the time zone +01h central Europe:
command:
AT+CCLK="02/11/07,12:24:30+04"<cr>
response:
OK
In this case, the time was successfully set.

3.6.1.2 Read the current date/time

- Send command **AT+CCLK?<cr>**

- wait for response in the format:

+CCLK: <time>
OK

Note: the three last characters of **<time>** are not returned by **+CCLK?** because the **ME** doesn't support time zone information.

For example:
1- Let's assume you want now to read the current time:
command:
AT+CCLK?<cr>
response:
+CCLK="02/11/07,12:26:47"<cr>
OK
In this case the current date/time is: 7th November 2002 12h 26m 47s

3.6.2 Alarm function

Once the current time has been set, the alarm function can be setup.

3.6.2.1 Regulate the Alarm time & behavior

- Send command **AT+CALA="<time>",0,<type>,"<text>"<cr>**

where:
<time> is the Alarm time string in the same format of the clock setting command:
yy/MM/dd,hh:mm:ss±zz
where:
yy : two digits year (00-99)
MM : two digits month (01-12)

dd : two digits day (01-31)
hh : two digits hour (00-24)
mm : two digits minute (00-60)
ss : two digits seconds (00-60)
±zz: signed two digits timezone (-11 - +11)

<type> is the Alarm behavior:

0 - reserved for other equipment use.

1 - the MODULE simply wakes up fully operative as if the **ON/OFF** button had been pressed. If the device is already **ON** at the alarm time, then it does nothing.

2 - the MODULE wakes up in "alarm mode" if at the alarm time it was off, otherwise it remains fully operative. In both cases the MODULE issues an unsolicited code every 3s:

+ALARM: <text>

where **<text>** is the **+CALA** optional parameter previously set.

The device keeps on sending the unsolicited code every 3s until a **#WAKE** or **#SHDN** command is received or a 90s timeout occurs. If the device is in "alarm mode" and it does not receive the **#WAKE** command within 90s then it shuts down. (default)

3 - the MODULE wakes up in "alarm mode" if at the alarm time it was off, otherwise it remains fully operative. In both cases the MODULE starts playing the alarm tone on the selected path for the ringer (see command **#SRP**)
The device keeps on playing the alarm tone until a **#WAKE** or **#SHDN** command is received or a 90s timeout occurs. If the device is in "alarm mode" and it does not receive the **#WAKE** command within 90s then it shuts down.

4 - the MODULE wakes up in "alarm mode" if at the alarm time it was off, otherwise it remains fully operative. In both cases the MODULE keeps it in this state until a **#WAKE** or **#SHDN** command is received or a 90s timeout occurs. If the device is in "alarm mode" and it does not receive the **#WAKE** command within 90s then it shuts down.

5 - the MODULE will make both the actions as for **<type>=2** and **<type>=3**.

6 - the MODULE will make both the actions as for **<type>=2** and **<type>=4**.

7 - the MODULE will make both the actions as for **<type>=3** and **<type>=4**.

<text> - unsolicited alarm code text string. It has meaning only if **<type>** is equal to 2 or 5 or 6.

- wait for response **OK**

TIP: Remember that the string time has to be encapsulated in double brackets, furthermore the Alarm time will not be computed for different timezone, therefore the alarm time will always refer to the same timezone as the clock setting regardless the timezone set in the +CALA command.

NOTE: if you use the unsolicited codes +ALARM: <text> , then you must fix the port speed rate and store it in the active profile (see command &W), in order to make the Telit GC864-C2 boot with the desired port speed, otherwise at the alarm wakeup, the module will start with the default port speed that may differ from yours.

3.6.2.2 Stop the Alarm activity

When the alarm time expires, the module starts the alarm activity according to the alarm behavior parameter <type> selected.
To stop the Alarm activity there are three ways, you can either decide to exit from alarm and shutdown the device or exit from alarm and entering the normal operational status; otherwise you can leave the alarm go on until the 90s timeout is reached.

3.6.2.2.1 Exit from the alarm status and shutdown

- Send command **AT#SHDN<cr>**
- wait for response **OK**

At the OK result code, the device will end alarm activity and shutdown.

3.6.2.2.2 Exit from the alarm status and enter the normal operating mode

- Send command **AT#WAKE=0<cr>**
- wait for response **OK**

At the OK result code, the device will end alarm activity and enter normal operating mode. If the device was already in normal operating mode (alarm has started when the module was already ON), then with the command only the alarm activity is terminated.

3.6.2.3 Querying the Alarm status

When the device awakes by means of an alarm time expire, the module starts the alarm activity but not the network activity, permitting some operations to be done by the controlling application without registering the mobile in the network.
To check if the mobile is in the "alarm status" and therefore no network activity is done or if the device is in the normal operating status:

- Send command **AT#WAKE?<cr>**
- wait for response in the format:

+WAKE: <status>
OK

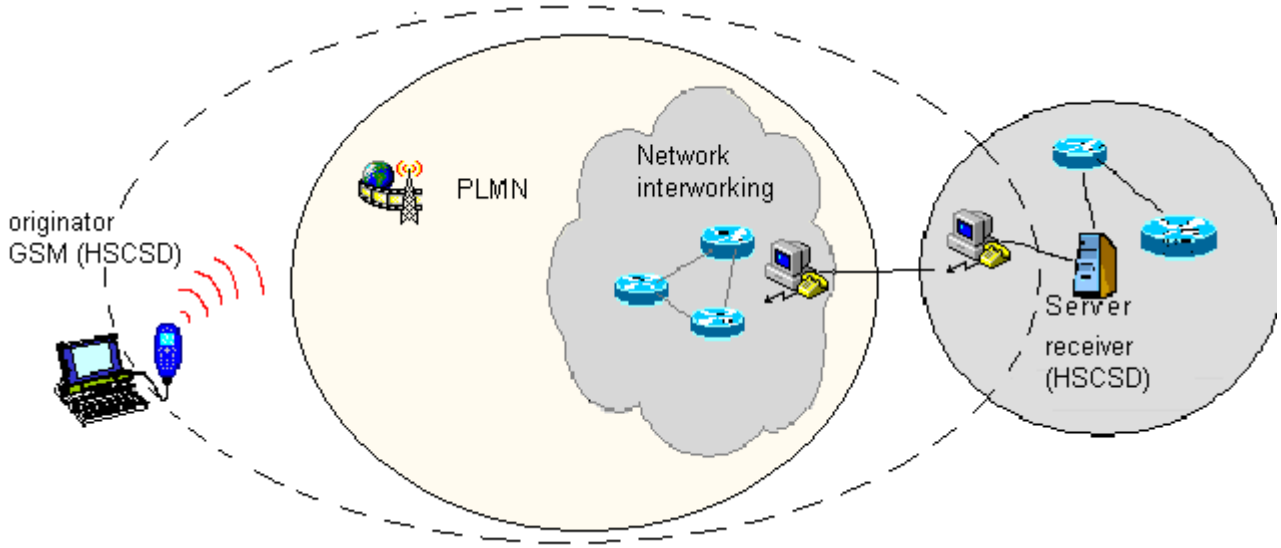
where:
<status> is the operating mode:
0 - normal operating mode
1 - alarm mode

NOTE: if the device is in the alarm mode no network activity is done, therefore the only commands that are accepted are the #WAKE and #SHDN ones.
When in the alarm mode, no operation is allowed towards the network, therefore it is not possible to receive or send calls, SMS and whatever GSM/GPRS services.

4 GPRS operations

4.1 Introduction

The General Packet Radio Services (GPRS) standard permits DATA transfers in a completely different way with respect to previous point-to-point communications made with Circuit Switch Data (CSD) GSM modems.
In CSD operations the modem establishes a connection with the other party (another modem) in such a way that all the Network devices in between are transparent to the data exchanged, simulating a real point-to-point connection, just as if the other party is directly connected with the controlling application of the modem. The other party can be either an Internet Service Provider (ISP) or a private server, but in any case, the arrival point must have a modem to connect to (Landline, ISDN or GSM CSD). The connection establishment procedure defines a particular path where all the information exchanged between the two peers flows and this path is reserved for exclusive use of these 2 peers for all the time the connection is active.
This approach has the drawbacks of a long time to set-up the link between the two peers (up to a minute) and a time counting bill which proceeds even if no data is exchanged because the path resources are reserved anyway; furthermore the speed of the data transfer is limited to 14400 bps.
An example of this kind of operation is shown in the following picture, where the point to point connection is between the two peers as if all the devices inside the dashed line are not present:

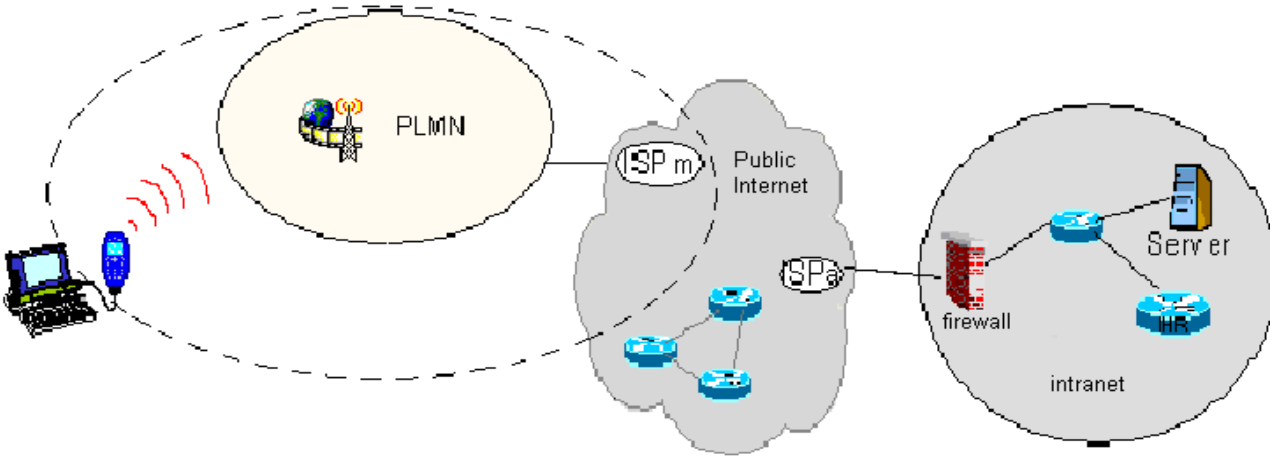


CSD interconnectivity

In GPRS operations instead, the connection is made directly towards Internet as if the GPRS modem was a network IP socket interface. There's no data path reserved for the data exchange between the two peers, instead the resources are allocated dynamically on demand and the data exchanged is

organized into packets typically TCP/IP, furthermore the maximum transfer speed can be much faster than GSM CSD.

An example of GPRS connection is shown in the following picture, where the GPRS connection is between the GPRS modem and the Internet as if all the devices inside the dashed line are not present:



GPRS interconnectivity

Due to this kind of connection, when activating the GPRS connection you must provide the network parameters to enter through the Internet point of the GPRS network ISP (Internet Service Provider) and not the phone number to be dialed; therefore it is not possible to establish a direct point to point GPRS connection between two modems as in CSD case, instead an Internet tunneling must be done to achieve a point to point connection between two peers.

This approach as the immediate advantage of projecting the controlling application of the GPRS modem directly on the internet, ready to be accessed virtually from anywhere in the world at the same cost on the GPRS; actually the billing of the GPRS connection is based on the amount of data exchanged (number of packets transferred) independently from the time the connection is active or where these packet must be delivered. Therefore, it is possible to leave the controlling application always connected and ready to receive/send data on demand, while paying only for the data really exchanged.

The drawback of the GPRS connection is that the controlling application must have its own TCP/IP protocol stack embedded to decode the packets that arrive from GPRS and encode the ones to be sent through the Internet.

There are few considerations than must be done on the GPRS connections:

- the GPRS connection speed with a GPRS class 8 multislot device is asymmetrical, 4 time slots in reception (57600 bps max) and 1 time slot in sending (14400 bps max).
- The controlling application of the module must have a TCP/IP - PPP software stack to interface with the GPRS modems.
- The controlling application must relay on some ISP that may be the Network Operator of the SIM to gain access to the Internet through the GPRS connection.
- Because of the point before, the receiving application must have Internet access either.
- Since the communication is based upon TCP/IP packets, then it is possible to talk contemporarily with more than one peer.

- When required, the data security in Internet shall be guaranteed by security protocols over the TCP/IP that must be managed by the controlling application.

A GPRS modem can be in 3 different states:

- DETACHED, which corresponds to the "not reachable" condition of a GSM mobile;
- ATTACHED, which corresponds roughly to the "registered" condition of a GSM mobile;
- CONNECTED, which roughly corresponds to the connected status of a GSM mobile;

A thing that must be noted on the GPRS connect, is the fact that, if the mobile IP address (the Internet address) is assigned by the ISP dynamically, then when the device is not connected it has no address and therefore it cannot be reached by Internet requests. The same thing occurs in the case the GPRS device has a static IP address assigned to it by the ISP, but it is DETACHED or attached but it does not support network initiated context activation.
In these cases there's no possibility for the Internet peer to "call" the GPRS device through Internet, the only way to alert it is to call it in GSM mode (either a Data or a Voice call are suited) and the GPRS module application must recognize the caller, eventually abort the GSM call and connect to the Internet in GPRS to receive the packets from the Internet peer.

To explain further the differences between CSD and GPRS an example application made in both ways will be shown.

4.1.1 CSD application example

Let's suppose you have several remote meteorological measurement units spread around the territory, and you want to access them wirelessly through a GSM module in CSD operation.
For each remote unit, there's a modem to connect with the server application, with its own SIM card and unique phone number.
Now there are two possibilities:

- the server application calls on demand the remote units, provided it has stored their phone numbers in a private database.
- the remote units call the server application modem when needed and eventually retry in the case they found it busy; this time the phone number to be stored is only one, the server number which must be stored on the remote units.

In both cases, once connected, the remote unit sends the meteorological data to the server, which places it in a central database for further reading by anyone who accesses the meteorological Internet site for example.
The drawback of this approach is that the CSD modem needs about 30s to establish the connection and, depending on the amount of data to be transferred (usually few hundreds bytes), some seconds to transfer them. So let say we pay a 40s call while we need only 10s to transfer data.

4.1.2 GPRS application example

The same application can be made with the GC864-C2 using the GPRS feature.
The remote unit is always connected to the Internet (at no charge) taking advance of the features of the GPRS system, when it needs to send data to the server application it simply fills the TCP/IP packets for the server with the meteorological data and gives them to the GC864-C2 to be delivered. The central server has a single modem to connect to the Internet, receives the TCP/IP packets from all the remote units and places the contained data in the central database.
The advantage of using GPRS is that the remote unit is always connected and reachable and it pays only for the amount of data (small) transferred and not for the connection time as in CSD operations; in addiction the call billing is equal for devices placed anywhere in the Network Operator State and the server can be anywhere in the World.
Furthermore, in the CSD operation the server shall have a set of modems and multiple phone lines to ensure that the calling units will not find it busy, while a single modem is enough for GPRS operation. The speed at which the packets can be transferred is up to 57600 bps (class 8 device), 4 times faster than CSD.

Now that's clear how GPRS works let see how to establish a connection:

4.2 Preliminary GPRS context parameters setting

For more information refer to *GE864-QUAD/PY Software User Guide 1vv0300741*.

4.3 GPRS context activation and data state entering

This operation corresponds to the dial and connect of a CSD GSM data call issued to an Internet service provider.

For more information refer to *GE864-QUAD/PY Software User Guide 1vv0300741*.

4.4 GPRS data state exit

For more information refer to *GE864-QUAD/PY Software User Guide 1vv0300741*.

4.5 Easy GPRS applications

For more information refer to *GE864-QUAD/PY Software User Guide 1vv0300741*.

5 FTP OPERATIONS

A set of AT commands is available to support the FTP activities.

For more information refer to *GE864-QUAD/PY Software User Guide 1vv0300741*.

6 Document Change Log

Revision	Date	Changes
ISSUE #0	28/09/06	First release
ISSUE #1	26/10/06	Added product GC864-PY-C2 in the document Changed footnote (1) from "*GC864-QUAD*" to "*GC864-PY/QUAD*" Pag.2 Added P/N 3990250686