GG863-SR Gateway Software User Guide

1vv0300836 Rev.2 - 05/05/2010

Making machines talk.

Applicable Products

Product
GG863-SR ZigBee
GG863-SR Plus
GG863-SR Pro

For further information on how to order **GG863-SR** specific model, please contact your Telit representative

This User Guide refers to the following software versions:

Uboot/Linux OS	Gateway SW
221.07.2000	SR: 2C.00.01

Contents

DISCLAIMER

The information contained in this document is the proprietary information of Telit Communications S.p.A. and its affiliates (“TELIT”). The contents are confidential and any disclosure to persons other than the officers, employees, agents or subcontractors of the owner or licensee of this document, without the prior written consent of Telit, is strictly prohibited.

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

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

Telit reserves the right to make modifications, additions and deletions to this document due to typographical errors, inaccurate information, or improvements to programs and/or equipment at any time and without notice. Such changes will nevertheless be incorporated into new editions of this document.

1 INTRODUCTION

1.1 Scope

This SW User Guide serves the following purpose:

- describes **GG863-SR** software architecture;
- describes the Telit Demo Gateway Application

1.2 Audience

This User Guide is intended for **GG863-SR** users who want to use Gateway functionalities.

For further information on how to design, develop and deploy custom application for **GG863-SR** please refer to section **§2.1 "Open platform**"

1.3 Contact Information, Support

Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.

For general contact, technical support, report documentation errors and to order manuals, contact Telit's Technical Support Center at:

TS-EMEA@telit.com
or
http://www.telit.com/en/products/technical-support-center/contact.php

Telit appreciates feedback from the users of our information.

1.4 Open Source Licenses

1.4.1 GNU General Public License

Linux system is made up of many Open Source device drivers licensed as follows:

GNU GENERAL PUBLIC LIC+
ENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Please refer to the following web page for the full text of the license:

http://www.gnu.org/licenses/gpl-2.0.html

1.5 Product Overview

The Telit **GG863-SR** Gateway brings together the GSM/GPRS and short range technology hosting a programmable **GE863-PRO^3** and any of the Telit short range modules.
Gateway functionalities are supported by the applications running in the ARM9 integrated in the **GG863-SR**. The applications manage data sending and receiving, set up and other operations related to GSM/GPRS or short range network and web interface.
Gateway application implements a protocol for data exchange between an IP Host and each node of the short-range domain, which is completely transparent to an external IP Host, including all short range functionalities inside the gateway. The application configuration is done through a Web based Management control panel.

1.6 Related Documents

[1] 80322ST10058a GG863-SR Product Description
[2] 1vv0300835 GG863-SR Hardware User Guide
[3] 1vv0300781 GE863-PRO^3 Linux Software User Guide
[4] 1vv0300777 GE863-PRO^3 U-BOOT Software User Guide
[5] 1vv0300819 Telit M-ONE Protocol Stack User Guide
[6] 1vv0300820 Telit Z-ONE Protocol Stack User Guide
[7] 1vv0300782 Telit GE863-PRO^3 Linux GSM Library User Guide
[8] 1VV0300780 Telit GE863-PRO^3 Linux Development Environment User Guide
[9] 1vv0300823 Telit Mesh Manager User Guide
[10] 1vv0300824 Telit TinyTools User Guide
[11] 1vv0300859 Telit ZigBee Democase Getting Started
[12] 80000nt10035 ZE modules configuration over serial port Application Note
[13] 1vv0300846 Telit ZTC User Guide

All documentation can be downloaded from Telit's web site www.telit.com if not otherwise indicated.

1.7 Document Organization

This manual contains the following chapters:

- Chapter 1 “INTRODUCTION”: provides a scope for this manual, target audience, technical contact information, and text conventions;
- Chapter 2 “GG863-SR Overview”: describes HW and SW architecture of the **GG863-SR** module;
  - **Chapter 2.4 “Getting started” can be used as a quick start guide for first time users;**
- Chapter 3 “Software applications” describes the set of applications that run on the **GG863-SR**;
- Chapter 4 “IP Gateway Application” describes the application that allows to use the SR Gateway functionalities
- Chapter 5 “Web Administration Panel” describes the **GG863-SR** Demo Gateway Application
- Chapter 6 “Auto Application” explains how the Auto Application works.
- Chapter 7 “Demo Gateway Application – Remote Configuration” describes how to configure the PC demo gateway application when using a GPRS connection
- Chapter 8 “APPENDIX” provide quick reference guide for common tasks on management of short range devices

1.8 Text Conventions

Danger – This information MUST be followed or catastrophic equipment failure or bodily injury may occur.

Caution or Warning – Alerts the user to important points about integrating the module, if these points are not followed, the module and end user equipment may fail or malfunction.

Tip or Information – Provides advice and suggestions that may be useful when integrating the module.

1.9 Document Change Log

Revision	Date	Changes
ISSUE#0	09/07/09	First release
ISSUE#1	23/07/09	MODIFIED: 1.4.1 GNU General Public License 2.4.1: Firmware filenames 2.4.2: Added note for Ethernet gadget 4.3 and subsections: table formatting style, notes are highlighted
ISSUE#2	05/05/10	MODIFIED: 2.3 Software Architecture 2.5 Demo Gateway Application – Local Configuration

		4.1 Overview 4.2 Control Interface 4.3 Data interface 5.3 How it works 6 Auto Application 7 Demo Gateway Application – Remote Configuration 8.1 Appendix.1 – Gateway Configuration 8.2 Appendix.2 – How to add a new Zigbee node 8.3 Appendix.3 – How to add a new Mesh Lite node ADDED: 8.5 Appendix.5 – How to send and receive raw data using MeshLite technology 8.6 Appendix.6 – Code example for Mesh Lite echo implementation

2 GG863-SR Overview

The **GG863-SR** is an all-in-one gateway solution which routes data between short range networks and IP networks. The **GG863-SR** brings together the GSM/GPRS and short-range technology, hosting **GE863-PRO³** with dedicated application processor and any of the short-range modules from Telit's wide product offer.
GG863-SR gateway application runs on a Linux OS, equipped with dedicated ARM9 32 bit processor running at 200MHz, a GSM module and a short range module, making it a turnkey solution for quick short range network setup and remote management.

2.1 Open platform

Thanks to the fully programmable **GE863-PRO³**, its dedicated development environment and a wide range of libraries and APIs, the **GG863-SR** is the perfect platform for running custom applications.

This guide is focused on all information related to the gateway application only. For further information onto custom application development on **GE863-PRO³** please refer to the documentation cited below.

If you are new to **GE863-PRO³** development, you are invited to read documentation in the following order:

- 80285ST10036a Telit GE863-PRO³ Product Description
- 1vv0300781 Telit GE863-PRO³ Linux Software User Guide
- 1vv0300780 Telit GE863-PRO³ Linux Development Environment User Guide
- 1vv0300782 Telit GE863-PRO³ Linux GSM Library User Guide

2.2 Hardware architecture

Figure below shows the hardware architecture of **GG863-SR** along with indications about available connections and peripheral. For detailed information about hardware implementation, please refer to [2].

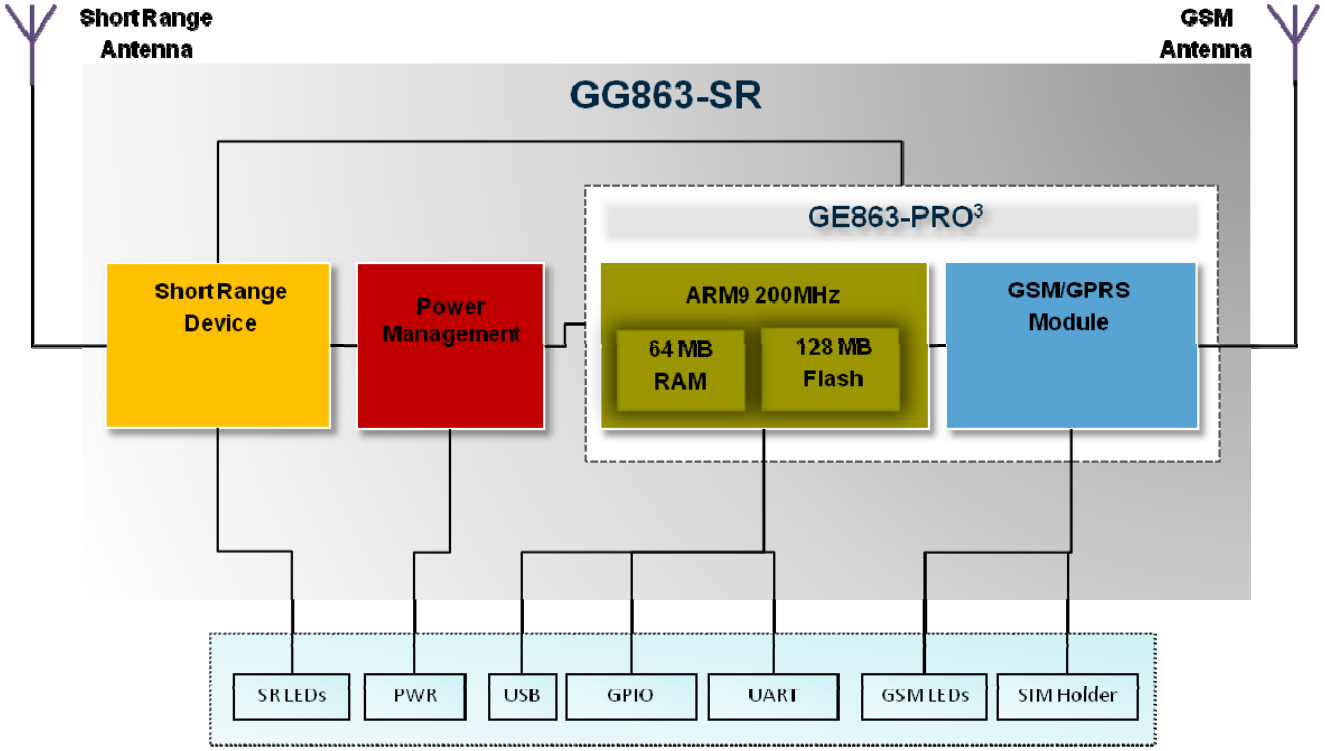


Figure 2.1 - Hardware Architecture

As specified above, **GG863-SR** provides a variety of connectors:

- USB type mini B (device)
- power supply (4 PIN molex connector)
- Sim holder
- RJ11 with UART and 2 GPIOs
- status and power LEDs, for GSM and Short Range.

For detailed information on connectors pinout and LED indication, please refer to [1].

2.3 Software architecture

The **GG863-SR** software architecture follows a modular stacked approach, consisting in a full featured Linux Operating System, which allows complete and highly flexible deployment of add-ons and/or specific custom applications.
The Telit Linux Software development kit has been used to build the Gateway application, and it is available on the Download Zone, under Software >> Cellular >> Software Tools GSM/GPRS section of the Telit web site www.telit.com.

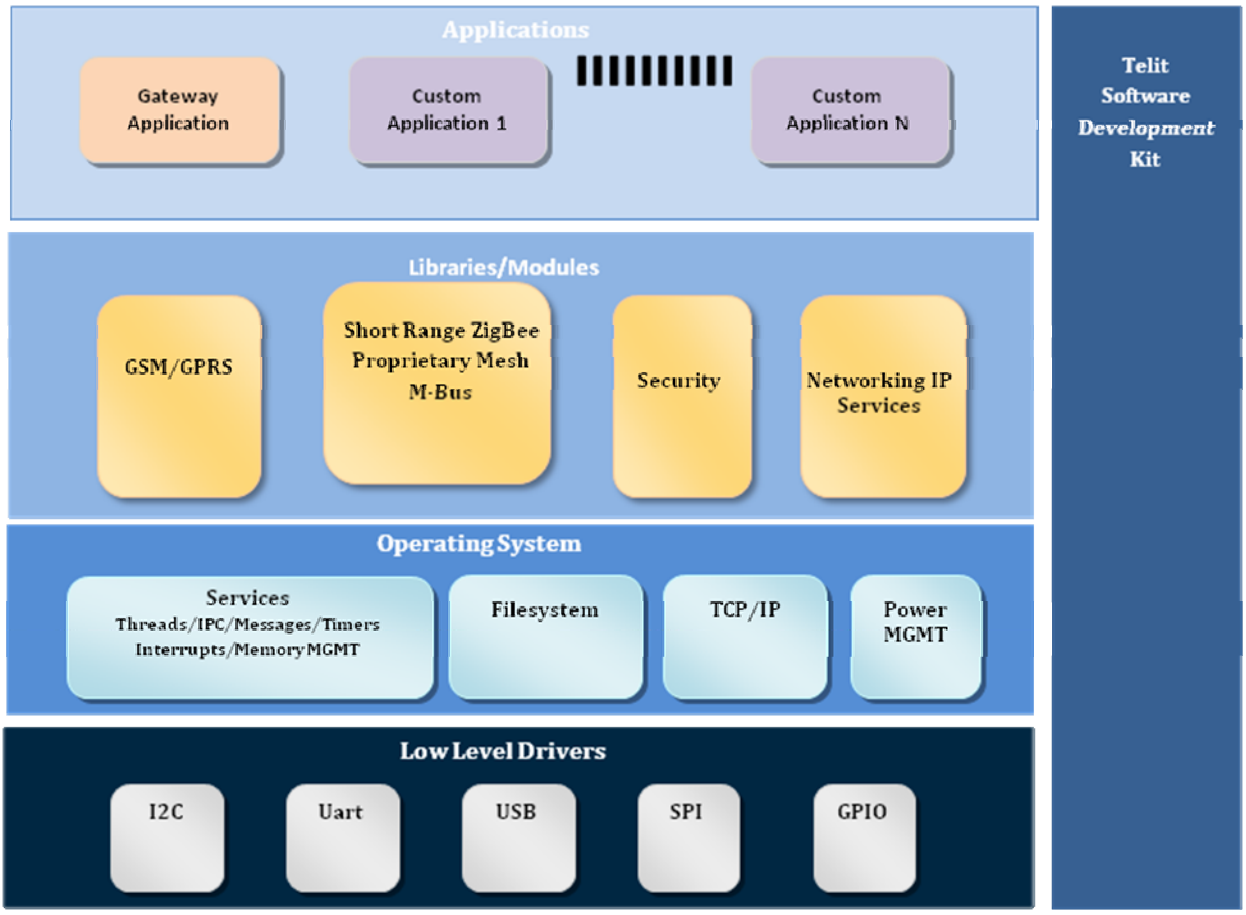


Figure 2.2 - Software Architecture

2.4 Getting started

This section serves as an introduction for **GG863-SR** first-time users. The following instructions give guidance on how to configure the gateway application and local PC in order to provide a hands-on approach to Short range Network gateway management and configuration.

Note: Users are strongly advised to carefully read the entire manual for full understanding of Short Range Network management and gateway application by means of GG863-SR.

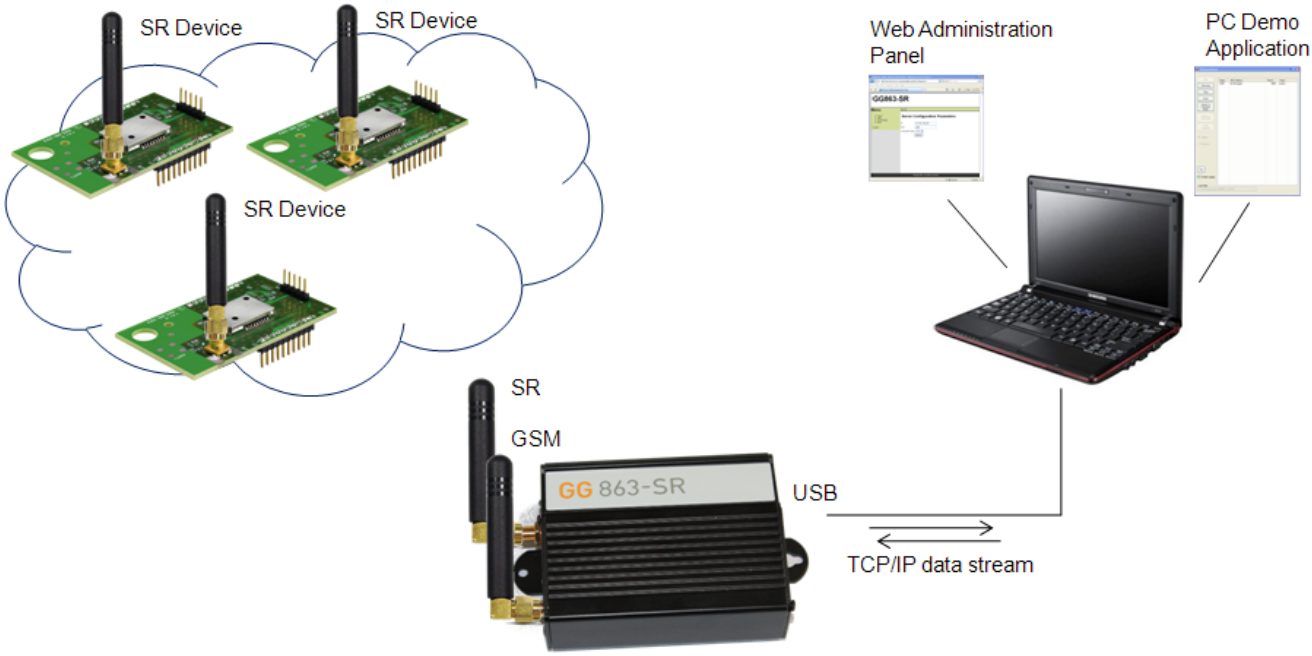


Figure 2.3 - Getting started scenario

2.4.1 Before you start

The minimum equipment to use the gateway functionalities of the **GG863-SR** is:

- The **GG863-SR** gateway
- One or more SR modules (of the same type of that mounted on the **GG863-SR**) programmed to act as Router or End Device with one of the following firmware, depending on device type:

  ZigBee
  - *xx.W31.11.32-B008-Router.s28*
  - *xx.W32.11.32-B008-EndDevice.s28*

  MeshLite
  - *DV.M0E.03.10-RC2.s19*
  - *DV.M0R.03.10-RC2.s19*
  - *DY.M0E.03.10-RC2.s19*
  - *DY.M0R.03.10-RC2.s19*

  In order to get details about flashing and configuring ZigBee modules the user shall refer to [9] and [10], for MeshLite modules refer to [11] and [13]

Warning – Please note that the IP Gateway application behavior is not warranted if different short range firmware versions are used

- The *SRGatewayDemo.exe*
- The Linux USB Ethernet/RNDIS Gadget
- A PC, which will host the demo application and which will be used to configure the Gateway

All the required software is available in the Telit website www.telit.com section *Download Zone > Software >Short Range >Software Tools Short range*

2.4.2 Connections setup

Before turning on the **GG863-SR** be sure to follow the connections detailed below:

- connect the antennas to RF connectors of the **GG863-SR**
- insert the SIM card in the card holder of the **GG863-SR**
- plug the Mini USB cable into the **GG863-SR** USB Device port (B-type)
- connect the power supply adapter to main power socket

Now the device is ready to be turned on.

2.4.3 Turning On GG863-SR

To turn on the **GG863-SR**, simply attach the power adapter cable to the **GG863-SR** and wait 20 seconds for full system startup.

Note – When the power supply cable is disconnected, it is recommended to wait for approximately 5 seconds before applying the power again.

2.4.4 Turning Off GG863-SR

To turn off the device you can simply unplug the power supply cable.

Warning – Please note that hardware power off should be done only after a proper GSM logoff. Any GSM device is requested to issue a "detach" request at turning off.

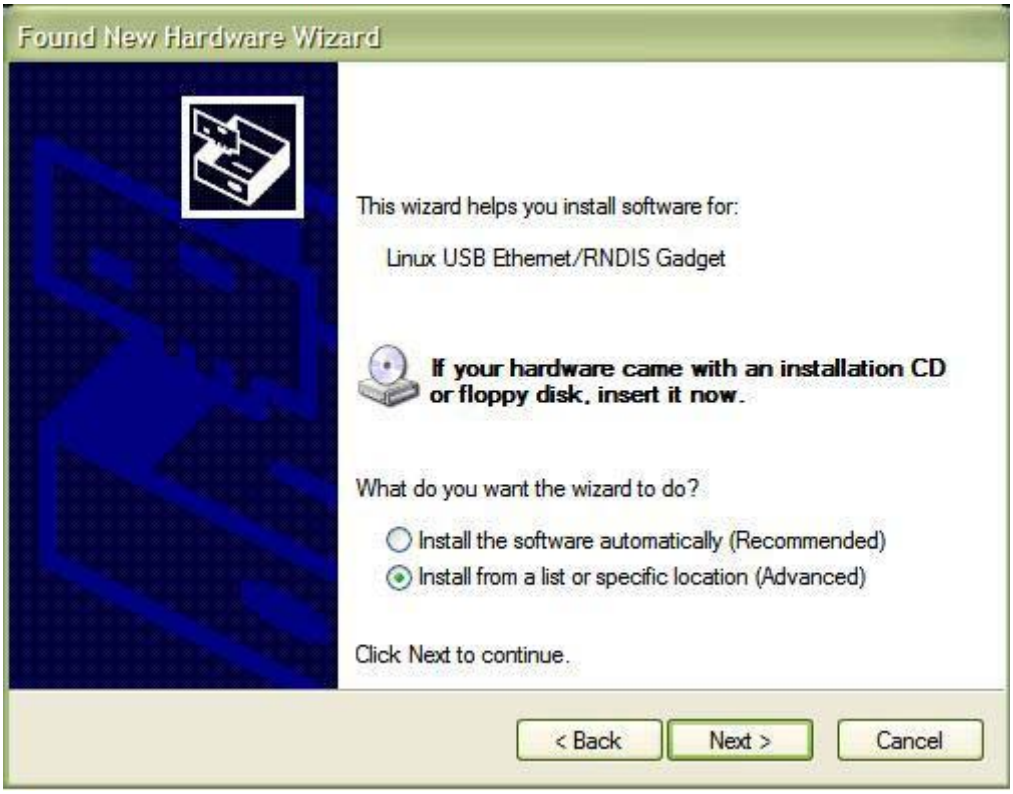
Normal shutdown sequence will take up to 8 seconds. An additional delay of up to 10s is experienced as the GSM module logs off the network.

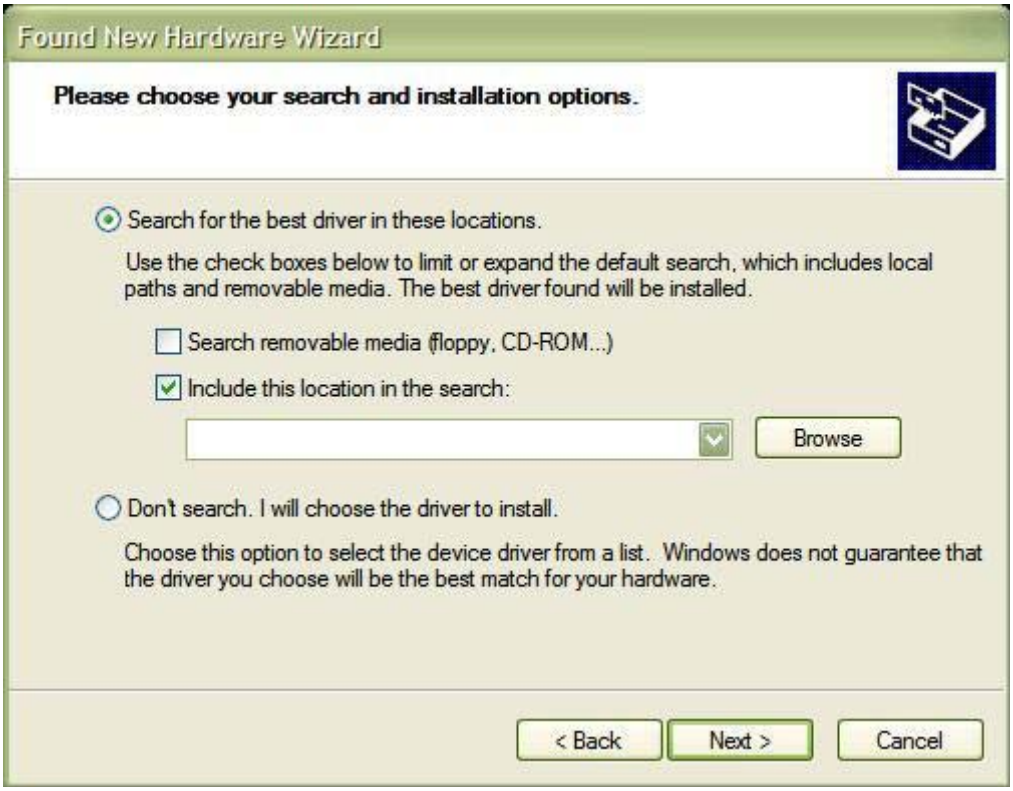
2.4.5 Connecting GG863-SR to the local PC

To connect the **GG863-SR** to the local PC follow this steps:

- Turn ON the **GG863-SR** (see section 2.4.3)
- Wait 20 seconds for the system to startup
- Plug the USB cable into the PC (A-type):
  The first time you connect the **GG863-SR**, you will be asked for a driver "Linux USB Ethernet/RNDIS Gadget". Follow the actions described by the screenshots:

Warning – If you have already installed the Telit Linux Development Environment, you already have the Ethernet Gadget installed and you are connecting the GG863-SR to the same USB port used for EVK-PRO[3], then the USB connection is automatically established. In this case you can continue reading the present document from 2.5.
If you are connecting the GG863-SR for the first time or to a different USB port please follow next steps.


Found New Hardware Wizard
This wizard helps you install software for:
Linux USB Ethernet/RNDIS Gadget
If your hardware came with an installation CD or floppy disk, insert it now.
What do you want the wizard to do?
Install the software automatically (Recommended)
Install from a list or specific location (Advanced)
Click Next to continue.
< Back
Next >
Cancel


Found New Hardware Wizard
Please choose your search and installation options.
Search for the best driver in these locations.
Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.
Search removable media (floppy, CD-ROM...)
Include this location in the search:
Browse
Don't search. I will choose the driver to install.
Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.
< Back
Next >
Cancel

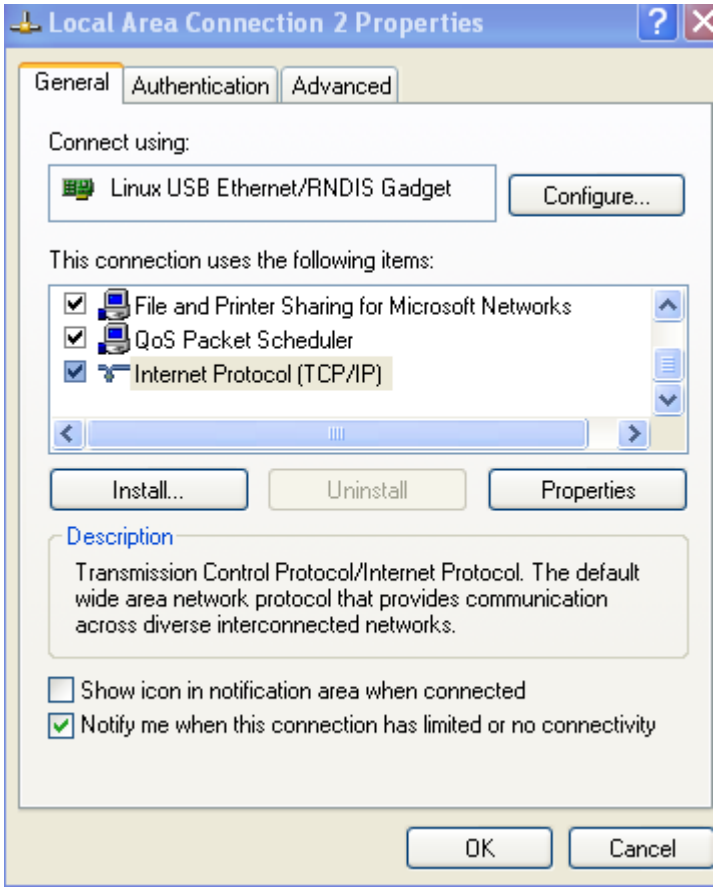
Click the **Browse** button and search for the file **linux.inf**, included in the Telit package.

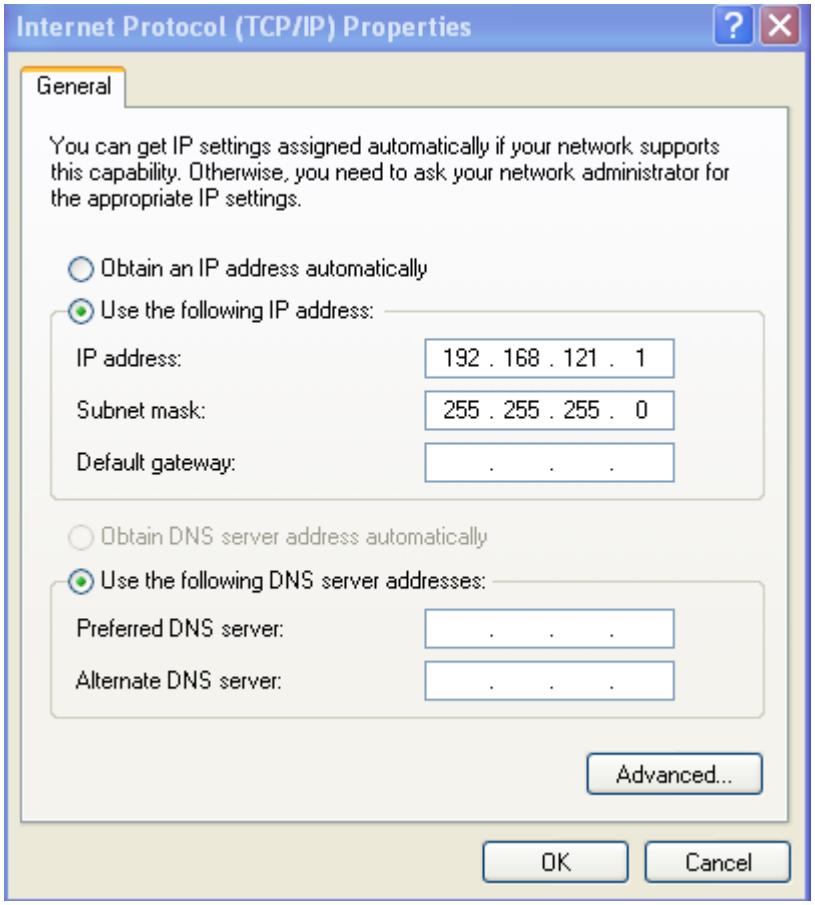
Click **Next**.

Click **Continue Anyway**.

Click **Finish**.

Go to **Network Connections**, right click on the device called **Linux USB Ethernet/RNDIS Gadget** and choose the menu voice **Properties**. Select the **Internet Protocol (TCP/IP)** and click on the button **Properties**; you should see the following windows:


Local Area Connection 2 Properties
General
Authentication
Advanced
Connect using:
Linux USB Ethernet/RNDIS Gadget
Configure...
This connection uses the following items:
File and Printer Sharing for Microsoft Networks
QoS Packet Scheduler
Internet Protocol (TCP/IP)
Install...
Uninstall
Properties
Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.
Show icon in notification area when connected
Notify me when this connection has limited or no connectivity
OK
Cancel



If this configuration is not available, you have to manually insert the following values:

IP address: 192.168.121.1
Subnet mask: 255.255.255.0

then click **Ok**.

At this point, Ethernet Gadget is installed on Local PC: Local PC and **GG863-SR** can communicate via IP with the following addresses:

- 192.168.121.1 (Local PC)
- 192.168.121.3 (**GG863-SR**)

2.4.6 Reboot GG863-SR

Before rebooting the **GG863-SR** you have to un-plug the USB cable from the PC if it is connected, reboot the **GG863-SR**, wait 20 seconds and plug the USB cable into the PC to make the connection work.

2.5 Demo Gateway Application - Local Configuration

The Demo Gateway Application is an example to show how SR devices communicate with an IP host and vice versa via the **GG863-SR** Gateway.

In “Local Configuration”, the PC used to configure the Gateway is the same hosting the PC Demo Application, and it is connected to the **GG863-SR** through the USB cable.
Please note that the Demo Gateway Application can run also in “Remote Configuration”.
Using the “Remote Configuration” the PC hosting the Demo Application is connected to the GG863 through the GPRS connection. In this case the PC used to configure the Gateway may or may not be the same used to host the PC Demo Application (Refer to §. 7 for a detailed description of the Demo Gateway Application in “Remote Configuration”).

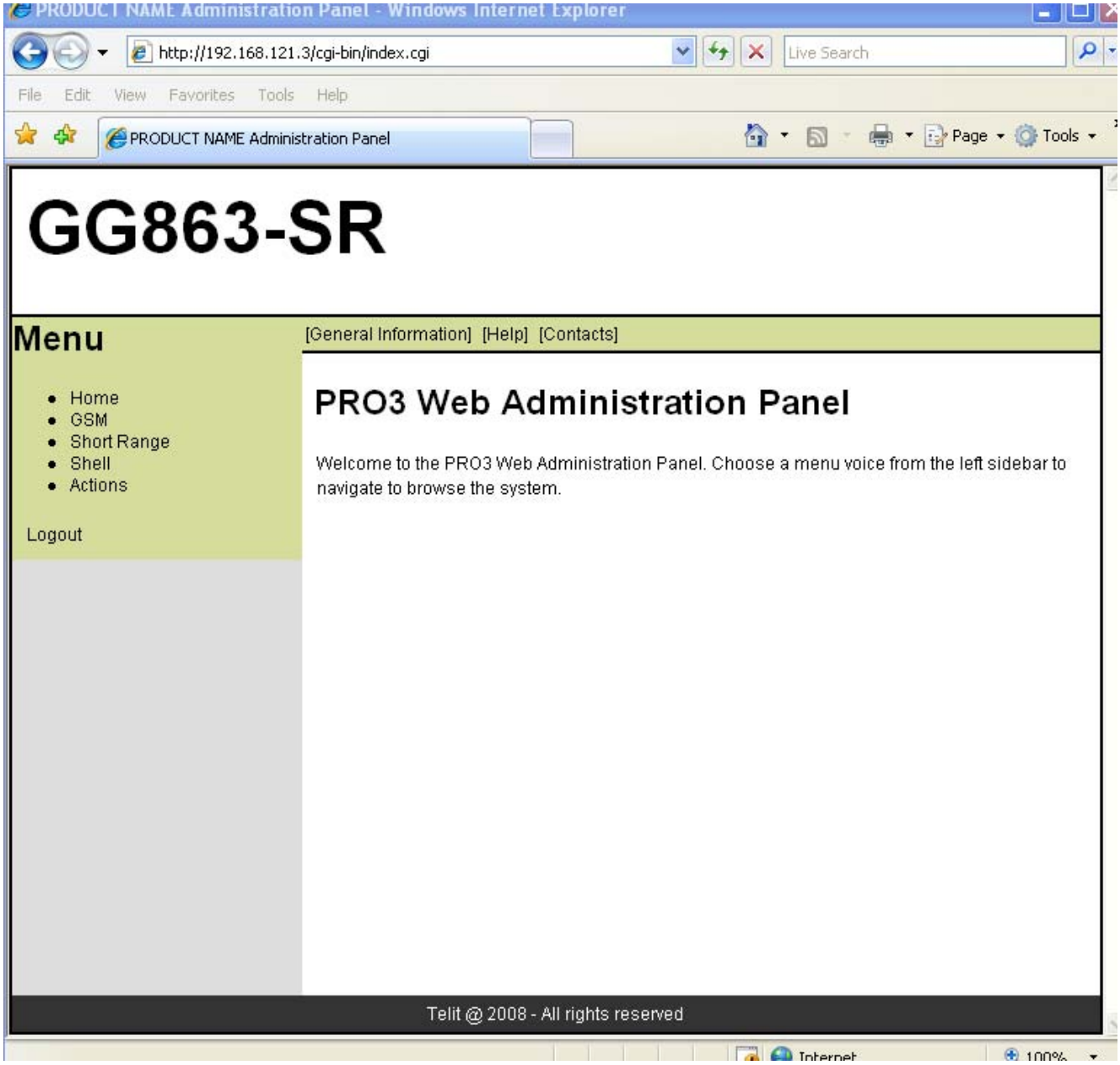
2.5.1 Start the IP GW application

- Turn on the **GG863-SR** and wait 20 seconds for the system to boot
- Connect the **GG863-SR** to the local PC
- In the local PC connected to the **GG863-SR** type the address:

  `http://192.168.121.3/cgi-bin/index.cgi`

where `192.168.121.3` is one of the IP addresses of the **GG863-SR.**

You can see the Home Page of the Web Administration Panel:



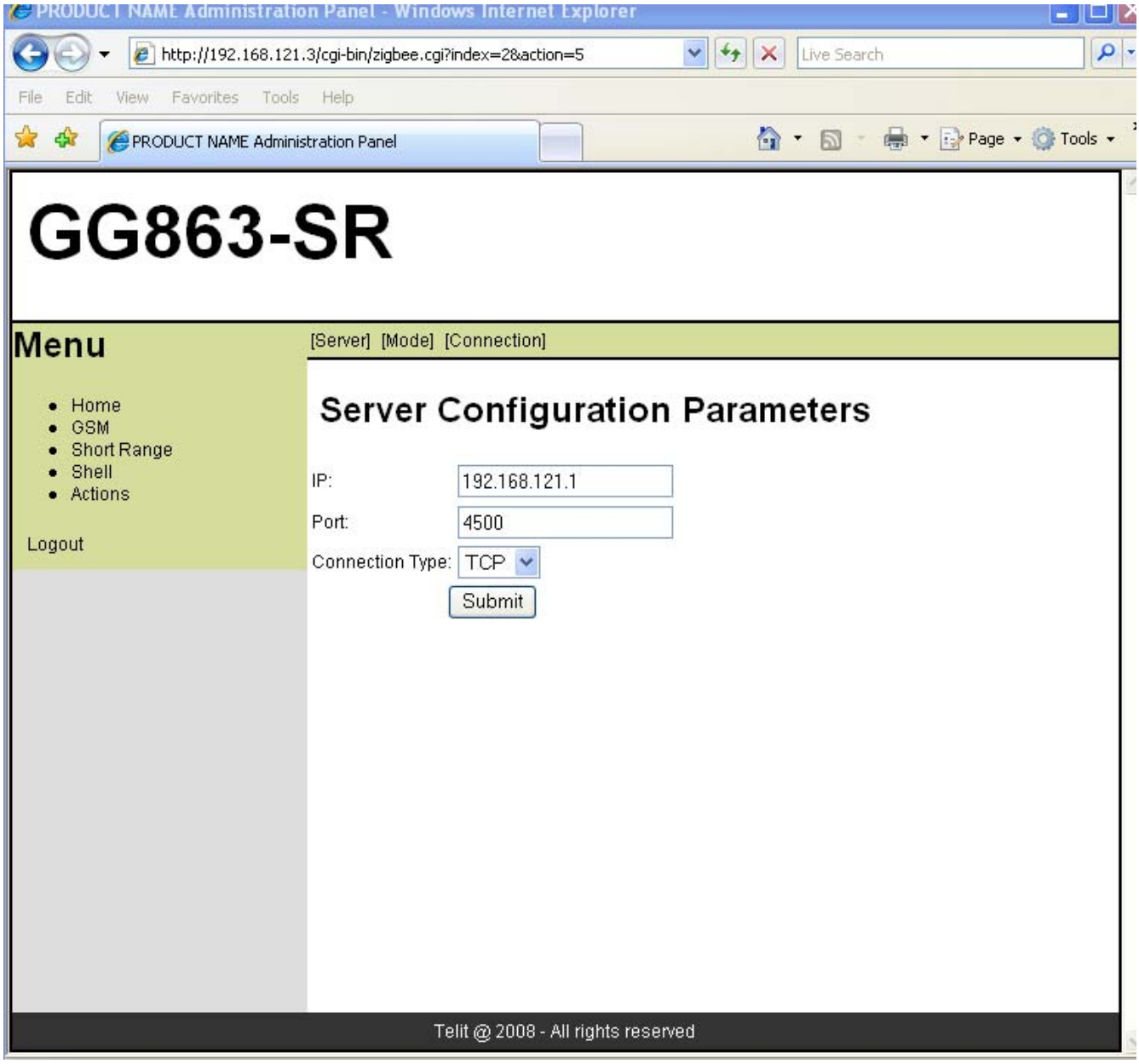
1. Set the IP Host parameters:

   1. Click on *Short Range*
   2. Click on [Server]

   Fill the form in the following way:

   - IP Host address: **192.168.121.1**
   - Port: **4500**

- Connection Type: **TCP**

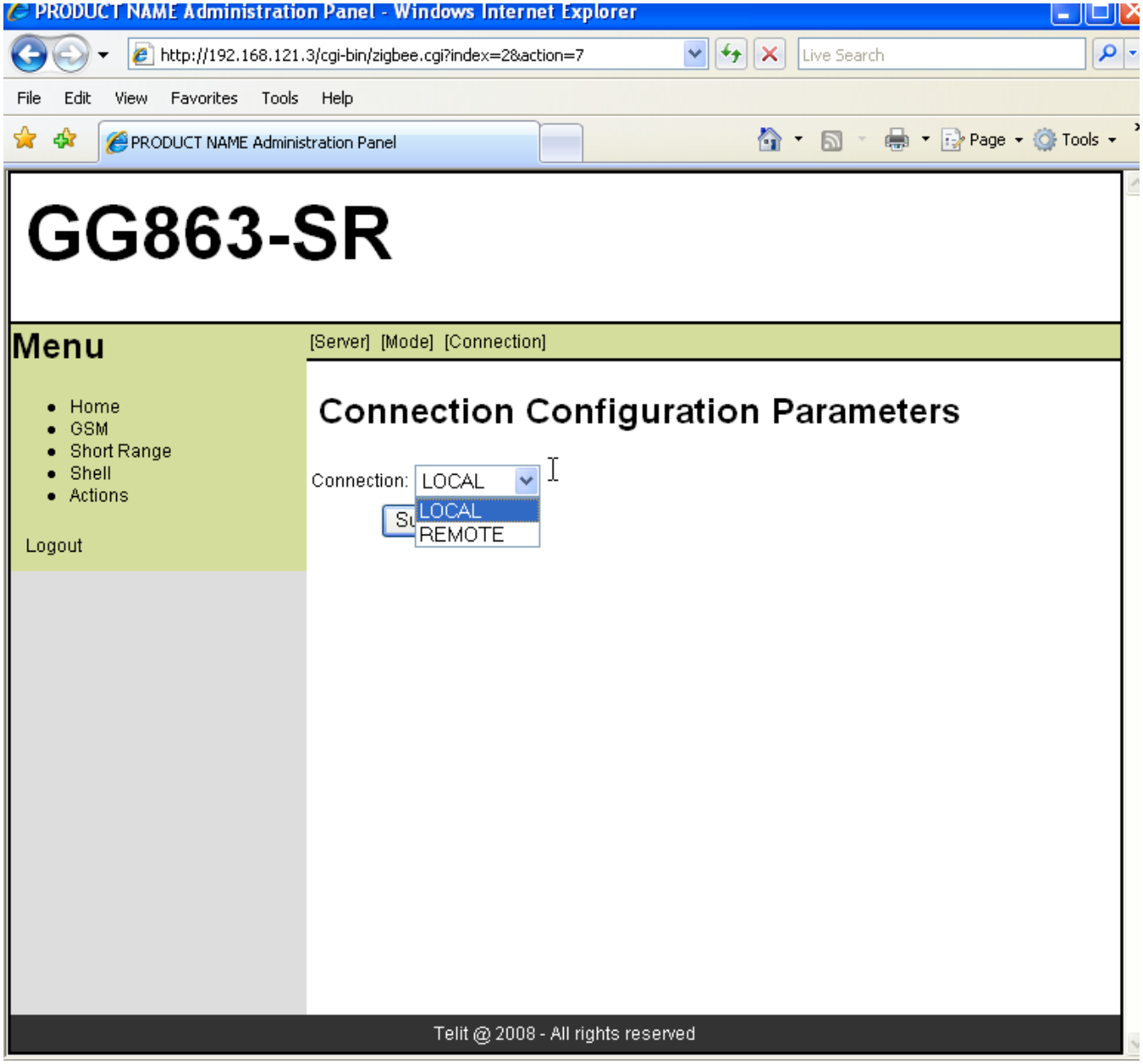


Click **Submit,** wait until the Web Panel shows the message:

Parameter configuration was successful

2. Click on [Connection]

– Connection: **LOCAL**

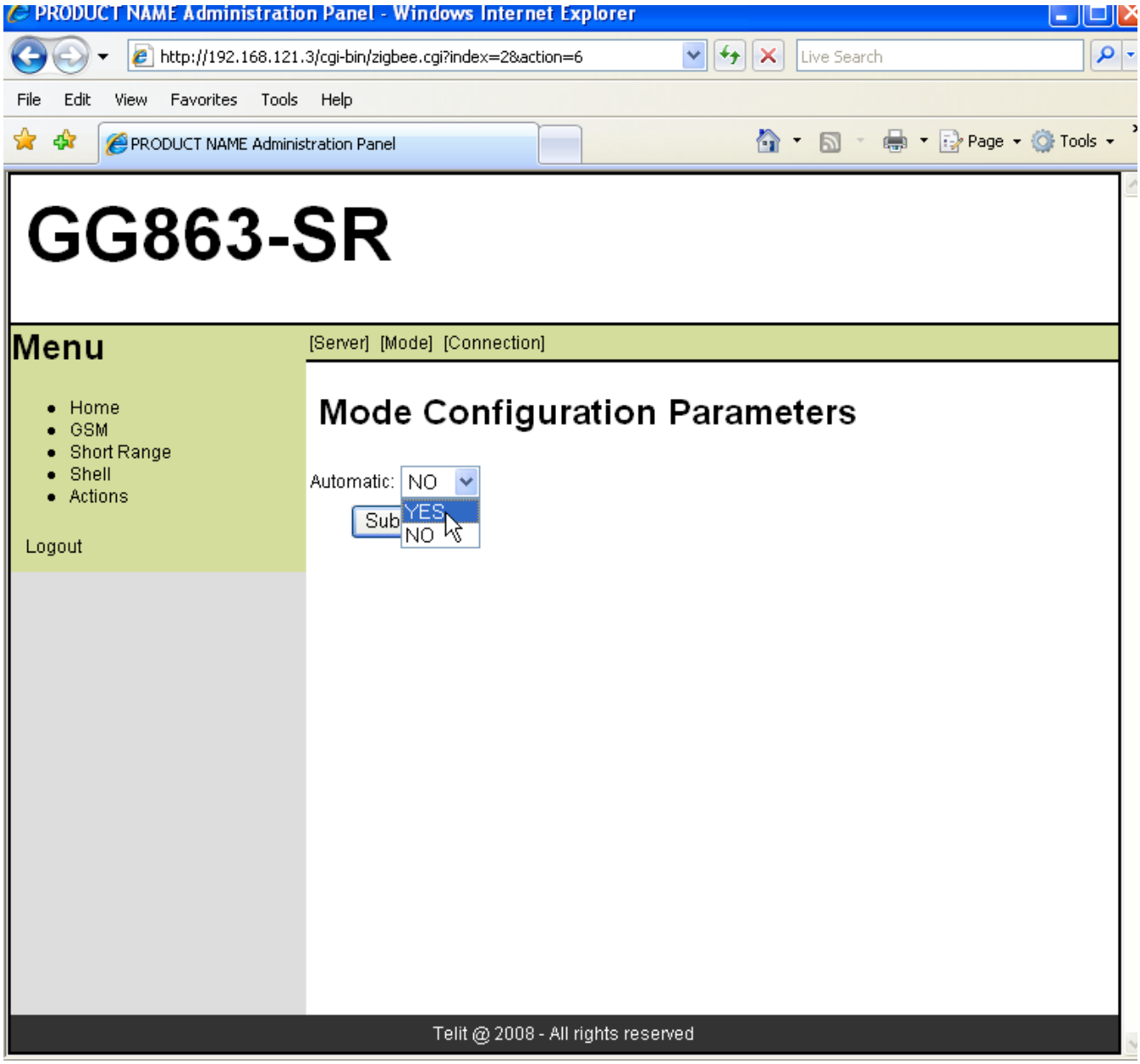


Click **Submit,** wait until the Web Panel shows the message:

Parameter configuration was successful

3. Click on [Mode]

– Automatic: **YES**



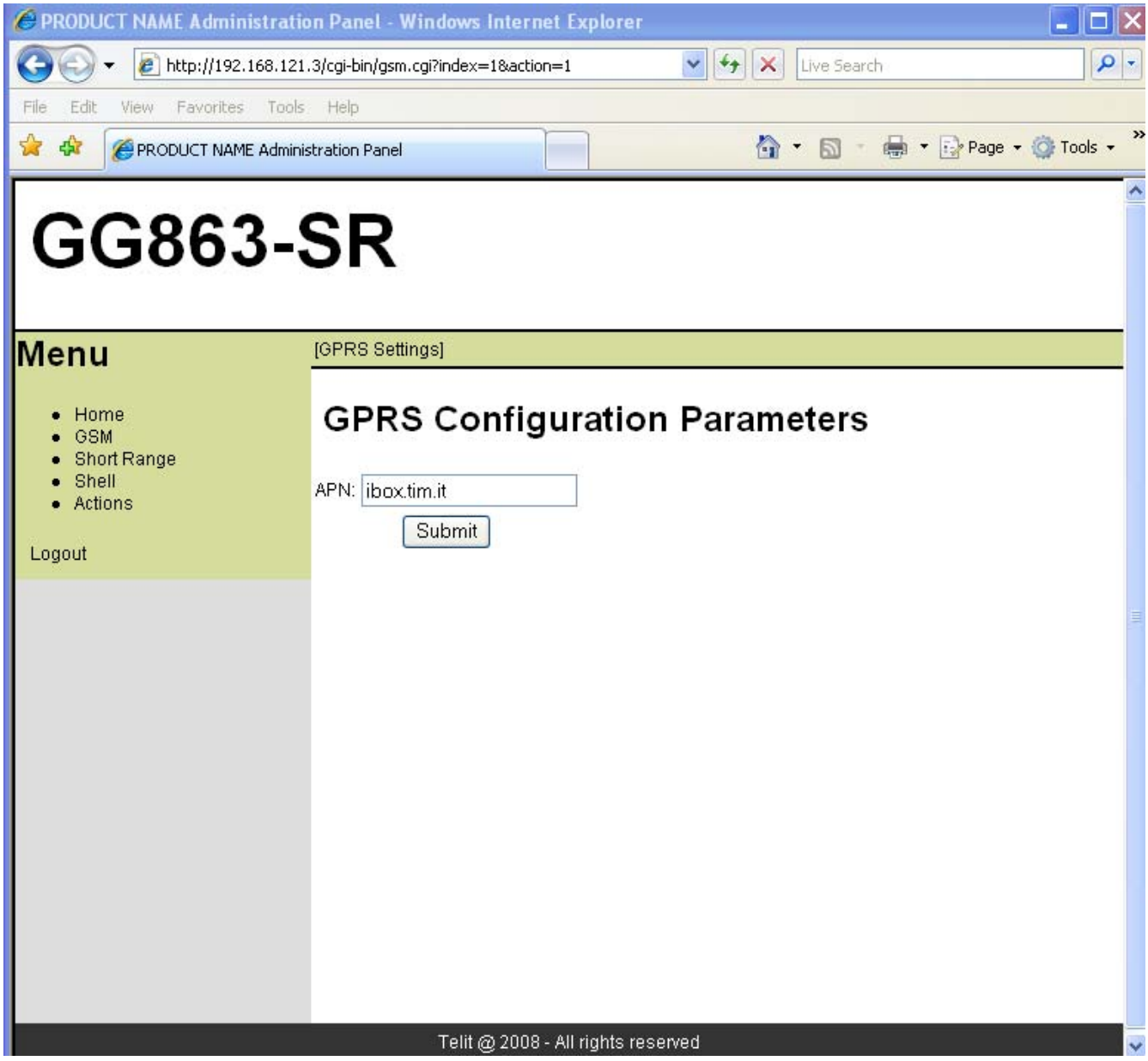
Click **Submit,** wait until the Web Panel shows the message:

Parameter configuration was successful

4. Click on GSM:

– Click on [GPRS Settings]

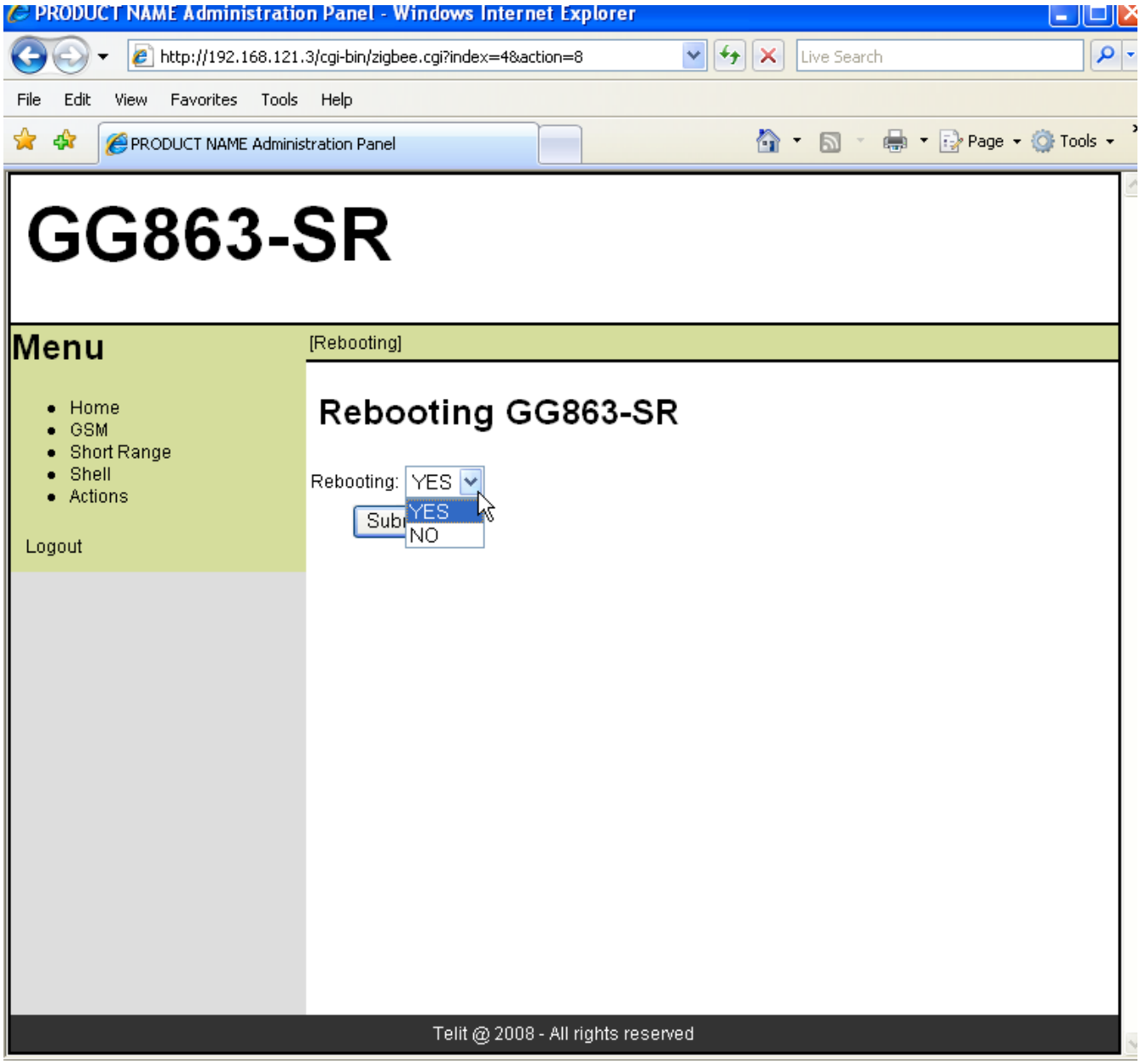
Fill the form with APN:



Click **Submit,** wait until the Web Panel shows the message:

Parameter configuration was successful

5. Click on Actions:
   - Click on [Rebooting]
   - Rebooting: **YES**



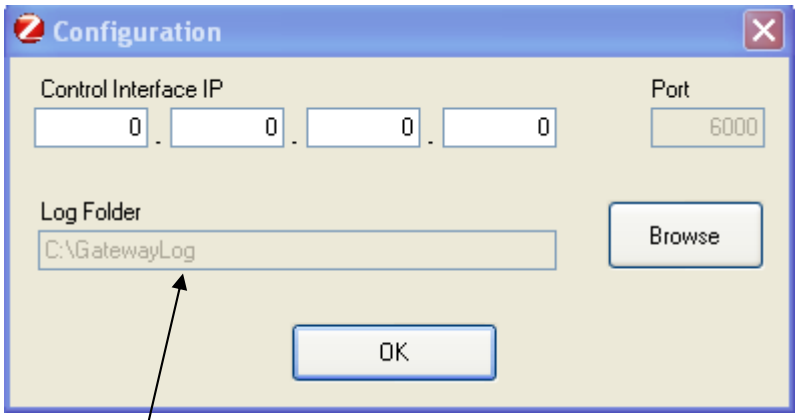
Click **Submit,** wait until the Web Panel shows the message:

Parameter configuration was successful

- Unplug the USB cable from the PC
- Wait 30 seconds
- plug the USB cable into the PC

2.5.2 Run the PC Demo application

After the previous USB plugging run the SRGatewayDemo.exe on the PC:

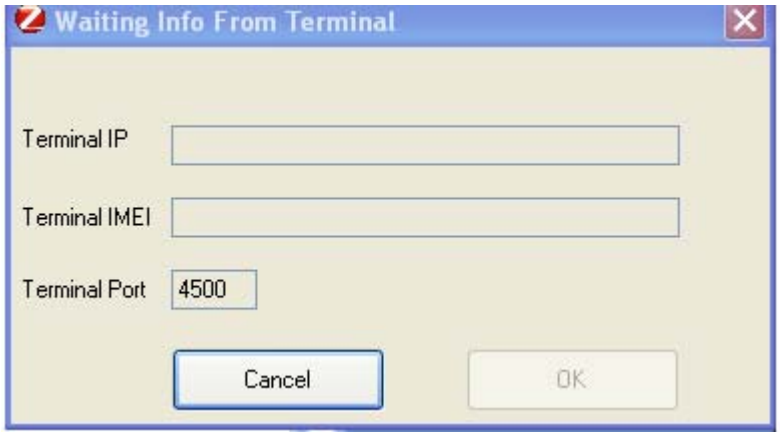


Choose the folder where to store the logs

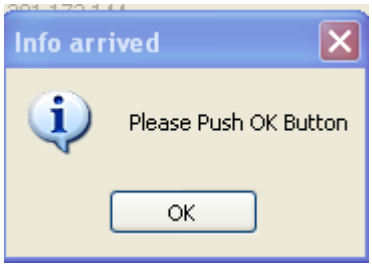
Not Delete the log folder while the SRGatewayDemo.exe is running

Click **OK**

- Clicking **Ok,** the IP Host waits for **GG863-SR** IP address and IMEI



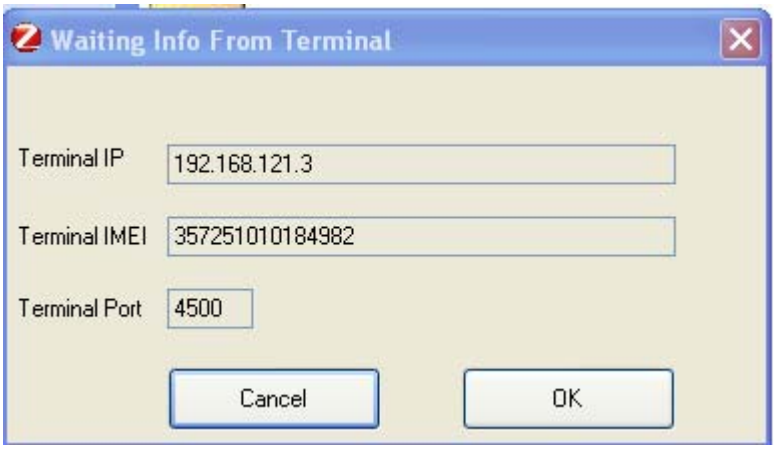
- When appears the following pop-up it means the connection between **GG863-SR** and SRGatewayDemo is established.



Click **Ok**

- Now you can see the information received from **GG863-SR**:

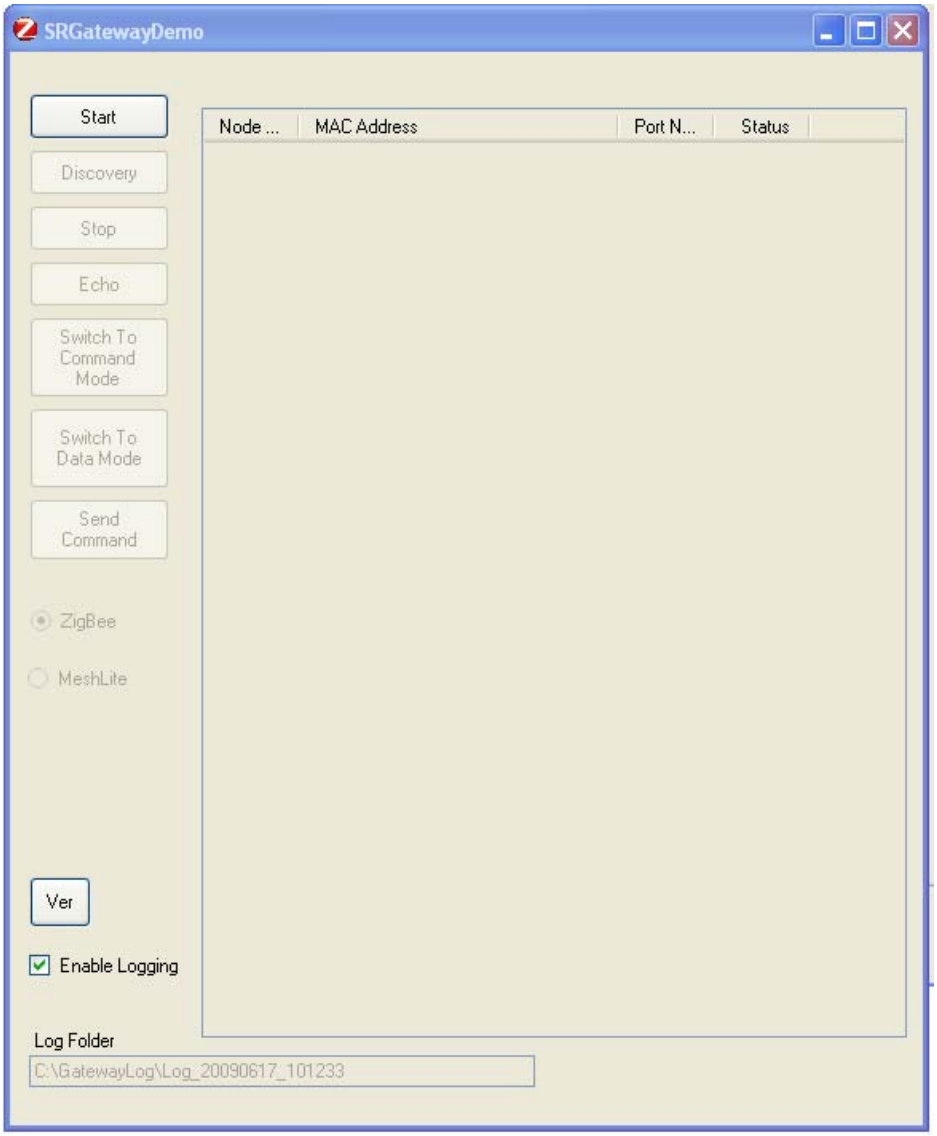
  For example:



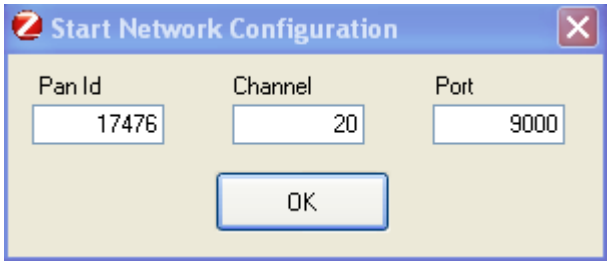
At this point the PC Demo is communicating with the IP Gateway Application running inside the **GG863-SR**.
Click **OK**.

- Start the SR Network (the technology radio button checked shows the Short Range Technology integrated in the **GG863-SR**):

  Click **Start**

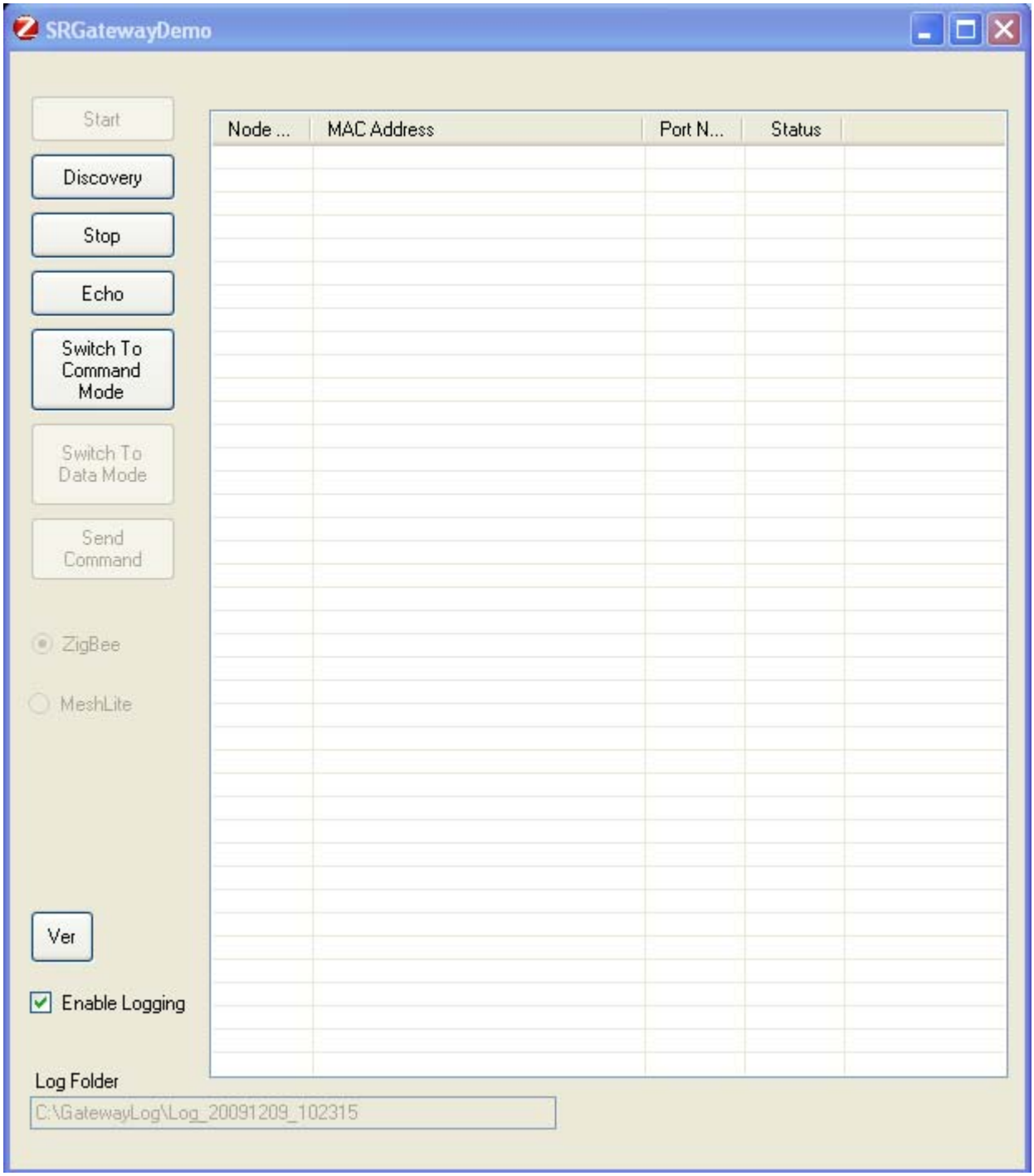


- The next window prompts for Pan Id, Channel and Port configuration used to Start the SR Network:
  - Pan ID → Short Range Network Identifier;
    ZigBee: any value from 0 to 65535
    MeshLite: any value from 0 to 255
  - Channel → RF channel for the SR Network;
    ZigBee: any value from 11 to 26
    MeshLite: the Channel field has not meaning and for this reason it is disabled
  - Port → the first TCP port used in *TCPport–nodeID* mapping
    ZigBee: any value from 6001 to 65535
    MeshLite: any value from 6001 to 65535

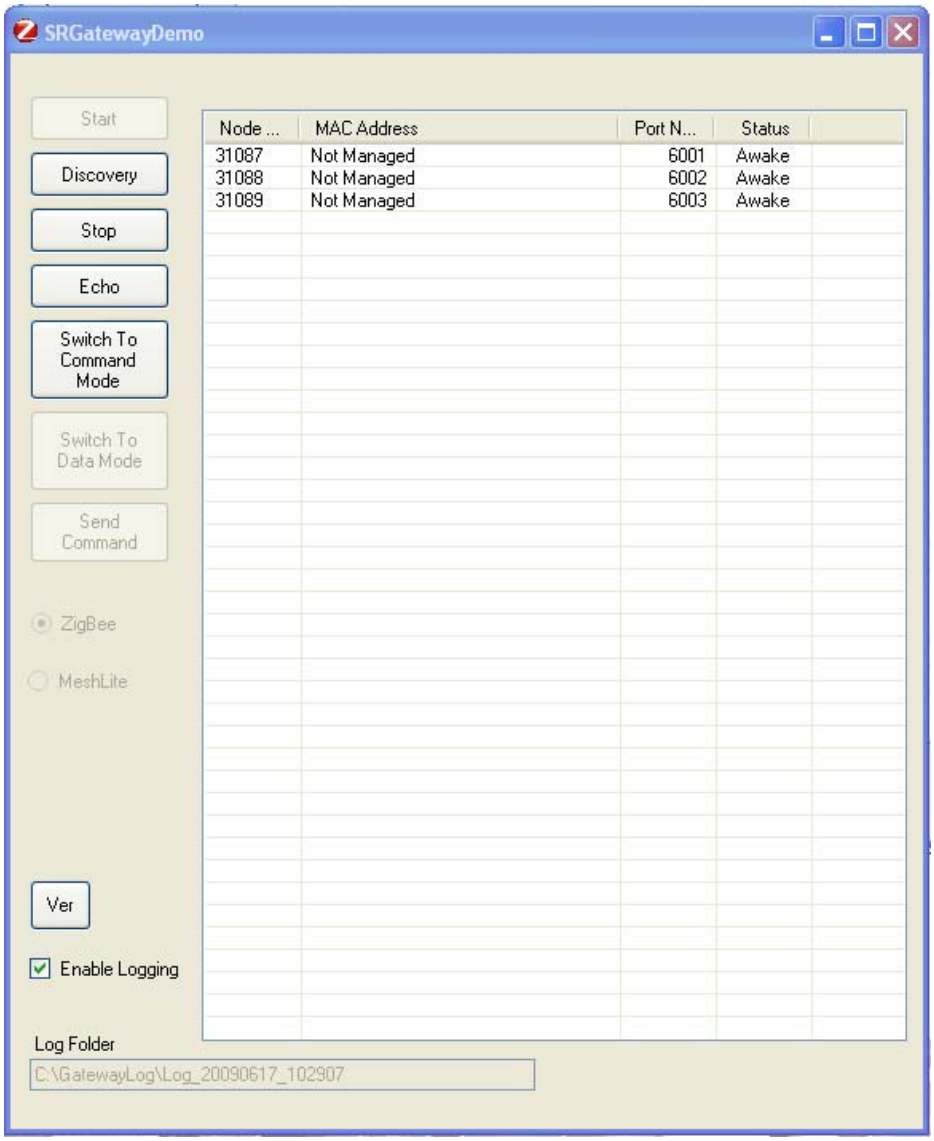


Click **Ok**

- Now you can discover the nodes associated to the network either before or after starting the SRGatewayDemo.exe application



- To see the nodes associated to the network (ZigBee or MeshLite) before running the SRGatewayDemo.exe application click the button **Discovery**:



- To see the nodes associated to the ZigBee Network after running the SRGatewayDemo.exe application follow this procedure:
  - ❖ Associate the SR modules to the SR Network by configuring the same channel and PAN ID previously used to start the network (see 8.2)
  - ❖ For any device associated to the network you will see a new Node in the Demo

- To see the nodes associated to the MeshLite Network after running the application SRGatewayDemo.exe follow this procedure:
  - ❖ Associate the SR modules to the SR Network by configuring the same PAN ID previously used to start the network (see 8.2 for)
  - ❖ Click the button **Discovery** and you will see the new Nodes in the Demo

For further information about the features of the PC Demo Application please refer to § 7

3 Software applications

Below it is depicted the **GG863-SR** software architecture at application level.

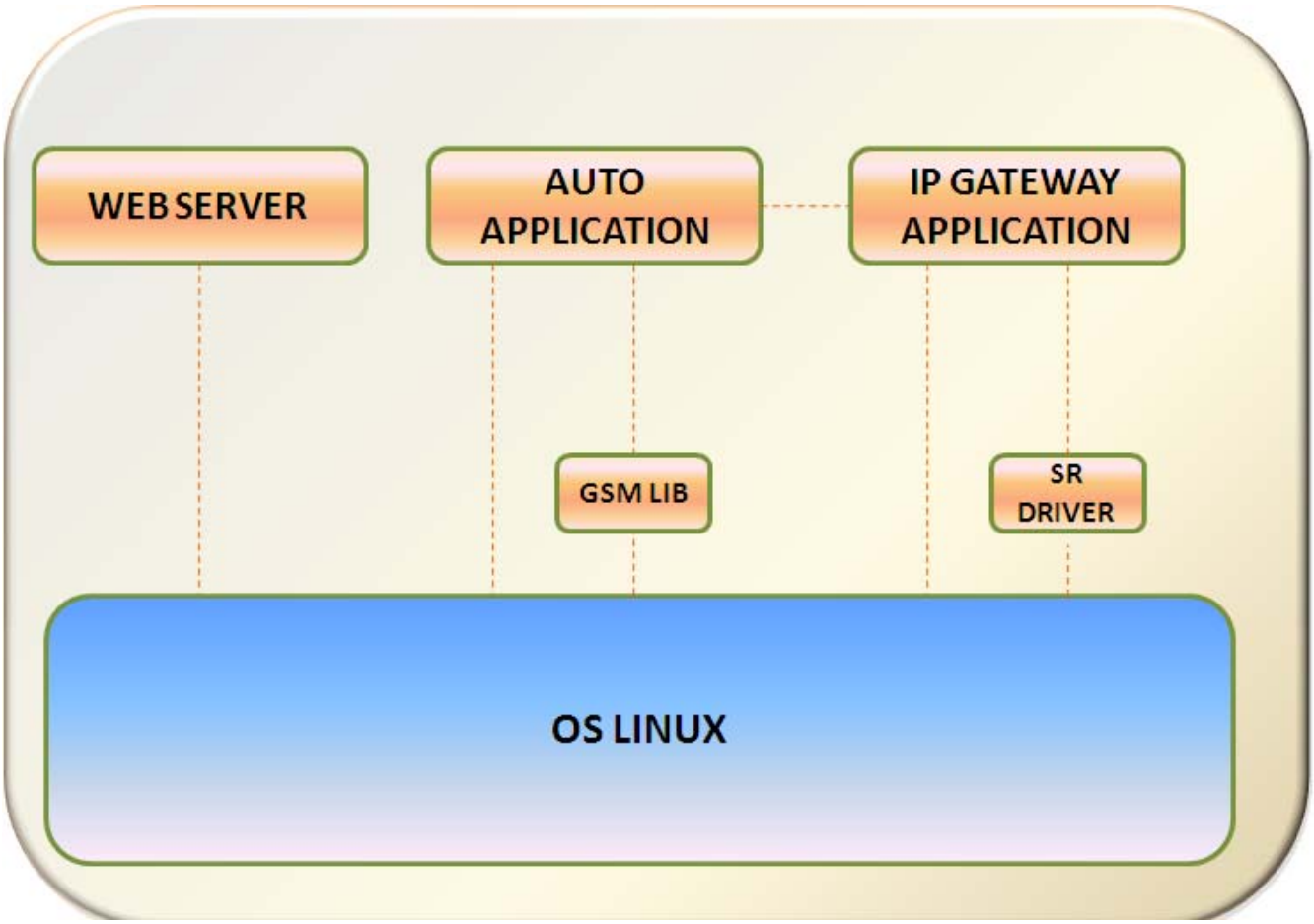


Figure 3.1 - GG863-SR software components

The following subsections contain a description of the available software components and their role in **GG863-SR** Gateway applications.

3.1.1 Operating System

The real-time OS LINUX gives developers access to an extensive library of drivers for different peripherals and to a complete development environment.
For a complete description of the **GE863-PRO^3** and its Linux OS refer to the **GE863-PRO^3** Linux Software User Guide [3].

3.1.2 Short Range Driver

The Short Range Driver provides a software interface between the IP Gateway application and the Short Range network coordinator. The SR Driver presents a set of API that allows the Application Level to control and configure the SR Coordinator.

3.1.3 IP Gateway Application

The IP Gateway Application is the core of the **GG863-SR**. It provides a link between the SR network and the TCP/IP domain allowing a remote IP Host to reach a node of the SR network and vice-versa.

3.1.4 Web Administration Panel

The Web Administration Panel is based upon the Web Server running on the **GG863-SR**. It allows setting the configuration parameters used by the Auto Application.
The Web Administration Panel also provides the possibility to communicate with the **GG863-SR** emulating the Linux shell.

3.1.5 Auto Application

The Auto Application manages the set up and configuration of the GSM/GPRS network and establishes a connection to a remote IP Host in order to provide the gateway functionalities.

3.1.6 GSM Library

The GSM Library is a set of APIs that grants access to all the GSM/GPRS functionalities provided by the **GG863-SR**. For more information on the GSM Library please refer to [7].

4 IP Gateway Application

4.1 Overview

The IP Gateway Application creates a direct connection between a remote IP Host and a node of the SR network. The IP Host views the connection with each Short Range node (SR node) as a pipe for exchanging messages between remote IP Host and SR network.

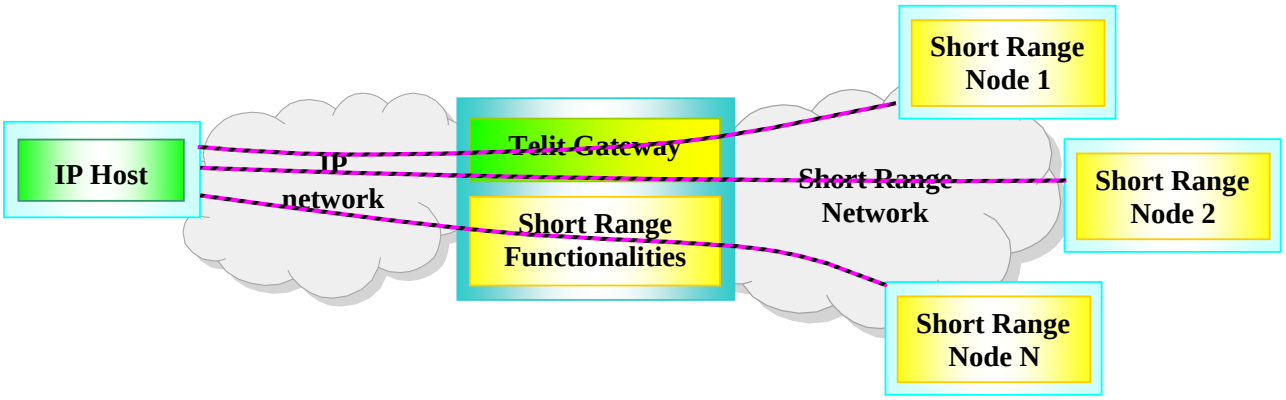


Figure 4.1 – IP gateway application pipelining

The IP Host interacts with the IP GW Application through an IP interface, available via a standard IP address.

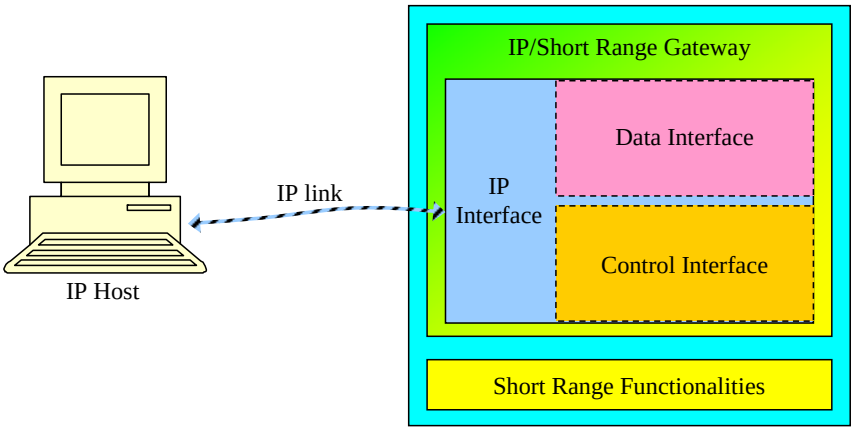


Figure 4.2

This IP interface is internally split into two interfaces, both available through specific TCP ports:

1. Control (or management) Interface, available at TCP port 6000. This interface receives commands from IP Host and sends back information/responses.
2. Data Interface, covered by a variable range of ports that depends on the number of SR nodes joined to the SR network. This interface assigns to each node an exclusive TCP port (port_1, port_2, port_n in Figure 4.3). The IP Host is notified of the pair *TCPport–nodeID* via messages issued by Control Interface.

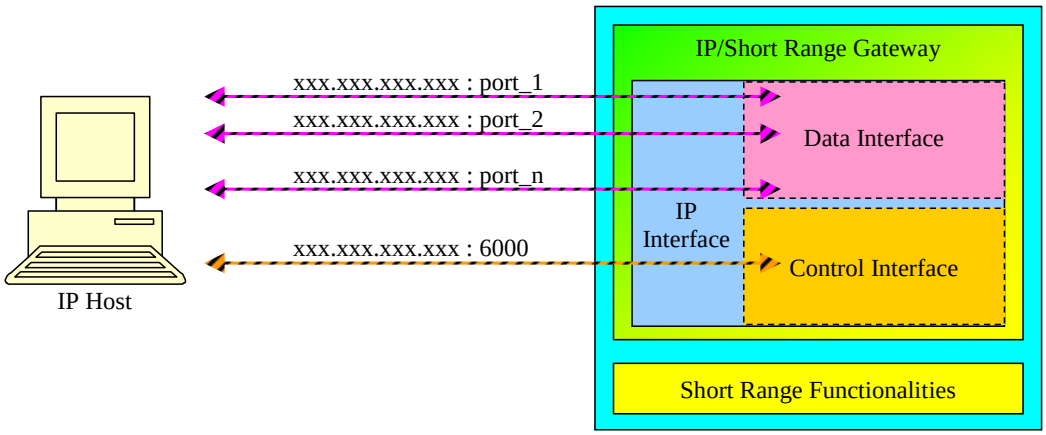


Figure 4.3 – Data and control channels/interfaces

4.2 Control Interface

Once IP GW Application has been started, the Control Interface will be available at TCP port 6000.
The Control Interface receives from the IP Host the TCP packets, converts them to internal control messages and sends the responses to the IP Host.
All frame formats in this clause are routed in the order in which they are transmitted, that is, from left to right. For every field the left most byte is the least significant byte and the right most byte is the most significant byte.
The TCP packets shall be formatted as illustrated in Figure 4.4.

1 Byte	Variable	1 Byte
START	Message	STOP

Figure 4.4

Every packet has a START byte and a STOP byte.

4.2.1 Stuff algorithm

When using START and STOP bytes in the Message, it must be ensured that they will not be confused with the true START and STOP. To achieve this result, an escaping technique called “stuff algorithm” has to be used. Every time the Message field contains a byte equal to the START or the STOP, the

same byte must be preceded by a STUFF byte and incremented of an OFFSET. Every time that in the Message field there is a byte equal to the STUFF byte it must be preceded by another STUFF byte. Values of START, STOP, STUFF and OFFSET bytes are shown in Table 4.1.

Name	Value
START	0xab
STOP	0xcd
OFFSET	0x01
STUFF	0xef

Table 4.1

4.2.2 Message protocol

The message protocol is used by the IP Host to configure and to manage the Gateway. It is an acknowledged protocol, so that every time the IP Host sends a command to the Control Interface of the Gateway, it will acknowledge the command with an ACK (Figure 4.5).
Therefore there are two types of messages:

- Command/Response messages
- ACK messages

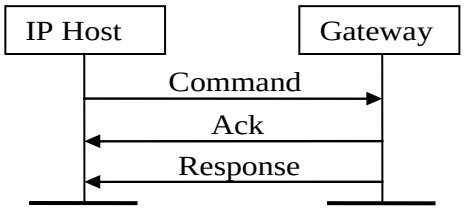


Figure 4.5

If the Command do not respect the protocol or if its parameters are out of range, the Control Interface of the Gateway will send only a NACK (acknowledge with status set to ERROR) and no response.

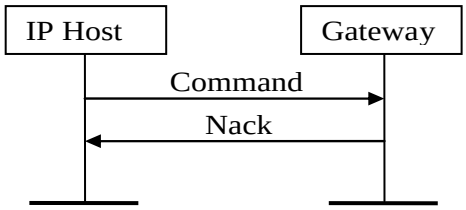


Figure 4.6

4.2.3 Command and Response messages

Command and Response messages format is shown in the Figure 4.7.

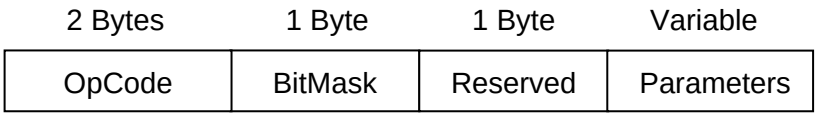


Figure 4.7

The OpCode is the code of the Command sent by the IP Host or the Response sent by the Gateway.
The field BitMask explains which parameters are active in the Parameters field. Note that all the parameters are always present in the packet, so a specific Command/Response has always the same size. If a parameter is not active or an invalid BitMask field has been set, the respective default value will be used.
The field Reserved is ignored and will be used in the future for new features. In the Command messages the field Reserved can be set to every value between 0x00 and 0xff (for consistency, in next sub paragraphs, this field will be set to 0xff).
The field Parameters holds the parameters of the command. It has a specific format for every command depending on the OpCode. Every parameter could be optional (O) or mandatory (M), if it is optional and the parameters is not enabled the Gateway will use a default value.
Every response has a Status field.
Table 4.2 shows the available commands and responses.

OP Code	Command/Response Name	Description
0x0001	Start Network Command	Start the SR Network and set the first TCP port used in *TCPport–nodeID* mapping
0x0004	Close Pipe Command	Close a data pipe
0x0005	Reset Command	Reset the gateway system
0x0008	Set Mode Command	Set the running mode of the IP GW Application (NORMAL or COMMAND)
0x0009	Stop Network Command	Stop the SR Network and close all data pipes
0x000b	Global Version Command	Retrieve version of IP Gateway software
0x0041	Start Network Response	Response to the Start Network Command
0x0042	Scan Network Response	Provide Information about TCP port/SR node association
0x0044	Close Pipe Response	Response to the Close Pipe Command
0x0045	Reset Response	Response to the Reset Command
0x0048	Set Mode Response	Response to the Set Mode Command
0x0049	Stop Network Response	Response to the Stop Network Command
0x004b	Global Version Command	Response to the Global Version Command

Table 4.2

IMPORTANT - When switched in command-mode the protocol used to communicate with the control interface will be the same of the normal-mode, but Reset Command and Start Network Command will not be available. If those commands will be issued during command-mode, a response with ERROR will be returned.

Table 4.3 shows the values of the Status codes for the responses.

Status	Value	Description
SUCCESS	0x0001	The command succeeded
ERROR	0x0002	Generic error
CLOSE_PORT_ERR	0x0004	The port is already closed
NETWORK_ALREADY_RUNNING	0x0005	A SR Network is already running. Stop this network before to start a new one.
NETWORK_ALREADY_STOPPED	0x0006	No SR Network running. There are not SR Networks to stop.
END_OF_SCAN_NETWORK	0x0007	End of scan network procedure
NEW_DEVICE_ANNOUNCE	0x0008	A new device has joined the network.

Table 4.3

Table 4.4 shows the definition of BitMask.

Name	Value	Description
BIT_0	0b00000001	The first parameter is enabled
BIT_1	0b00000010	The second parameter is enabled
BIT_2	0b00000100	The third parameter is enabled
BIT_3	0b00001000	The fourth parameter is enabled
BIT_4	0b00010000	The fifth parameter is enabled
BIT_5	0b00100000	The sixth parameter is enabled
BIT_6	0b01000000	The seventh parameter is enabled
BIT_7	0b10000000	The eighth parameter is enabled

Table 4.4

4.2.3.1 Start Network Command

The *Start Network Command* starts the SR Network, and set the first TCP port used in *TCPport–nodeID* mapping. Figure 4.8 shows the Parameters field of this command.

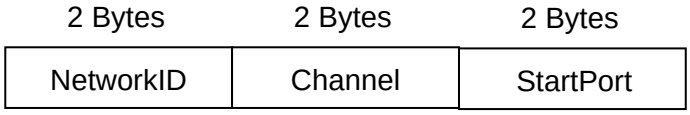


Figure 4.8

Table 4.5 shows the valid range for the values of the parameters.

Field	Type	Valid Range	Description	Default
NetworkID	Int	From 0x0000 to 0xffff (ZB) From 0x00 to 0xff (ML)	Network Identifier (NTW ID) of the SR Network. (E.g. For ZigBee technology it is the PanID value of the desired network)	0x1234 (ZB) 0x30 (ML)
Channel	Int	From 0x000B to 0x001A	RF channel for the SR Network (not	0x13 (ZB)

			used by MeshLite technology, don't care about its value)	
StartPort	Int	6001-65535	Number of the first TCP Port	6001

Table 4.5

Important - After the network is started, to add a node to the ZB network follow the steps described in §8.2, to add a node to the ML network follow the steps described in §8.3

4.2.3.2 Start network Response

On receipt of the *Start Network Command* the Gateway starts the SR Network using the ID of the network equal to NTW ID and the physical channel "Channel". The first TCP port assigned is StartPort, the smallest port available. Then it will send a *Start Network Response* to the IP Host. Figure 4.9 shows the Parameters field of *Start Network Response.*

2 Byte	2 Byte	2 Byte	2 Byte
Status	NetworkID	Channel	StartPort

Figure 4.9

The *Start Network Response* reports to the IP Host the result of the *Start Network Command* , Table 4.6 shows the valid range for the values of the parameters of the *Start Network Response*.
This command is not available when the IP Gateway is in "Command-mode", if *Start Network Command* is issued in this situation a *Start Network Response* with ERROR will be returned.

Field	Type	Valid Range	Description
Status	Int	SUCCESS: SR Network has been started ERROR: Generic Error NETWORK_ALREADY_RUNNING: A SR Network is already running. Stop this network before to start a new one	Status of the command
NetworkID	Int	0xFFFF	Unused
Channel	Int	0xFFFF	Unused
StartPort	Int	0xFFFF	Unused

Table 4.6

Note: The fields NetworkID, Channel and StartPort are reserved for future use.

IMPORTANT - Once a SR Network has been started, if the GG863-SR will be powered-down, the same SR Network (NetworkID and Channel) will be restored automatically at the next power-up. Only the port mapping will be lost, hence any successive discovery of the SR Network will start association from default StartPort 6001. For more details about SR Network discovery refer to 4.2.3.7.

4.2.3.3 Stop Network Command

Due to features of Mesh Lite technology, this command should not be used with Mesh Lite. If a Stop Network Command will be issued when Mesh Lite is used an ERROR response will be returned.

The *Stop Network Command* stops the running SR Network. If there is not a running SR Network it will return an error. If there is a running SR Network, this command shall be issued before a *Start Network Command* in order to create a new SR Network.

If at least a Router is associated to the ZigBee network, this command can't stop the network.
In this case *Stop Network Command* removes association between GG863-SR and ZigBee network only.
In order to stop the network each router shall be switched off or resetted one by one, acting directly through its serial interface.

Figure 4.10 shows the Parameters field of this command.

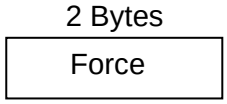


Figure 4.10

Table 4.7 shows the valid range for the values of the parameters.

Field	Type	Valid Range	Description	Default
Force	Int	0xFFFF	Unused	0xFFFF

Table 4.7

Note: Field Force is reserved for future use.

IMPORTANT: Due to features of Mesh Lite technology, this command should not be used with Mesh Lite. If a Stop Network Command will be issued when Mesh Lite is used an ERROR response will be returned.

4.2.3.4 Stop Network Response

On receipt of the *Stop Network Command* the Gateway will try to stop the running SR Network, if any. Then it will send a *Stop Network Response* to the IP Host.
If this command is issued when the IP Gateway application is *command-mode* the SR Network will be stopped, if any, and the IP Gateway application will return in *normal-mode* automatically. In order to notify this event to the IP Host a *Set Mode Response* will be sent through the Control Interface. For further details about this *Set Mode Response* refer to 4.2.3.11.

Figure 4.11 shows the Parameters field of *Stop Network Response.*

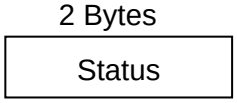


Figure 4.11

Table 4.8 shows the valid range for the values of the parameter of the *Stop Network Response*.

Field	Type	Valid Range	Description
Status	Int	SUCCESS: SR Network has been stopped ERROR: Generic Error NETWORK_ALREADY_STOPPED: SR Network already stopped	Status of the command

Table 4.8

4.2.3.5 Close Pipe Command

The *Close Pipe Command* closes a data communication between the IP Host and a SR node.

Note: The data communication can be close automatically also closing the data link from IP Host side.

Figure 4.12 shows the Parameters field of this command.

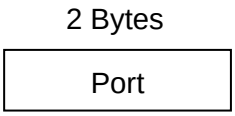


Figure 4.12

Table 4.9 shows the valid range for the values of the parameters.

Field	Type		Valid Range	Description	Default
Port	Int		6001-65535	TCP port to close	6001

Table 4.9

4.2.3.6 Close Pipe Response

On receipt of the *Close Pipe Command* the Gateway will try to close the pipe associated to the TCP port then will send a *Close Pipe Response* to the IP Host.
Figure 4.13 shows the Parameters field of *Close Pipe Response.*

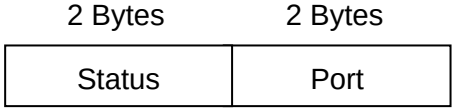


Figure 4.13

Table 4.10 shows the valid range for the values of the parameters of the *Close Pipe Response*.

Note: The Close Pipe Response is sent from the Gateway to the IP Host not only in response to the Close Pipe Command but every time a data connection is closed (E.g. after a Reset Command). This feature allows the IP Host to track the data connection status.

Field	Type	Valid Range	Description

Status	Int	SUCCESS: Port has been closed ERROR: Generic Error CLOSE_PORT_ERR: the port is already closed	Status of the command
ClosedPort	Int	6001-65535	The Port that the Gateway has tried to close

Table 4.10

4.2.3.7 Reset Command

The *Reset Command* resets the Gateway system.

Figure 4.14 shows the Parameters field of this command.

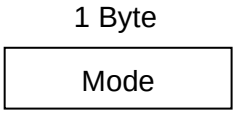


Figure 4.14

Table 4.11 shows the valid range for the values of the parameter.

Field	Type	Valid Range	Description	Default
Mode	Int	1 - Reset data connections and port mapping	Every data pipe is closed, the port mapping is reset and a new scan is made	1

Table 4.11

4.2.3.8 Reset Response

On receipt of the *Reset Command* every data pipe is closed, the port mapping is lost and a new scan is made.
After that the new scan request is made, the *Reset Response* is sent to the IP Host.
This command is not available when IP Gateway is in “Command-mode”.

Figure 4.15 shows the Parameters field of *Reset Response.*

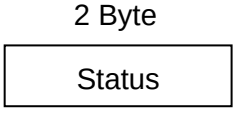


Figure 4.15

Table 4.12 shows the valid range for the values of the parameter of the *Reset Response*.

Field	Type	Valid Range	Description
Status	Int	SUCCESS: reset has been done	Status of the command

		ERROR: error resetting the system	

Table 4.12

During the scan the Gateway will send a *Scan Network Response* (one for each device) to the IP Host through the Control Interface.
When the Scan Network has been completed and all SR Nodes has been discovered, a *Scan Network Response* , with the Status field set to *END_OF_SCAN_NETWORK* value, will be sent to the IP Host.

Figure 4.16 shows the Parameters field of *Scan Network Response.*

2 Bytes	2 Bytes	2 Bytes	12 Bytes	1 Byte	2 Bytes
Status	Node ID	Port	MAC Address	Node Type	Spare

Figure 4.16

Table 4.13 shows the valid range for the values of the parameters of the *Scan Network Response*.

Field	Type	Valid Range	Description
Status	Int	SUCCESS, ERROR, END_OF_SCAN_NETWORK, NEW_DEVICE_ANNOUNCE	Status of the command. If the Status field has value SUCCESS, the NodeID, Port and MAC Address fields will contain informations about a device discovered during scan of the network. If the Status field value is END_OF_SCAN_NETWORK all other fields will be set to 0xFF. The Status field value NEW_DEVICE_ANNOUNCE is used to indicate *Scan Net Response* related to a new device that has just joined the network (not available for Mesh Lite). If there isn't a SR Network running, the Status will be set to ERROR and all other fields will be set to 0xFF.
NodeID	Int	From 1 to 65535	SR node ID. (E.g. for the ZigBee technology it will be the network address of the ZigBee node)
Port	Int	From 6001 to 65535	TCP port associated to the SR node
MAC Address	Int [12]	From 0 to 2^{96}-1	MAC address of the SR node terminated with 0x00 (null) byte.. For ZigBee, each bytes is the hexadecimal value of the correspondant cipher of the MAC Address. For Mesh Lite, each byte is the ASCII code of the correspondant cipher of the MAC Address.
Node Type	Int	From 1 to 255	Reserved for future uses.
Spare	Int	From 0 to 65535	Reserved for future uses.

Table 4.13

Figure 4.17 shows the messages exchanged between entities due a *Reset Command*. The *Close Pipe Response* will be sent only if there is a data pipe opened.
Other messages in dotted lines are not part of the reset procedure; they are transmitted only to exchange data with the SR Node.

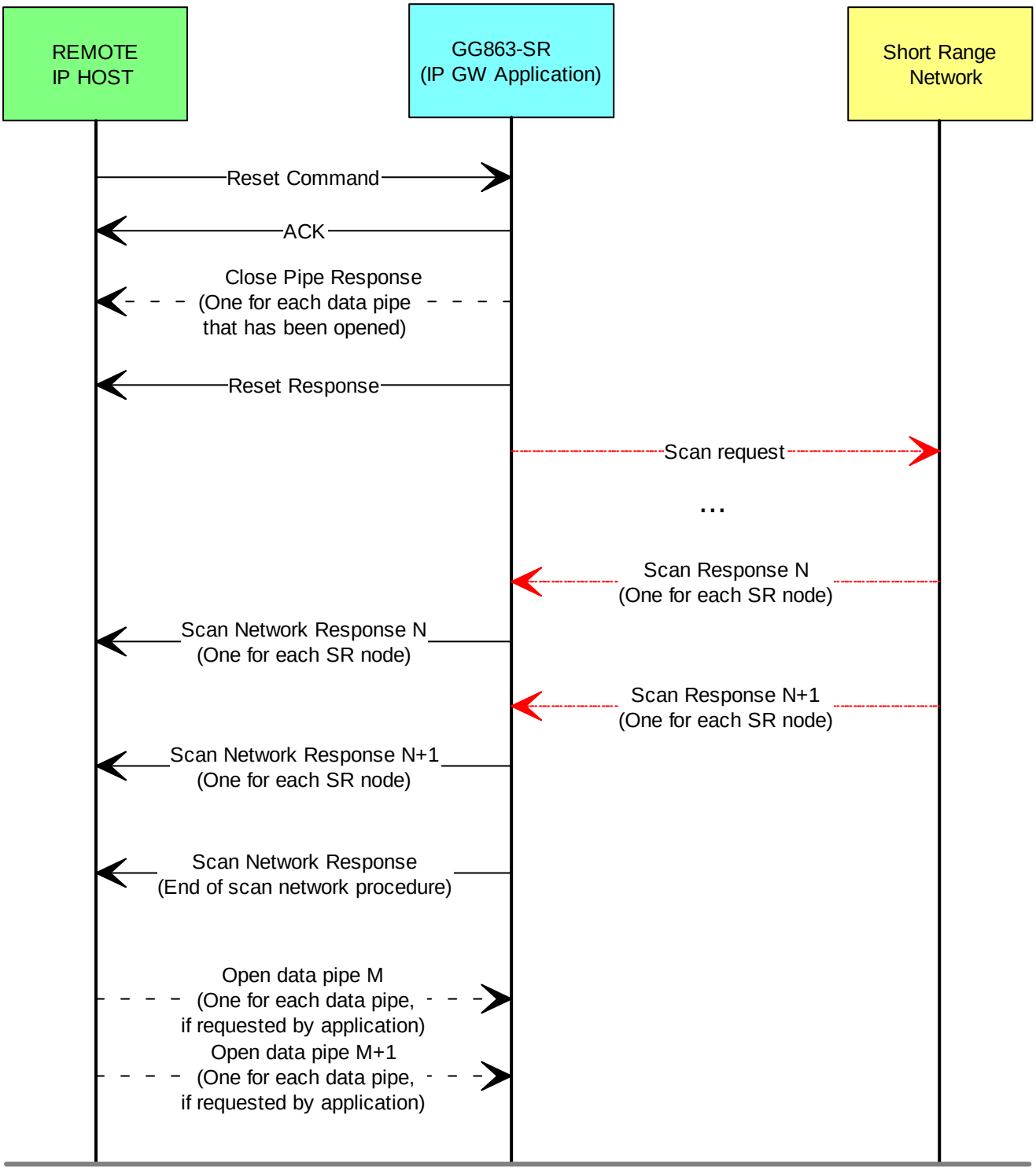


Figure 4.17

Figure 4.18 shows the messages exchanged between entities when a *Reset Command* has been issued without a SR Network running.
In this case it is not possible to perform a SR Network Scan but internal tables of IP Gateway Application can be erased as well.
Hence the *Scan Network Response* sent toward the Remote IP Host will contain the Status field set to ERROR. It means that the Scan Network procedure has ended with errors; then, if a SR Network will be started, the Scan Network procedure will be available again (through the *Reset Command*).

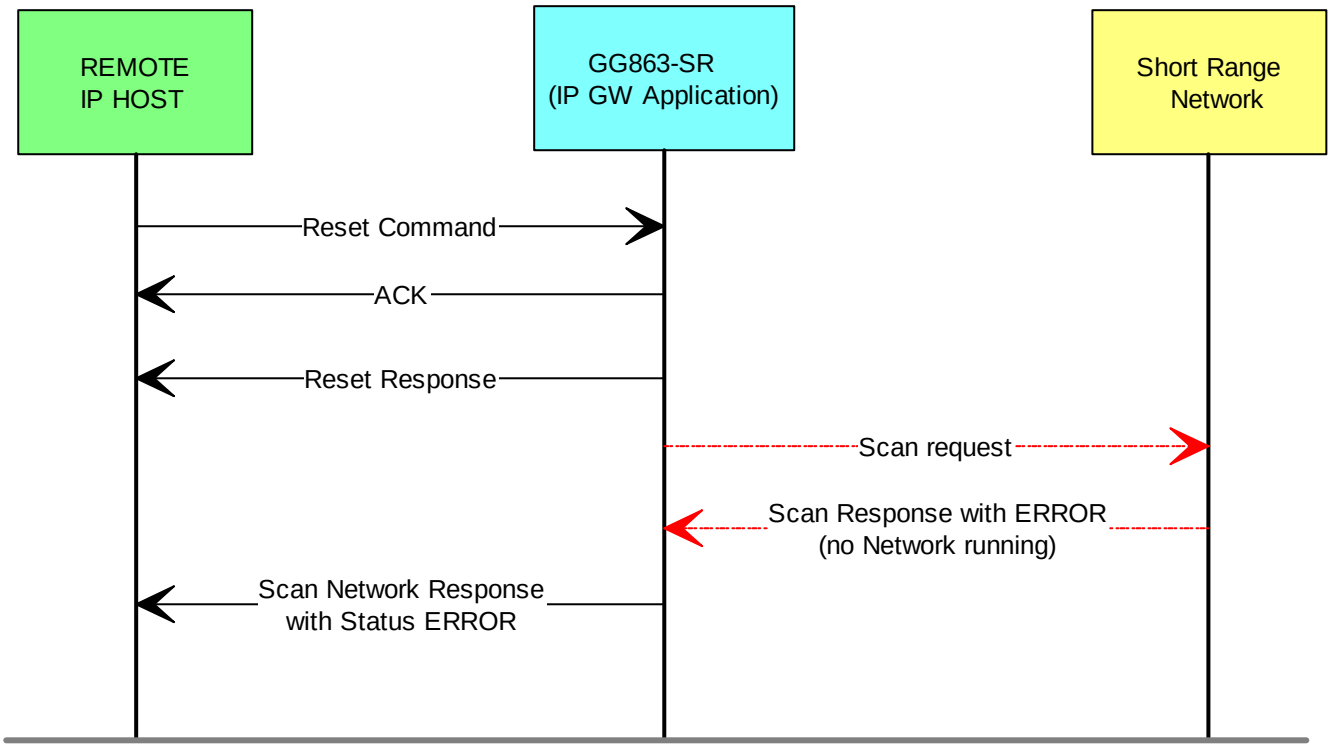


Figure 4.18

4.2.3.9 Scan Network Responses (unsolicited)

Unsolicited *Scan Network Response* messages will be sent to the IP Host also when a new SR node joins the network (one for each device). This feature is available only for the ZigBee technology, and its behavior is the same in both “Normal-mode” and “Command-mode”.

Figure 4.16 shows the Parameters field of *Scan Network Response* and Table 4.13 shows the valid range for the values of its parameters. In order to distinguish between device already in the network and new devices, the *Scan Network Response (unsolicited)* has Status field value set to NEW_DEVICE_ANNOUNCE value.
The sequence diagram in Figure 4.19 shows the messages exchanged between entities when a new SR node joins the network.

Messages in dotted lines are not part of the join procedure, they are transmitted only to exchange data with the new SR Node.

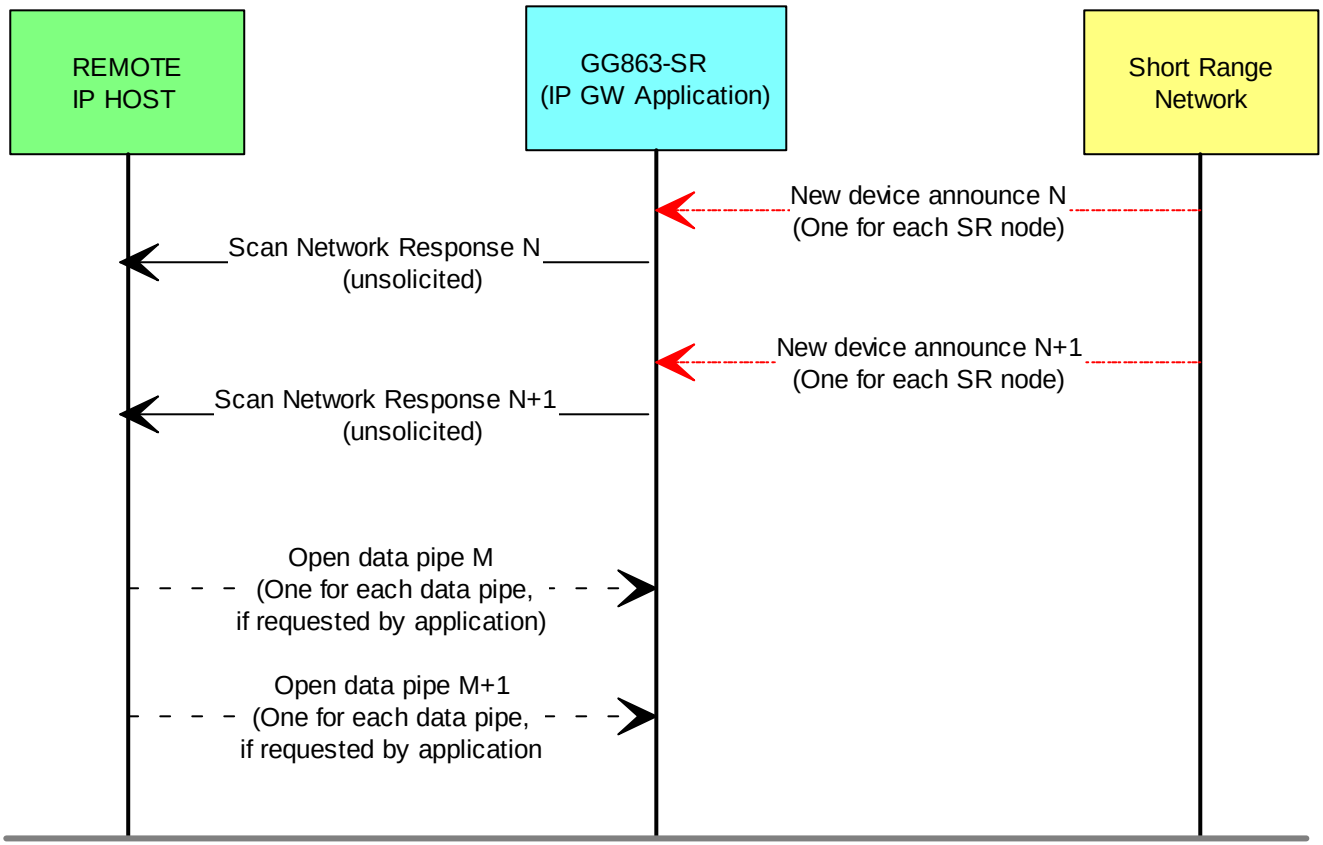


Figure 4.19

4.2.3.10 Set Mode Command

The Set Mode Command gives the possibility to switch the behavior of the Gateway system between Normal-mode to Command-mode and vice-versa.

Figure 4.20 shows the Parameters field of this command.

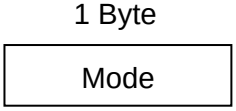


Figure 4.20

Table 4.14 shows the valid range for the values of the parameter.

Field	Type	Valid Range	Description	Default
Mode	Int	0 – Normal-mode 1 – Command-mode	The IP Gateway Application wil be switched to the specified mode.	0

Table 4.14

4.2.3.11 Set Mode Response

On receipt of the *Set Mode Command* the IP Gateway Application will switch its behaviour. After that the *Set Mode Response* is sent to the IP Host.

Note: Only for the ZigBee version:
An unsolicited *Set Mode Response* will be sent to the IP Host every time the IP Gateway Application changes its behavior.

Figure 4.21 shows the Parameters field of *Set Mode Response.*

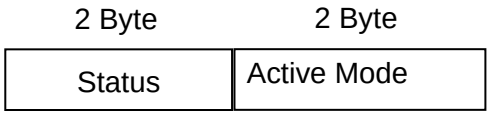


Figure 4.21

Table 4.15 shows the valid range for the values of the parameter of the *Reset Response*.

Field	Type	Valid Range	Description
Status	Int	SUCCESS: IP GW Application has been switched to desired mode ERROR: generic error	Status of the command
Active Mode	Int	0 – Normal-mode 1 – Command-mode	Actual mode of the IP GW Application

Table 4.15

The *Set Mode Response* will contain a ERROR if the *Set Mode Command* will be issued when there is no SR Network running.

4.2.3.12 Global Version Command

The Global Version Command retrieves the version of the IP Gateway software running in the **GG863-SR**.

Global Version: ***2C.MM.mm***
- ***2C***: represents the product IP Gateway
- ***MM***: Major Version
- ***mm***: Minor Version

Example: 2C.00.01

Figure 4.22 shows the Parameters field of this command.

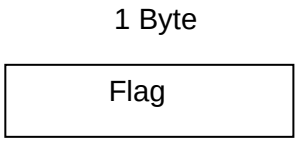


Figure 4.22

Table 4.16 shows the valid range for the values of the parameter.

Field	Type	Valid Range	Description	Default
Mode	Int	0x00	Information about global version will be showed.	0x00

Table 4.16

4.2.3.13 Global Version Response

On receipt of the *Global Version Command* the IP Gateway Application will send the *Global Version Response* to the IP Host.
Figure 4.23 shows the Parameters field of *Global Version Response.*

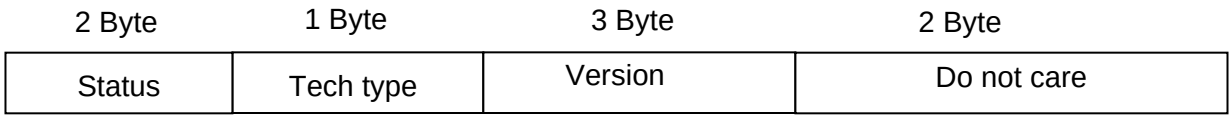


Figure 4.23

Table 4.17 shows the valid range for the values of the parameter of the *Global Version Response*.

Field	Type	Valid Range	Description
Status	Int	SUCCESS: The response shows the requested information. ERROR: generic error	Status of the command
Tech Type	Int	0x22 – ZigBee 0x23 – Mesh Lite	Short Range Technology Type in use. It depends on the Short Range chip used.
Version	Int	0x000000 - 0xFFFFFF	Global Version of the IP Gateway software
Do not care	Int	0xFFFF	Do not care

Table 4.17

4.2.3.14 Ack message

Ack message format is shown in Figure 4.24

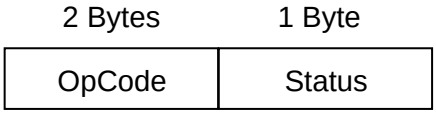


Figure 4.24

The OpCode is the same of the Command that has to be acknowledged. If an OpCode shorter than 1 byte has been issued, then the default OpCode 0XFFFF will be used. Status explains if the Command sent by the IP Host is correct.
Table 4.18 shows the valid range for the values of the parameter.

Field	Type	Valid Range	Description
OpCode	Int	See Table 4.2	Operation code of the command
Status	Int	SUCCESS: command is correct ERROR: command is not correct	Explains if the Command sent by the IP Host is correct

Table 4.18

4.3 Data Interface

When a pair TCPport – nodeID is got, Data interface creates a pipe between the TCP port and the SR node. Once created, each pipe can be used to exchange raw bit stream between IP Host and SR node in both directions.
That is to say, each byte sent from IP Host toward a data pipe will be transferred to the SR node without any modification, and vice-versa.

IMPORTANT – A limitation to the statement above applies when ZigBee technology is used: due to a limitation of the ZigBee firmwares, if the length of the raw bit stream is bigger than 84 bytes, the IP-Gateway-Application breaks it up in packets of length <= 84 bytes. The SR Node Application should reassemble the original raw bit stream according [12].

IMPORTANT – A limitation to the statement above applies when Mesh Lite technology is used. In order to understand how to send/receive data to/from an end device using normal-mode and command –mode when the Mesh Lite technology is used, please refer to Appendix.5 – How to send and receive raw data using MeshLite technology.

The IP GW Application supports two different operating modes: the normal mode and the command mode
When the IP GW Application is used in *normal-mode*, the IP Host needs only a minimal knowledge of the Short Range functionalities and the data sent through those pipes are raw-data. The *normal-mode* is explained in further details in § 4.3.1.
When the IP GW Application is used in *command-mode*, the IP Host might be used to send commands to the SR node using data pipes. In the latter case, the IP Host needs to know the set of commands available for the Short Range technology and their functionalities.
Once switched in *command-mode* all data pipes of the Data Interface can be used to send commands directly to the SR node associated to the data pipe. These commands vary accordingly to the Short Range technology used. Further details on the *command-mode* are provided in § 4.3.2.

4.3.1 Normal-mode

If the normal-mode is used, **GG863-SR** Gateway is completely transparent. Data messages flow in both directions as a raw bit stream.

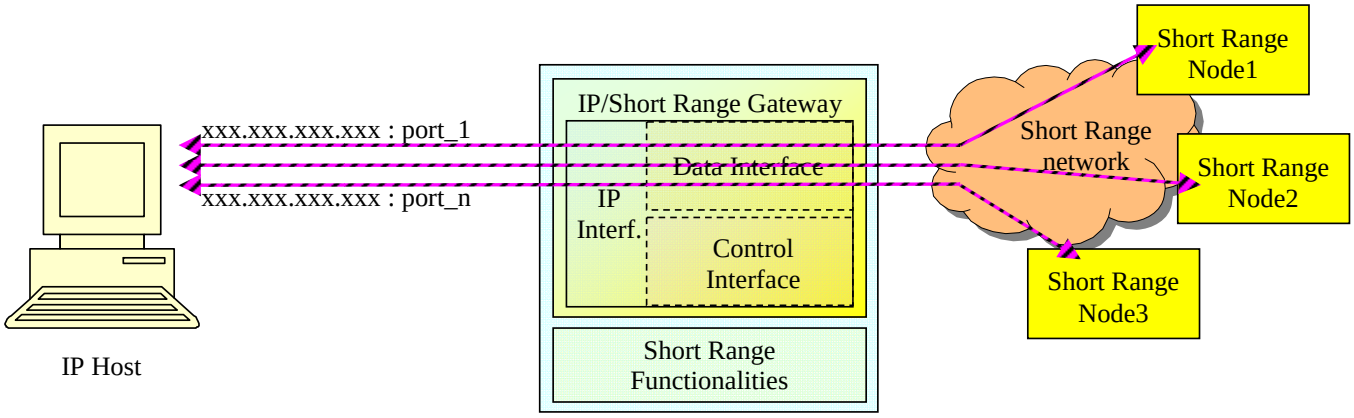


Figure 4.25 – Gateway in normal mode

The way data messages are used is totally user dependent. From a logical point of view the responsible for data managing are applications running on IP Host and each SR node connected to the gateway.

Figure 4.26 – Normal Mode: data flow at application layer

4.3.2 Command-mode

If command-mode is used, the IP Host may send specific commands to a specific SR node, which is addressable through a data pipe with a specific IPaddress:port pair. Messages exchanged through these pipes shall be formatted according to the protocol used by Short Range technology. For the ZigBee refer to [6], for the MeshLite refer to [5].

4.3.2.1 Specific ZigBee commands

All the commands/responses/indications supported by the IP GW Application, when switched in command-mode, are listed in the following table:

Commands	Description
Nwk Addr request	Retrieve the 16-bit address (network address) of the ZB node using its known IEEE address.
Nwk Addr confirm	Confirm to Nwk Addr request
IEEE Addr request	Retrieve the IEEE address of the ZB node using its know network address
IEEE Addr confirm	Confirm to IEEE Addr request
Node desc request	Retrieve the node descriptor of the ZB node
Node desc confirm	Confirm to Node desc request
Power desc request	Retrieve the power descriptor of the ZB node
Power desc confirm	Confirm to Power desc request
Simple desc request	Retrieve the simple descriptor of a specific endpoint of the ZB node.
Simple desc confirm	Confirm to Simple desc request
Active EP request	Retrieve the list of endpoints of the ZB node
Active EP confirm	Confirm to Active EP request
User desc request	Retrieve the user descriptor of the ZB node
User desc confirm	Confirm to User desc request
User desc set request	Set the user descriptor of the ZB node
User desc set confirm	Confirm to User desc set request
Match desc request	Check simple descriptor support for a ZB node
Match desc confirm	Confirm to Match desc request
AF Direct request	Send data to a ZB node

AF Direct indication	Receive data from a ZB node
AF Direct confirm	Confirm to AF Direct request

Table 4.19

Note: In order to work correctly each command listed in Table 4.19 shall be sent from data pipe to the ZB node associated to it. Broadcast addresses and data pipe/ ZB node mismatch may cause unexpected behaviour.

For further details about each command/response/indication refer to [12] . Pay attention at the use of commands, they may lead to change critical parameters for the SR network.

It is recommended to use Data Interface to exchange raw bit stream between IP Host and SR node.

At the moment, due to a limitation of the ZigBee firmwares, using the commands AF Direct request and AF Direct indication the length of the Afdu shall not be bigger than 84 bytes to prevent uncorrect behaviours .

4.3.2.2 Specific MeshLite commands

In the MeshLite technology, there is a specific format of transmitted serial frames:

TYPE	RECIPIENT	CONTENT

Field	Length (Bytes)	Description
TYPE	1	Type of Frame: • 0x65 → data • 0x6A → ping • 0x6D → AT command
RECIPIENT	2	Recipient ID (LSB first)
CONTENT	-	Message content: • Data to be sent (for 0x65) • Blank (for 0x6A)

		• AT command (for 0x6D)

Below is described the format of received serial frames:

TYPE	SENDER	CONTENT

Field	Length (Bytes)	Description
TYPE	1	Type of Frame: • 0x65 → data • 0x6B → ping • 0x6E → AT command
SENDER	2	Sender ID (LSB first)
CONTENT	-	Message content: • Data to be sent (for 0x65) • Blank (for 0x6B) • AT command (for 0x6E)

NB: In order to work correctly each command shall be sent from data pipe to the ML node associated to it. Data pipe/ ML node mismatch may cause unexpected behaviour.

4.3.2.2.1 AT command

The 'AT' protocol or Hayes mode is used to configure the Mesh Lite device parameters, based on the following principles:

- A data frame always begins with the two ASCII 'AT' characters, standing for 'Attention'
- Commands are coded over one or several characters and may include additional data
- A given command always ends up with a <CR> Carriage Return

Below is described the Content of AT command:

'AT'	**Command**	**Additional data**	**<CR>**

Below is described the complete list of the 'AT' commands available

Command	Description	Response
/V	**Modem's firmware version** 'AT/V' command displays the modem's firmware version number.	Version <Product>: vX.YZn
Sn?	**Register interrogation** 'ATSn?' command displays the content of Hayes register number n (Refer to the register description table).	Sn=x
Sn=m	**Register modification** 'ATSn=m' command configures Hayes register number n with the value m. The	OK or ERROR

	value is automatically stored in the EEPROM memory.	
R	**Parameters reset** 'ATR' command resets all modem's parameters to their default values.	OK
C	**Children table rest** 'ATC' command clear the module's children table.	OK
O	**Save and Reboot** 'ATO' command reboots the module to apply all previous setting.	
BL	**Switch to Bootloader** 'ATBL' command escapes from the main program and run the bootloader. This command is useful to update the firmware.	

Below an example of 'AT' command used to set a specific register:

In ASCII 0x6D is 'm' and 0x6E is 'n'; we suppose to set the register 300 on the device 'E1' (0x45,0x31) with the value '0'. The complete sequence of command and response is:
Command: 'mE1ATS300=0\r' – in hexadecimal – 0x6D, 0x45, 0x31, 0x41 ,0x54, 0x53, 0x33, 0x30, 0x30, 0x3D, 0x30, 0x0D
Response: 'nE1OK\r' – in hexadecimal – 0x6E, 0x45, 0x31, 0x4F, 0x4B, 0x0D

4.3.2.2.2 Data

If a Data packet is sent, the user shall not expect any answer from the recipient module. The data transmission will depend on the payload chosen (S330 register). The payload defines the maximum amount of data that can be sent in one radio frame. The transmission of higher amount of data is possible (multi-frames transmission): in that case, it is necessary to manage the serial flow control in order to avoid buffer saturation.

A typical data packet is shown below:
0x65, 0x45, 0x31, … PAYLOAD …, 0x0D

For more details refer to Appendix.5 – How to send and receive raw data using MeshLite technology

4.3.2.2.3 Ping

The typical Ping command consists of a very simple packet; if the user wants to send a ping to a specific network element (i.e. the device ID is 0x45,0x31 (ASCII 'E1')) the command to send is:

Ping: 0x6A – 0x45 – 0x31 – 0x0D

The answer expected is:

Pong: 0x6B – 0x45 – 0x31 – 0x0D

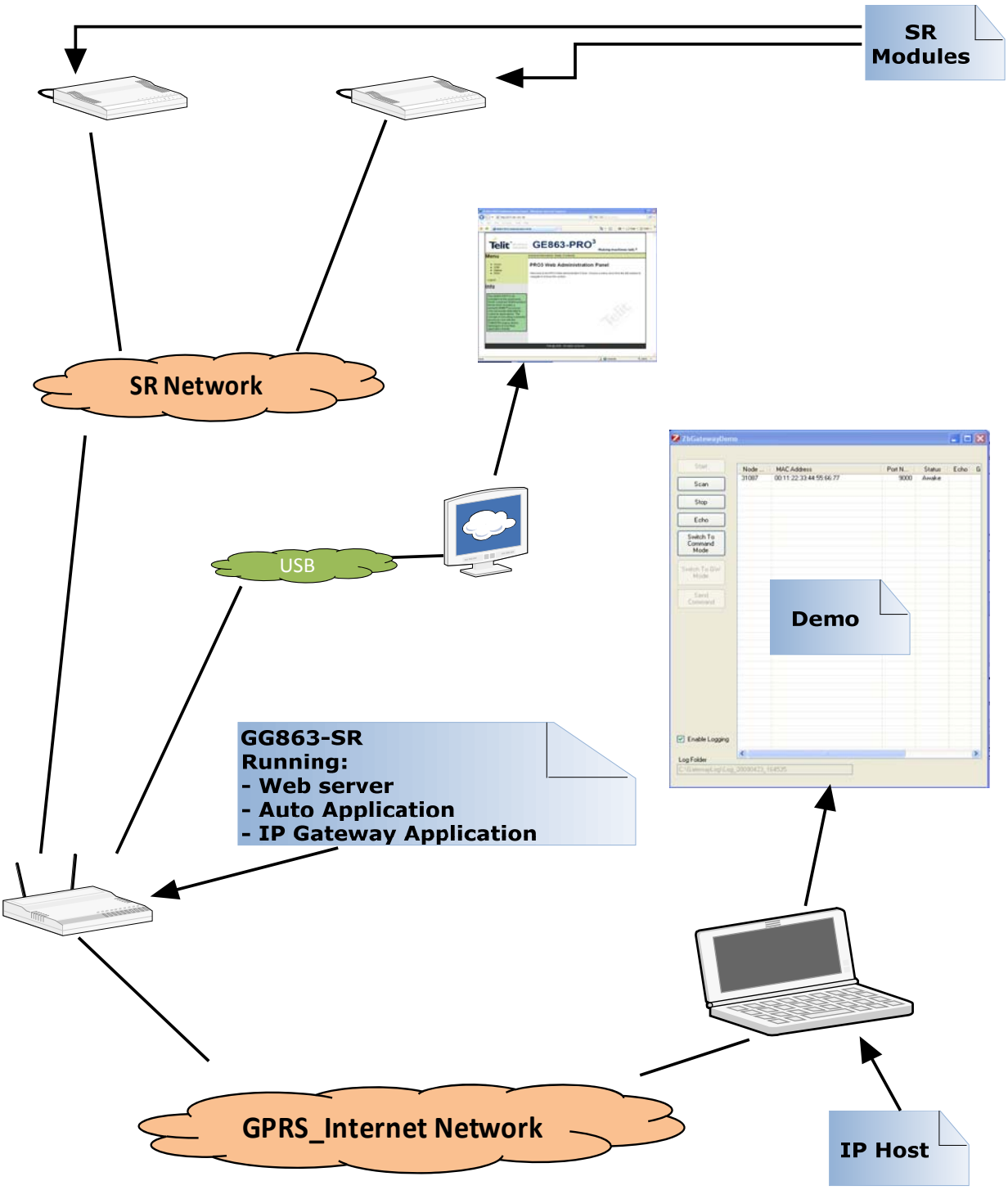
5 Web Administration Panel

5.1 Overview

The Web Server running in the **GG863-SR** gives the user the possibility to configure/customize all the parameters used by the Auto Application.
The Web Administration panel provides the access to the **GG863-SR** thanks to a window emulating the Linux Shell.

5.2 Scheme

In the picture below it is depicted how **GG863-SR**, local PC and IP host are connected:


SR Modules
SR Network
USB
Demo
GG863-SR
Running:
- Web server
- Auto Application
- IP Gateway Application
GPRS_Internet Network
IP Host

5.3 How it works

To use the Web Server functionalities:

- Connect the Local PC to the **GG863-SR** (see § 2.4.5)
- In the local PC connected to the **GG863-SR** type the address:

`http://nnn.nnn.nnn.nnn/cgi-bin/index.cgi`

where nnn.nnn.nnn.nnn is one of the IP addresses of the **GG863-SR**.

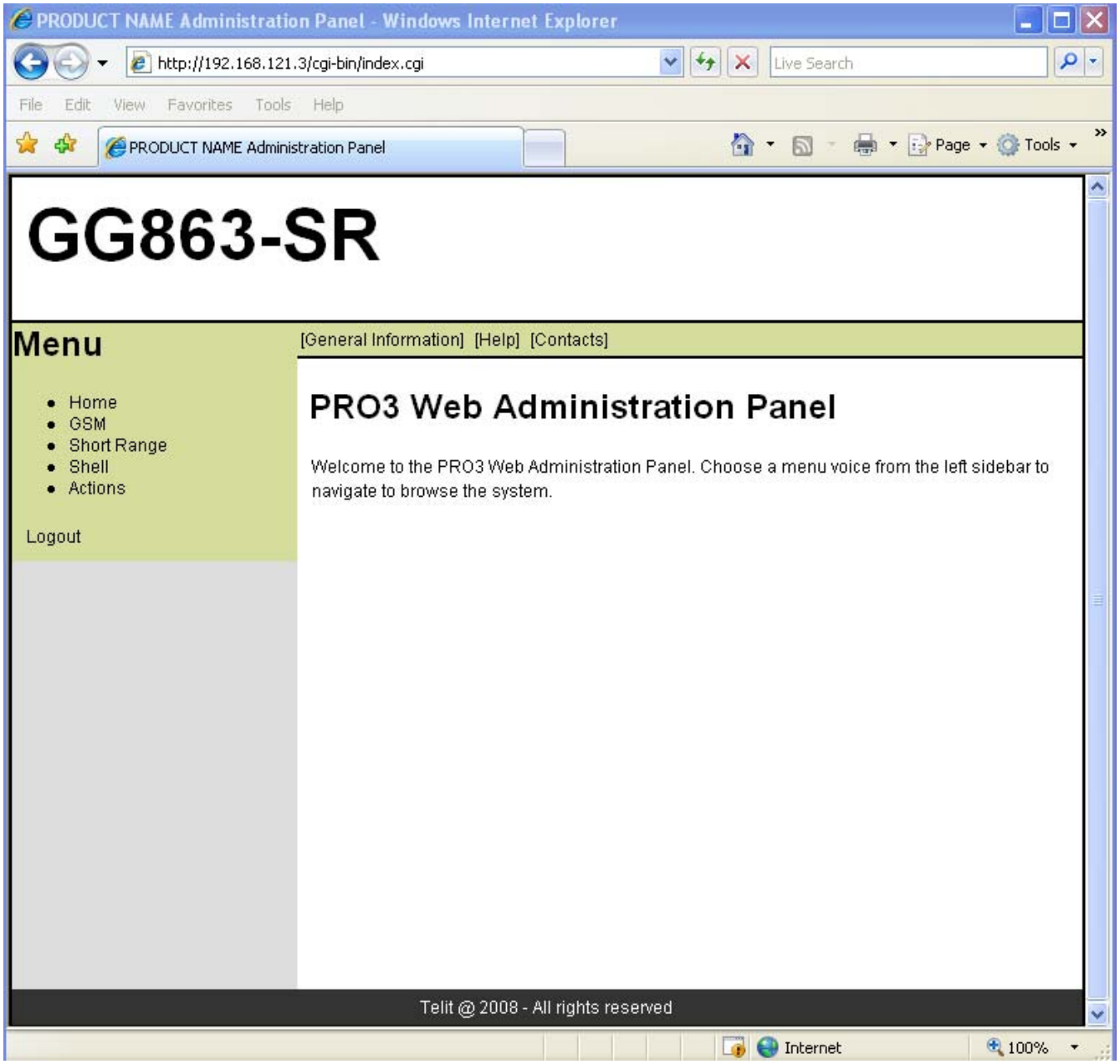
The GG863-SR Administration Panel might be reached from a remote PC connected through a GPRS connection. In this case the IP address is the one assigned to the GG863-SR by the Mobile Network Operator.

Note – When connected via USB (local configuration), the address is usually 192.168.121.3

Warning– The GG863-SR IP address must be directly accessible on TCP port 80 for HTTP connection to be established.
For GPRS connections, the IP address is usually assigned by the Mobile Network Operator. Please check with your MNO for further information.

For example:

You can see the Home Page of the Web Administration Panel:



5.3.1 Short Range menu

In order to connect a remote IP Host to the **GG863-SR**, you have to set the IP Host parameters:

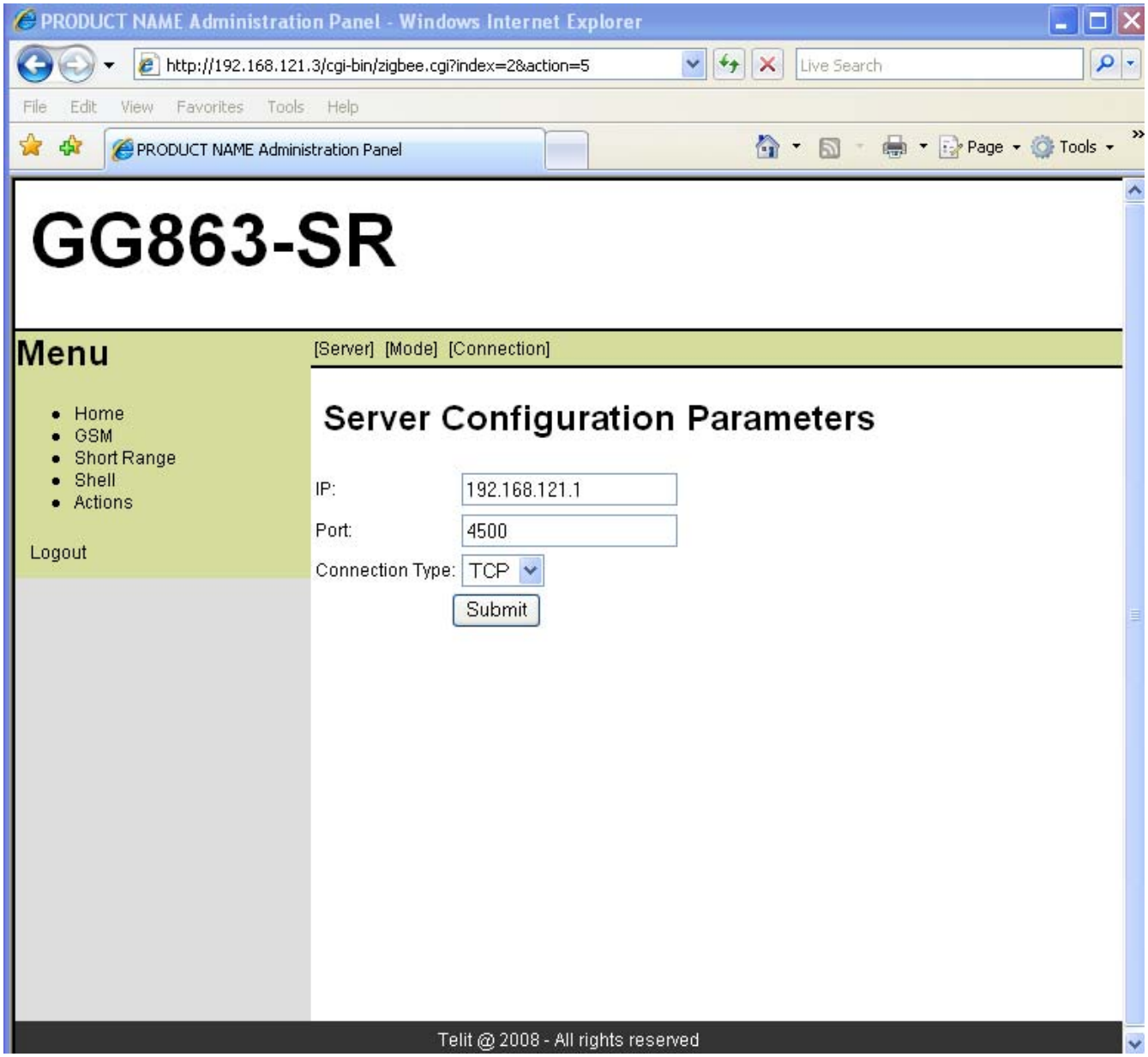
1. Click on ShortRange

2. Click on [Server]

By filling the form you set the configuration files parameters used then by Auto Application (The Auto Application is described in the next section).

- IP: Host address
- Port: please specify the port on which the IP host is waiting for network registration of **GG863-SR**
- Connection Type: specify TCP

For example:

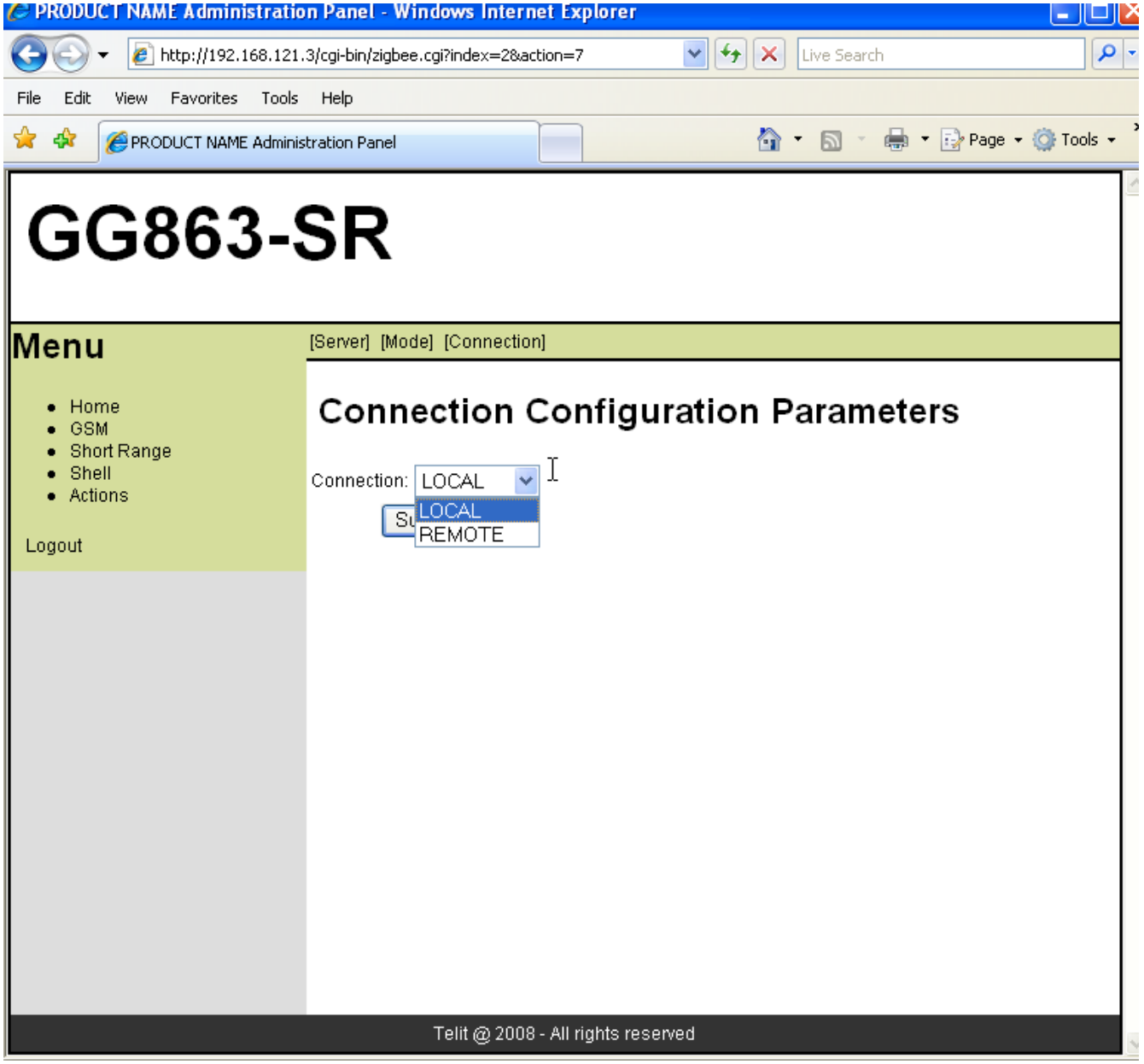


3. Click **Submit,** wait until the Web Panel shows the message:

   Parameter configuration was successful

4. Click on [Connection]

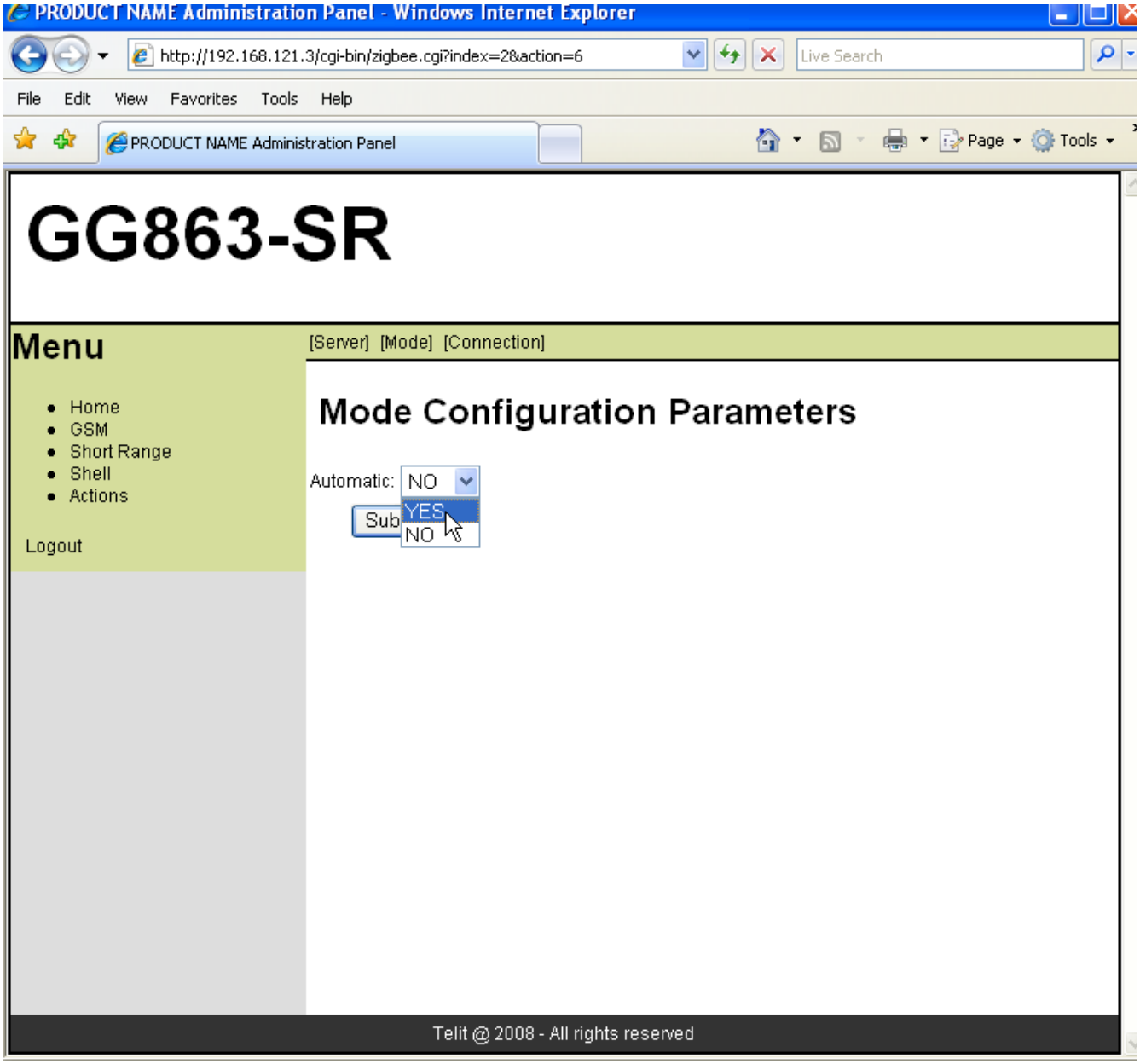
– Connection: the type of connection between GG863-SR and the IP host (**LOCAL/REMOTE**)



5. Click Submit, wait until the Web Panel shows the message:

Parameter configuration was successful

6. Click on [Mode]
   - Automatic: allows to set if Auto Application will be started automatically or not at next reboot of GG863-SR (**YES/NO**)

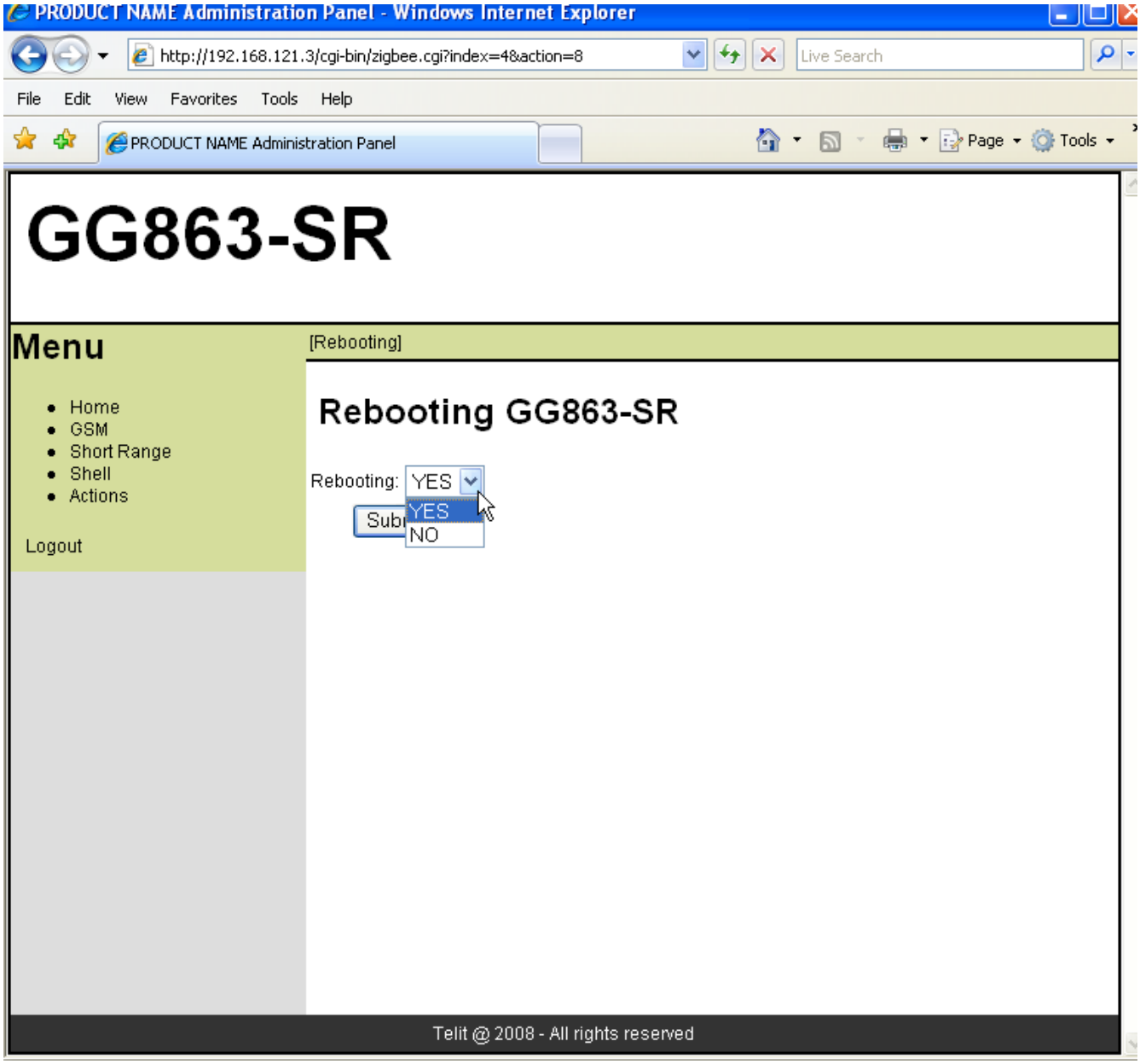


Click **Submit,** wait until the Web Panel shows the message:

Parameter configuration was successful

7. Click on Actions:
   - Click on [Rebooting]
   - Rebooting: allows to reboot or not the GG863-SR (**YES/NO**).

   Note: you should select **YES** only if you really want to reboot the GG863-SR.



Click **Submit,** wait until the Web Panel shows the message:

Parameter configuration was successful

5.3.2 GSM/GPRS Menu

- In order to connect the **GG863-SR** to the GPRS Network, you have to specify the APN:

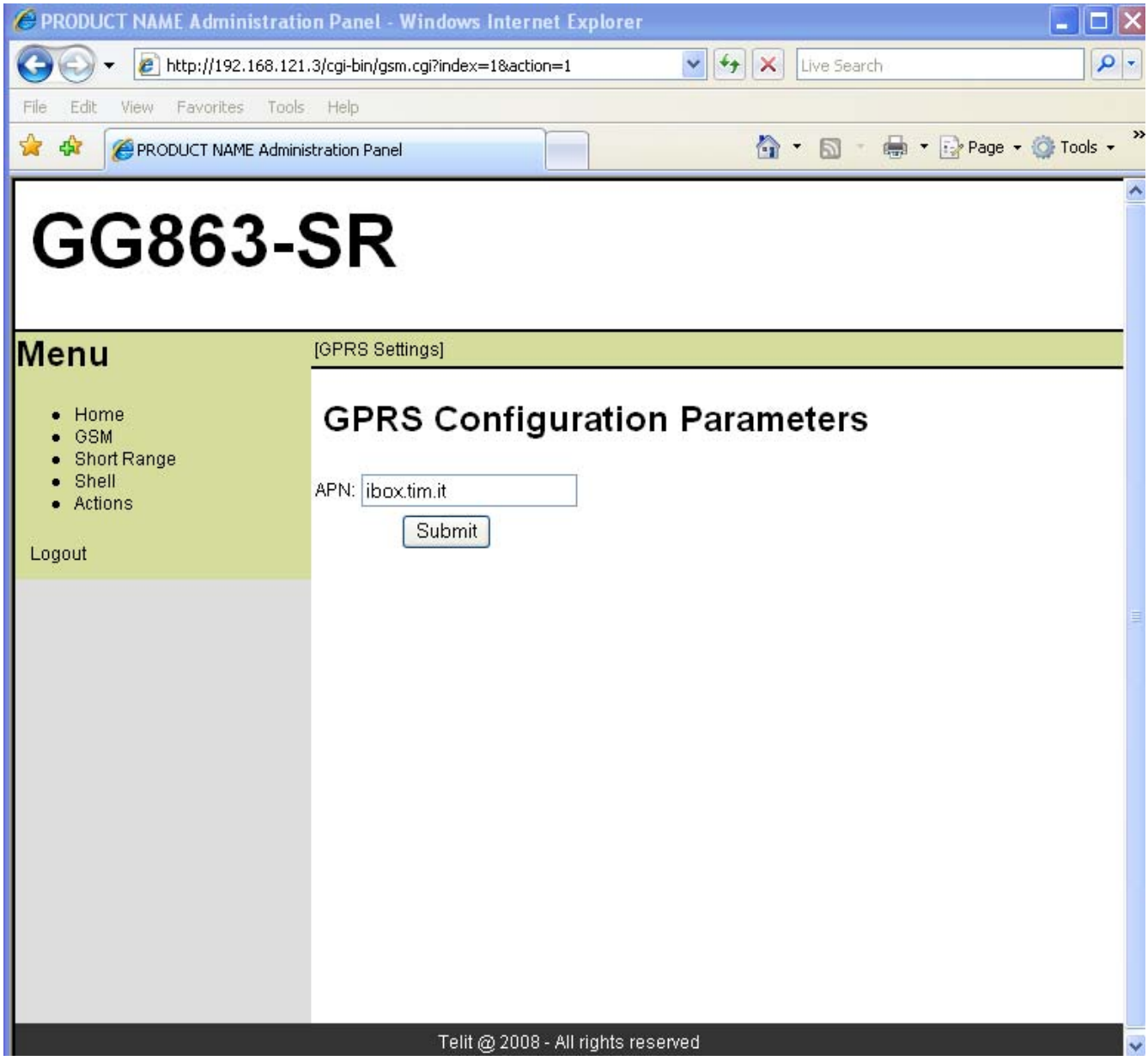
.

- Click on GSM
- Click on [GPRS Settings]

  By filling the form you set the configuration parameters used by Auto Application.

  APN is the Access Point Name provided by the Mobile Network Operator

For example:

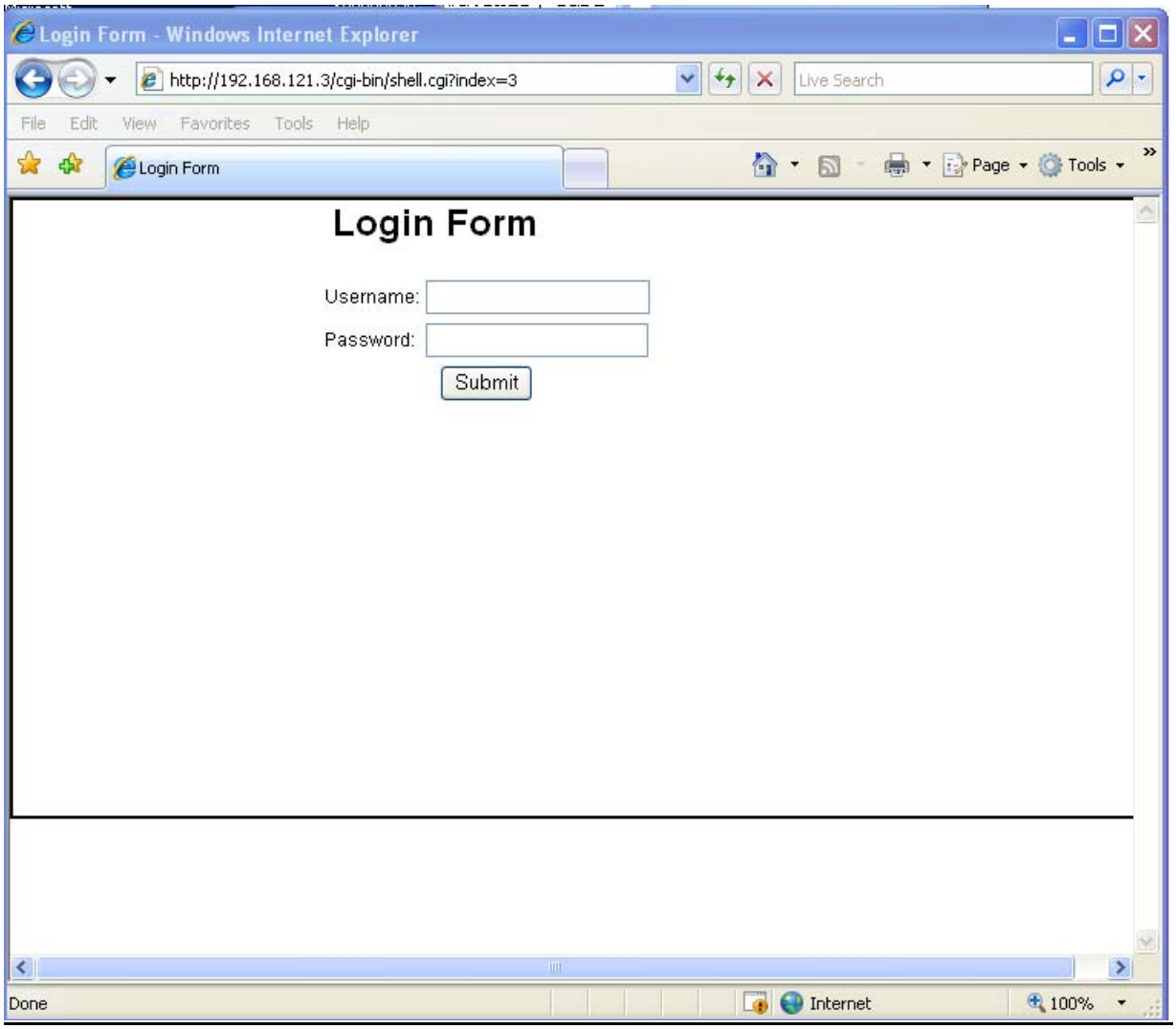


5.3.3 Shell menu

- In order to use the shell on the **GG863-SR** given by the Web Administration Panel, go to the Home Page Menu and click on:

  ❖ [*Shell*]

  Type Username and Password and click Submit



Tip–Default user/password pair is: admin/123456

- After the login phase click on the Tab:
  - [Shell form]

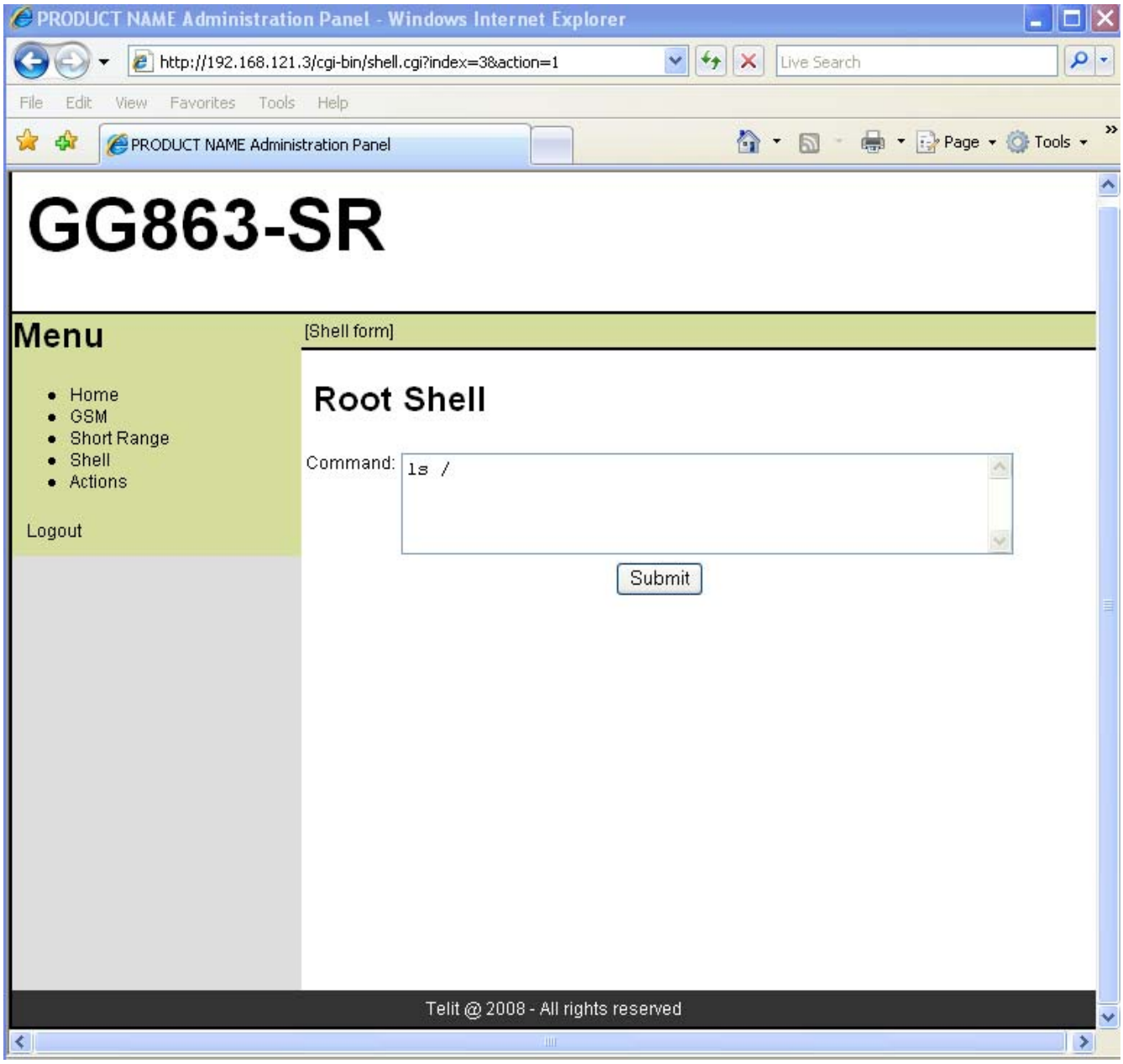
- You can use this form as **GG863-SR** shell:
  - Click on [Shell form]

- Here there are reported some examples:

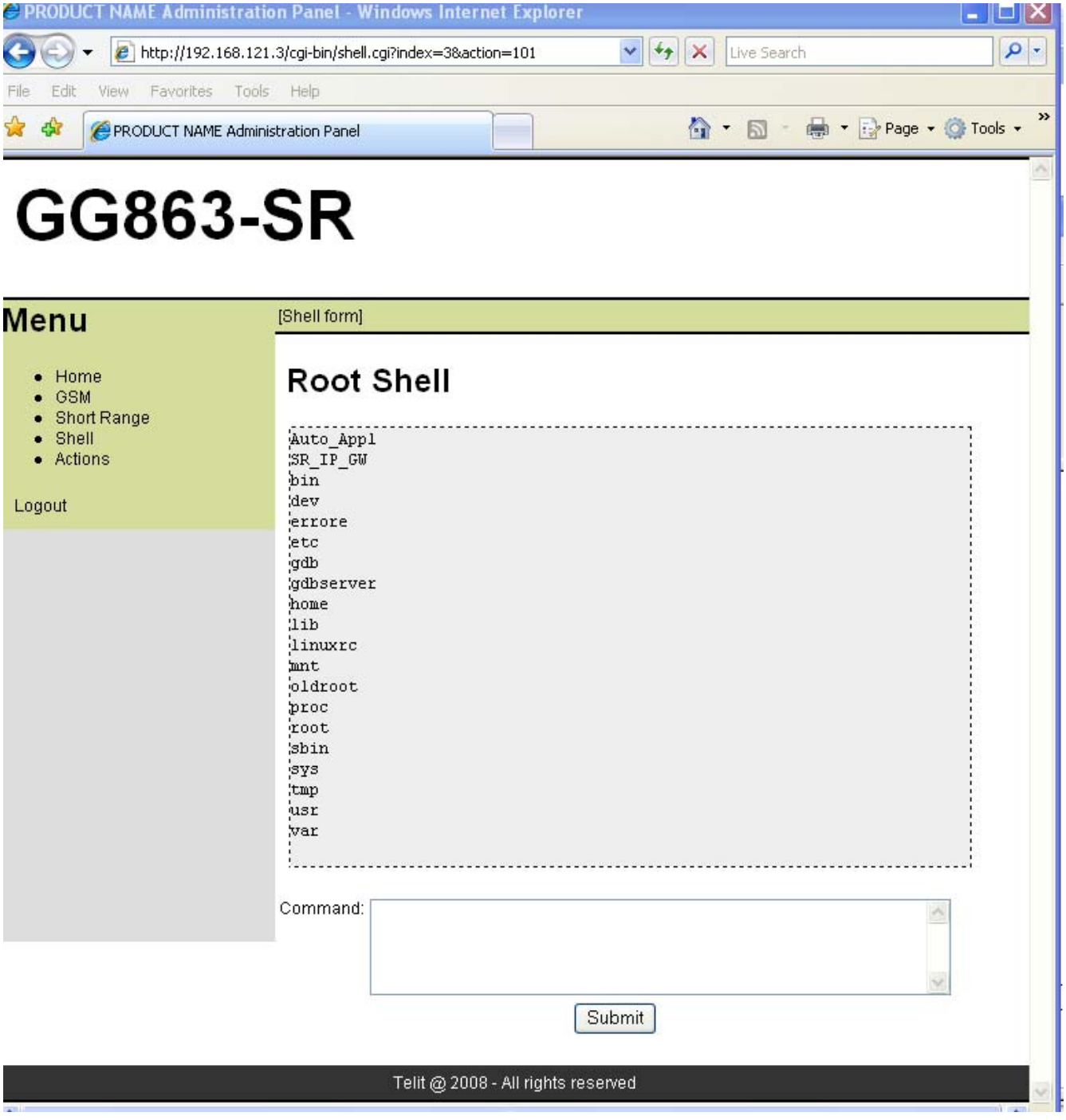
    1) Fill the text area with the following text:

       `ls /`

       Click Submit



You can see for example:



- Write in the text area:

```
ifconfig
```

Click Submit

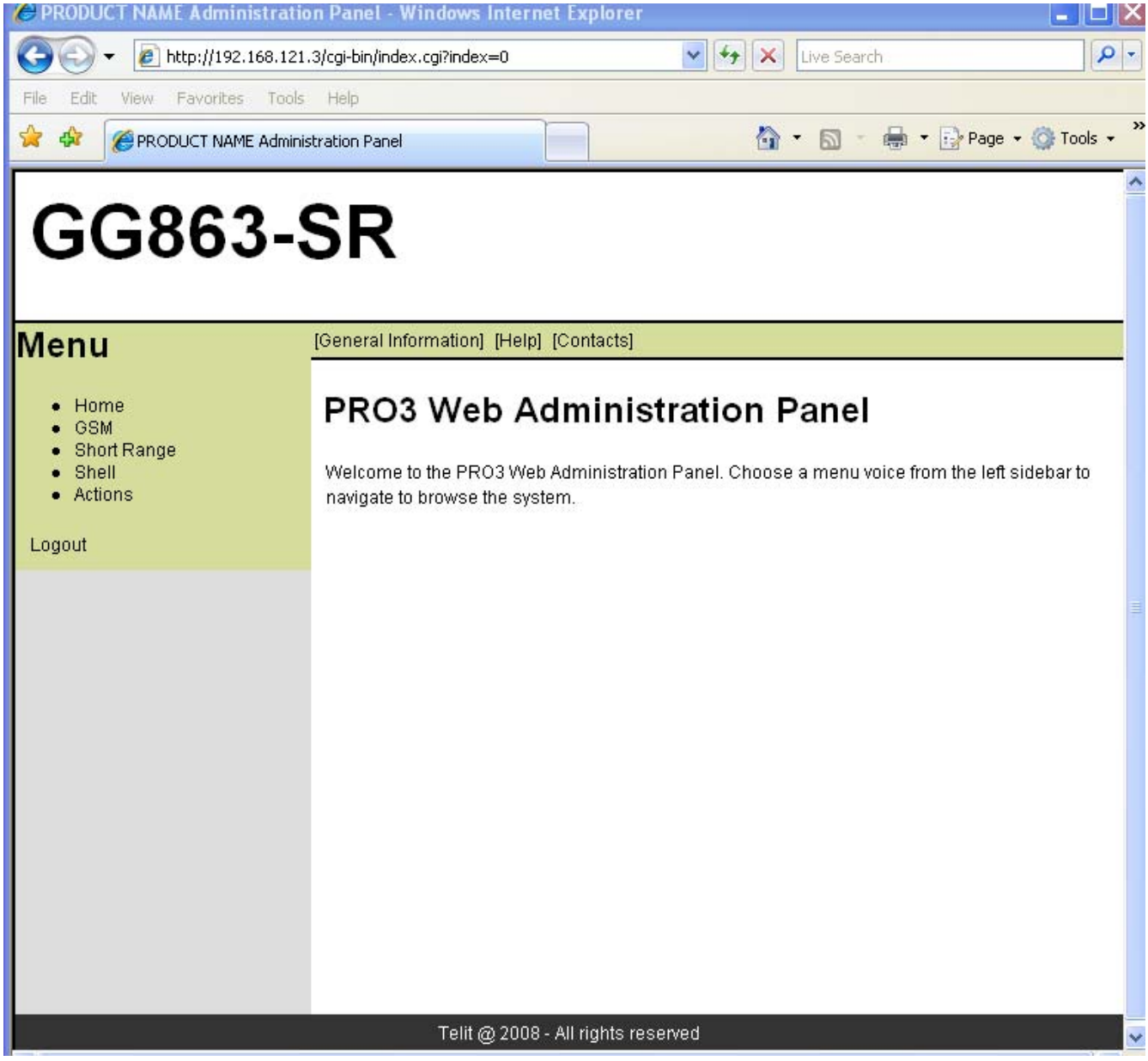
You can see for example:

Below it is reported a list of commands supported by the Web Administration Panel shell:

- `cat`
- `echo`
- `ifconfig`
- `ipcs`
- `ls`
- `lsmod`
- `netstat`
- `printenv`
- `ps`
- `who`
- `whoami`

- To log out from Web Administration Panel go to the Home Page Menu:
  - Click on Logout

You will be returned to the home page


PRODUCT NAME Administration Panel - Windows Internet Explorer
http://192.168.121.3/cgi-bin/index.cgi?index=0
Live Search
File Edit View Favorites Tools Help
PRODUCT NAME Administration Panel
Page
Tools
GG863-SR
Menu
Home
GSM
Short Range
Shell
Actions
Logout
[General Information] [Help] [Contacts]
PRO3 Web Administration Panel
Welcome to the PRO3 Web Administration Panel. Choose a menu voice from the left sidebar to navigate to browse the system.
Telit @ 2008 - All rights reserved

6 Auto Application

The goal of the Auto Application is to connect a remote IP Host to the **GG863-SR** in order to use the gateway functionalities.

6.1 How it works

Following the set up phase described in the “Appendix.1 – ” the Auto Application starts at each Linux boot.
The Auto Application performs the following actions:

1. Read IP Host parameters as configured by the Web Administration Panel:
   - **IP**: IP Host Address
   - **port** : PORT
   - **connection**: Connection Type (TCP)

2. Read GSM parameters as configured by the Web Administration Panel:
   - **APN**: acces point name

3. Open a GPRS connection using the chosen APN

4. If the GPRS connection has been activated with errors the application goes into a loop that tries again to opening the GPRS connection every 24 hours and for 10 times every 10 minutes. The loop breaks when the GPRS connection has been activated with no errors.

5. Run the IP Gateway Application (described in §4).

6. At this point the IP Host has not yet the possibility to connect to the IP Gateway Application because it does not know the **GG863-SR** address.
   For this reason the Auto Application will proceed performing these tasks:

   - Try to connect to the IP Host with IP, PORT and Connection Type configured by Web Server as described in § 5.3.1

     - If the connection to the IP Host has not been activated the application goes into a loop that tries again to connect to the IP Host for 10 times every 3 minutes. The loop breaks when the connection has been activated or when the max number of tries is reached, in this last case the application performs a reboot and goes back to the point N. 1.

     - If the connection to the IP Host has been established the Auto Application will send to the IP Host the **GG863-SR** information: IMEI and IP address needed to connect the IP Host to the IP Gateway Application.

- Listen for an ACK from IP Host every 5 seconds for 10 times:
  - if the max number of tries is reached without receiving an ACK from IP Host, the application performs a reboot and goes back to the point N. 1.
  - If the ACK has been successfully received the Auto Application goes into a loop that checks every 30 seconds if the GPRS Linux interface is up/down. If GPRS is down it performs a reboot and goes back to the point N.1, this is necessary because if the GPRS connection is broken the Auto Application should be restarted.

Before to perform the reboot the system waits about 2 minutes

6.2 Flowchart

Here is the flowchart that describes the Auto Application workflow:

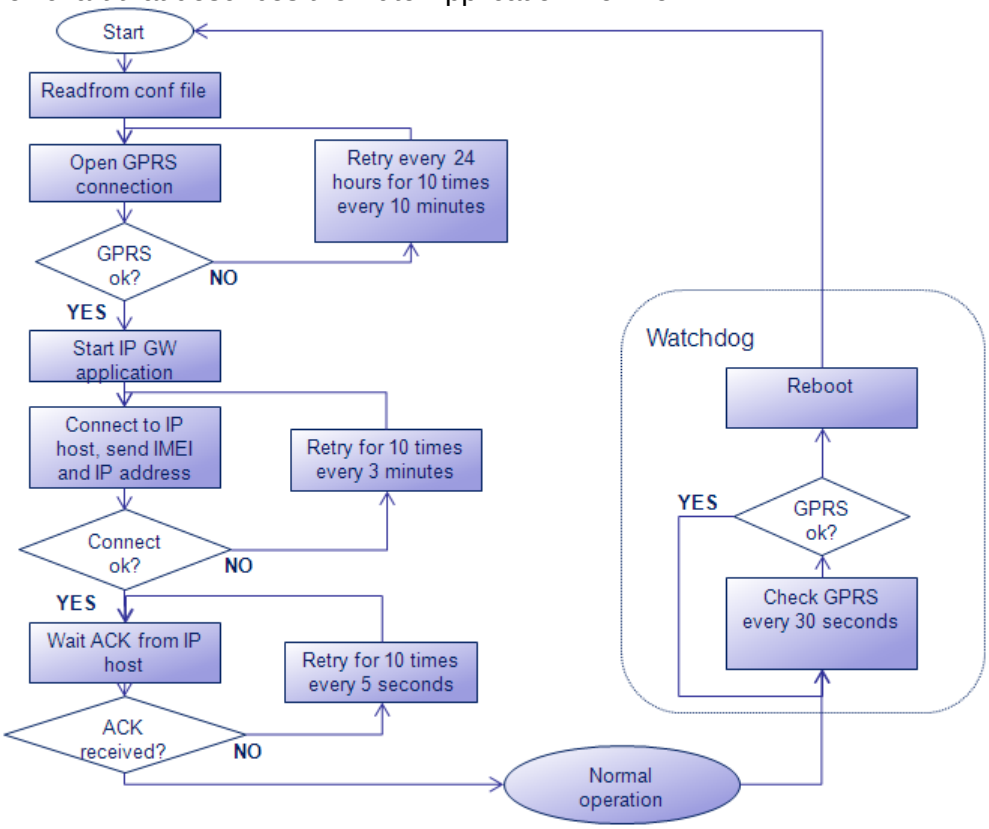


Figure 6.1 - Auto Application flowchart

7 Demo Gateway Application – Remote Configuration

7.1 Demo overview

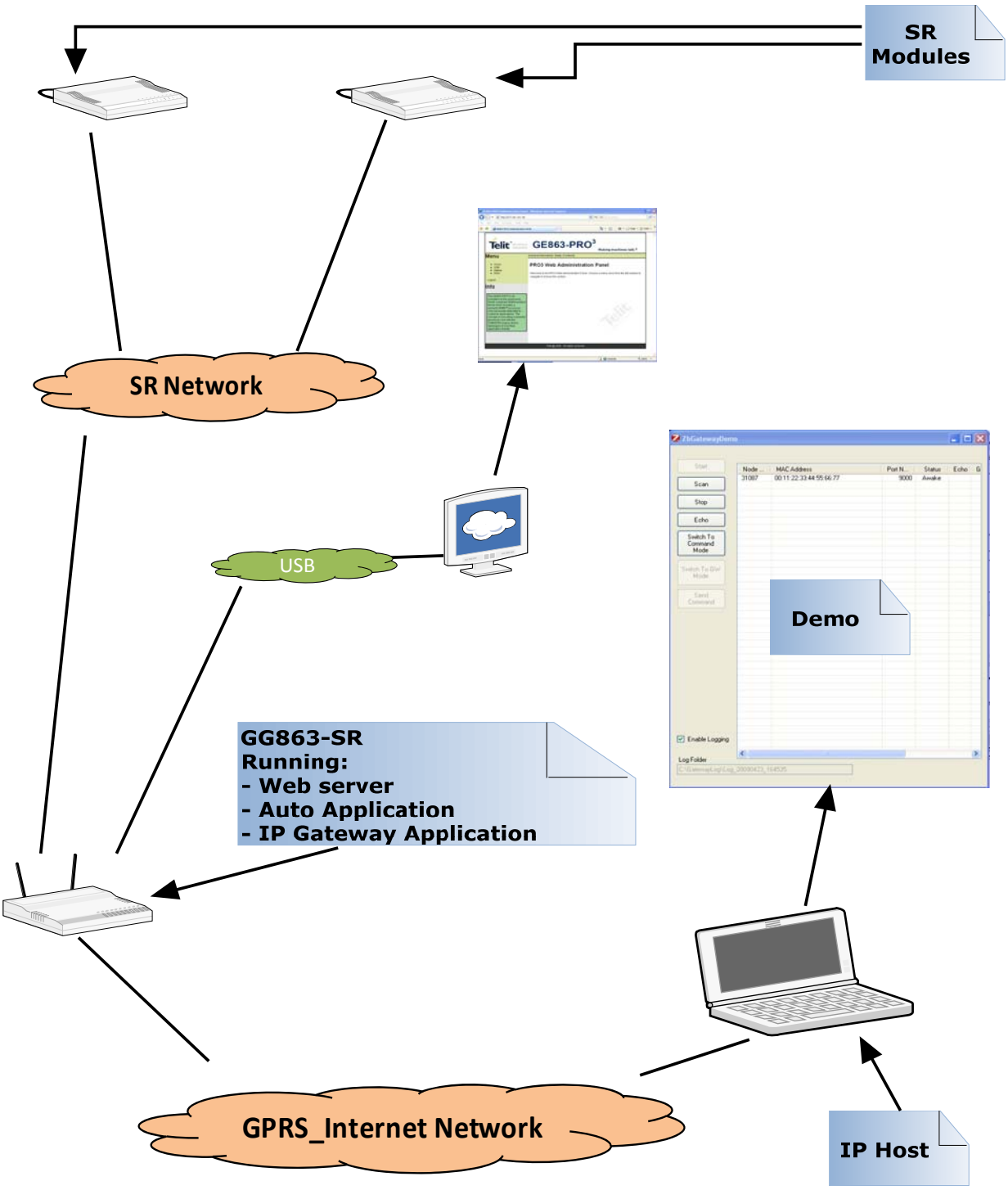
The Telit Demo Gateway is intended as an example showing one of the possible scenarios of gateway application deployments on the **GG863-SR** Telit Terminal.
The Demo Gateway Application shows how an IP Host and a SR node exchange data and how the IP Host can send commands to each SR node and receive responses.
Data exchanged between the IP Host and the SR nodes are completely transparent to the **GG863-SR** Gateway Terminal.

The Demo uses the following components:

- 1 **GG863-SR**
- 1 local PC
- 1 IP Host
- 1 or more SR modules

7.2 Demo Scheme

The following picture shows the components of the Demo and their relationships:



7.3 Demo User Guide

To set up and use the Telit Demo Gateway follow these steps:

1) Setup of the IP Host and get public IP address:

   - ❖ Connect the modem to the IP Host and start the GPRS connection
   - ❖ Get the public IP address, using for example the `ipconfig` command:

     ```
     IP address.............  nnn.nnn.nnn.nnn
     ```

2) Ensure that the SR devices are ready to connect to the network: follow the steps needed to start a SR Network and to associate to it End Devices (for more details see §8.2 for ZigBee Network or §8.3 for MeshLite Network, here the ZigBee example is presented)

   A. Hard/Soft Reset of all SR modules present in the Demo
   B. Choose a valid Pan Id to start or to associate to a network
   C. Choose a valid Channel in which the devices will try to associate
   D. Set Pan Id and Channel on all SR modules present in the Demo

3) Follow the set up phase described in §8.1and be sure that:

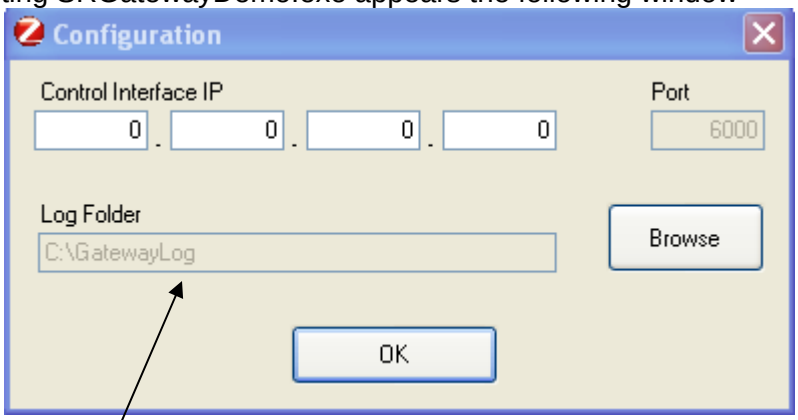
   ShortRange → [Server]:
   - ❖ IP → **IP Host address got in step 1**
   - ❖ Port → **4500**
   - ❖ Connection Type → **TCP**

   Connection:

   - ❖ Connection → REMOTE

4) Go to the IP Host and start the application SRGatewayDemo.exe

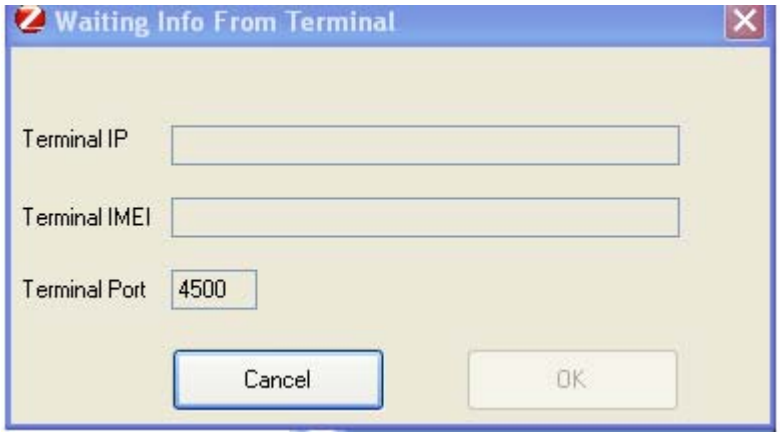
- Starting SRGatewayDemo.exe appears the following window



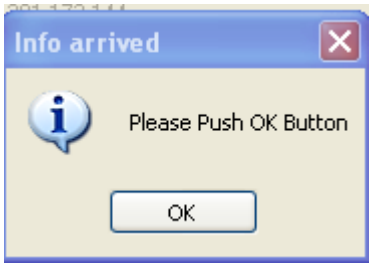
Choose the folder where to store the logs

Not Delete the Log Folder while the SRGatewayDemo.exe is running. If you delete the Log Folder the file logs can not be write unless you create the log folder with the previous name.

- Clicking **Ok,** the IP Host waits for **GG863-SR** IP address and IMEI



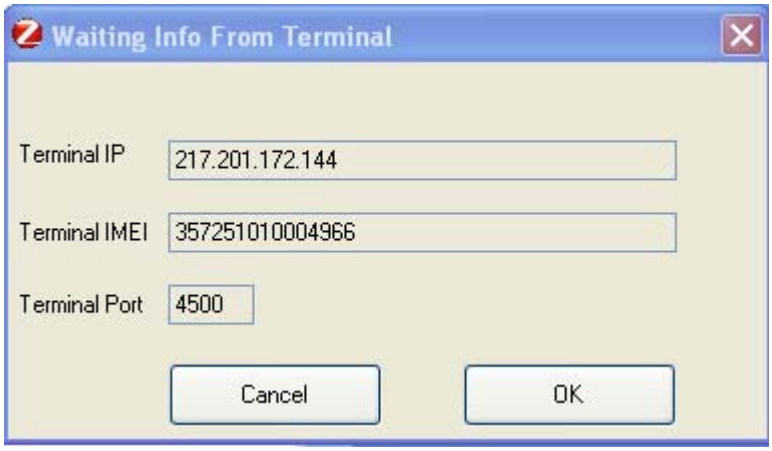
- When appears the following pop-up it means the GPRS connection between **GG863-SR** and IP Host is established.



Click **Ok**

- Now you can see the information received from **GG863-SR**:

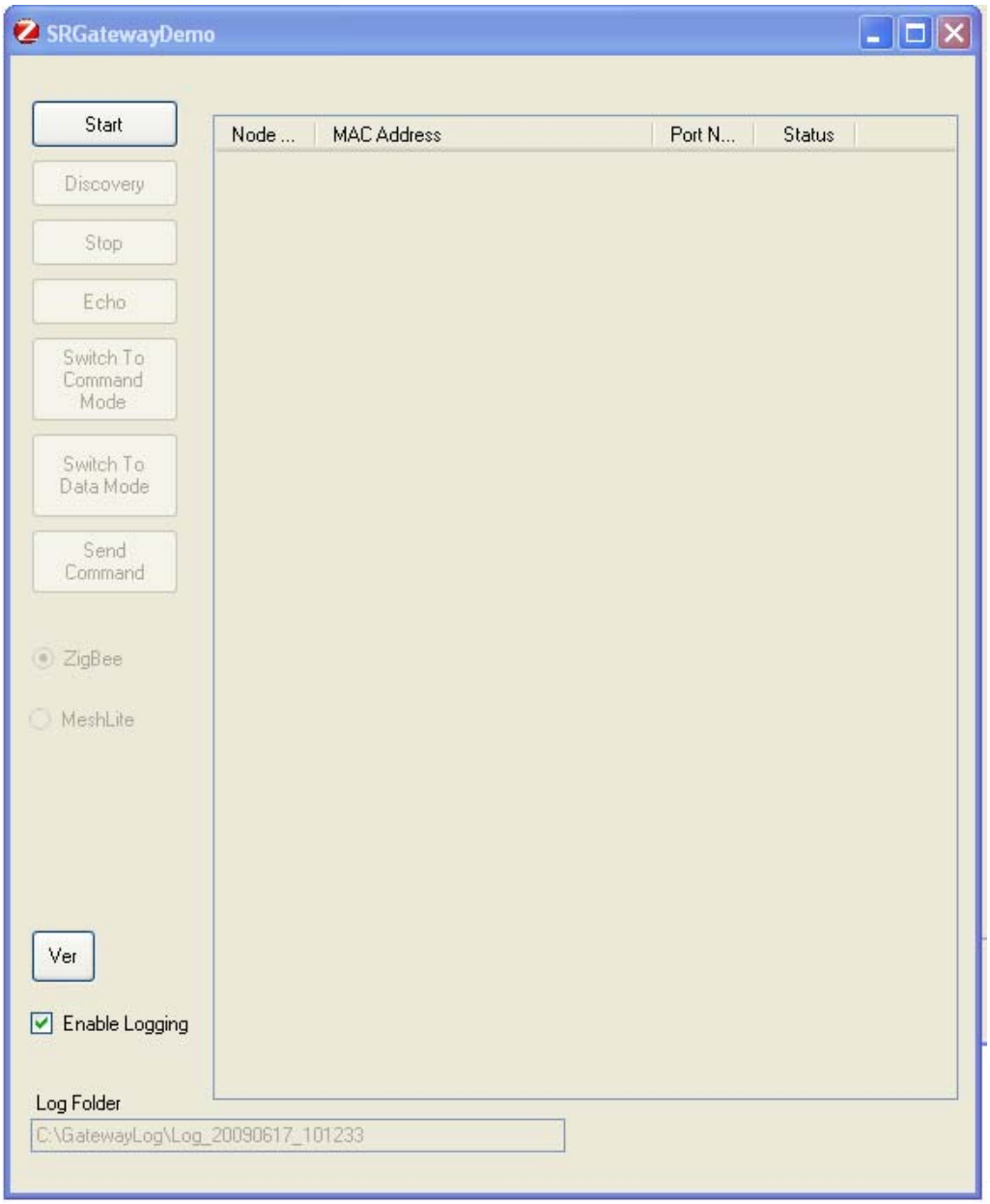
  For example:



- Click **OK**

- At this point the SRGatewayDemo.exe allows to use Gateway functionalities running inside the **GG863-SR**:
- First of all Start the SR Network (the technology radio button checked shows the Short Range Technology equipped in the **GG863-SR**):

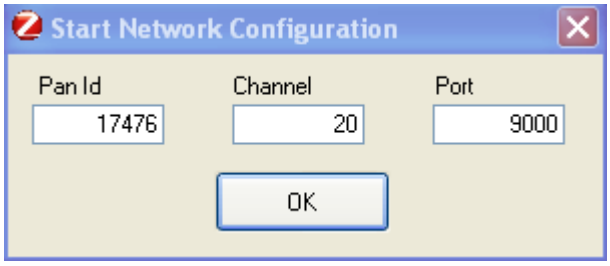
Click **Start**



- The next window shows Pan Id, Channel and Port used by the SRGatewayDemo.exe to Start the SR Network:

– Pan ID → Short Range Network Identifier;
ZigBee: any value from 0 to 65535;
MeshLite: any value from 0 to 255

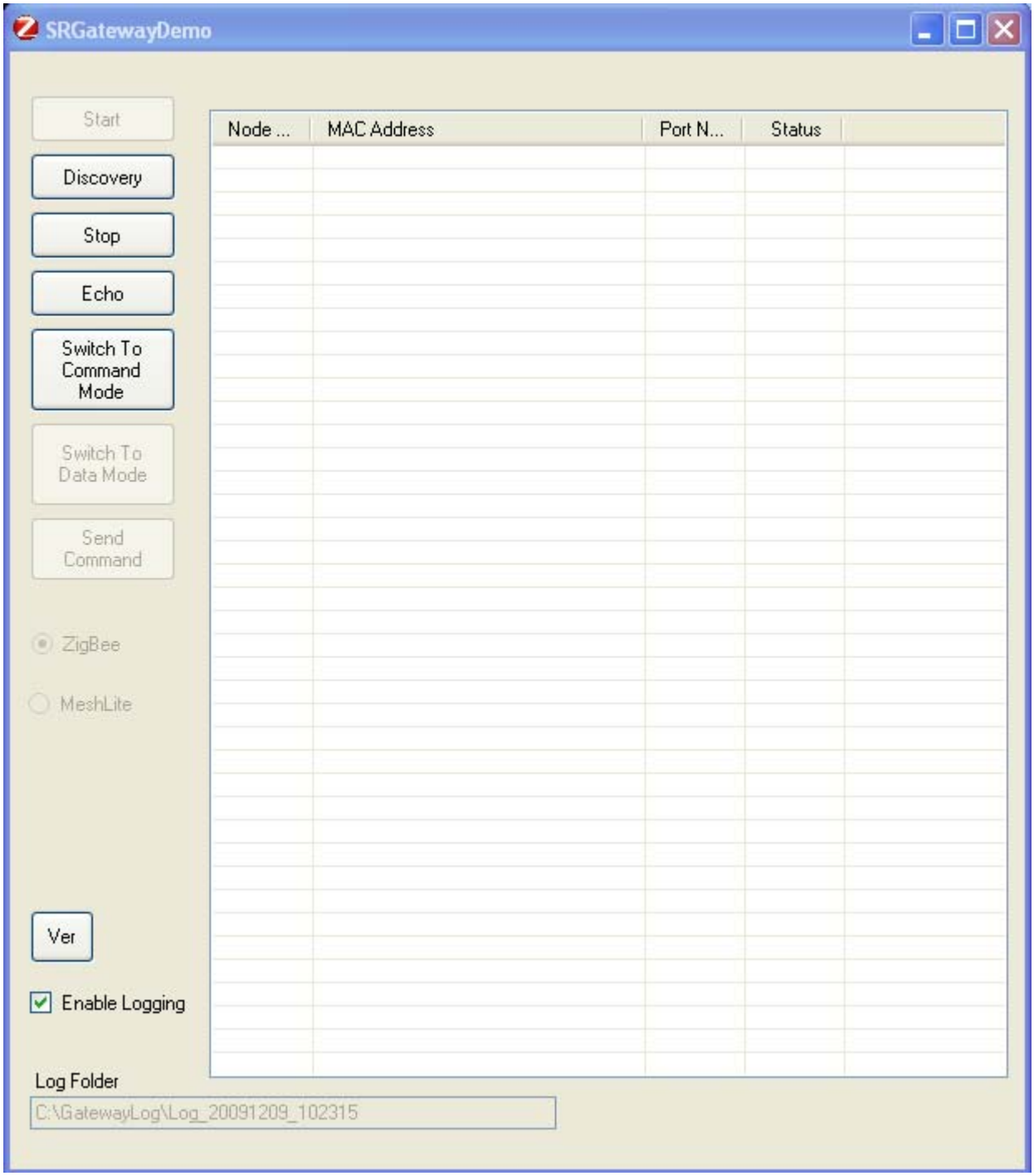
- Channel → RF channel for the SR Network;
  ZigBee: any value from 11 to 26
  MeshLite: the Channel field has not meaning and for this reason disabled
- Port → the first TCP port used in *TCPport–nodeID* mapping
  ZigBee: any value from 6001 to 65535
  MeshLite: any value from 6001 to 65535



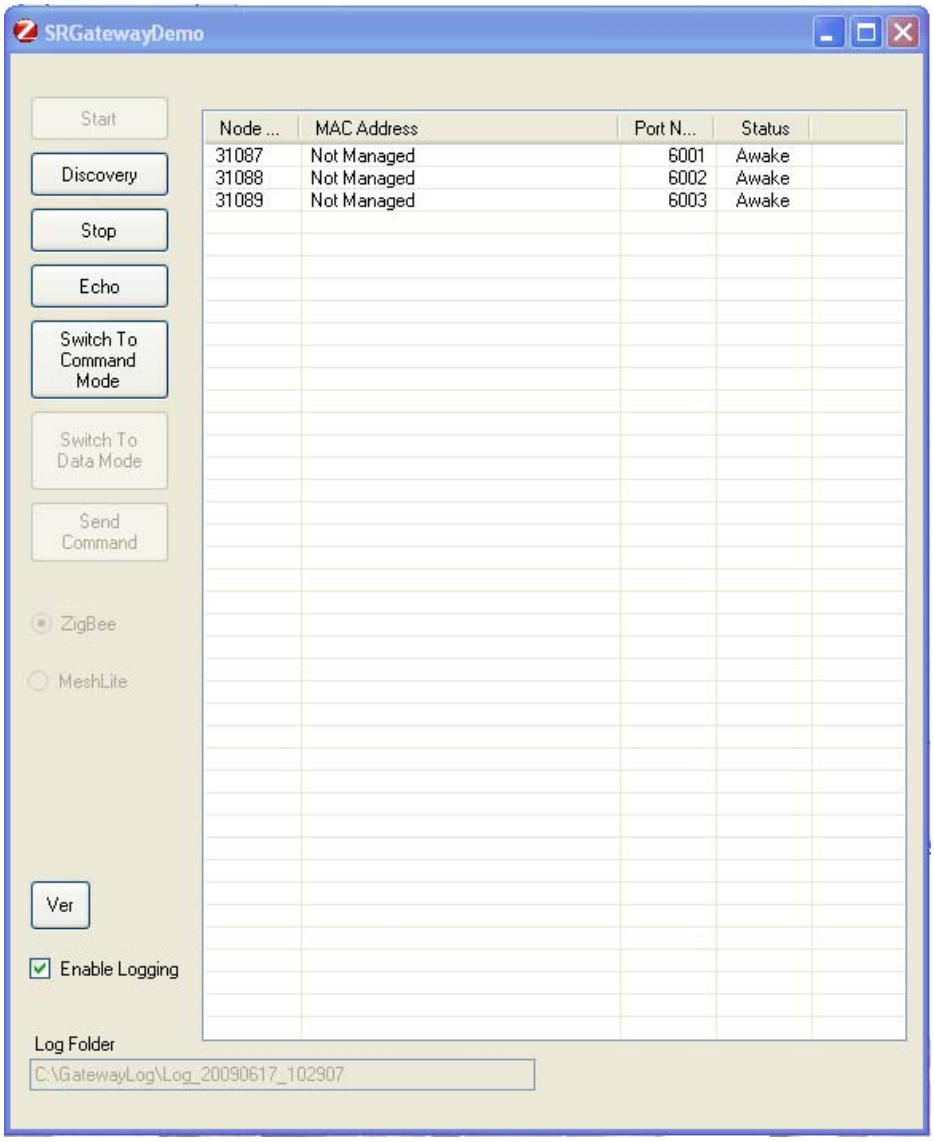
Click **Ok**

- At this point the Demo is using the IP Gateway Application running inside the **GG863-SR** in *normal-mode*, the IP Host does need a minimal knowledge of the Short Range functionalities.

- Now you can discover the nodes associated to the network either before or after the SRGatewayDemo.exe application has been started



- To see the nodes associated to the network (ZigBee or MeshLite) before starting the SRGatewayDemo.exe application click the button **Discovery**:

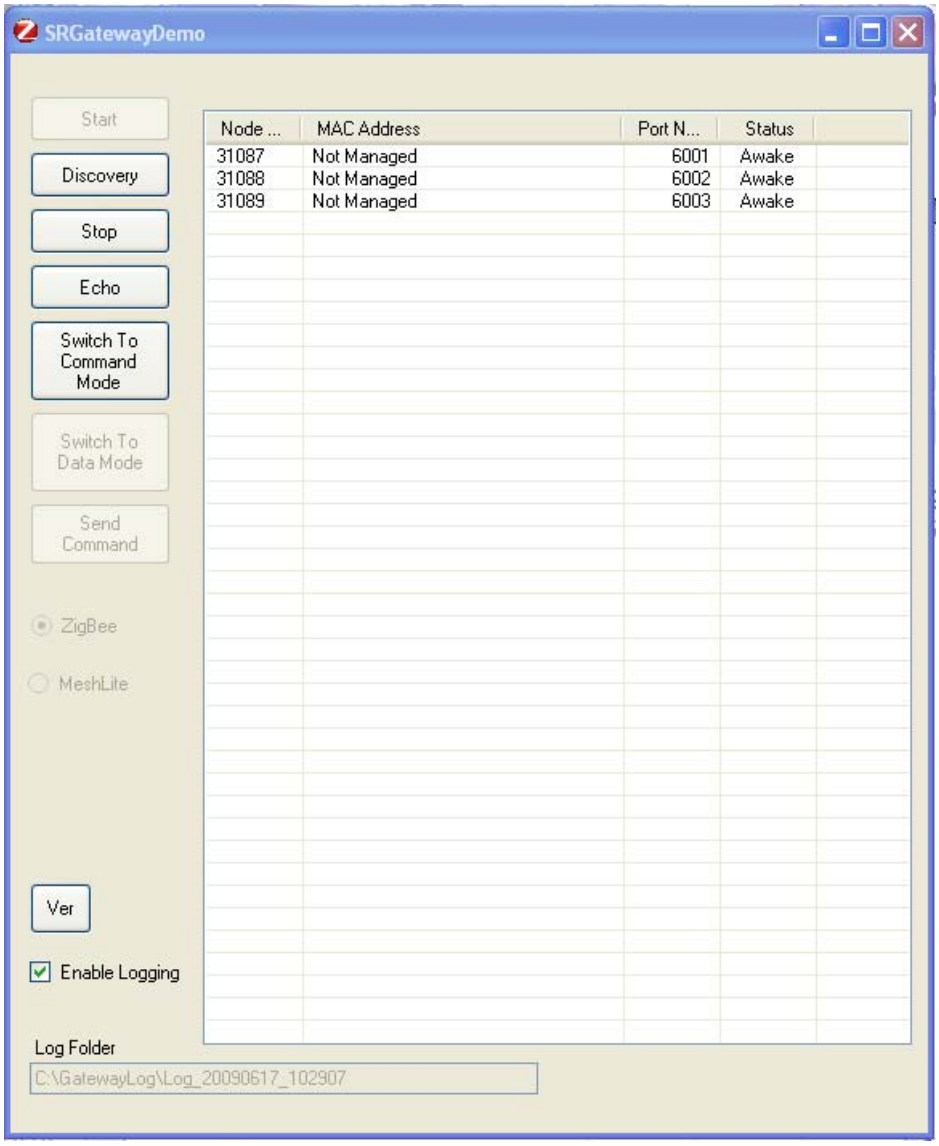


- To see the nodes associated to the ZigBee Network after the SRGatewayDemo.exe application has been started, follow this procedure:
  - ❖ Associate the SR modules to the SR Network by configuring the same channel and PAN ID previously used to start the network (see 8.2 for ZigBee technology)
  - ❖ For any device associated to the network you will see a new Node in the Demo

- To see the nodes associated to the MeshLite Network after the SRGatewayDemo.exe application has been started, follow this procedure:
  - ❖ Associate the SR modules to the SR Network by configuring the same PAN ID previously used to start the network (see 8.23 for MeshLite technology)
  - ❖ Click the button **Discovery** and you will see the new Nodes in the Demo

At any moment you can discover if changes happened on the Network:

Click **Discovery**



Any listed device represents a pipe through send raw-data.
Here it is suggested a scheme for a very simple application that makes use of the pipes.
It is an echoing application: the IP Host sends data to SR nodes which are echoed back by the SR node to the IP Host.

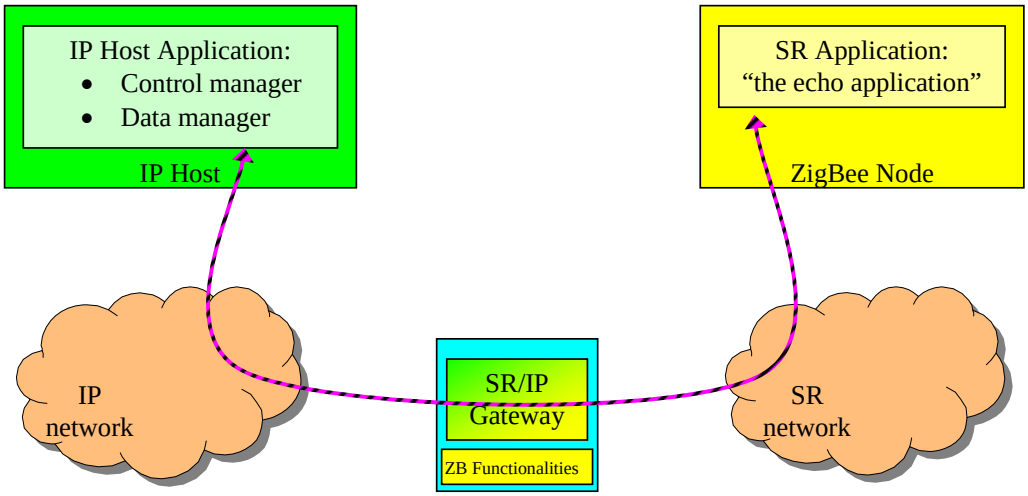


Figure 7.1

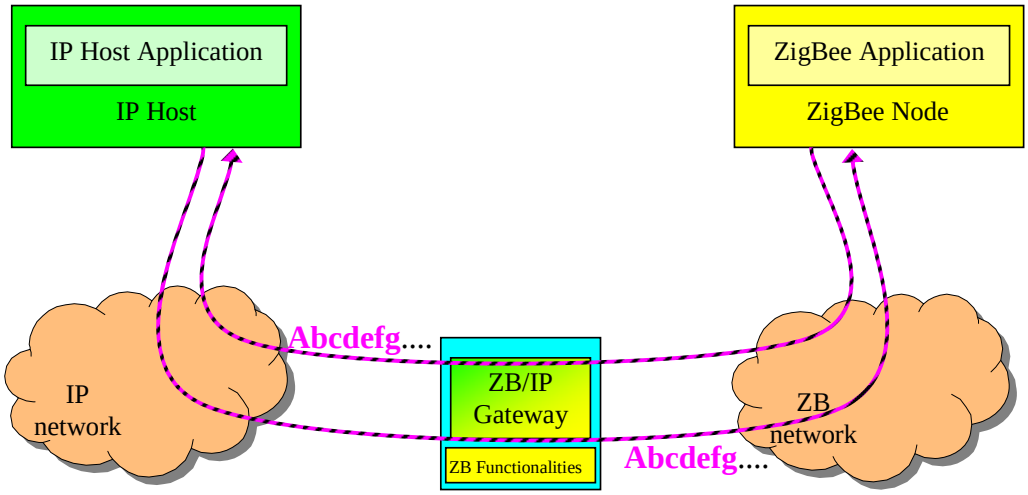


Figure7.2

Realizing this application is very easy if the following points are considered:

ZigBee:

- SR nodes have been load with a firmware version supporting the echo functionality
- The “Echo attribute” in the SR nodes where to use the echoing application has to be set to ON (see Appendix.4 for ZigBee modules)

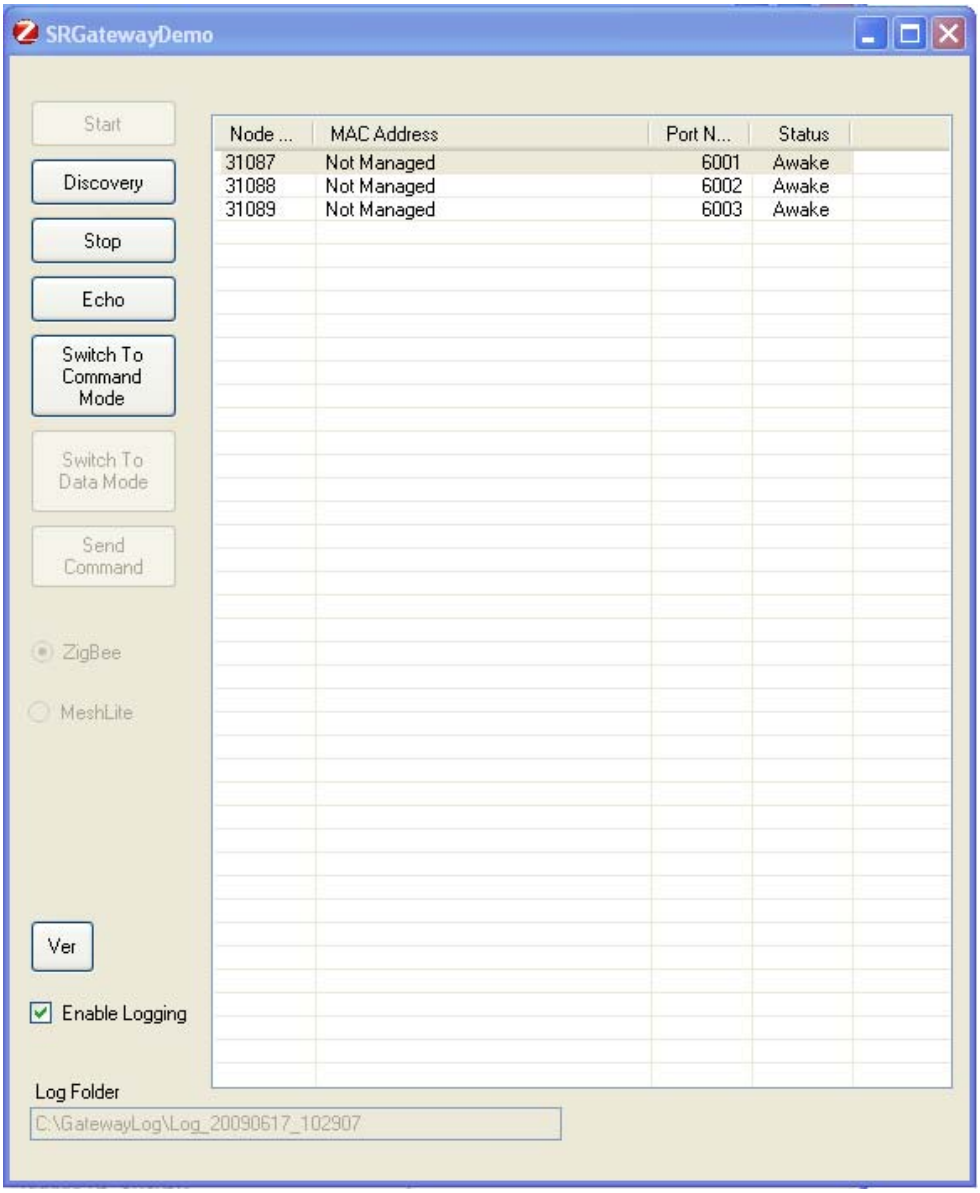
MeshLite:

- Since the MeshLite does not allow any customized software, in order to implement the Echo feature the SR module needs to be connected through the serial port to an external CPU (Pc or embedded system) where the Echo application will be implemented. The external application shall read from serial port the incoming data (maximum 50 bytes) and then it will write all this data on the same serial port. A little example code is reported in Appendix.6 – Code example for Mesh Lite echo implementation

If previous listed points are achieved, the basic steps are:

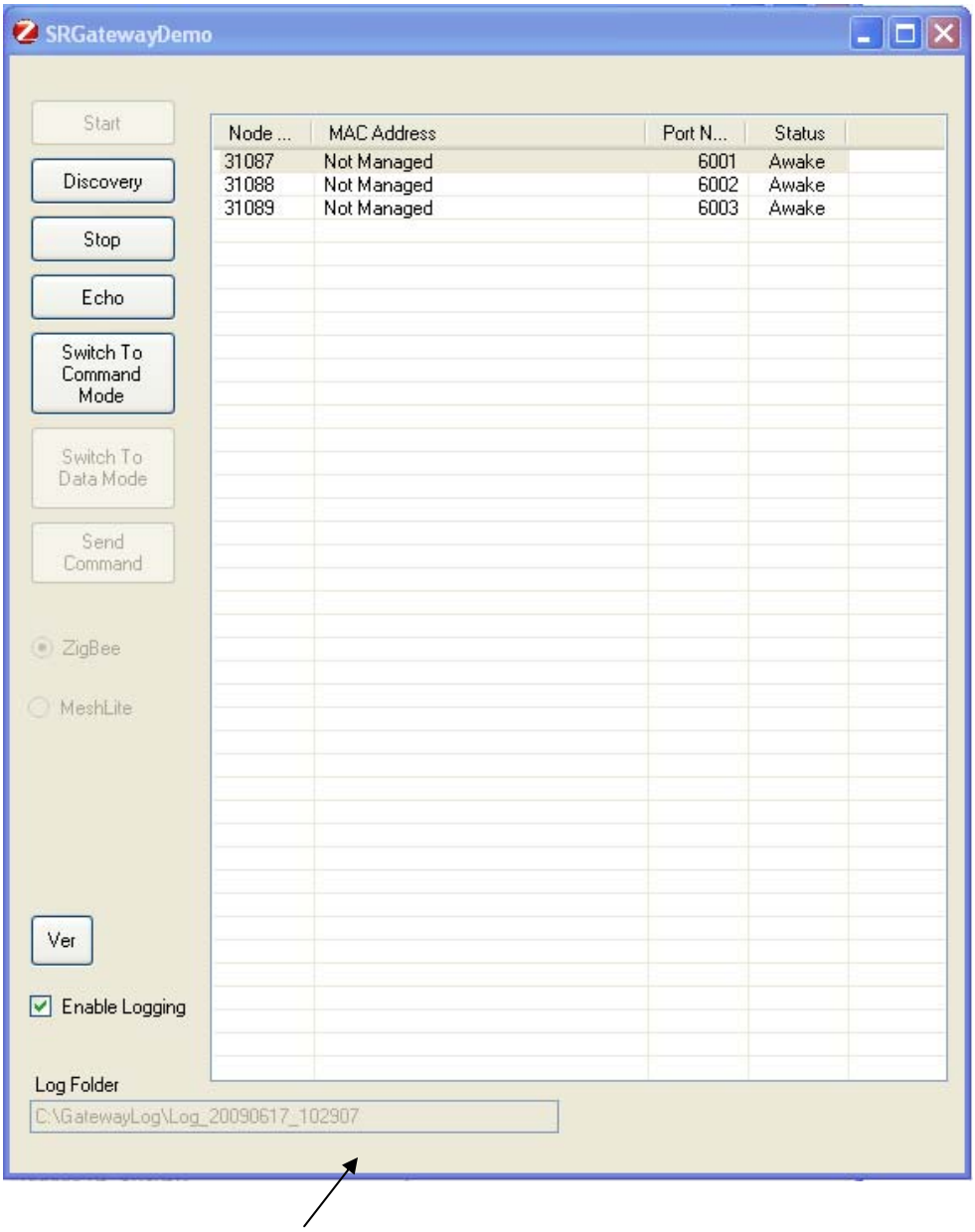
- Turn on the SR nodes (at least the **GG863-SR** and one SR node).
- Add the nodes to the SR network (see Appendix.2 for ZigBee or Appendix.3 for MeshLite)
- If you are using ZigBee technology Set to ON the “Echo attribute” in the SR nodes (see Appendix.4)
- If you are using the MeshLite technology be sure that the external CPU has been connected to the SR Node and the external application is running.

❖ Select the node in the list:



❖ Click **Echo** button

❖

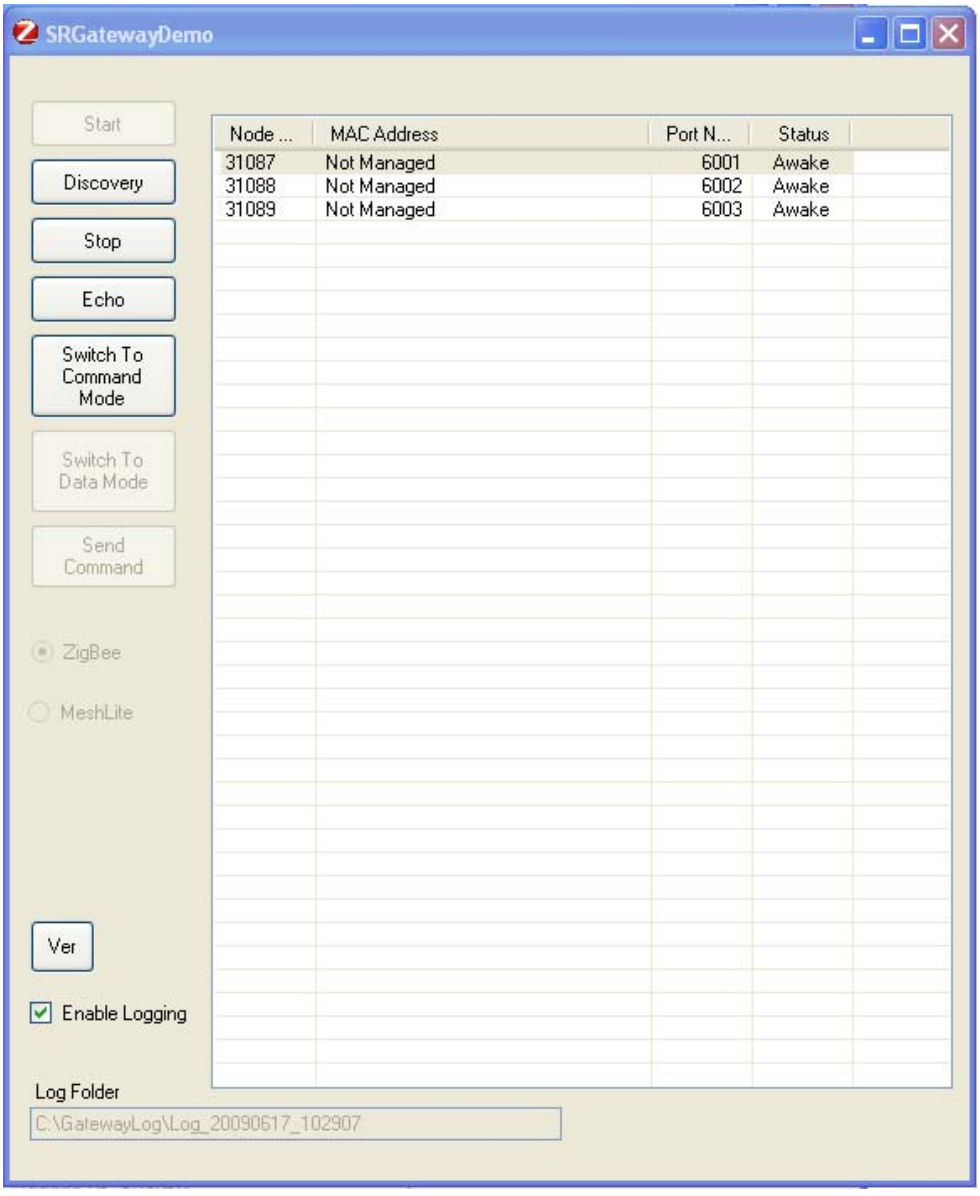


❖ Go to the Log Folder showed in the SRGatewayDemo :

❖ In that folder you will find the LogInput_xx.bin and the LogOutput_xx.bin:
The LogOutput_xx.bin contains the data sent by the IP Host to the SR node and the LogInput_xx.bin contains the data sent by the node to the IP Host. xx stands for the Port Number of the selected Node. If the Echo Application runs correctly, the files will be identical.

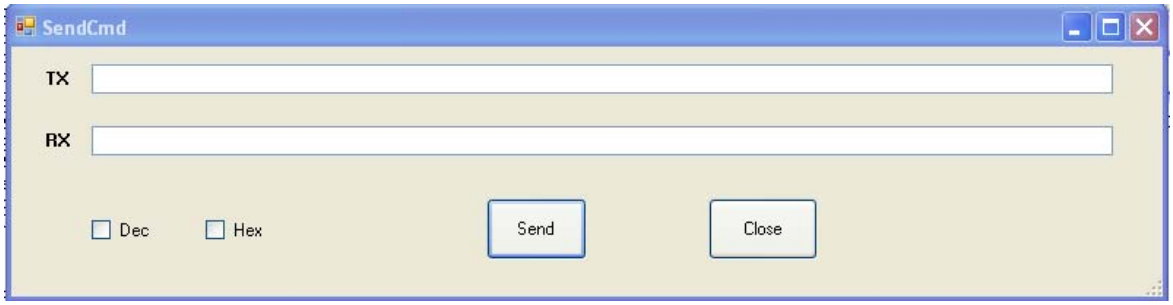
- The Demo allows to use the Command Mode functionality described in § 4.3.2

Click on **Switch To Command Mode**



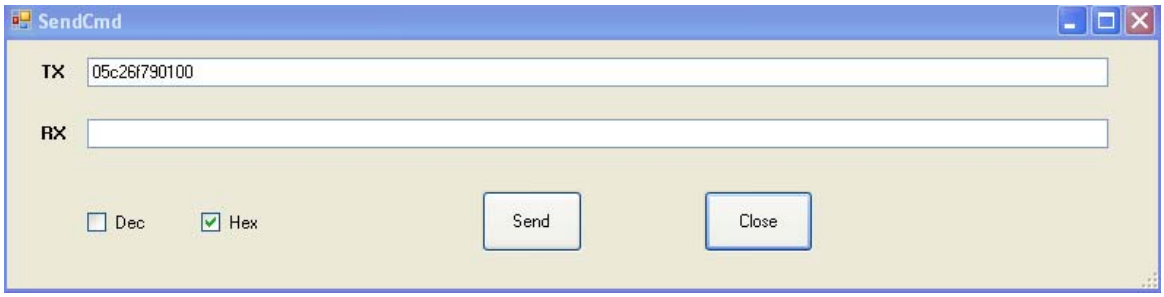
If the command-mode is used, the IP Host could send specific commands to the SR node associated with each data pipe. Messages exchanged through these pipes shall be formatted according to the protocol used by Short Range technology.

Select the device and click on **Send Command**



- Check the format (Dec or Hex) in which write the command and then write a valid command to send to the SR node:

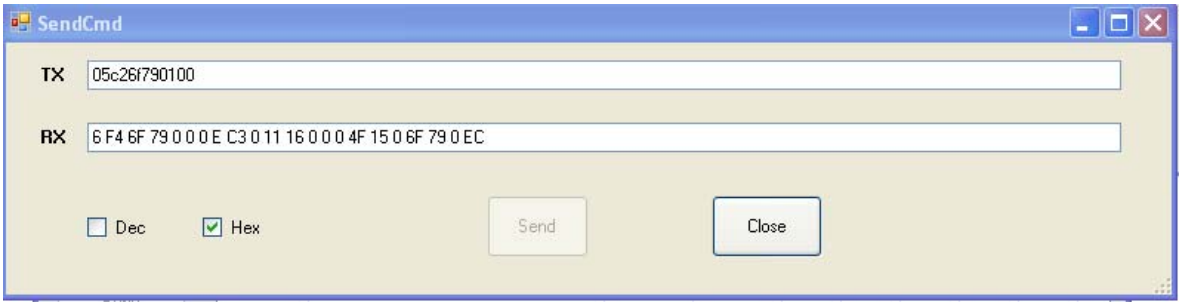
  For example:



  Click **Send**

  - Wait few seconds for the response from the Device:

  For example:

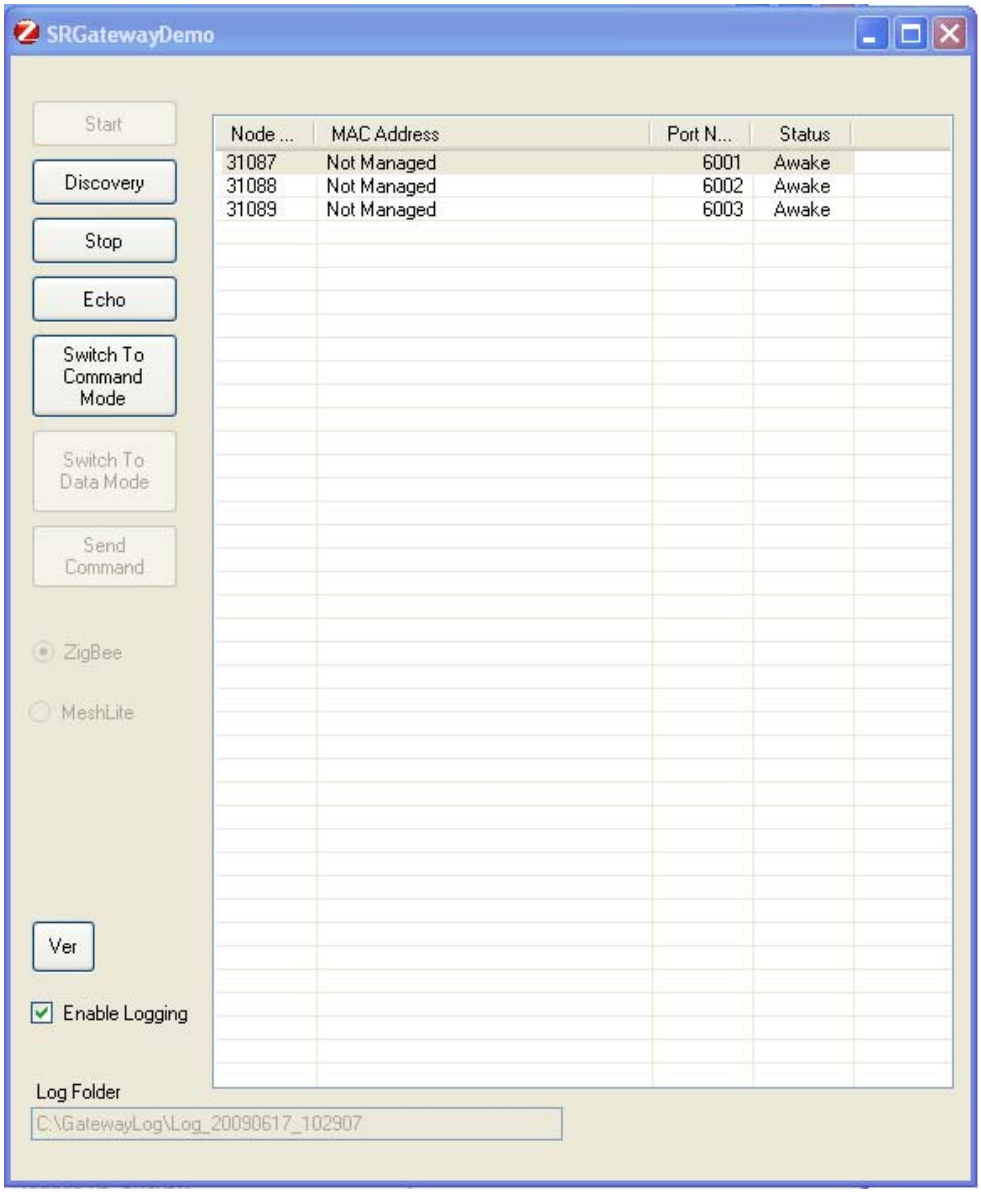


  Note: if after few seconds no response has been received it can be due to one of these reasons:
  - Wrong command sent to SR Node
  - No response expected

  - At any moment is allowed to switch again to Normal Mode:

  If the normal-mode is used, **GG863-SR** is completely transparent managing the pipes, which means that data messages flow in both directions as a raw bit stream.

Click **Switch To Data Mode**

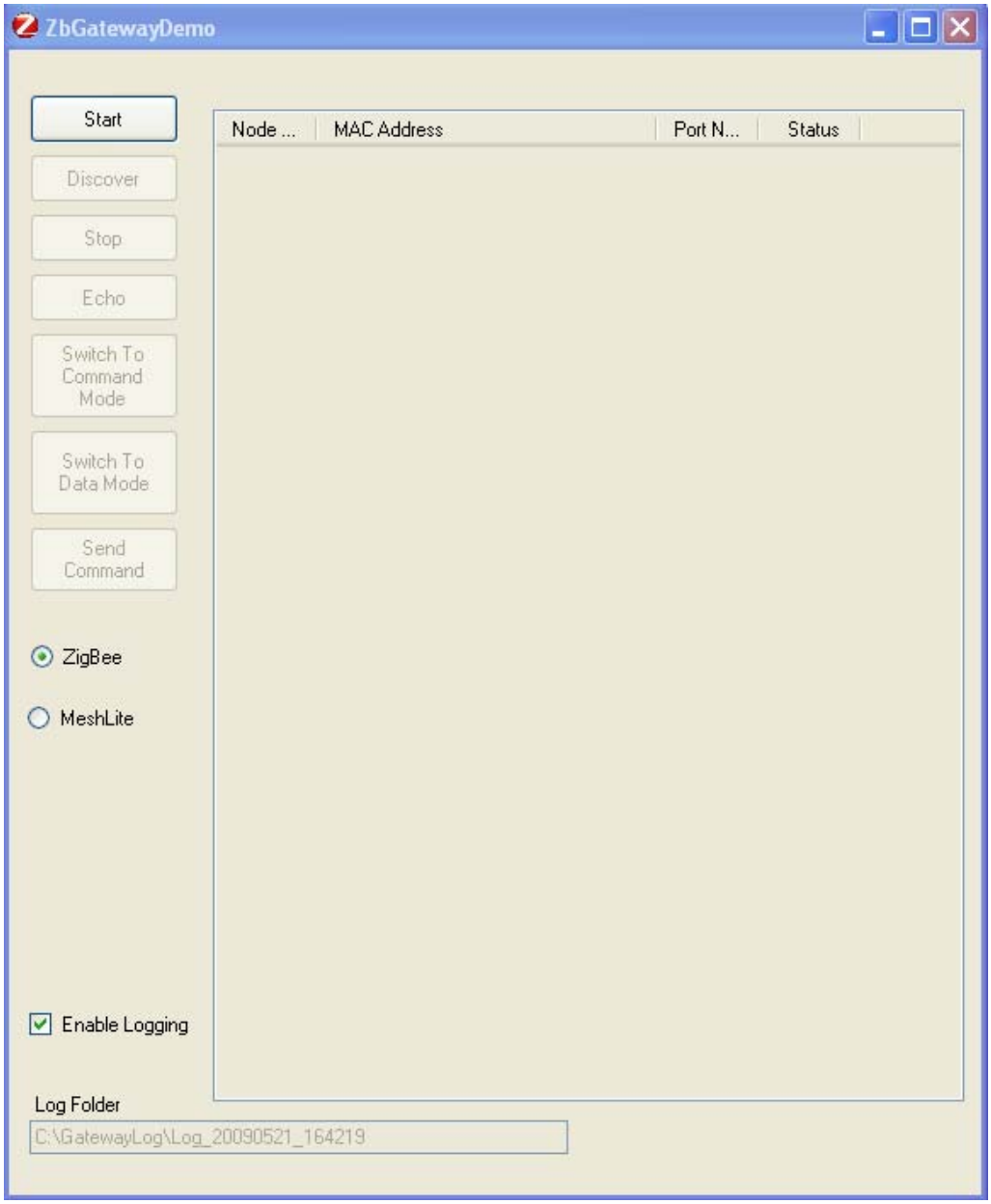


The button **Stop** allows stopping the running SR network only for ZigBee technology. This command shall be issued before creating a new SR Network with a Start Network Command.

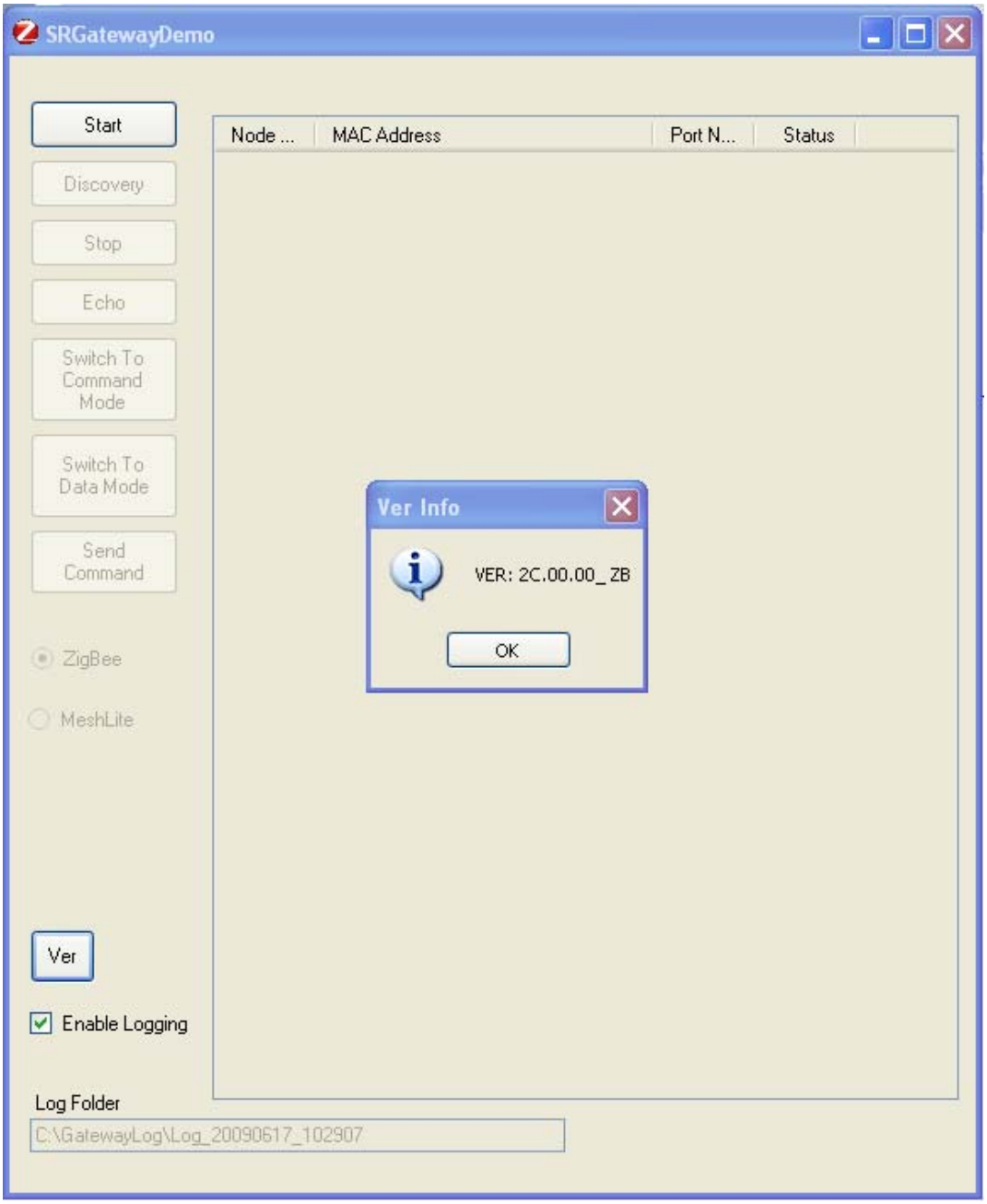
Click **Stop**

This operation requires some seconds. A pop-up shows the Stop Network response.

Now a new SR network can be started.



At any moment you can see the version info about the **GG863-SR** in use:
Click **Ver**


SRGatewayDemo
Start
Discovery
Stop
Echo
Switch To Command Mode
Switch To Data Mode
Send Command
ZigBee
MeshLite
Ver
Enable Logging
Log Folder
C:\GatewayLog\Log_20090617_102907
Node ...
MAC Address
Port N...
Status
Ver Info
VER: 2C.00.00_ZB
OK

8 APPENDIX

8.1 Appendix.1 – Gateway Configuration

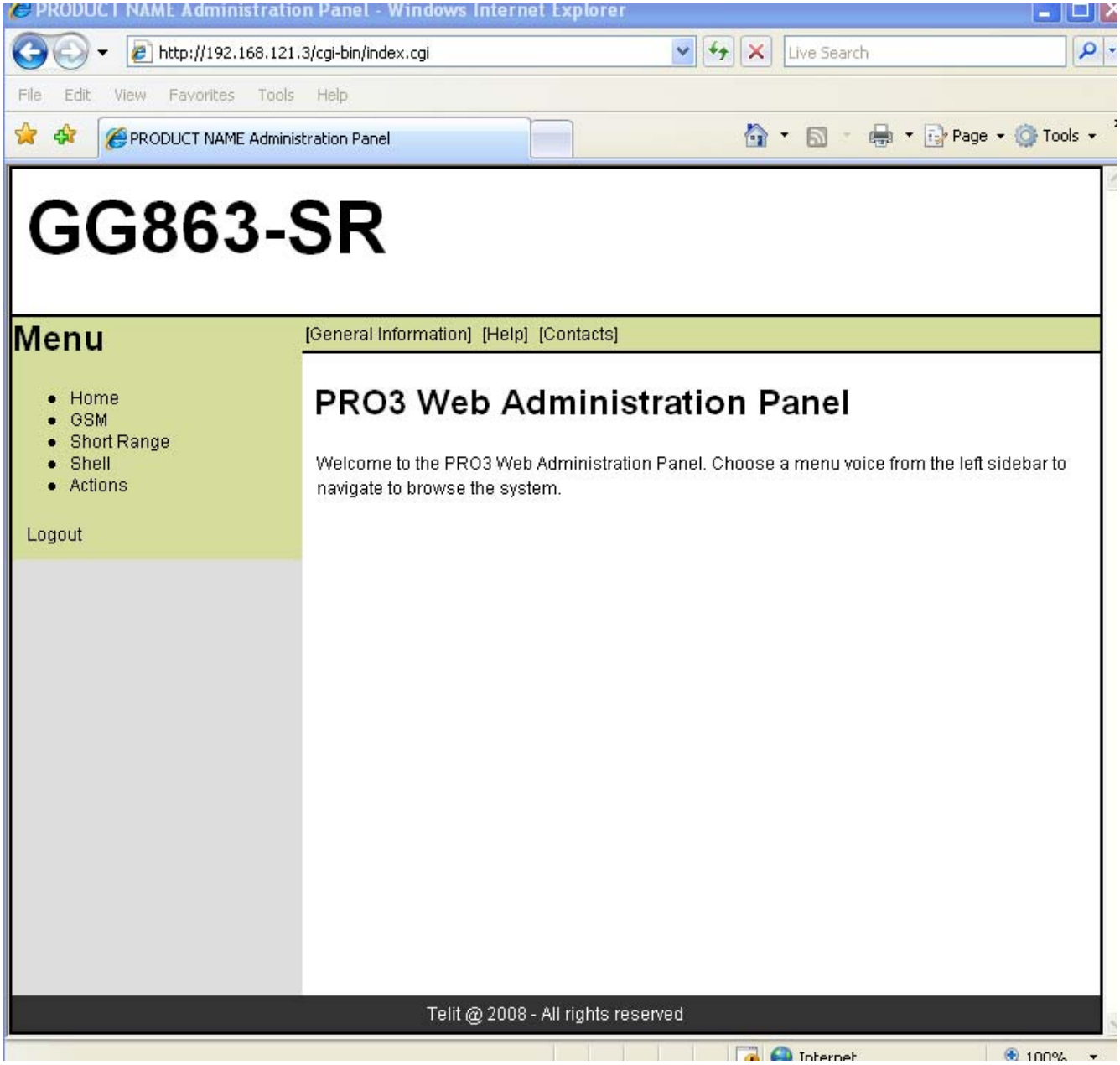
After the boot of the **GG863-SR** the user has to follow this Configuration Gateway phase to set the parameters needed to connect an IP Host to the **GG863-SR**.

- Turn on the **GG863-SR** and wait 20 seconds for the system to boot
- Connect the **GG863-SR** to the local PC
- In the local PC connected to the **GG863-SR** type the address:

  `http://192.168.121.3/cgi-bin/index.cgi`

where `192.168.121.3` is one of the IP addresses of the **GG863-SR.**

You can see the Home Page of the Web Administration Panel:



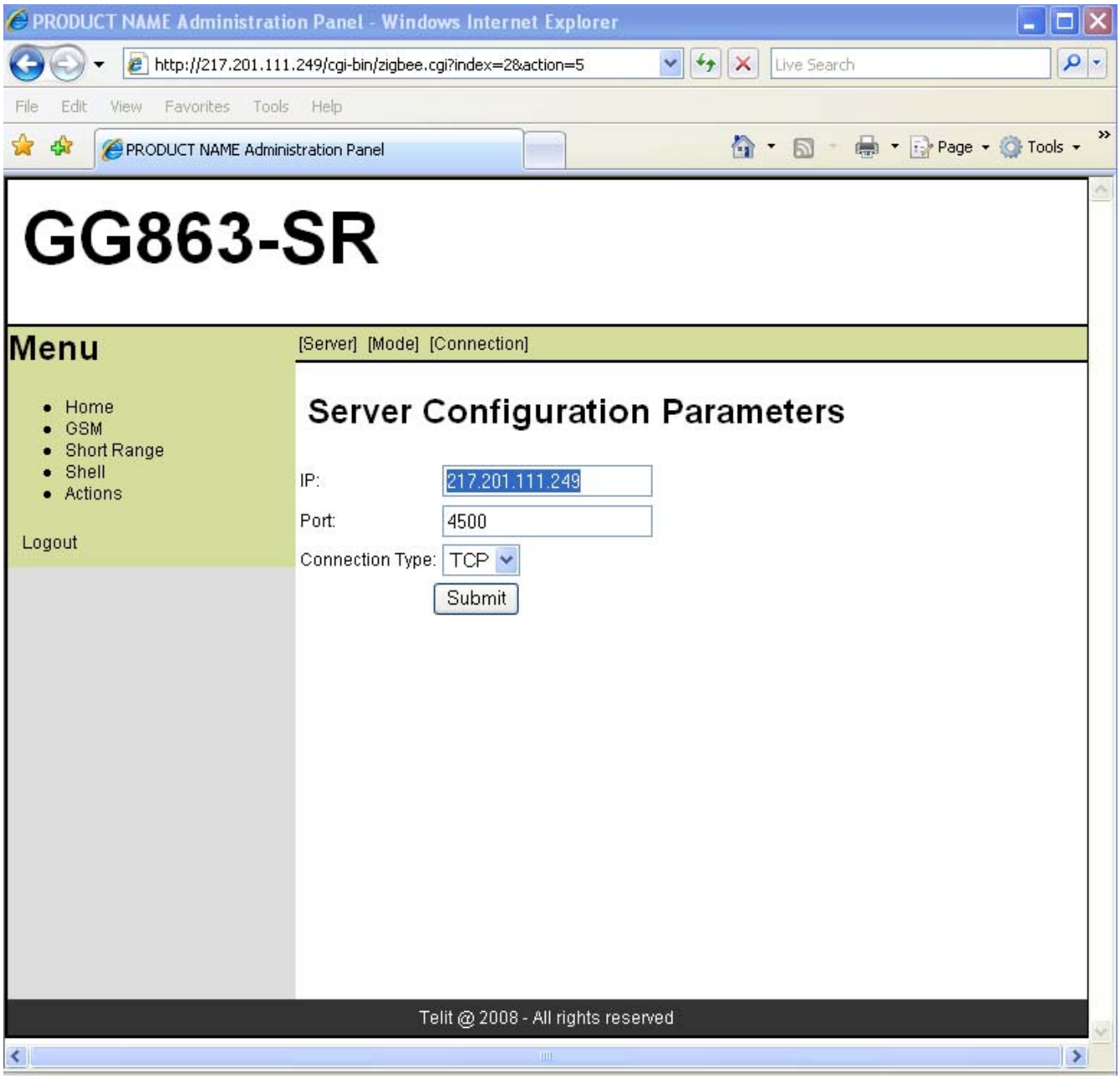
1. Set the IP Host parameters:

   – Click on *Short Range*
   – Click on [Server]

   Fill the form in the following way:

   - IP Host address: *Remote Host IP Address*
   - Port: **4500**

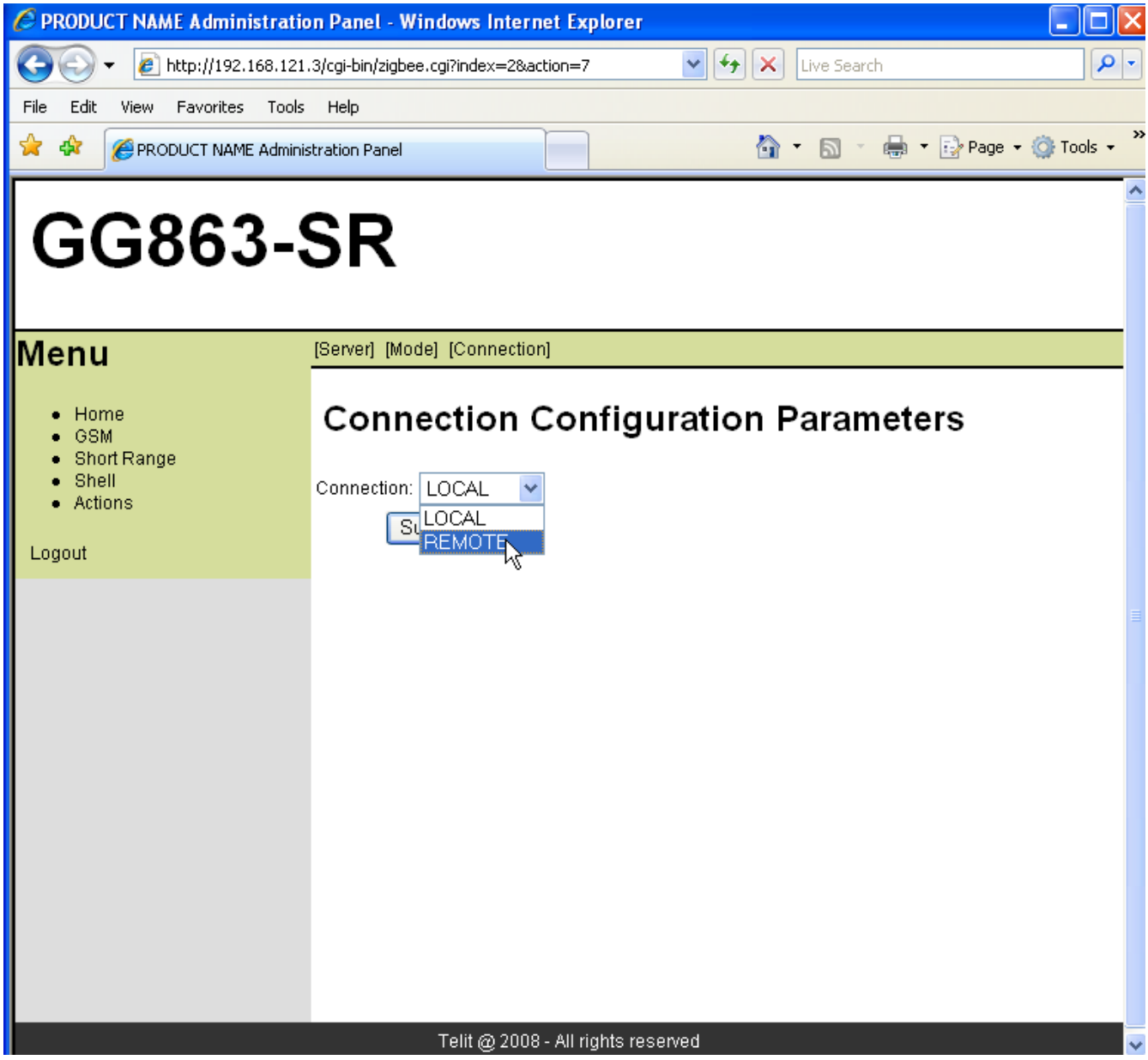
- Connection Type: **TCP**



Click **Submit,** wait until the Web Panel shows the message:

Parameter configuration was successful

2. Click on [Connection]
   - Connection: **REMOTE**

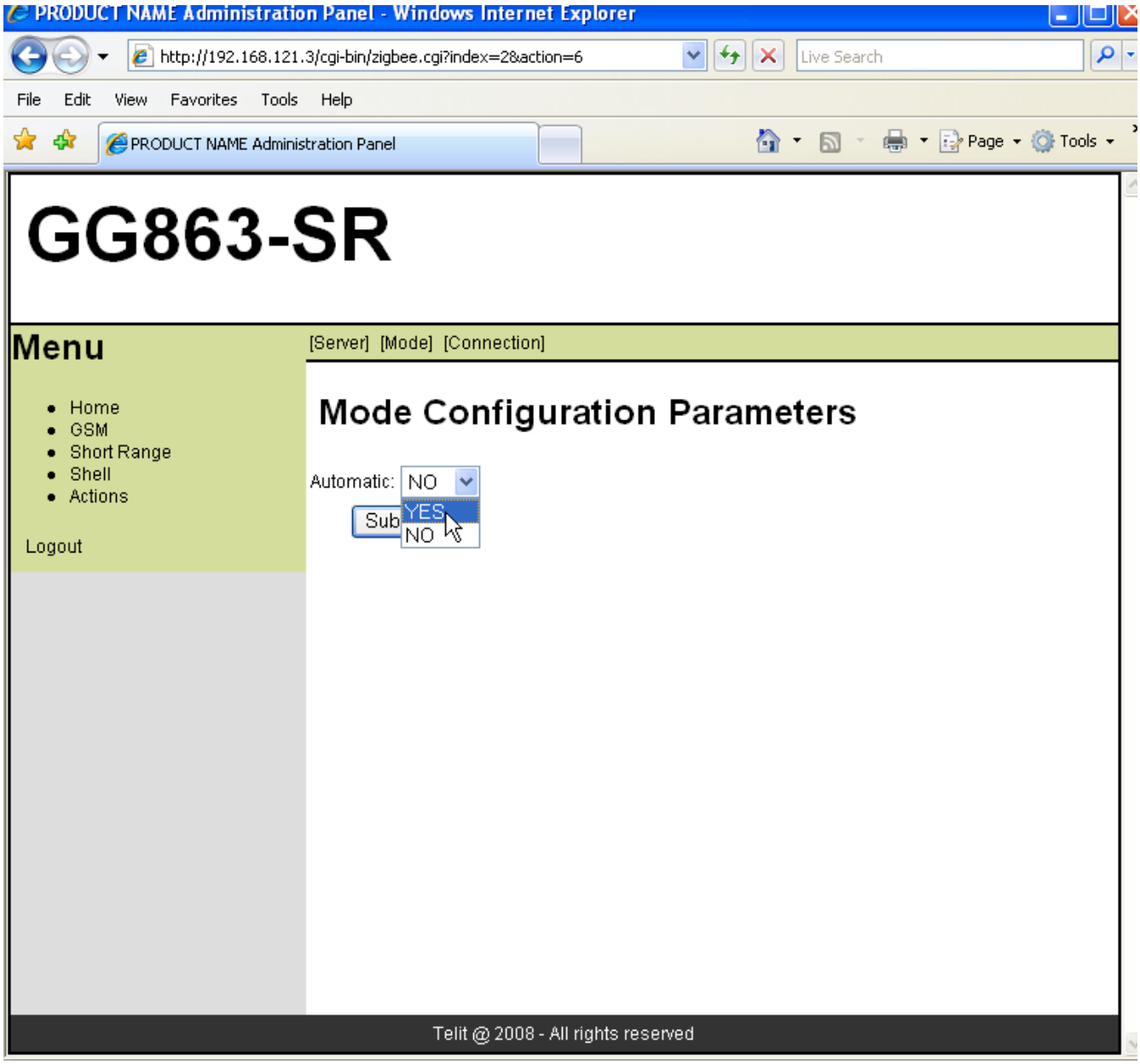


Click **Submit,** wait until the Web Panel shows the message:

Parameter configuration was successful

3. Click on [Mode]

– Automatic: **YES**



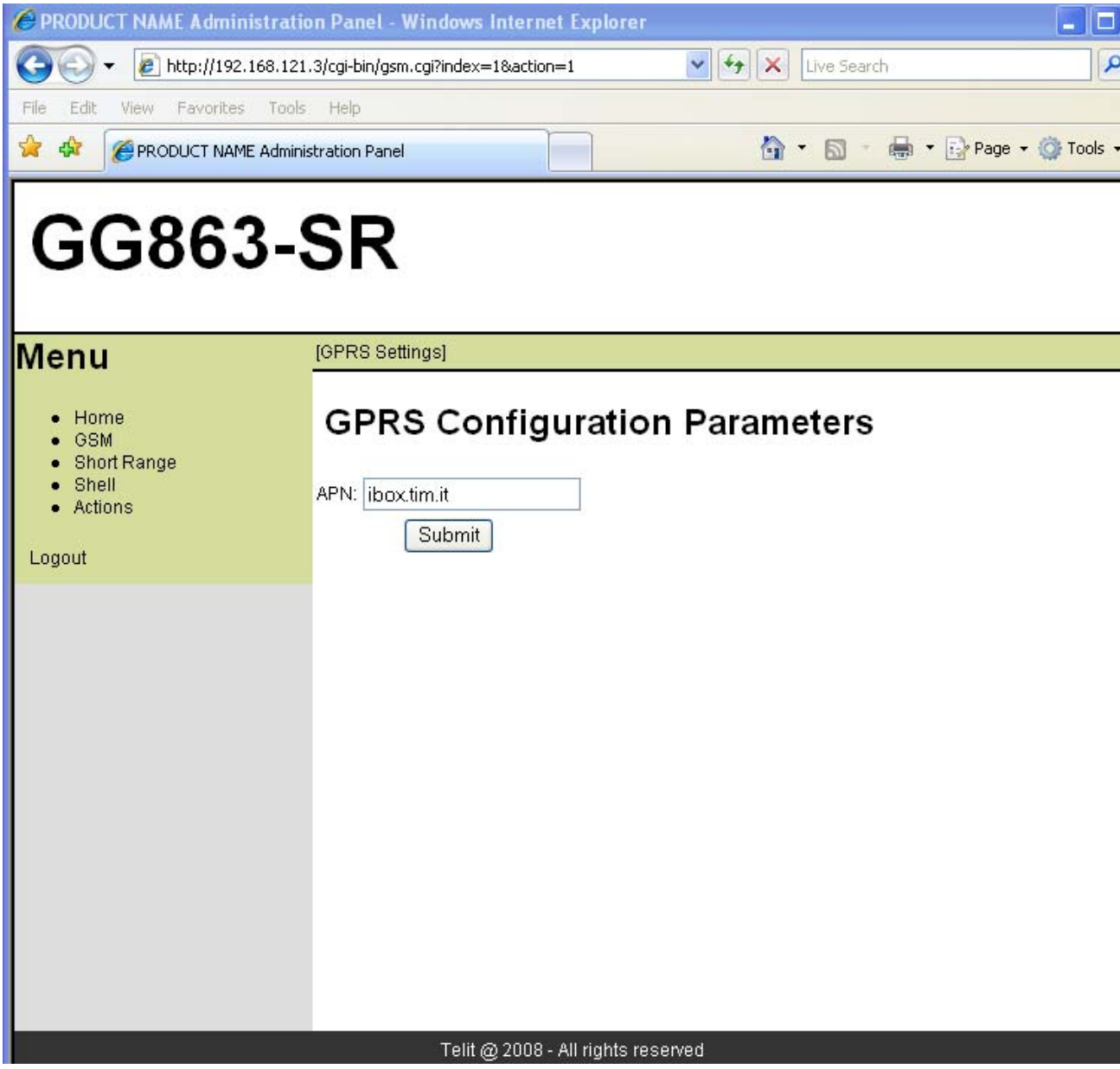
Click **Submit,** wait until the Web Panel shows the message:

Parameter configuration was successful

4. Click on GSM:

– Click on [GPRS Settings]

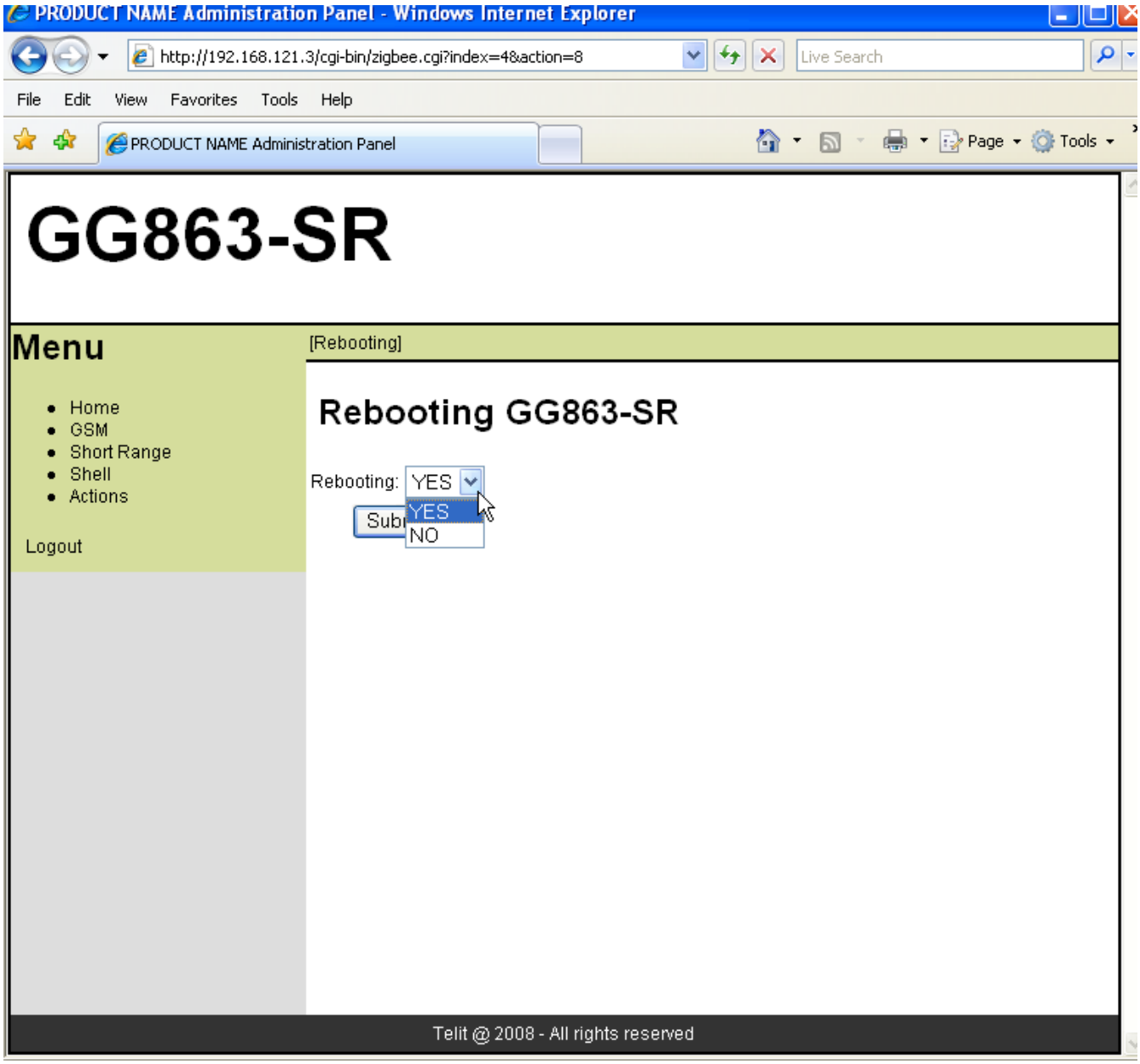
Fill the form with APN



Click **Submit,** wait until the Web Panel shows the message:

Parameter configuration was successful

5. Click on Actions:
   a. Click on [Rebooting]
   b. Rebooting: **YES**



Click **Submit,** wait until the Web Panel shows the message:

Parameter configuration was successful

- Unplug the USB cable from the PC
- Wait 30 seconds
- plug the USB cable into the PC

Note: After the use of the Demo described in this section, if you want to use the Local Configuration described in section 2.5 you have to use the Web Panel to set Connection → LOCAL, to change the Sever IP and reboot the system.

8.2 Appendix.2 – How to add a new Zigbee node

To add a new TelitRF node to the ZigBee network created by the gateway, next steps shall be followed:

1. Switch on the ZigBee node and connect to its serial interface (serial configuration has to be 115200,8,1,N)
2. Enter in command mode issuing the command:
   `2B 2B 2B`
   The device will respond:
   `0D`
3. Send a Hard Reset Command:
   `02 10 00`
   The device will not return any code, but it will exit from command mode.
4. Re-enter in command mode:
   `2B 2B 2B`
   The device will respond:
   `0D`
5. Set the Extended PanID register:
   `0B 12 C4 08 XX XX XX XX XX XX XX XX`
   where `XX XX XX XX XX XX XX XX` is the Extended Pan ID used by the network.
   The device will respond:
   `03 13 00 C4`
6. Set the Radio Channel register:
   `05 12 01 02 YY YY`
   where `YY YY` is the Radio Channel used by the network
   The device will respond:
   `03 13 00 01`
7. Start the network:
   `01 16`
   The device will respond:
   `02 17 00`

After these steps, the ZigBee node will be associated to the existing ZigBee network.
If the IP Gateway Application is running, a Scan Network Responses (unsolicited) will be sent toward the remote IP host and a new device will appear in the main window.
For further details about commands listed above refer to [12] .

8.3 Appendix.3 – How to add a new Mesh Lite node

To add a new TelitRF node to the MeshLite network created by the gateway, next steps shall be followed:

1. Connect the module to the serial port
2. Switch on the module and before 500ms move the SwitchProg in “on“ position
3. Set the Network Id (i.e.: 48) sending on the serial port the string “m00ATS310=48” followed by carriage return
4. If there are no problem the module replays with “OK” string to the previous command
5. If the user needs other configurations he can send other commands across the serial connection
6. Switch off the module and move the SwitchProg in “off” position
7. Switch on the module

For more details about the usage of Telit TinyOne modules refer to [5].

8.4 Appendix.4 – How to enable the “Echo” feature on Zigbee

Zigbee nods have the “Echo” feature disabled by default. In order to enable the “Echo” feature the following steps shall be followed:

- To set the “Echo attribute” to ON in the ZigBee node send the following command (no responses will be send back):
  `10 F3 XX XX 31 31 02 00 06 00 01 05 04 00 01 00 00`
  Where:
  XX XX = Dst Addr in LSB

- To Read the “Echo attribute” set in the SR node send the following command to the SR node:
  `0F F3 XX XX 31 31 02 00 05 00 01 00 04 00 00 00`
  where:
  XX XX = Dst Addr in LSB

- The previous command will return the following message:
  `12 F5 31 XX XX 31 02 00 08 08 01 01 04 00 00 10 YY 00`
  where:
  XX XX = Src Addr in LSB;
  YY = 00 → Echo Off
  01 → Echo ON

Where Dst Addr is the network address of the ZigBee node to enable for “Echo”.

Each command can be sent from the serial port of a generic node associated to the ZigBee network; the “Echo” feature set-up will be lost at the power-off, hence it should be re-enabled at the next power-on, if required.

8.5 Appendix.5 – How to send and receive raw data using MeshLite technology

As reported in [5], it is not possible to add any customized software on the Mesh Lite module. The only way to use these modules is to connect another “external CPU” to the serial port and then implement a custom protocol using Mesh Lite serial protocol features. As written in previous chapters, the Mesh Lite serial protocol has the following format:

Byte Header	LSB Address	MSB address	PAYLOAD	CR

In this format there is not any information about packet length or CRC, than the only way to recognize the end of packet is to wait for a carriage return character. For this reason the user shall not insert the '0x0D' as data into the data packet; if it will be necessary he shall implement and use a bit stuffing/destuffing algorithm to hide the '0x0D' character into the data stream packet, both when he sends data from gateway to end device and when he sends data from end device to gateway.

Due to Mesh Lite firmware some raccomandations are suggested to avoid loss of data:

- packet data maximum length → 250 bytes.
  To send more than 250 bytes it is recommended to split them in more packets of maximum 250 bytes size.
- Minimun delay between packets → 100ms.

Starting from these considerations, the user shall differentiate between the communication in normal-mode and command-mode.

8.5.1 Normal-mode

When the gateway is set in normal-mode, if the user want to send data to an end device he shall write on the socket only the data he want to send (using stuffing algorithm to hide 0x0D character). The carriage return at the end of data packet will be added by the gateway application. On the end device the following data stream will arrive:
0x65 | 0xFF | 0xFF | USER STREAM | 0x0D

- 0x65 is the Data type packet
- 0xFF and 0xFF are the two address bytes of the sender (coordinator)
- 0x0D is the end of packet byte

If the user software has to reply or send data to the gateway, it shall encapsulate the answer into a Mesh Lite data packet (using stuffing algorithm to hide 0x0D character) in the following way:
0x65 | 0xFF | 0xFF | DATA_FOR_COORDINATOR | 0x0D

- 0x65 is the Data type packet
- 0xFF and 0xFF are the two address bytes of the recipient (coordinator)
- 0x0D is the end of packet byte

8.5.2 Command-mode

When the user wants to use the gateway in command-mode the rules of data sending from end device to the coordinator are the same of the normal-mode. The communication changes when the data are sent by the user from the gateway to the end devices. In this working mode the gateway application does not apply any control on the data stream sent by the user across the socket. The user shall send on the socket a complete Mesh Lite packet; his task will be to encapsulate data into a Mesh Lite packet and if it will be necessary he will use a stuffing algorithm.

8.6 Appendix.6 – Code example for Mesh Lite echo implementation

In the following lines is reported a code example used on a Linux platform connected to a Mesh Lite module.

```
#define SERIAL_NAME "/dev/ttyS1"
#define BUFFER_SIZE 60

static int i32OpenSerial(char *port)
{
  /* Function for opening serial port */
  int fd;
  struct termios options;
  fd = open(port, O_RDWR | O_NOCTTY);

  if (fd < 0)
  {
    return -1;
  }
  else
  {
    fcntl(fd, F_SETFL, 0);
    tcgetattr(fd, &options);

    /* Set Baud Rate */
    cfsetispeed( &options, B19200);
    cfsetospeed( &options, B19200);

    // Set the Character size
    options.c_cflag &= ~CSIZE; /* Mask the character size bits */
    options.c_cflag |= CS8; /* Select 8 data bits */
```


```
    // Set parity - No Parity (8N1)
    options.c_cflag &= ~PARENB;
    options.c_cflag &= ~CSTOPB;
    options.c_cflag &= ~CSIZE;
    options.c_cflag |= CS8;

    // Disable Hardware flowcontrol
    //options.c_cflag &= ~CRTSCTS; // if supported
    //Enable HW FlowControl
    options.c_cflag |= CRTSCTS; // if supported

    //Enable Raw Input
    options.c_lflag &= ~(ICANON | ECHO | ECHOE | ISIG);

    //Disable Software Flow control
    options.c_iflag &= ~(IXON | IXOFF | IXANY);

    //Disable newline/carriage return conversion

    options.c_iflag &= ~(ICRNL | INLCR | IGNCR);

    //Chose raw (not processed) output

    options.c_oflag &= ~OPOST;

    //Opzioni aggiunte per far funzionare la select() !!!
    options.c_iflag |= IGNBRK;
    options.c_cflag |= (CLOCAL | CREAD | CS8 | HUPCL);
    options.c_cflag |= B19200;
    //Opzioni aggiunte per far funzionare la select() !!!

    tcflush(fd, TCIOFLUSH);

    if (tcsetattr(fd, TCSANOW, &options) == -1)
      printf ("Error with tcsetattr = %s\n", strerror ( errno ) );

    return (fd);
  }
}

int SerialInit(int *pi32Fd, char *pu8Dev)
{
  unsigned char u8single = 0x00;
  unsigned char u8Ret;
  if (pi32Fd == NULL)
  {
    return -1;
  }

  *pi32Fd = i32OpenSerial(pu8Dev);

  if (*pi32Fd <= 0)
  {
    return -1;
```


```
  }

  /*Workaround Problem on Mesh Module Serial*/
  u8Ret = write(*pi32Fd, &u8single, 1);

  tcflush(*pi32Fd, TCIOFLUSH);
  return 0;
}

int i32ReceiveData(int i32Fd, unsigned char *pu8Buffer, unsigned int
u32Length)
{
  int i32ByteRcv = 0;
  fd_set rfds;
  struct timeval tv;
  tv.tv_sec = 1;
  tv.tv_usec = 0;
  int sel = 0;

  FD_ZERO(&rfds);
  FD_SET(i32Fd, &rfds);

  sel = select((i32Fd + 1), &rfds, NULL, NULL, &tv);

  if(sel > 0)
  {
    if((FD_ISSET(i32Fd, &rfds)))
    {
      i32ByteRcv = read(i32Fd, pu8Buffer, u32Length);
    }
  }

  return i32ByteRcv;
}

int i32SendData(int i32Fd, unsigned char *pu8Buffer, unsigned int u32Length)
{
  int i32ByteSent;

  i32ByteSent = write (i32Fd, pu8Buffer, u32Length);

  return i32ByteSent;
}

void vCloseSerial(int i32Fd)
{
  close(i32Fd);
}

char boolean;


void exitFunction(int sig)
{
```


```
  printf("Received end application signal\n");

  boolean = 0;
}

int main()
{
  int fd;
  int ret;
  unsigned char rcvBuff[BUFFER_SIZE];

  ret = SerialInit(&fd, SERIAL_NAME);

  if (ret <0)
  {
    printf("Error: open serial port\n");
    printf("Error Code %d: %s\n", errno, strerror(errno));
    return -1;
  }

  signal(SIGINT, exitFunction);
  boolean = 1;

  while(boolean)
  {
    memset(rcvBuff, 0, BUFFER_SIZE);
    ret = i32ReceiveData(fd, rcvBuff, BUFFER_SIZE);

    if (ret > 0)
    {
      printf("Received %d bytes\n", ret);
      i32SendData(fd, rcvBuff, ret);
      printf("Sent %d bytes\n", ret);
    }
  }

  vCloseSerial(fd);
  printf("Application terminated\n");
  return 0;

}
```