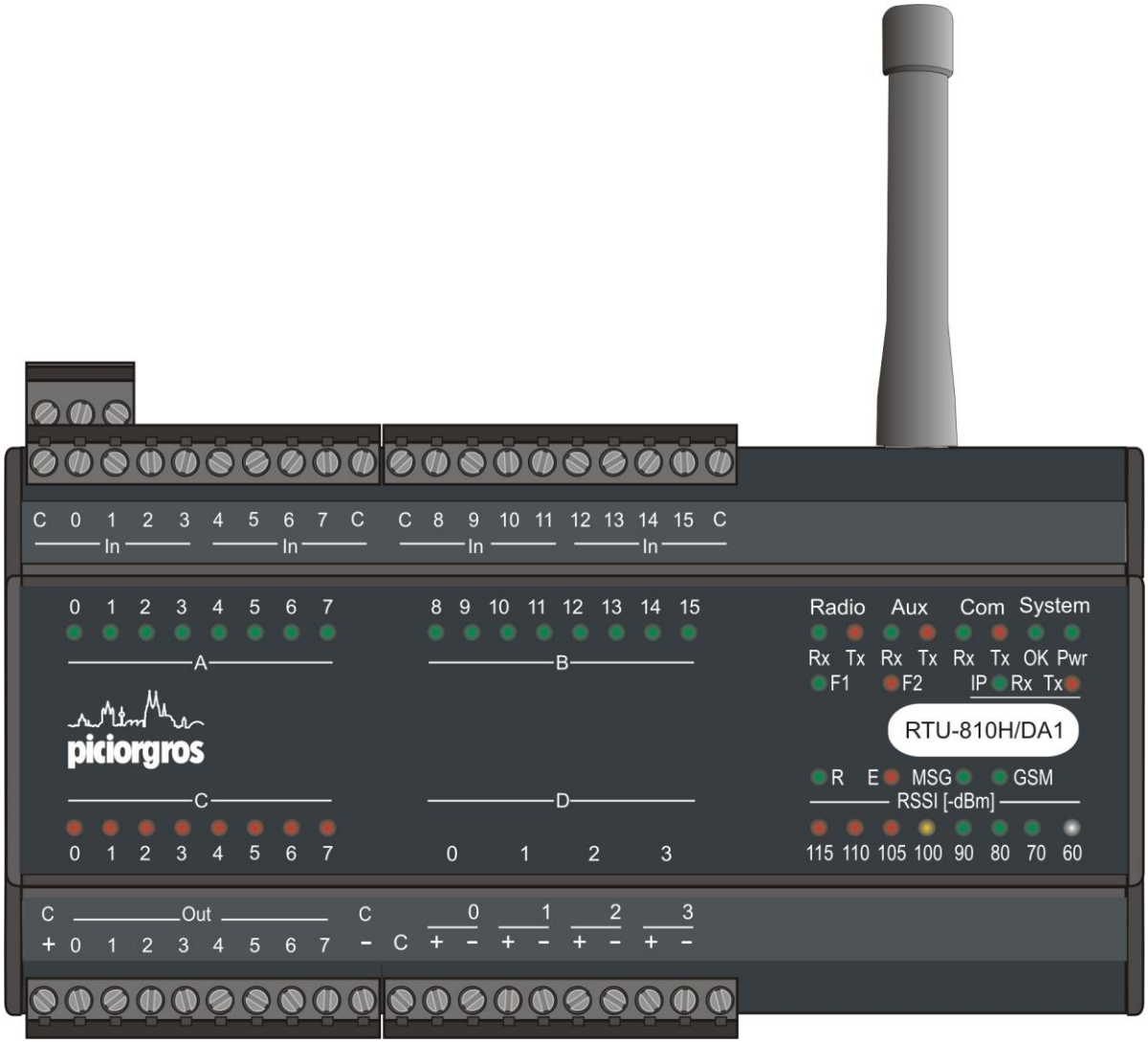
TRM-810 / RTU-810
TRM-81 / RTU-81/DA
User Guide
10/2018



Funk-Electronic Piciorgros GmbH
Claudiastrasse 5
51149 Cologne
Germany

1 Introduction

1.1 Overview

This document contains information about installation, settings, and operation of the RTU-810 Radio Modem. Additional information is also available over the Internet, on the website www.piciorgros.com, in the FAQ pages. It includes practical guidance relating to antenna selection and installation, operating range, extension modules, software support, etc.

1.2 Safety Precautions

This equipment transmits radio waves in the frequency range 420 to 470 MHz or -800 MHz. Under certain circumstances, these radio waves could be harmful to any living being or electronic equipment near it. It should be ensured that the radio and antenna systems are installed and commissioned by trained individuals only.

This radio equipment should not be used in life support systems or in safety systems without Piciorgros' prior written permission.

1.3 Disclaimer

Piciorgros has carefully checked the contents of this document, and the hardware and software described in it, for compatibility. However Piciorgros cannot exclude possibilities of deviations and cannot guarantee complete conformity of the document with the equipment it describes. If any corrections or improvements are to be made, they will be taken into consideration in the next edition of this document.

Important instructions are marked by the expressions "Important", "Note" or "Caution!". These should be carefully observed. Explanations regarding these precautions can be found in the website www.piciorgros.com, in the Login Area pages.

1.4 Versions TRM-810 vs. RTU-810 vs. RTU-81

The name "TRM-xxx" stands for data communication devices, which are interfacing to external equipment via serial or IP interfaces at Piciorgros on a historical basis.

The TRM-810 does have the same software platform and functions as the RTU-810 except for functions, which do relate to the internal digital and analog inputs and outputs. The focus of the TRM-810 is to connect external equipment like SCADA-servers, PLC's, RTU's etc. while the focus of the RTU-810 is to connect sensors directly to the embedded I/O's.

Nevertheless, the RTU-810 also has serial interfaces like a TRM-810 and can perform all functions of the TRM-710 plus the functions around the embedded I/O.

As a conclusion the TRM-810 is the "little sister" of the RTU-810.

The RTU-81 is a radio device in a smaller DIN rail enclosure for low power applications. It is only supporting the free ISM band frequency 433-434 MHz with a transmit power of up to 10mW. It also only has a fixed internal I/O setup of eight binary inputs, eight binary outputs, and two analog inputs.
Apart from this, it has the same features as the RTU-810.

In this manual and from that point on we are referring generally to the name "RTU-810" for simplicity and to describe the full feature set.

1.5 Functions and Features

The RTU-810 is a radio data communication modem. These modems allow transparent data communication between two or more nodes. The RTU-810 conveniently combines the functions of a controller, router, modem, and radio transceiver in a single compact enclosure.

The RTU-810 supports standard serial and IP-based data communication protocols. It has two serial data interfaces (RS-232 or RS-485/422) and an Ethernet (10/100 Mbits/sec) port.

Optionally the RTU-810 can also be provided with built-in inputs and outputs that can be read and set remotely using the MODBUS-RTU protocol, IEC-680870-5-101/104 and DNP3 (future option).

The RTU-810 has a rugged aluminum housing compatible with standard DIN rail mounting. The wide power input voltage range of 12-24 VDC [+/- 20%] makes it easy to integrate the unit into monitoring and control systems.

1.6 Hardware Options

The RTU-810/RTU-81 is available in different hardware (I/O) versions with various additional software options that can be enabled by license keys.
The following hardware options are available:

- **TRM-810M, TRM-810H**
  Radio modem with two serial ports and one Ethernet port.

- **RTU-810M/DA1, RTU-810H/DA1**
  Radio modem with two serial ports, one Ethernet port, 16DI, 8DO, 4AI (0-20mA, 4-20mA) and an expansion port connector for expansion modules with additional I/O.

  RTU-810M/DA2, RTU-810H/DA2
  Radio modem with two serial ports, one Ethernet port, 16DI, 8DO, no embedded AI and an expansion port connector for PEM-type expansion modules with additional I/O.

  RTU-810M/DA3, RTU-810H/DA3
  Radio modem with two serial ports, one Ethernet port, 16DI, 16DO, no embedded AI and an expansion port connector for PEM-type expansion modules with additional I/O.

- **RTU-810M/DA4, RTU-810H/DA4**
  Radio modem with two serial ports, one Ethernet port, 16DI, 8DO, 2AI (0-20mA, 4-20mA), 2AO (0-20mA, 4-20mA) and an expansion port connector for expansion modules with additional I/O.

- **RTU-810M/DA5, RTU-810H/DA5**
  Radio modem with two serial ports, one Ethernet port, 16DI, 8DO, 4AO (0-20mA, 4-20mA) and an expansion port connector for expansion modules with additional I/O.

- **TRM-81L**
  Radio modem with one serial port, one Ethernet port and a 1/10mW ISM band radio (433.1 – 434.7 MHz).

- **RTU-81L/DA**
  Radio modem with one serial port, one Ethernet port, 8DI, 8DO, 2AI (0-20mA, 4-20mA) and an expansion port connector for expansion modules with additional I/O. 1/10mW ISM band radio (433.1 – 434.7 MHz).

The RTU-810/RTU-81 can be delivered in three power classes, Low, Medium and High Power- The following hardware options are available:

- **/L-Versions**
  These devices contain a medium power RF device, defined as a product with 10 mW – 500 mW RF output power. Available for TRM-81 and RTU-81

- **/M-Versions**
  These devices contain a medium power RF device, defined as a product with 10 mW – 500 mW RF output power. Available for TRM-810 and RTU-810

- **/H-Versions**
  These devices contain a high power RF device, defined as a product with 100 mW – 6 W RF output power. Available for TRM-810 and RTU-810

By default the RTU-810 will be delivered with two serial RS-232 ports.

RS-422 or RS-485 are options, which must be provided with the order. They cannot be activated or changed after the purchase in the field!

1.7 Software Options

License keys can enable a number of additional options. Each key is individually valid for one single RTU-810 device.
The activation of the license keys can easily be done with the embedded web server using the “Service / Configuration” menu.
When ordering an additional license, the user has to submit the serial number of the RTU-810 to be updated.

1.7.1 General Activation

By default a purchased RTU-810 is activated for use. Demo devices may be delivered to only run a certain time.

1.7.2 IP Communication

Option to use the RTU-810 as an IP router to connect IP devices via the radio link (Future option)

1.7.3 PicoLogo – embedded User Application Interface

The PicoLogo option is a powerful User Application Interface that can be used like a Micro PLC to generate Alarms, Text-Messages, M2M-Communication or to monitor and supervise digital or analog values.

An additional graphical editor for developing PicoLogo applications is also available.

1.7.4 IEC-60870/DNP3 support

This option allows the access of the internal I/O of a RTU-810/DAx via IEC60870-5-101/104 and DNP3 protocol.

At a single time only one of these protocols can be active as these are different firmware versions.

1.7.5 RTU master mode

Enables an RTU-810/DAx to be a radio master for I/O exchange with up to 4 other RTU-810 (Point-to-point up to Point-to-four-points link)

1.7.6 Serial master modes

Enables an RTU-810/TRM-810 to act as a serial master device to connect PLC's, SCADA- and OPC servers for operating radio networks of up to 239 slave RTU 710/810 and TRM-710/810 (Operation modes: ZZ, ZZT, ZZTR).

1.8 Future options and devices

This document describes options and devices, which are already in development and will be available in short to mid-term.

These features are described in blue paragraphs.

Related hardware devices and hardware options will be:

- SIG-810 Serial-to-IP Gateway for tunneling serial connections via IP networks to the IP Application Interface of the RTU-810.

- RGW-810 Radio Gateway to enable parts of the radio network being remotely offset via IP connection

1.9 Document Changes

The history of this documentation is listed below:

Firmware Version	Document Version	Comments / Changes
2.30	2.30	Initial Document Release
2.36	2.36	• Removed outlooks for the GSM variant (which will become an independent product) • Added description for registers 953-955 in compatibility mode which will be BCD-coded with version 2.36 (to ensure compatibility to the format of the TRM-7xx
2.40	2.40	• Added T1X radio mode
2.45	2.45	• Added ZZT and S2U radio modes for serial communication
2.51	2.50	• The Radio Layer Address assignment method has been changed. View chapter 1.10.1 for more information! • The method of performing a Hardware Reset To Factory Default has been changed. See chapter 2.4.2
2.55	2.55	• "PLS"-Mode for serial slave implemented • "T1X+RSSI" interface mode implemented

Software Release Notes are distributed with firmware packages.

1.10 Important changes

1.10.1 Radio Layer Address Change (from Firmware 2.50 / July 2018)

With firmware version 2.50 / July 2018 the Radio Layer Address (RLA) is no longer assigned by Funk-Electronic Piciorgros GmbH. The RLA can be freely configured in the radio settings in the web server. The primary target is a much enhanced isolation of networks, even operated by the same customer. Also extensions of existing networks are simplified by eliminating the fear of wrong assigned RLA's.

All new delivered RTU/TRM-81(0) have their own serial number set by default as the Radio Layer Address!

Recommended procedure for new radio networks:
Each new deployed radio network should use the serial number of the master device as its dedicated Radio Layer Address!
The RLA of the master device can therefore be left unchanged.
For each slave device, the serial number of the master should be entered as the RLA in the slave's radio configuration.

Procedure for extending existing networks:
For existing networks the actually used RLA must be converted, especially if they contain equipment of the types:

- RTU-700 / RTU-710
- TRM-700 / TRM-710
- SS20F
- MDP-200, MDP-300, MDP-310.200

The RLA of new deployed RTU/TRM-81(0) must be changed to match the existing Radio Layer Address of the network.

The old, fixed assigned RLA are using the hexadecimal format! This format must be converted to decimal to be used with the new method.

Example:
An RTU-810 should be added into an existing network with the assigned RLA "1234".
The converted value which must be entered as the Radio Layer Address in the RTU-810 configuration must be:
HEX 1234 → Decimal **4660**

If the firmware of an existing RTU/TRM-81(0) is upgraded to V2.50 or higher, the currently configured (old) radio layer address is automatically converted to the new format!

With a Reset To Factory Default the RLA is always defaulting to the device serial number!

2 Connections and Hardware Installation

2.1 Mechanical Details

The dimensions of the RTU-810 housing conform to DIN 43880 and therefore it can be mounted on a standard 35mm DIN rail [DIN EN 50022]. Two serial interfaces are provided for connecting the RTU-810 to a PC/PLC or other local terminal equipment. The serial interfaces are using a female DB9 connector. On the bottom side of the enclosure an RJ-45 connector for the Ethernet port allows the RTU-810 to be connected with IP based equipment such as PLC's, RTU's, SCADA servers etc.

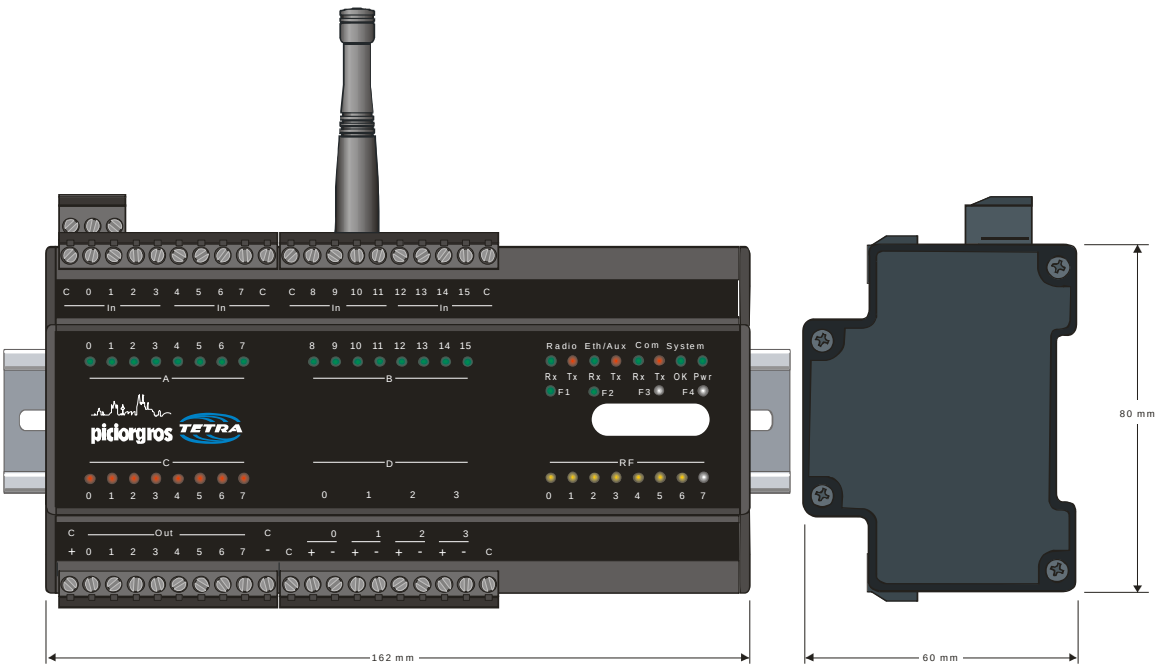
On the upper side of RTU-810/DAx (units with embedded I/O), a RJ-45 connector is provided to connect PEM I/O expansion modules.

Also on the upper side of the unit the plug-in terminal connector for power supply (12-24 VDC +/-20%) and a BNC socket for the antenna can be found.

LED lamps on the front panel provide information about the operating condition of the unit: e.g., received Radio RF signal strength, error conditions, etc.

2.1.1 Dimensions

The dimensions of the RTU-810 are as follows:
162mm (9T) wide x 80mm high x 62mm deep
All dimensions exclude connectors and antenna.

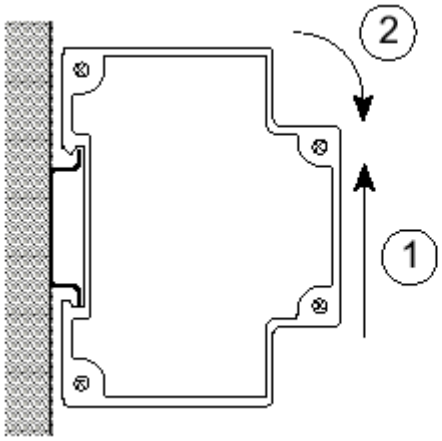


2.1.2 Mounting

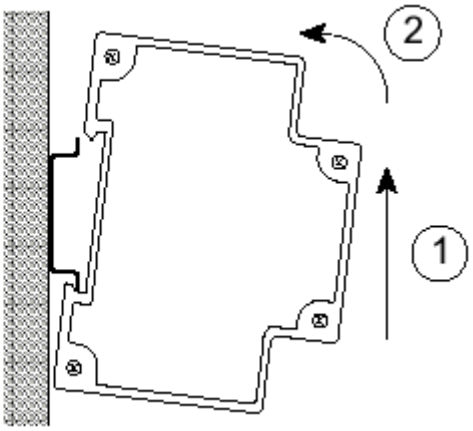
The DIN-rail mounting clip is at the bottom of the module. First the lower lip (spring-loaded) of the clip is engaged with the lower flange of the DIN rail, with the module tilted downward slightly. The module is then pushed upward (1) and rotated backward (2) until the upper lip of the clip snaps onto the upper flange of the DIN rail.

2.1.3 Dismounting

To dismount the module, force it upwards (1), and then rotate its upper end outward (2) until the upper lip of the module's clip disengages from the upper flange of the rail. Then move the module down slightly to disengage its lower lip from the rail flange.



Mounting



Dismounting

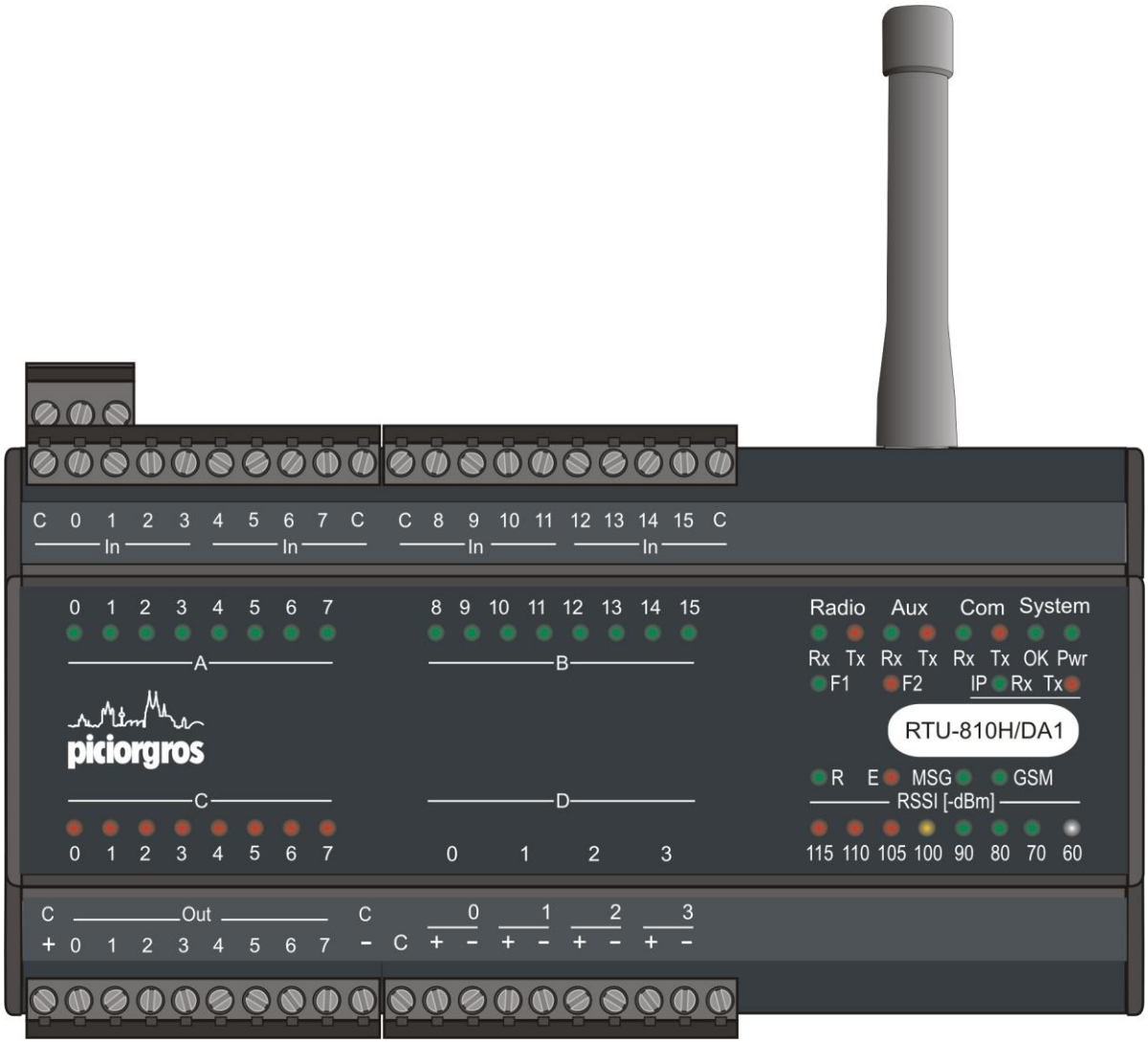
2.2 Electrical Connections

2.2.1 Power Supply Input

The required supply voltage (12-24 VDC +/-20%) is connected through 3-way screw terminal connector located on the upper side of the enclosure.

The terminals are assigned as follows (viewed from the front of the module, facing the front panel):

Outer (left): Enclosure Ground (electrical earth)
Middle: + 12 Volt to + 24 Volt (+/- 20%)
Inner (right): GND, 0 Volt from Power Supply



Configuration: Ethernet, embedded I/O and two serial ports

2.2.2 Serial Interfaces

The RTU-810 has two serial data interfaces. Each interfaces has a 9-pin D-sub connector on the upper side of the unit. The COM interface can be either RS-232 or user-selectable RS-422/485 (RS-422/485 must be initially ordered as this is a hardware option). The AUX interface is RS-232.

The following parameters are user adjustable: baud rate in the range 300 - 57600 bps, data word length 7 or 8 bits, odd / even / no parity, and 1 or 2 stop bits. The factory setting is 9600 bps, 8 data bits, no parity, 1 stop bit.

If a frame error is detected, or if the parity bit does not conform to the setting, the received data block is rejected.

Pin No.	**Pin Assignment: RS-232**	
2	TxD	Send data RTU-810 → peripheral
3	RxD	Receive data RTU-810 ← peripheral
4	DTR	Shorted to Pin 6
5	GND	
6	DSR	Shorted to Pin 4
7	RTS	Handshake RTU-810 ← peripheral
8	CTS	Handshake RTU-810 → peripheral

Pin No.	**Pin Assignment: RS-422 (COM only)**	
2	Y	Transmitter + (output)
3	Z	Transmitter – (output)
5	GND	
7	A	Receiver + (input)
8	B	Receiver – (input)

Pin No.	**Pin Assignment: RS-485 (COM only)**	
2	A	Transceiver +
3	B	Transceiver –
5	GND	

For the connection of the COM interface to a PC or PLC, use a standard 1:1 connector-terminated cable (9-pin D-sub male to 9-pin D-sub female).

2.2.3 Ethernet Interface

The Ethernet interface is provided via an RJ-45 socket on the bottom side the unit. This is a standard 10/100 Mbit/s interface. Two LEDs indicate the operating condition of this interface:

- Green LED: Lights up when an Ethernet network is connected (LINK)
- Yellow LED: Flashes when data transfer is taking place (DATA)

Network parameters such as IP address, net mask, and gateway address can changed via the embedded configuration web server.

The interface supports Auto MDI-X, so the RTU-810 can be directly connected to a computer without the need of a crossover cable.

2.2.4 Extension Port for additional I/O modules (only for devices with embedded I/O)

The I/O expansion connector is located at the upper right part of the RTU-810 for all versions with embedded I/O. The connector is grey and has the same color as the expansion port modules connector.

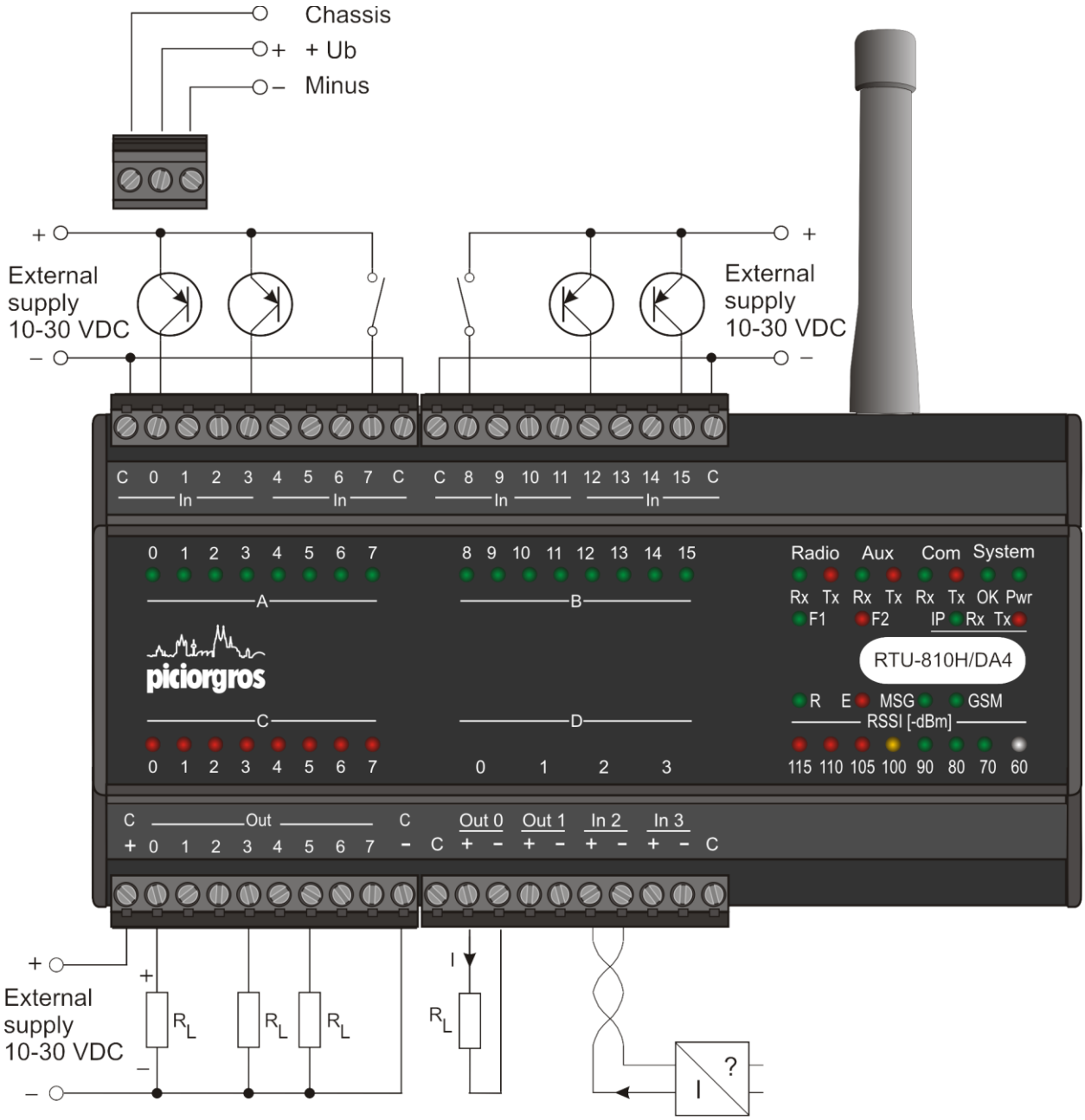
Up to 16 expansion modules can be connected to the RTU-810 to enlarge the number of inputs and outputs. The access to all I/O can be performed by using the Modbus RTU or Modbus TCP protocol.

The following I/O modules are available:

PEM-16DI	16 digital inputs
PEM-32DI	32 digital inputs
PEM-16DO	16 digital outputs
PEM-32DO	32 digital outputs
PEM-16DIO	16 digital inputs and 16 digital outputs
PEM-08AI	8 analog inputs (0-20mA 4-20mA)
PEM-04AO	4 analog outputs (0-20mA 4-20mA)

I/O modules should only be connected to or disconnected from the RTU-810 when power supply is disconnected!

2.2.5 Connecting the I/O (RTU-810/DAx types only)



Binary inputs:
The binary inputs are divided into two groups per port. Each group contains 4 inputs with one common terminal. The inputs are bipolar, so they can be switched with an active "+" or active "-" signal.
If the "C" terminal is connected to the ground, the inputs will be active if the positive voltage (12-14V) applies. If the "C" terminal is connected to the positive voltage, the inputs are activated by applying the ground connection.
Please note that the input voltage should not exceed 24V +20%!

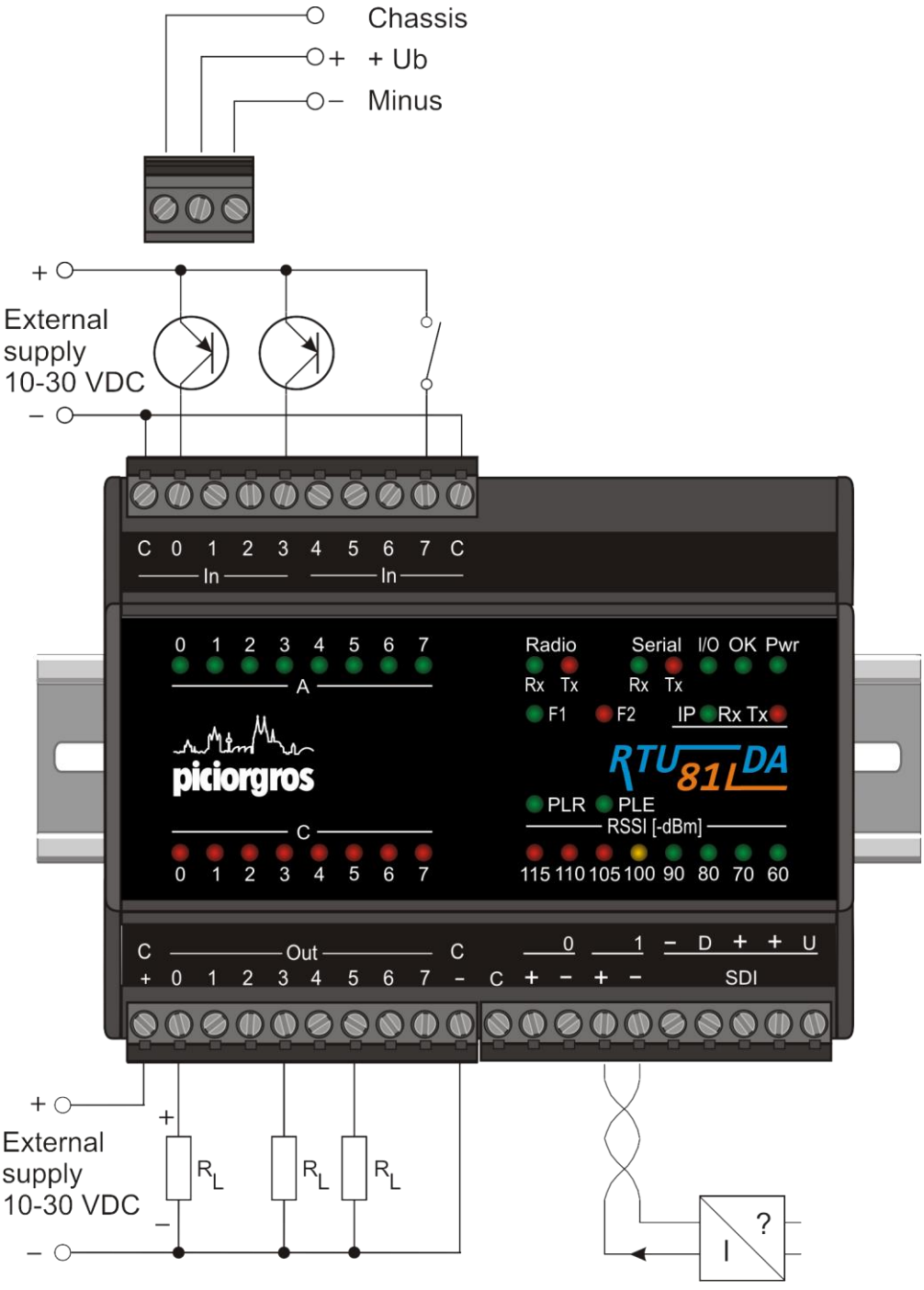
Binary outputs:
The supply voltage for the binary outputs must be connected to the "+" and "-" terminal of port C. Active outputs will have the positive supply switched to the related terminal. The maximum load is 0.5A per outputs.

Analog inputs:
The analog inputs need to be in the current loop, the power supply must be provided externally in the loop. The input range is 0-20mA with a resolution of 12 bit.

Analog outputs (DA4/DA5 type):
The analog outputs are an active source for 0-20mA, which means that i.e. an Ampere meter can be directly connected to the "+" and "-" pin of the output. The resolution is 12 bit.

Note: All I/O are electrically isolated!

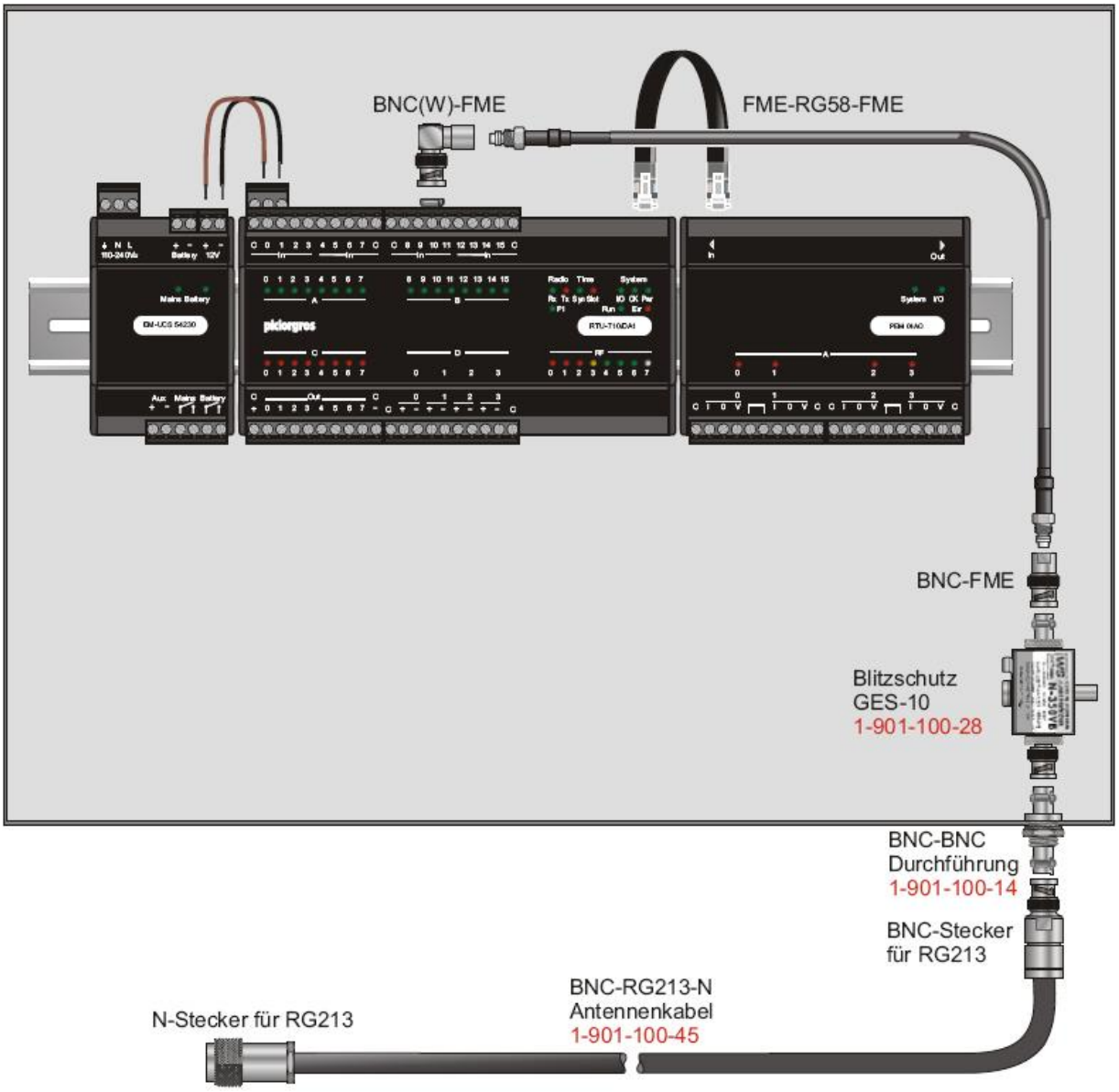
2.2.6 Connection diagram RTU-81



2.2.7 Connecting the Antenna

A BNC socket is provided for the connection of the antenna. Ensure that the antenna is selected correctly for the operating radio frequency. The antenna cable (50 ohm) should not be longer than 10 meters, and should be of good quality if possible low-loss type (RG-213 or Aircell).

Overvoltage protection to protect the modem is available. This should be attached to the modem using a feed through bush (see diagram). Ensure that the modem's grounding point, antenna input lightning protection, and the antenna system are connected to a common grounding point.



2.2.8 DCF77-Option

The DCF77-Option is another method to synchronize the time base of the TRM/RTU-81(0). In addition of using an active DCF77 antenna which needs to be in the reception range of the German DCF77 transmitter, a PTS-100 time server can be connected which generates the signal from the GPS system and which works world wide.

The DCF77 option is a hardware option. Devices which do have the DCF77 option are equipped with an additional jack plug to connect the active antenna or the PTS-100. The DCF77-option works as an additional time source to the SNTP function. This means that both methods can work side by side, i.e. to enable time synchronization redundancy.

2.2.8.1 Setting up the DCF77 Antenna

The DCF77-Antenna must be connected to the jack plug of the TRM/RTU-81(0). The antenna must be mounted horizontal. First, the antenna should be turned in a way that the red LED on its top is completely off or nearly completely on – this is the position of the worst reception quality. From this position the antenna should be turned 90° to get the optimal direction.

As soon as the signal quality is good, the TRM/RTU-81(0) will start to decode the received signals and the green "Time SYN" LED will start to flash. This shows that the synchronization phase has started and that the device is decoding the received time signal.

After less than two minutes the flashing of the "SYN"-LED should turn into a permanent light. The device is time synchronized now.

If time slot operation has been set up, the "Slot"-LED now indicates when the operating time slots are active.

In case the time synchronization is interrupted, the TRM/RTU-81(0) can continue in time slot operation even unsynchronized for around three days. Unsynchronized operation is being indicated by a lit SYN-LED which flashes off once per second for a short time.

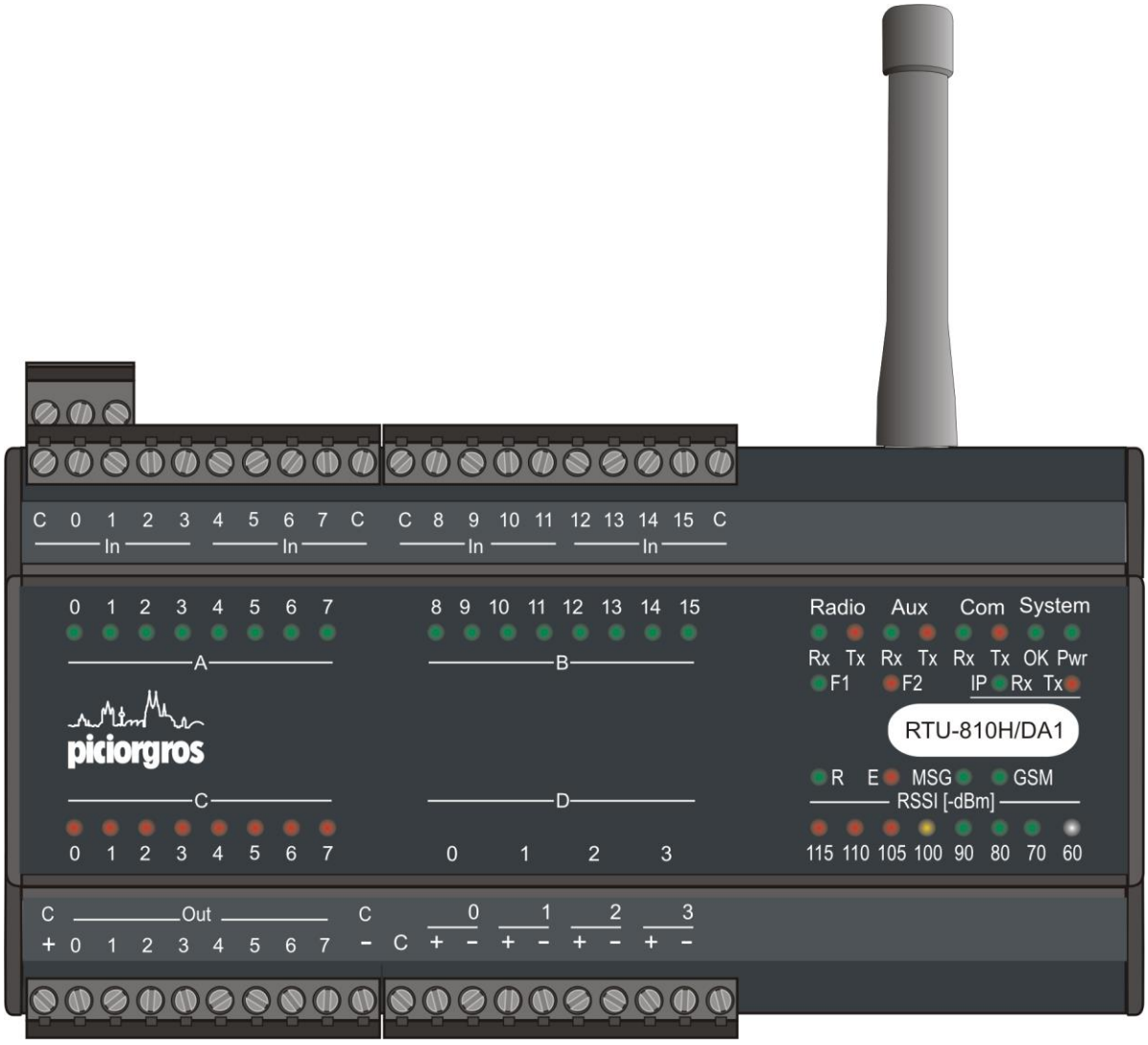
SYN-LED	SLOT-LED	Description
Off	Off	The device is not decoding any time information
Flashes 1:3	Off	The device is in synchronization mode, searching the start of the next minute
Flashes 1:1	Off	The start of a minute was found, the device is decoding the time information. Every correctly decoded bit flashes the SYN LED.
On	On during active time slot	The device is time synchronized
Short off-blip every second (3:1)	On during active time slot	The device was not synchronized for some time and is running on internal time backup (max. 70 hrs)

Important:
The SYN- and Slot-LED are used to show the general status of the time synchronization. In case that the SNTP and the DCF sources are active, the LED will show the synchronized status as soon as the time sync was performed by SNTP, even when the DCF signal is still not fully decoded.

The detailed status of each time source can be checked in the configuration web server on "Radio" → "Timeslot operation".

2.3 LED Functions

Groups of LED lamps on the front panel of the RTU-810 indicate the operating condition of the modem, the field strength of the received radio signal, and error conditions if any occur. Specific LED functions are described below.



LED	Function
System Pwr	Power supply input
System OK	Indicates the "Ready" status of the modem (continuously lit). Also indicates error conditions (coded blink sequences). A 1:1 blinking means that the device is forced to IP address 192.168.0.199/20 (default). This mode can be switched by pressing into the small hole on the back of the RTU-810 for one second.
COM Tx	Lights up while the RTU-810 sends data via the COM interface
COM Rx	Lights up while the RTU-810 receives data via the COM interface
AUX Tx	Lights up while the RTU-810 sends data via the AUX interface
AUX Tx	Lights up while the RTU-810 receives data via the AUX interface
Radio Tx	Indicates that the RTU-810 is currently transmitting on the radio frequency.
Radio Rx	Indicates that the RTU-810 is receiving a data frame on the radio frequency. It only lights up if the RTU-810 is receiving data from another Piciorgros device. The Radio Rx LED can also be configured to only show the reception of Piciorgros data frames with the same customer (radio layer) address.

RF	These eight LEDs indicate the field strength of the received radio signal. They always show the current occupation of the configured radio frequency, which includes any transmitter on that frequency. This can be used to watch the occupation by other radios and/or interferences. A configurable Peak Hold display (default on) shows the RF field strength of the last data telegram, which was addressed to a certain RTU-810. This always gives a view about the relevant RSSI value of data addressed to the device. If the latest received data is older than 15 minutes or if the device was restarted, this peak hold value flashes. The RSSI display is calibrated to show RSSI values between -115dBm and -60dBm and above.
F1 (green)	If the RTU-810 operates in time slot operation, this LED shows the sync status of the time source (SNTP or DCF-77/PTS-100 time server): • Permanently on: The time synchronization is up to date and time slot operation can be used • Mostly off with an on-blip every second (DCF/PTS-100 only): The RTU-810 is waiting for the second 0. Time slot operation will not work in this state. • 500ms flash every second (appears as 1:1 flash, DCF/PTS-100 only): The RTU-810 is decoding time information from the DCF-77 antenna or PTS-100 time server. Time slot operation will not work in this state. • Mostly on, blipping off every second: The last time synchronization is overdue and the RTU-810 time slot operation is running on internal reserve. In this mode the time slot will get smaller continuously to balance the drift of the internal clock. Without synchronization the timeslot operation will stop after around 3 days. • Off: No time synchronization active, time slot operation will not work in this state.
F2 (red)	If the RTU-810 operates in time slot operation, the LED shows a currently active time slot. The RTU-810 is allowed to transmit as a master if the LED is on.

IP Rx	Indicates, when data is being received over the local Ethernet port
IP Tx	Indicates, when data is being transmitted to the local Ethernet port
R	If the green "R"-LED lights up, a PicoLogo application is loaded, set to "RUN" mode and running properly.
E	If the red "E"-LED is lit, a PicoLogo application is loaded, but in error state. Check the web server on page "PicoLogo" → "Run switch / status" for details.
Port A/B LED	Shows the current state of the binary inputs
Port C LED	Shows the current state of the binary outputs
Port D LED (DA3 type only)	Shows the current state of the binary outputs

2.3.1 OK LED: Blink Pattern Error Indication

When the RTU-810 is powered up, the OK LED should be constantly “on” to indicate the device is operated properly.
If this LED is flashing or off, it indicates an exception. Different “blinking codes” can indicate different problems as shown in the table below:

Blink Pattern	Meaning
LED off	Controller (CPU sub-module) error or the modem is not in "Ready" state.
Continuously lit	Device in "Ready" state, no error conditions
Slow blinking, 1:1 tempo	The device is in programming mode and configured to the IP address 192.168.0.199/20, regardless of the configured Ethernet IP address
Long flash – short flash alternating	The device is processing or performing a firmware update
4 blinks, then pause	The internal radio module does not respond
5 blinks, then pause	The programmed radio frequency does not fit to the radio module's frequency range

2.3.2 Other LED codes

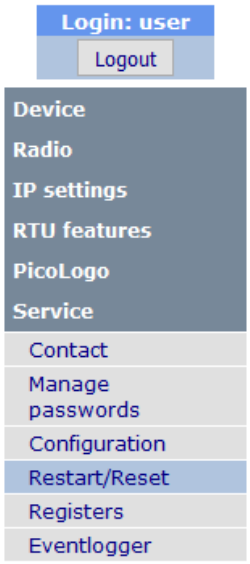
Blink Pattern	Meaning
ALL LED's are flashing fast after power-on	The startup could not be performed due to a failure. The RTU-810 will try a restart after 12 seconds in this state. If this state persists, even after a power cycle, a hardware issue might be the cause and the RTU-810 must be sent in to service.
All I/O-LED's (Input and output ports) are flashing up periodically (/DAx devices only)	This indicates a problem with the I/O board. The RTU-810 must be sent in to service.

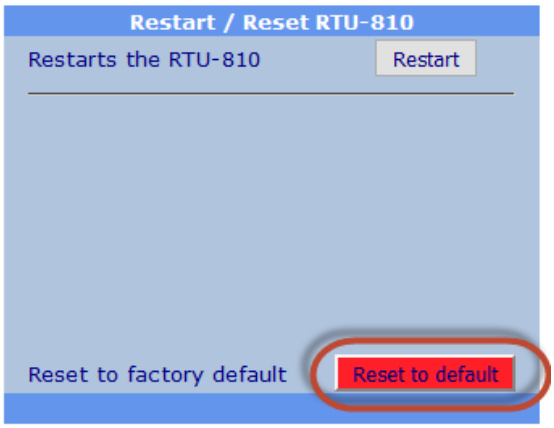
2.4 Reset to factory default

The RTU-810 can be reset to factory default configuration. This can be done in two ways:

2.4.1 Reset via Web Server

A button on the page "Service" → "Restart/Keyloader" can perform a Reset-to-Factory-Default:





2.4.2 Reset via Configuration Button

New method (Firmware V2.51+):

- Power down the RTU/TRM-81(0)
- Press the button behind the small hole in the back of the device with a paperclip or similar tool and keep it pressed
- Power up the device
- Keep the button pressed for at least 5 seconds until the middle 6 LED's of the RSSI bar start flashing
- The device has now started with the default configuration. The button can be released.

Old method (prior to Firmware V2.51):
If the configuration button behind the small hole at the back of the RTU-810 is pressed for at least 5 seconds (using a Paperclip or SIM-Extractor of a smartphone), the LEDs of the RF display begin to flash alternatingly. The RTU-810 will then perform a reset to factory default.

2.5 Temporary reset to default IP address

If the IP address of an RTU-810 is unknown, it can temporarily be reset to the default address:

- Press for one second on the configuration switch behind the small hole at the back of the RTU-810, until the "OK"-LED starts to flash in a 1:1 cycle
- Release the configuration button (Take care: A 5-seconds-press resets the complete device to default!)
- The RTU-810 can now be accessed on its default IP address 192.168.0.199/20. The originally configured IP can be seen and changed in the webserver on "IP settings" → "Ethernet".
- To resume to normal mode, press again for 1 second on the configuration switch, until the OK-LED resumes to be permanently on. The RTU now reacts to the configured IP address

3 Configuration of the RTU-810 using the embedded web server

All parameters of the RTU-810 are configured via an embedded web server. Recommended browsers are Mozilla Firefox or Chrome.

3.1 Accessing the embedded web server

To access the RTU-810, a computer must be connected to the RTU-810's Ethernet port – either directly or via a switch / LAN. The computer must be in the same IP subnet as the RTU-810.

The default IP address of the RTU-810 is 192.168.0.199 with netmask 255.255.240.0.

If the IP address of the RTU-810 is typed into the web browser, the login page will appear:

The first login will be done using the default username and password (take care, as both are case sensitive):

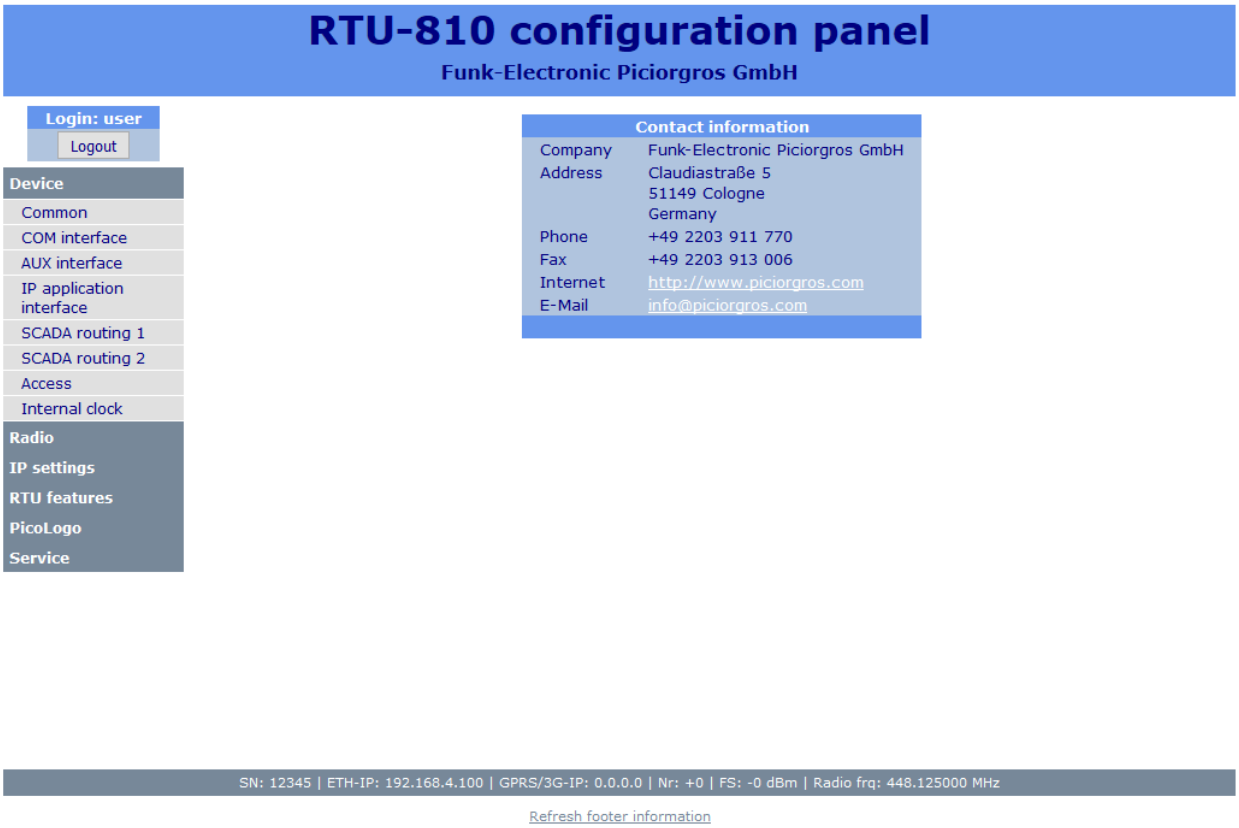
Username: “user”
Password: “user“

These passwords can be changed any time using the menu “Service" → "Manage Passwords”.

After a successful login, the configuration menu of the RTU-810 will be displayed. A navigation menu on the left side of the page allows access to the different configuration pages.

A separate footer displays the most important device parameters.

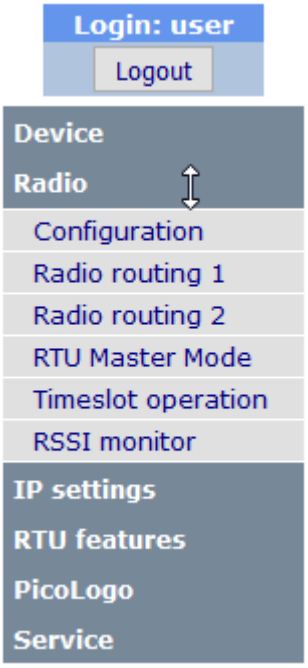
Please note that the menu and footer line are only loaded once after login to reduce traffic load. The footer line has a "Refresh" link to update the information of the footer on demand.



After 5 minutes without activity the user will automatically be logged out.

3.1.1 Navigation using the Menu

To select a menu, move the cursor over the desired headword and click it. Then the menu will open displaying the next options



3.1.2 Applying changes

On the bottom of each page two buttons can be seen, which give the user the ability to apply the changes, which are made on that page.

Each value, which is entered or changed will not immediately be applied by the RTU-810. Once the changes of the settings on given page are done, the button "Apply" must be clicked – the RTU-810 will then immediately apply the new values.

If any wrong entry has been made on the page, a click on "Reset" will restore the original values, which have been transmitted initially. Note that a "Reset" will not work after the changes have been applied.

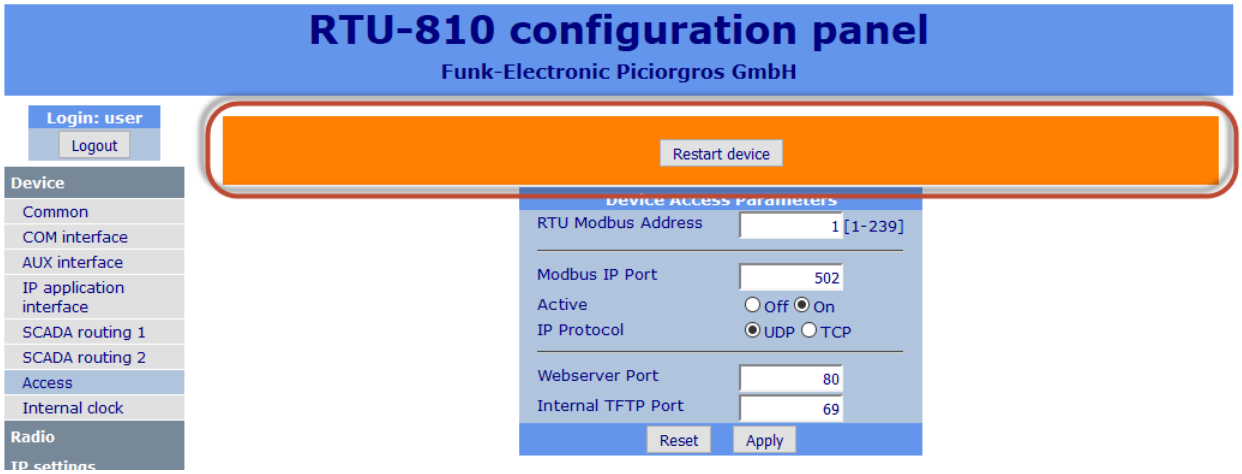
Note: Instead of clicking "Apply" pressing the "Enter" key will have the same function on most pages. **Care must be taken on pages where other buttons are used**: In this case the first button on the page will be activated! On pages with buttons it is recommended to use the "Apply" button at the end to prevent unwanted actions!

3.1.3 Requesting Restart of the Device

Some configuration changes need a device restart before they will take effect.

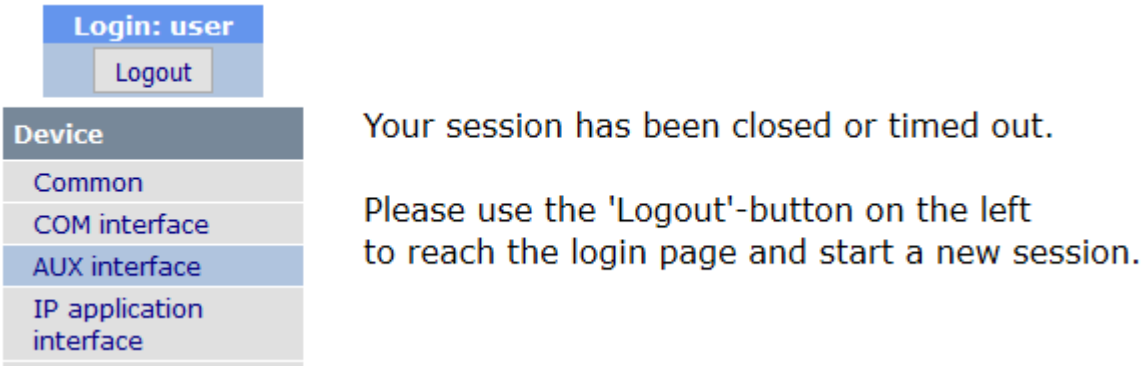
If the user has made one of these changes, an orange button with the text “Restart Device” is displayed.

It is not necessary to press this button immediately as long as other parameters need to be changed also.



3.1.4 Session timeout and page reload

After 5 minutes of inactivity the configuration session will time out.



An easy way to continue the configuration session is to reload the page by hitting the F5 key.

3.1.5 Recommended Web Browsers

For the configuration of the RTU-810 these Browsers are recommended:

- Mozilla Firefox
- Google Chrome

Microsoft Internet Explorer is not recommended as with this Browser single parameters on some pages are not completely transmitted to the RTU-810!

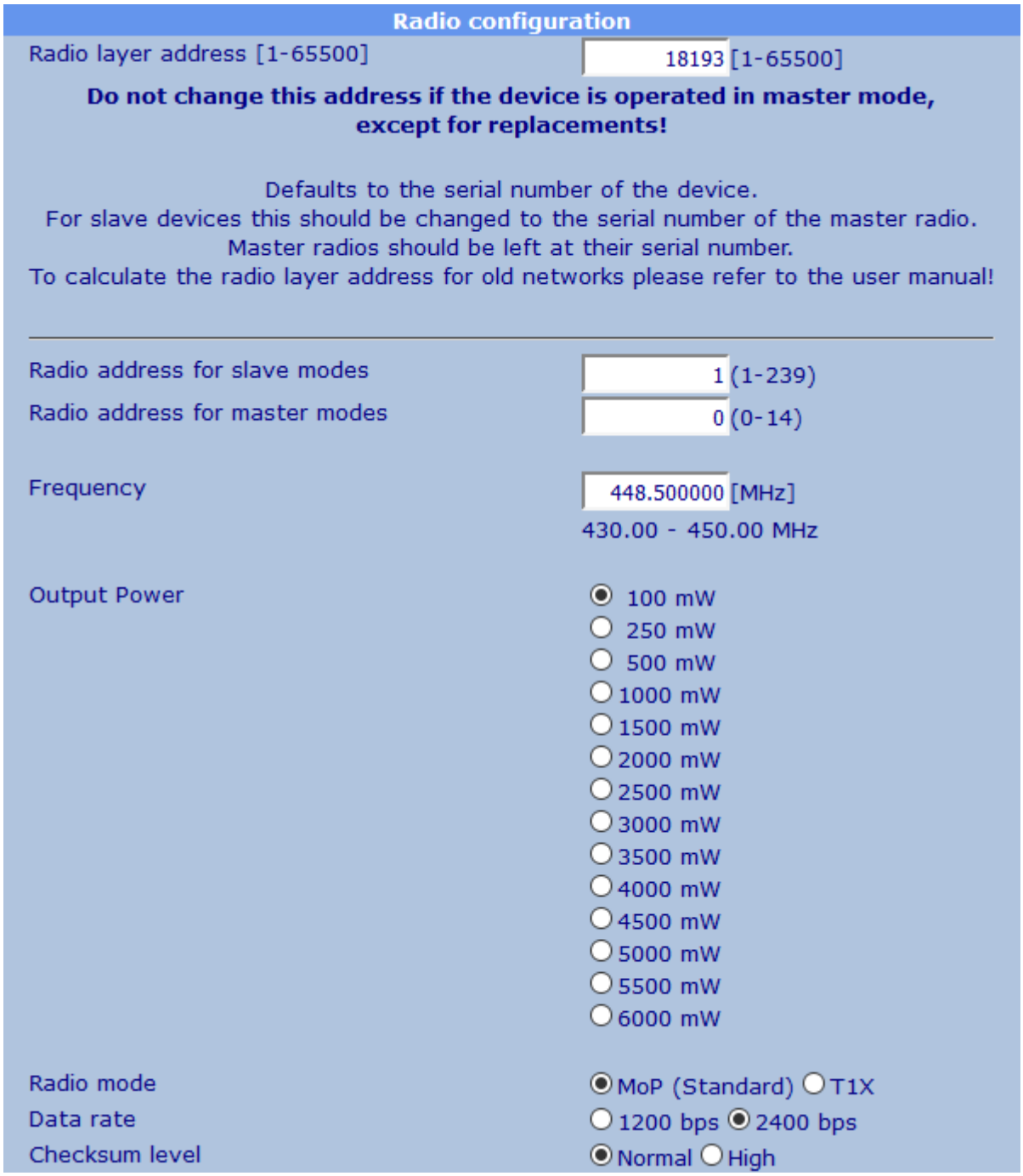
3.2 Radio configuration

This section explains the radio configuration of the RTU-810. Most important thing to start with is to set the correct radio frequency and transmit power.

3.2.1 Configuration

The basic radio configuration is on the page "Radio" → "Configuration":

3.2.1.1 Radio Configuration



Radio layer address

Only radios with the same Radio layer address (RLA) can communicate together. The RLA defaults to the serial number of the device.

It is recommended to leave the RLA on master devices untouched for new radio networks. In this case the serial number of the master radio must be entered into this field on each slave device which communicates with the master.

For existing networks which are using the old fixed assigned RLA, a conversion of the old address is needed.

Please see chapter 1.10.1 "Radio Layer Address Change (from Firmware 2.50 / July 2018)"

Radio address for slave modes	This is the address, which the RTU-810 uses in the radio network for slave functions. Slave functions are: • Being addressed for I/O access by a master radio (RTU-810 in RTU master mode or by a TRM-810 using the serial master modes) • Being addressed for serial slave operation on a serial interface (S1U mode) • Being used as a radio relay This is the "main" radio address of the RTU-810 and must be unique in a radio network.
Radio address for master modes	This is the "sender's address" for all radio telegrams where the RTU-810 acts as a master: • Telegrams sent out via serial interface in one of the master modes (ZZ, ZZTR, ZZTRM) • Telegrams sent out in RTU master mode to poll up to four RTU-810 slaves If only one master is present in a radio network, the address can be kept at the default. If more that one master is acting on the same frequency and with the same radio layer address, each master should have its own master address configured.
Frequency	This is the frequency given in MHz where the RTU-810 should transmit and receive the data. The frequency range, which is supportd by the hardware, is listed below this field. The frequency must be a multiple of 12.5 kHz. If this is not the case, the entered frequency will be automatically corrected when the changes are applied.
Output power	This sets the RF output power for transmissions. The selection is different between medium power (500mW) RF modules and high power (6W) RF modules.
Radio Mode	This selects the main radio mode between the standard MoP method and the T1X method. In MoP mode each device has an own radio address, the use of routing and radio relays in the network is possible as well as the operation of time slot operation with just a single time reference in the master station. This mode is the powerful standard. In T1X mode the radios do not have an address and all communications which is sent from one radio is received and processed all other radios with the same radio layer address. This mode enables real peer-to-peer networks where any device can directly talk to any other.

Check received T1X for MODBUS data	This item is only visible when the device is set to T1X radio mode! If switched to "On", the device will check any valid received data if a MODBUS-RTU telegram is embedded. If this is the case and the MODBUS address matches to the one configured in "Device" → "Access", the RTU will process the MODBUS data locally.
Data rate	This selects the air interface data rate. 2400 bit/s is set by default. Higher data rates will be available in a future high speed version of the RTU-810.
Checksum level	The RTU-810 always sends a 24 bit CRC checksum for the radio telegram to ensure that no erroneous telegrams are interpreted. TRM-700/710 and RTU-700/710 radio networks are using an 8 bit CRC over the radio telegrams where in rare cases of a bad receive RSSI even a wrong telegram could match the CRC by random. Most of these wrong matches are later trashed by an additional checksum in the application data, but in some customer applications without an own checksum the transmission security only relies on the 8 bit radio checksum. The checksum level setting determines what checksum the RTU-810 expects for reception. If this is set to "normal" (default) it will only check the 8 bit checksum, which makes it compatible to radio networks with existing TRM-7xx/RTU-7xx and SS20F solar devices. If a radio network only uses RTU-810/TRM-810 the checksum level can be set to "high" and the full 24 bit checksum is required to match.

3.2.1.2 T1X operation

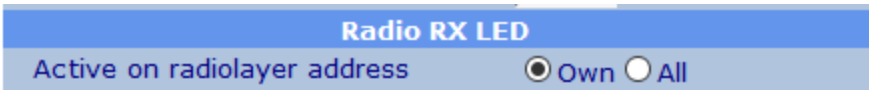
If the radio is set to "T1X" mode, a couple of limitations are applied:

- The only available radio mode for the serial interfaces and the IP Application Interface is "T1X" now.
- The radio routing menus and relay options are not available
- The RTU Master mode is not available
- The RSSI monitor is not available
- In general, a timeslot operation is possible, but in contrast to the standard MoP mode each radio must have a local time synchronization.

3.2.1.3 RSSI display

Display mode	Reserved
RSSI peak hold	If the peak hold is set to "on", a single LED in the RSSI bar will "hold" the receive RSSI of the last radio telegram, which was directly addressed to this RTU-810 for procession. This is an easy way to identify the RSSI of the relevant telegrams addressed to the own radio among the general RSSI of the data traffic on this frequency.
Peak hold expire	If for this amount of minutes no access to the RTU-810 has occurred, the peak hold LED will start to flash. A flashing peak hold LED means, that this RTU-810 did not receive any data addressed to it for given time. Also after a device restart or power on the RSSI value of the last received data is displayed as the value is stored nonvolatile.

3.2.1.4 Radio RX LED



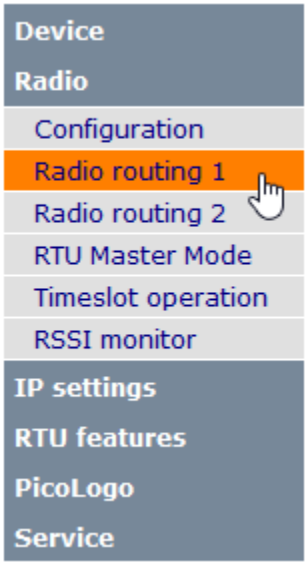
Determines whether the green "Radio RX" LED on the front panel should only indicate the reception of data, which is using the same radio layer address as currently used for the RTU-810, or if it should indicate the reception of any Piciorgros radio telegrams, even if different radio layer addresses are used.

3.2.1.5 RTU/TRM-710 compatibility mode

If the compatibility mode is set to "On", the RTU/TRM-810 will behave like an old RTU/TRM-710 device in some parts for compatibility:

- The radio RSSI in the MODBUS register 1 and in the "*"-commands in the serial "ZZ" mode will be returned in an uncalibrated "%"-Value instead of a calibrated –dBm value
- The device ID will not be 0x0800 to indicate it is an RTU-810, it instead reports the old ID from the 710-series:
  RTU-810/DA1 → 0x0011 (RTU-710/DA1)
  RTU-810/DA2 → 0x0010 (RTU-710/DA2)
  RTU-810/DA3 → 0x0012 (RTU-710/DA3)
  RTU-810/DA4 → 0x0013 (RTU-710/DA4)
  RTU-810/DA5 → 0x0014 (RTU-710/DA5)
  TRM-810 → 0x0028 (TRM-710/S1U)
- The status register has the old behavior of the 710 series that any writing to this register clears all bits of the register. Normally for the RTU-810 only bits, which are set in the write access will be cleared.
- The time read registers 953-955 will be read back BCD-coded instead of the current time format YYMMDDhhmmss.

3.2.2 Radio Routing tables 1 & 2



The Piciorgros radio layer can use up to 31 radio relays for each data transmission.

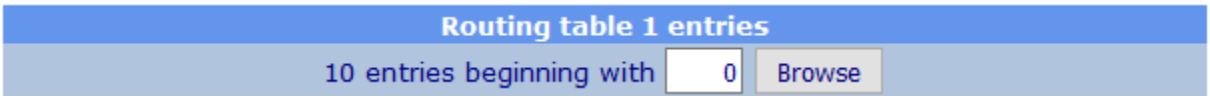
Two routing tables are available in the RTU-810, which are used to configure the routing for each of the 239 available radio slave addresses.

Whenever the RTU-810 sends out for a master functionality, the radio routing table will be considered to determine the routing to the given radio slave address. The routing table assignment is as follows:

If the RTU master mode is used to achieve point-to-point or point-to-multipoint connections, the routing table, which should be used can directly be selected for each slave.

For the serial master modes the used SCADA routing table will determine the radio routing table, which is used. If the serial interface is configured to use the SCADA routing table 1, this will also use radio routing table 1. If the SCADA routing table 2 is used, it will also use the radio routing table 2.

3.2.2.1 Browse page



As only 10 entries are displayed per page, but the routing table holds a maximum of 239 entries, the start index of each page can be entered here. A click on the "Browse" button will reload the page with the selected index number as first entry.

3.2.2.2 Routing entries

Slave Addr.	Routing (Relais addresses)														
	Enter radio relais to be used for each slave, ending with '0'														
1	2	7	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	4	12	8	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The routing table holds 239 lines, named "Slave Addr.". This address corresponds to the slave address of the radio where the data should be sent to.

Once a "0" is seen in the routing table, the relay chain ends and the original slave address will be added to it. This results that an entry with all zeroes will send the data directly to the addressed slave, without any relays in the chain.

If relay addresses are in the table, the data is sent exactly in the order given by the relay entries.

For the example above, the routing would be:

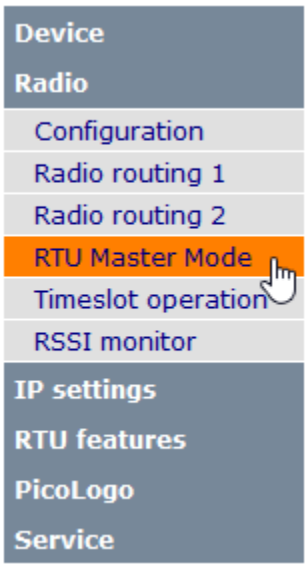
- Data , which should be sent to slave address 1 will take this route:
  RTU-810 → slave addr. 2 → slave addr. 7 → slave addr. 1
- Data , which should be sent to slave address 2 will take this route:
  RTU-810 → slave addr. 4 → slave addr. 12 → slave addr. 8 → slave addr. 2
- Data , which should be sent to slave address 3 will take this route:
  RTU-810 → slave addr. 3

The returning data from the slave will automatically use the reverse way back to the RTU-810.

Note that using radio relays are multiplying the transmission time by the number of relays! If data is sent to a slave via 2 relays it'll take 3 times longer than sending it directly to the slave!
Rule of thumb:

- One slave (direct transmission): 1 x the transmission time
- Two slaves (one relay): 2 x the transmission time
- Three slaves (two relays): 3 x the transmission time
- And so on

3.2.3 RTU master mode



If the RTU master mode is active (this is a software option , which must be purchased), the RTU-810 can operate as a master for a point-to-point to a point-to-four-point operation.

This means, the RTU-810 can transfer any number of its internal digital and analog I/O to up to four other RTU-810's.

The slave RTU-810 does not need to have also the RTU master option activated, this is only needed for the RTU-810 , which is the communication master to the slave RTU-810's.

The RTU-810 will check the slave RTU-810's if they meet the configured requirements, i.e. if they provide at least enough digital and analog I/O as configured for exchange in the master RTU-810. If the requirements are not fulfilled, the concerned connection will change to an error state.

In error state or when the radio link to a slave breaks, these actions will be performed:

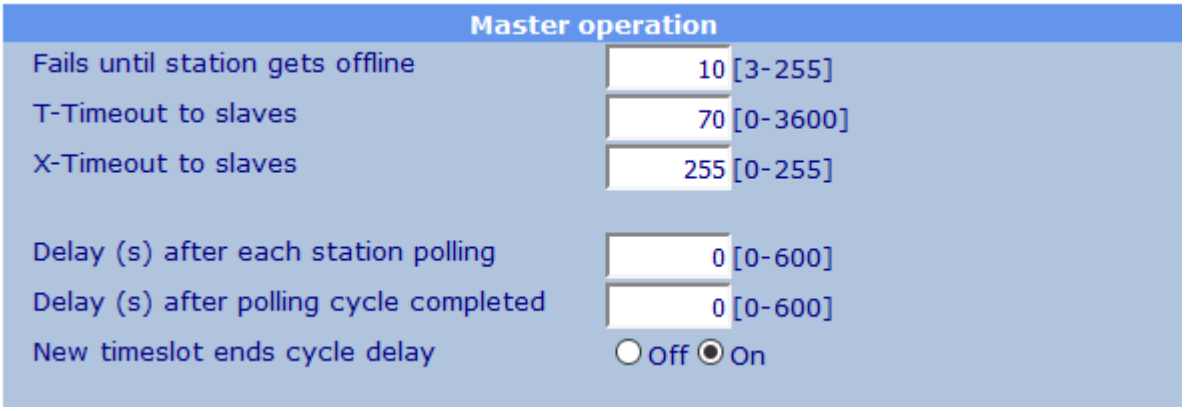
- The digital and analog outputs on the master RTU-810 , which are configured to the particular failed slave will be switched to OFF
- The digital and analog outputs of the respective slave are also written to "OFF" by the RTU-810 master
- If the radio link to the slave breaks, the slave will switch all its outputs to "OFF" if either the X-Timeout or the T-Timeout runs off (whatever happens first)
- The error will be displayed on the RTU master mode configuration page of the master RTU in the "ERROR messages" section
- The error will be written into the event logger of the RTU-810

The RTU-810 will periodically check if the slave is available and fulfills the requirements for the connection. In that case it will (re)establish the radio communication immediately.

The RTU-810 master will periodically exchange the configured I/O with the enabled slaves. The timeslot operation can be used to limit the transmission to a certain part of each minute, this fulfills the German requirements for time slot operation and can internationally be used to operate several systems on the same frequency within the same area.

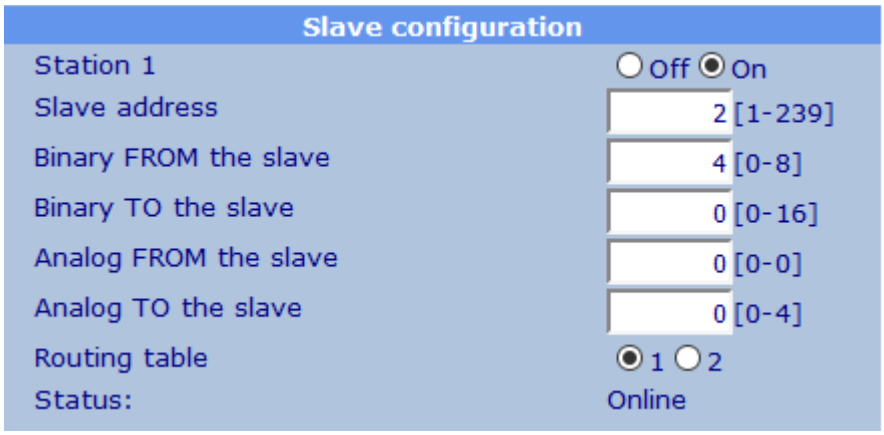
To fulfill duty cycle requirements a configurable pause can be inserted after each single polling and/or after each complete polling round.

3.2.3.1 Master operation



Fails until station gets offline	This is the number of consecutive communication failures before a station is tagged offline. When a station goes offline, this is an error state and all outputs will be deactivated (see 3.2.3 RTU master mode) Deafult is 10.
T-Timeout to slaves	This is the T-timeout value , which is transmitted to the slaves. After this seconds without being polled by the master, the slave will set its outputs to "OFF" state. 0 disables the T-timeout at the slaves.
X-Timeout to slaves	This is the X-timeout value , which is transmitted to the slaves. When the slave received this number of radio messages with the same radio layer address without being polled itself it will set its outputs to "OFF" state. 0 disables the X-timeout at the slaves.
Delay after each station polling	If this value is greater than 0, the RTU-810 will wait this number of seconds after a station is been polled before polling the next slave.
Delay after polling cycle completed	If this value is greater than 0, the RTU-810 will wait this number of seconds after a complete polling round (all slaves polled once) has been completed.
New timeslot ends cycle delay	If this is set to "On", in time slot operation the start of a new time slot will break an ongoing cycle delay. This can be used to poll each slave only once per time slot. It i.e. one time slot is active per minute, the cycle delay can be set to 70 seconds. After a completed polling cycle the polling delay starts, but with the start of the next time slot in the next minute the remaining cycle delay time will be cancelled and again one polling round will be performed.

3.2.3.2 Slave configuration



There are 4 identical sections to configure up to 4 slaves , which should be polled by the RTU-810.

Station 1-4	This enables or disabled the polling of that particular slave
Slave address	Radio address of the slave radio. This can be any valid radio address between 1 and 239.
Binary from the slave	This is the number of binary channels , which are transmitted from the remote slave to the master RTU-810.
Binary to the slave	This is the number of binary channels , which are transmitted from the master RTU-810 to the remote slave
Analog from the slave	This is the number of analog channels , which are transmitted from the remote slave to the master RTU-810
Analog to the slave	This is the number of analog channels , which are transmitted from the master RTU-810 to the remote slave
Routing table	Determines , which radio routing table should be used to route the radio telegrams to the particular slave. This will be changed from software version 2.40 on to use the SCADA routing table instead. The use of the SCADA routing table enables the ability to route the communication via IP networks.
Status	Status of the connection to the slave: • Offline: Slave is deactivated or tagged offline due to a broken radio link to that slave • Scanning: Slave is activated but is in progress of scanning. This state persists unless it's confirmed that it's an RTU-710 or 810 and the number of I/O to exchange with the master RTU is matching • Online: The slave is online and data will be exchanged

The numbers behind the parameter fields are the valid ranges for the entry. The range for the number of binary and analog I/O , which should be transferred are calculated from the number of available I/O of the RTU-810.

The I/O are used in the order of the configuration.

Example:

The master RTU-810 is a DA1 type and has 8 digital outputs on port C0. For slave 1, all 8 outputs are available. The valid range for the number of binary channels from the slave 1 to the master RTU-810 is given from 0-8 as the master RTU-810 has 8 available binary outputs:

The first slave will now be configured to transfer 4 binary channels to the RTU-810 master. After the change "Apply" will be clicked.

The binary inputs on port A0-A3 of the slave 1 will now be transferred to the outputs C0-C3 of the master RTU-810.

At the same time the valid range for the same parameter on the following slaves have been updated:

The following slaves now give the valid range from 0-4 channels as only 4 outputs are left available on the master RTU-810 (the first 4 are used for the slave 1).

We configure 2 binary outputs from the slave 2 to the master RTU-810:

In this case, the binary inputs A0 and A1 will be transferred to the outputs C4-C5 on the master RTU-810 (the outputs C0-C3 are already used for slave 1).

The following slaves 3 and 4 now show a valid range of 0-2 channels, as only 2 outputs are free on the RTU-810:

Due to the fact that this recalculation happens any time "Apply" is pressed, it is recommended to configure the I/O transfers for slave 1, then press "Apply" to update the information for the slaves 2-4.

Then slave 2 can be configured, pressing "Apply" after the configuration to update the ranges for the slaves 3 and 4 … and so on.

If all slaves are configured at once, it'll be possible to enter more I/O in total as are available on the master RTU-810. In this case, after pressing "Apply", an error message will indicate this fact on the page:

Error messages	
ERROR:	BINARY from slaves exceed available I/O

The communication to the slaves will not be established unless these errors are cleared!

3.2.3.3 Error messages

If there are any major configuration issues , which stops the master RTU mode from working, these are listed in the "Error messages" field:

Error messages	
ERROR:	ANALOG from slaves exceed available I/O

Note: Any events are logged into the Event Logger, also single station events , which do not affect the communication of the RTU-810 master to the other stations (like communication losses etc.

Event logger
31.03.17 16:46:46 Master mode: Station addr. 2 was set to ONLINE
31.03.17 16:46:45 Master mode: Station addr. 2 station has unsufficient number of I/O
31.03.17 16:40:02 Master mode: Station addr. 2 set to OFFLINE with failcounter: 10
31.03.17 16:39:53 Master mode: Station addr. 2 was set to ONLINE
31.03.17 16:39:39 Master mode: I/O configuration error - suspending polling
31.03.17 15:26:03 Master mode: I/O configuration error - suspending polling
31.03.17 15:07:21 Master mode: Station addr. 2 set to OFFLINE with failcounter: 10
31.03.17 15:07:12 Master mode: Station addr. 2 was set to ONLINE

3.2.3.4 RTU Master Mode via IP networks (Q2/3 2018)

From firmware version 2.50 on, the communication to the slave radios can be either the radio network or an IP network. This enables Point-to-(Multi)point connections also via LAN or WAN IP networks.

For this reason the use of the routing table will be changed to use the SCADA routing table 1 or 2, like for the serial master modes. The SCADA routing tables with its IP entry do have the ability to specify an IP address beside the radio slave address.

In the slave configuration a selection for each slave will be added to configure whether the slave is connected via radio network or via IP network.

If the radio network is used, the radio slave address will be determined from the entry in the selected SCADA routing table, for this slave address in a second step the associated radio routing table will be used.

As the SCADA routing table is defaulted to use a 1:1 entries for the radio slave addresses 1-239, a firmware update to version 2.40 will not break existing configurations.

3.2.4 RTU Master auto configuration (Q2/3 2018)

Apart from the detailed configuration of the RTU master mode as described above, the RTU Master Autoconfiguration will be a very easy way to do simple Point-to-(Multi)point setups with only a few actions.

The auto configuration works in these steps:

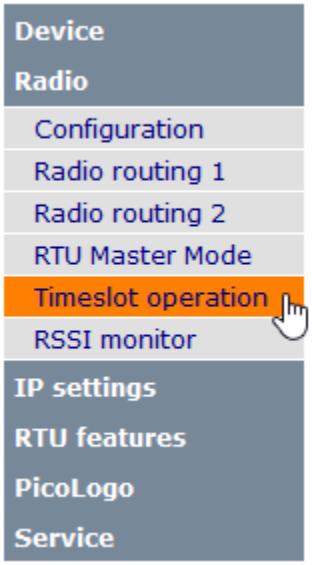
- The radio master and the up to 4 slaves are configured to use the same radio frequency
- On the auto configuration page, the number of slaves , which should be used will be entered
- The auto configuration will be started by the click of a button. The family address of the master will be set similar to its serial number to ensure a unique isolated network.
- The master radio is now searching for the first slave radio. One of the slave RTU-810 must be powered on and set into configuration mode (by pressing the configuration switch in the back of the device for one second until the "OK" LED starts flashing)
- Once the master can talk to the slave in configuration mode, it'll get the available information about the slave like available physical I/O etc.
- The slave will be automatically set to the same family address as the master and will have a slave address automatically assigned. Finally the configuration mode will be automatically ended on the slave device.
- This procedure will be repeated with any other slave RTU-810
- Once the last slave has been configured by the master, the master will automatically configure the I/O configuration to and from the slaves as follows:

  The binary and analog channels from the slaves to the master will be equally divided between them, depending on the available outputs on the master. Each slave will transfer the same number of binary and analog inputs to the master.

  The binary and analog channels from the master to the slaves will be configured in one of two configurable methods:
  Either, the binary outputs and analog outputs from the slaves will be equally divided between them, depending on the available inputs on the master. The master will transfer the same number of binary and analog inputs to the slaves.
  Or, the binary and analog channels from the master are sent to all slaves in parallel so that each slave will get the same data from the master.
- The radio polling will finally start.

The automatically configured setup will be available in the RTU Master Mode configuration page afterwards. It can be modified and "fine-tuned" anytime afterwards.

3.2.5 Timeslot operation



The RTU-810 fully supports time slot operation mode. For TS mode, each minute is divided into fixed slices of 6 seconds, staring with second 0. This is giving in total 10 time slots (TS 1-10).

Each 6 second time slot can be divided into 4 sub-timeslots with 1.5 seconds each, which is pretty sufficient for running a point-to-point-connection.

The time slot method fully fulfills the requirements for operation on the time slot frequencies in Germany.

In other countries the time slot operation can mainly be used to run multiple radio networks on the same frequency and within the same range. As the time slots are synchronized to fixed parts of a minute, networks on different time slots will not collide with each other.

Time slot operation requires a precise time source to synchronize the internal time base of the RTU-810. Possible sources are:

- SNTP: A SNTP time server , which is available via Ethernet connection
- GPS via PTS-100: This is a time server module , which includes a GPS antenna for time synchronization. The connection to the RTU-810 is a DCF-77 coded signal, hence it requires the DCF or PTS option at the RTU-810
- DCF-77: The DCF option is a hardware option and includes a socket for an active DCF-77 antenna and the antenna itself. The range of the DCF-77 reception is up to 1000km around Frankfurt/Germany.
  The PTS option includes only the socket, which can be used to connect a PTS-100 for generating the time sync information from a GPS signal.

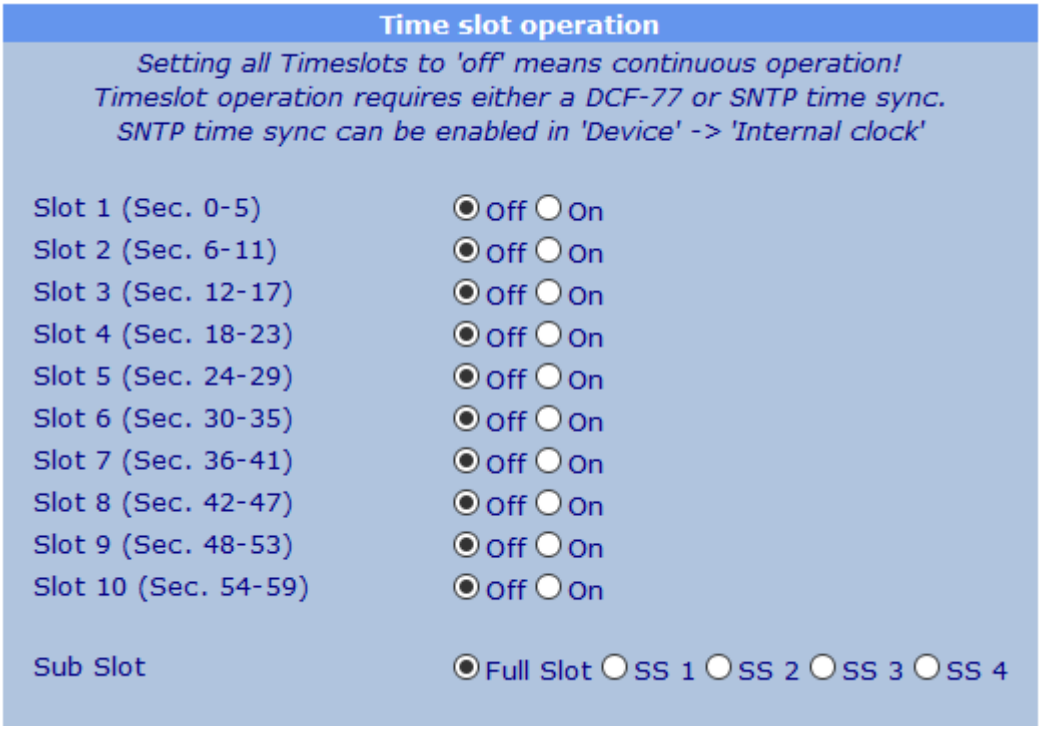
If both, SNTP and DCF time sources are available, the RTU-810 uses SNTP first. When a sync with SNTP is not successful, it'll use the DCF source. After selecting a sync source it will use this source as long as a synchronization fails, it'll then try to use the other source.

The RTU-810 will automatically calibrate its internal time base to the time source, which gives it a 3 days buffer in case all time sources will fail. During this reserve time the time slot is continuously capped at its start and stop to ensure that the deviation will not collide into neighbor time slots.

The time source is also used to set the internal battery buffered real time clock. However, the RTC does not maintain the precision to maintain the precise time base during power down. After a power down or device reset the RTU-810 must have a valid time sync to continue time slot operation.

In a radio network only the radio master(s) need a time synchronization source for time slot operation. Slave devices , which are not transmitting without being polled by a master do not need a time synchronization source as the information about time slot availability is passed from the master via the radio protocol.

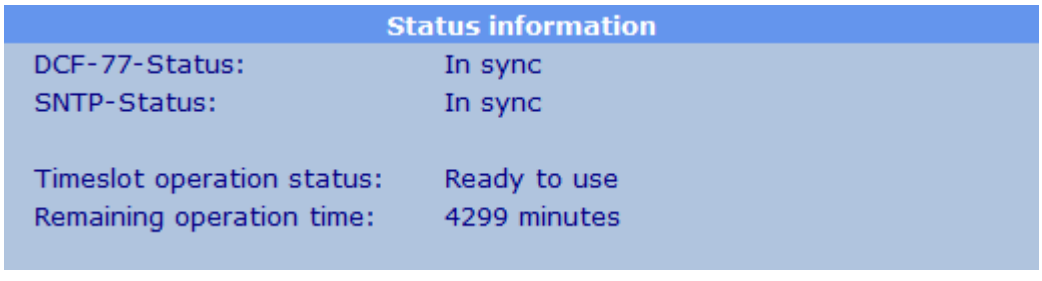
3.2.5.1 Time slot configuration



This selects any of the 10 available time slots for radio operation. If all time slots are set to "off", the RTU-810 operates in continuous operation.

Each 6 seconds time slot can be further divided into 4 sub slots with 1.5 seconds each. If several time slots are selected with also a sub slot configured, the sub slot will be used in each active main time slot.

3.2.5.2 Status information



This gives information about the availability and sync status of the time sources, the status of the time slot operation and the remaining operation time in case all time sync sources will fail.

3.2.6 RSSI monitor

The RSSI monitor acts as a network monitor and gives a complete overview over the radio network. Even more, it lists every station it can receive , which is using the same radio layer address.

This gives also a feedback about station in neighbor radio networks on the same frequency to see possible interferences if the networks are running in continuous operation or to determine if a station in a neighbor network can be used as a radio repeater for the own network if time slot operation is used.

For clarification: Any RTU-810 can only receive and "see" other Piciorgros devices if they are delivered to the same customer. It is not possible to monitor devices from other customers on the same frequency or to communicate with them. The RSSI monitor will not list any "alien" devices outside the own radio networks.

Radio RSSI monitor

Station	RSSI	Last contact	Low RSSI count
Master 1	- 98 dBm	15.03.17 16:06:11	222
Substation 2	- 50 dBm	03.04.17 14:18:33	255
Substation 3	- 50 dBm	17.03.17 16:59:59	0
Substation 4	- 50 dBm	15.03.17 16:05:51	0
Substation 177	- 50 dBm	15.03.17 16:26:29	0
Substation 239	- 50 dBm	08.02.17 14:57:06	0

The RSSI monitor lists each station it has ever seen in a list, started with master stations and followed with slave (sub) stations, ordered by radio address.

The time stamp of the last received data from the particular station is stored along with the RSSI value of the reception. A threshold value can be defined, and any data , which is received with a lower field strength as the defined threshold will count up the "Low RSSI count". This can be used to check whether the receive field strength of a station drops below a certain value from time to time.

Note: Old TRM-700/710 master radios can use a master address 0, although this address is not valid anymore in newer devices. For that reason a "Master 0" can appear in the list!

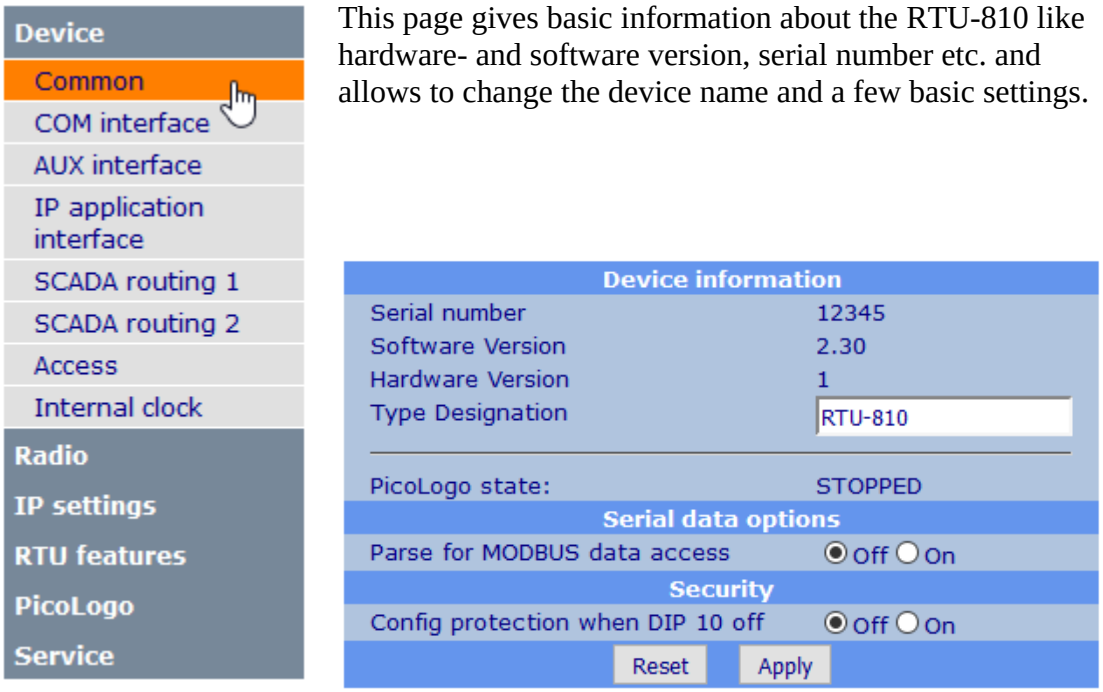
3.2.6.1 Low RSSI count threshold and stats reset

This section can be used to set the threshold for the Low-RSSI counts in a range between -60 and -140 dBm.

Note: Clicking "Apply" on this page will reset all collected statistics! If the RSSi monitor values just should be checked and not erased, you need to leave the page by clicking on a different page in the menu or via the "Logout" button.

3.3 Device configuration

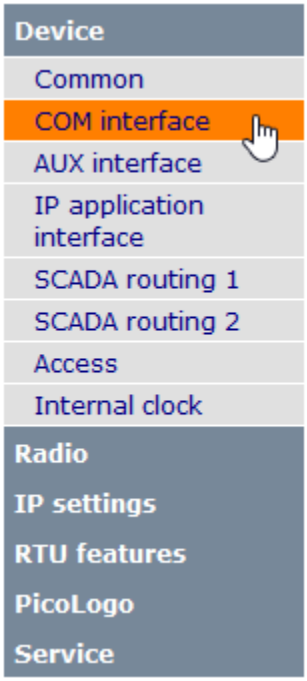
3.3.1 Common



This page gives basic information about the RTU-810 like hardware- and software version, serial number etc. and allows to change the device name and a few basic settings.

Serial number	The serial number of this RTU-810
Software version	The version of the main firmware.
Hardware version	The Hardware release information.
Type designation	The name of this device. Any name can be given here, i.e. the location of the device. This name is used in status and RSSI messages and can be used to identify the device. It can also be used as a placeholder in PicoLogo TextSend blocks.
PicoLogo state	State of the internal PicoLogo application. This can be "Stopped", "Running" or "Error".
Parse for MODBUS data access	If this is set to "on", any telegram , which passes over the serial interfaces or IP Application Interface will be checked if it's a MODBUS data telegram. If yes, the RTU-810 will process it locally if the MODBUS address is matching the address set in the RTU-810.
Config protection	If this is set to "on", configuration files and PicoLogo programs can't be sent to the RTU-810 via IPLoader or PicoLogo Editor unless the device is in programming mode (the key behind the small hole on the back of the TMO must be pressed for 1 second until the OK-LED flashes).

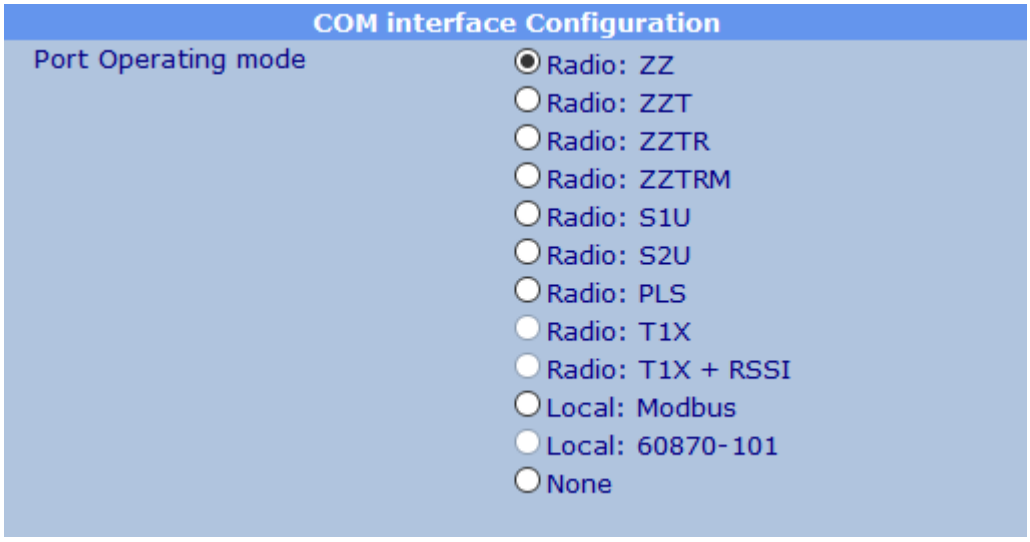
3.3.2 COM and AUX interface



The settings of the "COM" and "AUX" serial interfaces are identical, therefore both are described in this chapter.

Both interfaces can operate completely separated from each other, so two different services or serial protocols can run on a single RTU-810.

3.3.2.1 Interface configuration



Port operating mode

Selects the operating mode for this serial interface:

- Radio ZZ: This serial interface acts as a master interface in ZZ mode (i.e. operation for OPC server)

- Radio ZZT: This is the predecessor of the ZZTR mode. A ZZT master can only talk to S2U radios. The mode is fully transparent, but does not support routing like the ZZTR mode.
  Any telegram sent by a ZZT master radio will be put out on any S2U device in range which slave address is by 1 higher than the master address of the ZZT device. Any S2U radio can respond back within 6 seconds (or for the remaining time slot).
  This mode is supported for compatibility reasons, for new networks it is highly recommended to use the ZZTR mode which has more capabilities.

- Radio ZZTR: This serial interface acts as a master interface in ZZTR mode. This is a transparent mode , which includes protocol address detection and routing by using other radios as repeaters.
  In time slot operation the CTS hardware handshake output will be set to "Stop" if no active timeslot is available.
  ZZTR can be used for transparent data communication to RTU-810 interfaces in "S1U" mode and to communicate with RTU-810's via the MODBUS-RTU protocol.
- Radio ZZTRM: Same as ZZTR, but the data via the serial interface is treated as MODBUS-RTU data. The RTU-810 will answer them locally if a MODBUS-RTU telegram addressed to the local MODBUS address of the RTU-810 is seen on the interface.
  In time slot operation the CTS hardware handshake output will be set to "Stop" if no active timeslot is available.

- Radio S1U: This is the slave operating mode for serial data transmission. The respective master mode is ZZ (OPC-Server) or ZZTR/ZZTRM.

- Radio S2U: This is the slave mode for ZZT master stations.
  This mode is supported for compatibility reasons, for new networks it is highly recommended to use the ZZTR/S1U mode which has more capabilities.

- Radio PLS: This has the same functionality as the S1U mode, but supports special broadcast messages which have been built for car-park routing systems.
  This mode is only available for the serial COM port!
  A detailed description of the additional broadcast codes is available in German language only.

- Radio T1X: This mode can be activated once the device is set to "T1X mode" in radio configuration. T1X is a pure peer-to-peer mode without any routing capabilities or individual device addressesDiese F. Any data sent by a T1X device will be put out on the interface of any other T1X devices with the same radio layer address which can receive the data.

- Radio T1X + RSSI: This is the same as "Radio T1X", but each datagram is extended by the receiving device, adding one byte which contains the reception field strength of this datagram. The RSSI is a –dBm value by default, if the device is set to "TRM/RTU-710 compatibility mode" in the radio configuration, the byte gives a percentage value of 0-100 instead.

- Local MODBUS: The port is used to access the RTU-810 locally via MODBUS-RTU protocol.

- Local 60870-101 (Future option): If the RTU-810/DAx has the IEC60870-option active, the RTU-810 can be locally accessed by 60870-5-101 protocol via this interface.
  Note that only one communication path can be used for IEC-60870.
  To use a serial interface for local access it must be selected in "IEC60870" → "Basic settings" → "101 interface".

- None: This port is not used, incoming data will be discarded.

3.3.2.2 Radio routing via IP networks (Q2/3 2018)

From firmware version 2.50 on, the communication to the slave radios can be either the radio network or an IP network.

A selection will be added for each interface , which can select these routing options:

- Radio
- IP
- Mixed radio/IP

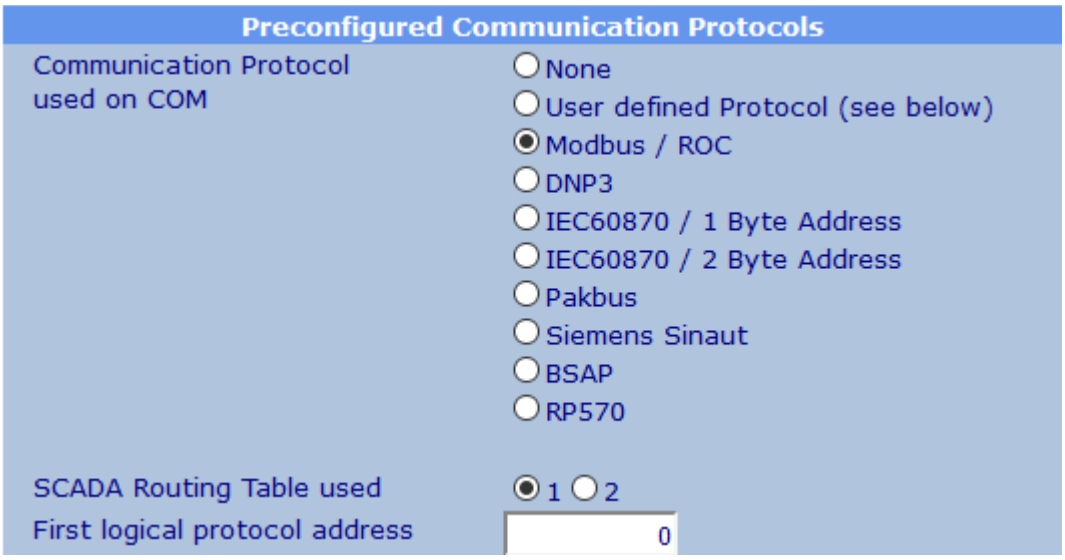
If "Radio" is selected, all traffic to this interface is will use the radio slave address , which is determined from the selected SCADA routing table. If the entry is "0", no data will be sent.

If "IP" is selected, all traffic to this interface will use the IP address , which is determined from the selected SCADA routing table. If the IP address is "0.0.0.0", no data will be sent.

In "Mixed" mode, the slaves can be partially in the radio network and partially in the IP network. If the radio slave address , which is determined from the selected SCADA routing table is not 0, the data will be sent to the radio, using the associated radio routing table.
If the radio slave address is "0", the data will be sent to the IP address given in the SCADA routing table. If the IP address is also "0.0.0.0", no data will be sent.

Apart from connecting to slaves , which are locally connected in a LAN or WAN network, this can be used in conjunction with the RGW-810 radio gateway to connect to remote radio networks via IP connection.

3.3.2.3 Preconfigured Communication Protocols



When the interface is set to "ZZTR" or "ZZTRM" mode, it is normally connected to a SCADA master, primary RTU etc.

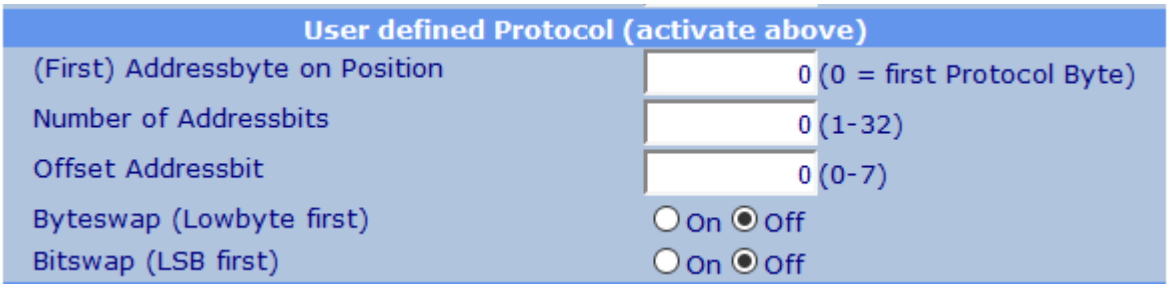
This means that the connected equipment will likely send out data , which is addressed to multiple slave RTU-810 in the field.

To determine the radio slave address of the RTU-810 to , which the data telegram should be sent, including the radio routing to reach the slave, a conversion from a SCADA address to a slave radio address must be done.

The RTU-810 can check the received data on the interface for a contained SCADA address and use this address as an index to one of the routing tables. To do this, the RTU-810 needs to know what SCADA protocol will run on the particular interface.

Communication protocol used on COM/AUX	Selects a known SCADA protocol. "User defined protocol" gives the ability to define the position and format of the address bits/bytes inside the data in the next paragraph.
SCADA Routing table used	Selects , which SCADA routing table should be used for the conversion from the SCADA address to the target slave radio address.
First logical protocol address	This can be used to shift the SCADA address range into the routing table index range from 0-1023. Many SCADA applications are using addresses , which are not in the range between 0 and 1023. The value in this field is subtracted from the SCADA address value , which was found in the data. Example: If the protocol detection returns a SCADA address of 10005 and this field contains 10000 as the first logical protocol address, the index 5 of the selected routing table is used to get the radio address where the data must be sent to.

3.3.2.4 User defined protocol

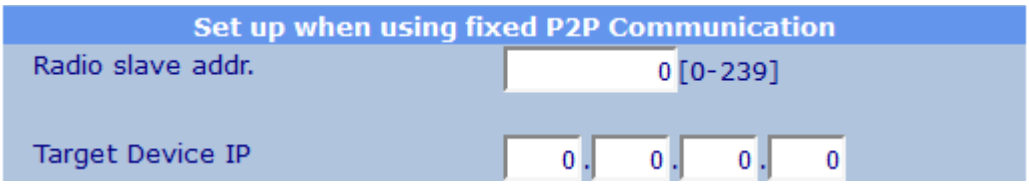


If a propriety protocol is used , which can't be selected by one of the protocol presets, the position of the address bits/bytes can be manually defined.

To use this feature, the "User defined protocol" must be selected in the Preconfigured Communication Protocols:

First addressbyte on position	Gives the Byte position where the address field starts. The first byte in a data telegram starts to count with 0
Number of address bits	This selects how many bits the address field uses. Usual the address is given in full bytes, which is 8,16,24 or 32 bits. But any number of bits between 1 and 32 will work here in case the address is only a part of a byte or a word.
Offset address bit	In case the address does not start with bit 0 (LSB) of a byte, an offset to the first bit of the address field can be defined here. The offset will be used to start the address field in the defined first address byte.
Byteswap	If this is set to "On", the first byte in the protocol data is treated as the least significant byte. In this case the address "10" in a 16-bit-field would be HEX "0A 00". If this option is set to "Off", the first byte is the most significant byte. The address "10" in this case would be HEX "00 0A".
Bitswap	If this is activated, the first bit in the defined address field will be treated as the least significant bit. If the option is set to "Off", the first bit will be treated as most significant bit.

3.3.2.5 Set up when using fixed P2P communication



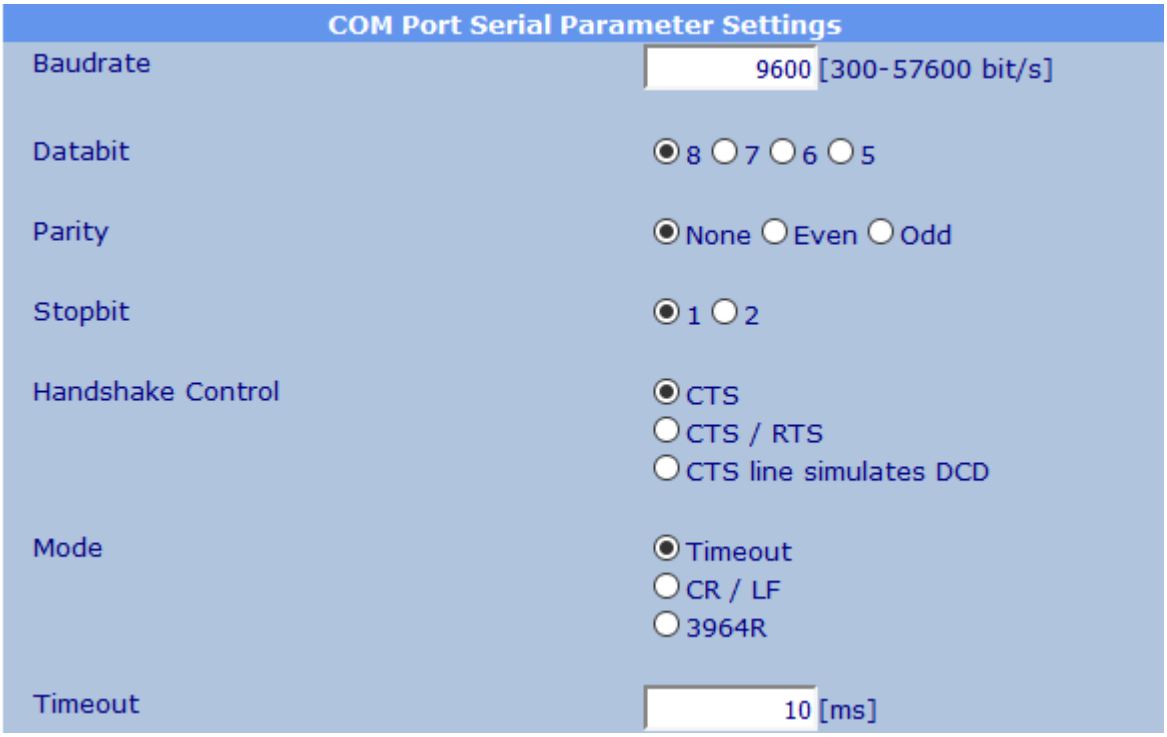
When the interface is set to "ZZTR" or "ZZTRM" mode, it requires the protocol address detection to extract the SCADA address from the data and the routing table to determine the radio slave address to , which the data should be sent.

This option can be used to override these mechanisms. If all fields are set to "0", the protocol extraction/routing tables are used for interfaces in master mode, and the address learning will be used for interfaces in slave mode.

Any given radio slave or IP address in this section overrides this and any data received on the interface will always be sent to just this radio slave or IP address.

3.3.2.6 Port serial parameter settings

This configures the serial parameters for the serial interface:



Baudrate	The baud rate of the interface in the range between 300 and 57600 bit/s
Databit	The number of data bits per byte
Parity	The parity bit setting
Stopbit	The number of stop bits per byte

Handshake control

The CTS output of the RTU-810 is always operated, regardless if it's needed/connected in the application or not. These settings are available **for RS-232 type interfaces**:

- CTS: No incoming handshake, the CTS output is operating (but must not be connected if not required by the connected equipment)
- CTS/RTS: The output of the serial interface can be paused by the RTS input. Use this option with care, if the interface is put on hold for a too long time incoming data for this interface can time out and get lost in the internal buffers of the RTU-810!
- CTS line simulates DCD: Some older RTU equipment may need a DCD signal , which is activated before any data is transmitted to the equipment and , which is deactivated after the transmission.
  To connect these kind of equipment, the CTS output of the RTU-810 can be used to simulate this DCD behavior.

  If this option is selected and applied, two additional fields will appear on the page to set the time between activation and data output and between end of data and the deactivation.

If the interface is a RS-422/485 interface on the COM port, a selection between RS-485 and RS-422 mode is presented instead:

Mode

- Timeout: This is the default and the setting for fully transparent communication. Once the first character is received on the interface, a timer is set counting down milliseconds.
  With each character , which is received before this timer runs out, the character is added to a buffer and the timer is reset to its initial value.
  Once no character is received and the timer runs down to 0, the buffered data is treated as a complete data telegram and will be processed by the RTU-810.
- CR/LF: Received bytes are collected in the input buffer until a "CR" is received. The "CR" is not added to the data and the buffer will be processed after the reception of the "CR" character.
  Trailing "LF" characters are ignored.
  Any transmitted data on the interface will end with "CRLF".
- 3964R: Local point-to-point protocol for serial lines, mostly used by Siemens PLC's.

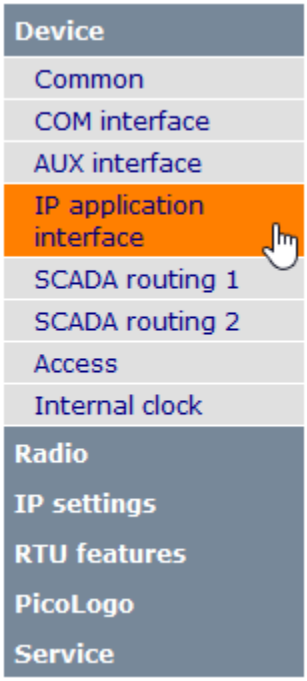
Timeout

In case the "Timeout" mode is selected, this gives the number of milliseconds for the timer. Once a gap of more than this time appears after a received character, the reception is treated as complete and the buffer will be processed by the RTU-810.

The default is 10 times of a complete character with the selected baud rate. If the baud rate is changed, the timeout value will be automatically recalculated by the RTU-810 to meet the 10-characters-default.

The timeout can also be modified manually. This may be necessary if a slow SCADA PC is connected , which sometimes delays the data flow on its interface due to task switching. In this case a value of 50ms @ 9600 bit/s is sufficient in most cases.

3.3.3 IP Application Interface



The IP Application interface is basically treated the same way as the two serial interfaces with the exception that the physics is TCP or UDP via Ethernet in this case.

Also the settings and functions are similar to the serial interfaces, including the address extraction of SCADA addresses and the use of the routing tables to determine the target RTU-810 where the data should be sent to.

It is important to know that the RTU-810 is the IP endpoint of the connection! The IP application interface can't be used for IP routing or for applications , which requires different slaves being addressed by different IP addresses.

The easiest way to explain the IP application interface is to name it as a serial interface but with a different physics. It can i.e. be used to use an IP LAN network where otherwise a very long serial cable would be needed to connect the RTU-810 to equipment.

The IP Application Interface can be used with the SIG-810 to remotely connect a serial connection to the RTU-810 via a LAN or WAN connection.

3.3.3.1 Interface status

IP Application Interface status	
Interface status	CONNECTED to 192.168.0.21:65278

This line shows if the interface is currently connected to a device via IP. In this case the IP address of the remote device and the remote port is shown.

3.3.3.2 Interface configuration

This is the same as for the serial interfaces, please refer to "3.3.2.1 Interface configuration"

3.3.3.3 Preconfigured Communication Protocols

This is the same as for the serial interfaces, please refer to "3.3.2.3 Preconfigured Communication Protocols"

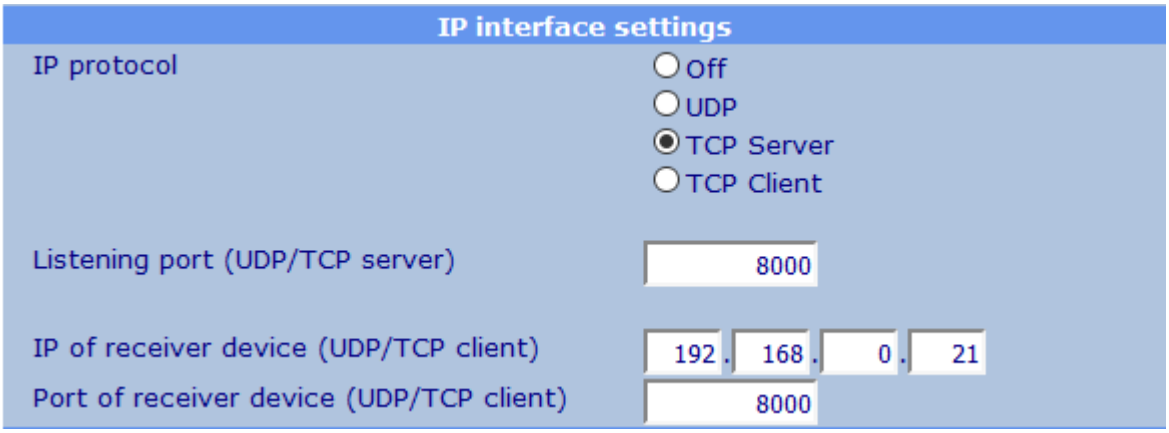
3.3.3.4 User defined protocol

This is the same as for the serial interfaces, please refer to "3.3.2.4 User defined protocol"

3.3.3.5 Set up when using fixed P2P communication

This is the same as for the serial interfaces, please refer to "3.3.2.5 Set up when using fixed P2P communication"

3.3.3.6 IP interface settings



IP protocol

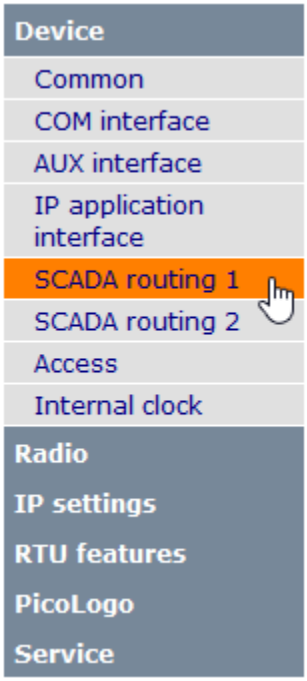
- Off: The interface is deactivated
- UDP: The interface sends and receives data via UDP protocol. The listening port and the IP and port of the receiver device (where the interface will send data to) must be specified further down.
- TCP server: The interface acts as a TCP server, this means it opens a port and waits for an incoming TCP connection. Only the listening port must be defined further down.
- TCP client: The interface will act as a TCP client and opens a TCP connection to a given IP address and port. If the connection can't be established, the RTU-810 will periodically retry to establish the connection.
  The IP address and port of the receiver device must be configured further down.

Listening port This specified the port on , which the interface listens for incoming UDP packets (if set to "UDP") or an incoming TCP connection (if set to "TCP server")

IP of receiver device IP address to , which the UDP data is sent (if set to "UDP") or to , which an outgoing TCP connection is established (if set to "TCP client")

Port of receiver device IP port to , which the UDP data is sent (if set to "UDP") or to , which an outgoing TCP connection is established (if set to "TCP client")

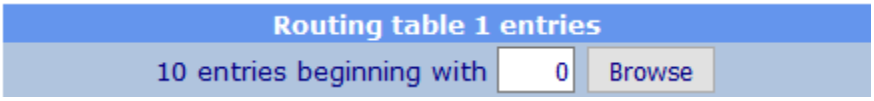
3.3.4 SCADA Routing 1/2



A routing table can hold up to 1024 entries with a radio address and an IP address in each entry. 10-20 entries are displayed per page.

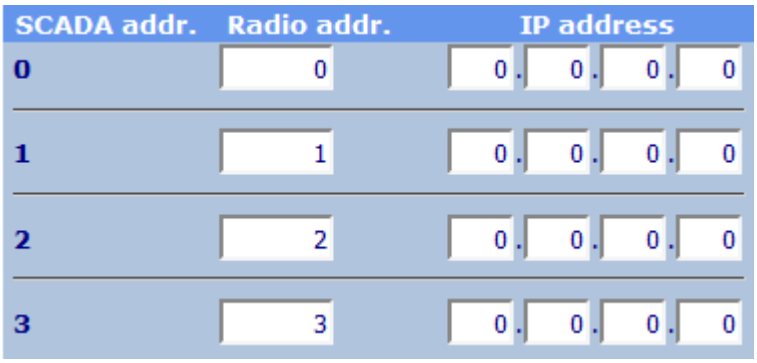
The population of the "Radio addr." and "IP address" fields are optional. By default, the SCADA addresses 1-239 are preset with radio slave addresses 1-239, all other fields are set to "0".

3.3.4.1 Browse page



As only 10-20 entries are displayed per page, but the routing table holds a maximum of 1024 entries, the start index of each page can be entered here. A click on the "Browse" button will reload the page with the selected index number as first entry.

3.3.4.2 Routing table entries



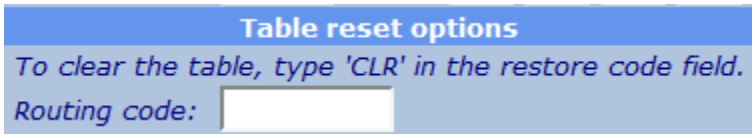
The routing table contains a line per possible slave device:

SCADA addr.	This is the index of the protocol address extraction , which was done on an interface. If the "first logical address" in the interface configuration is 0, the "SCADA addr." number exactly matches the SCADA address , which is extracted from the data received by the interface. If i.e. a MODBUS data is addressed to slave address 5, the line with SCADA addr. "5" must contain the radio address and/or IP address of the slave RTU-810 where the data for the MODBUS address 5 should be sent to. If the "first logical address" is set to a certain value, the ID number is the result of the extracted SCADA address minus the value of the "first logical address" field in the interface.
IP address	IP address of the slave RTU-810 where the data for the specified SCADA address should be sent to if "IP" is used as transmission method (future option)

Note:
The radio address , which is determined by the SCADA protocol detection using the SCADA routing table is the final radio address of the slave radio where the data should be sent to.

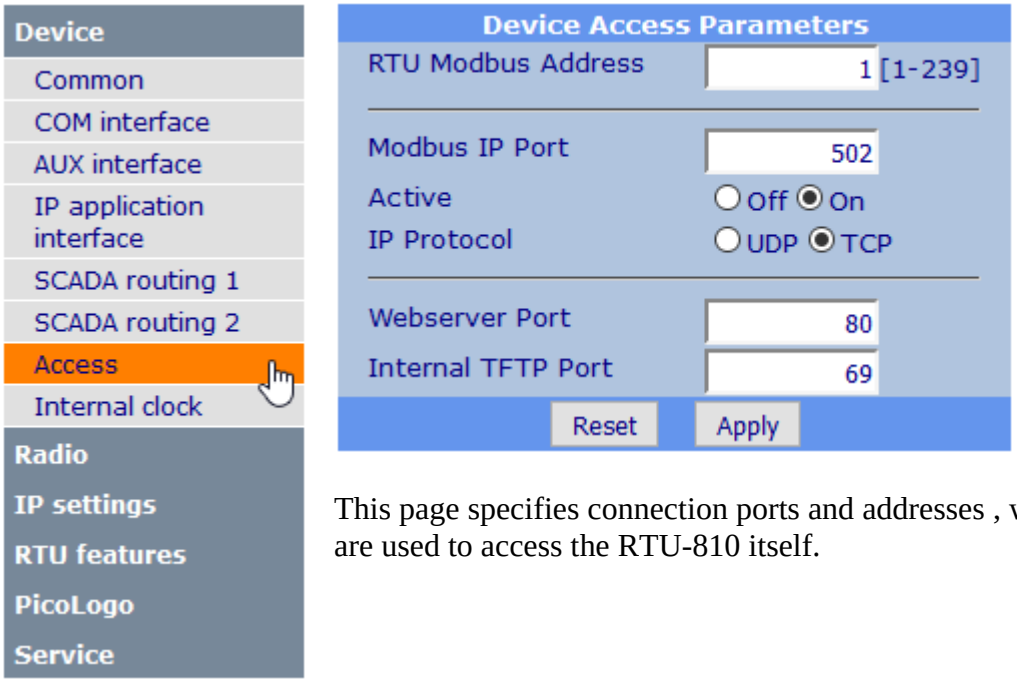
For each radio slave address, a radio routing table will be used to enable the use of radio relays. Any radio address , which is determined here will use the appropriate radio routing table for the final way to the slave radio. SCADA routing table 1 will use radio routing table 1 and SCADA routing table 2 will use radio routing table 2.

3.3.4.3 Table reset option



When "CLR" is types in this field and then "Apply" is clicked, the SCADA routing table will be restored to default state. This means a 1:1 routing for SCADA addresses 1-239 to radio slave addresses, all higher radio addresses and the IP addresses are set to 0.

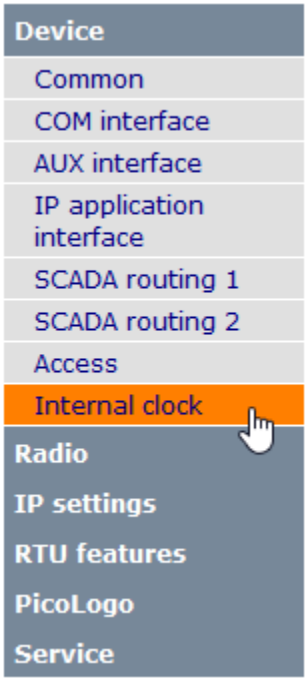
3.3.5 Access



This page specifies connection ports and addresses , which are used to access the RTU-810 itself.

RTU MODBUS address	The address , which is used to access the RTU-810 via MODBUS-RTU protocol, either via local serial port(s) or via radio network.
MODBUS IP port	The port number on , which the RTU-810 listens to incoming MODBUS/IP connections (TCP) or data (UDP). A change of the port needs a device restart.
MODBUS IP Active	Turns the MODBUS/IP function On or Off. A change needs a device restart.
MODBUS IP protocol	Selects the use of UDP or TCP for the MODBUS/IP access. Default is TCP. A change needs a device restart.
Webserver port	The port to , which the embedded configuration webserver is listening. Default is port 80, a change needs a device restart.
Internal TFTP port	The port to , which the internal TFTP server is listening. This is needed for firmware updates, saving and restoring the configuration and for transferring PicoLogo applications into the RTU-810. Default is port 69, a change needs a device restart.

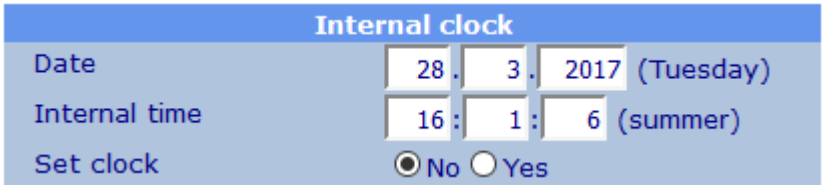
3.3.6 Internal clock



This page configures the internal, battery backed up real time clock and time synchronization methods.

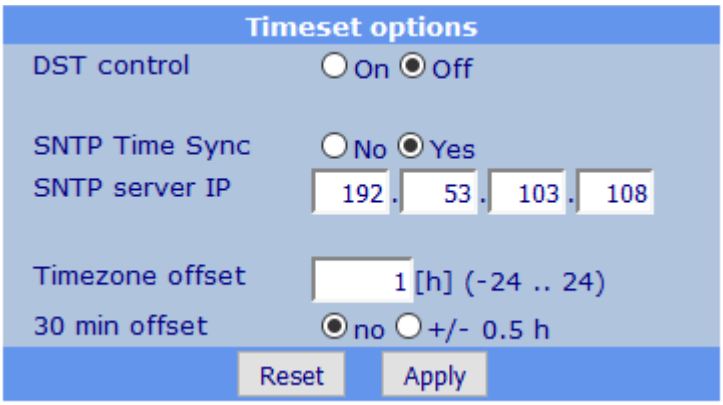
With each call of this page the current date and time of the internal clock is shown / updated.

3.3.6.1 Internal clock



Date	The current date in format DD.MM.YYYY To set the date to a different value, the data can be entered in these fields.
Internal time	Current time in 24h format. It requires a page reload to update this information. To set the time to a different value, the data can be entered in these fields.
Set clock	If this is set to "Yes", the values in the "date" and "time" fields will be written into the clock at the moment the "Apply" button on the bottom of this page is clicked.

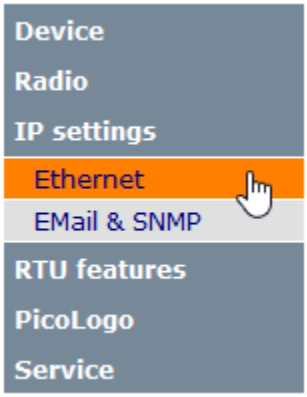
3.3.6.2 Timeset options

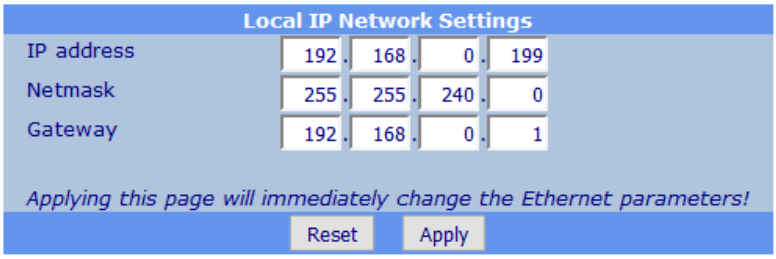


DST control	If this is set to "On", the clock will automatically change from summer time to winter time. The transition day will be the last Sunday in March 2:00h for winter→summer and the last Sunday in October 3:00 for summer→winter (European dates).
SNTP time sync	If set to "Yes", the RTU-810 will use a SNTP time server for synchronizing the internal real time clock and the internal time slot synchronization. **This method is eligible as a clock source for time slot operation mode!**
SNTP server IP	The IP address of the SNTP time server. This server must be available via Ethernet IP connection of the RTU-810 and must be reachable in the LAN where the RTU-810 is connected to (local time server) or via WAN/internet connection (public time server). The default value is a public SNTP server (ntp1.ptb.de) in Germany.
Timezone offset	Offset from GMT for the clock of the RTU-810 in full hours.
30 min offset	Adds 0.5 hours to the given offset for the use in some countries.

3.4 IP settings

3.4.1 Ethernet





This sets the IP parameters of the Ethernet interface of the RTU-810.

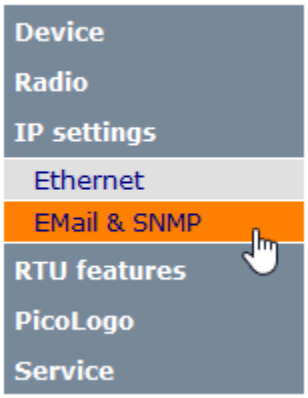
IP address	The IPv4 address of the Ethernet interface of the RTU-810
Netmask	The net mask of the Ethernet interface of the RTU-810
Gateway	Specifies the standard gateway for the RTU-810 for data sent via the Ethernet interface. This is only needed if the RTU-810 should be accessed via Ethernet from other IP networks or the internet or should access the internet or other devices in the LAN by itself using a router. In this case the IP address of the router must be entered here. The standard gateway must be in the same IP subnet as the Ethernet port of the RTU-810.

Once the "Apply" button is pressed the IP parameters will be immediately changed! To access the RTU-810 afterwards the web browser must be pointed to the new IP address of the RTU-810.

If the RTU-810 is "lost" during a misconfiguration or the IP address of a RTU-810 is not known, please refer to "0

Temporary reset to default IP address".

3.4.2 Email & SNMP

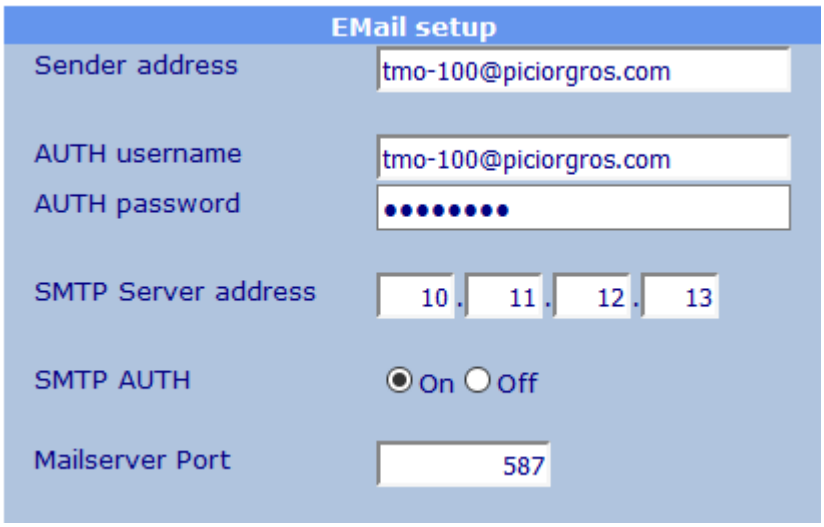


The RTU-810 can send out messages via eMail via a PicoLogo application (i.e. Alarm messages).

To have this function working via the TextSend block, the email sever where the mail is passed to, must be configured in the RTU-810.

The RTU-810 can also send Logger events, cold start messages and periodic alive messages to a syslog server via SNMP trap.

3.4.2.1 Email setup



Sender address	This is the email address , which is used in the "From:" field of the email and in the email envelope during transmission to the email server.
AUTH username	This is the username , which is used for SMTP authentication
AUTH password	This is the password , which is used for SMTP authentication
SMTP server address	This is the IP address of the server , which handles outgoing mails. The server must be reachable for the RTU-810, either via Ethernet connection.
SMTP AUTH	Enables SMTH AUTH to authenticate towards the email server
Mailserver port	The port on , which the RTU-810 connects to the mail server. This is usually the SMTP port 25 or the transmission port 587.

Note: The RTU-810 does not support TLS or SSL protocols, nor SMTP AUTH with challenge response mechanisms.

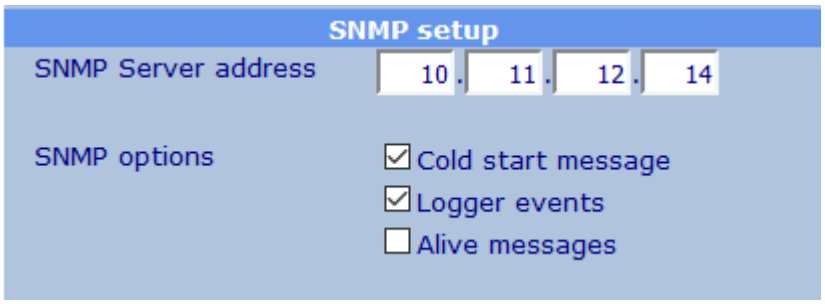
The communication with the email server will be plain unencrypted, the SMTP AUTH is using AUTH-Login with the password Base-64 encoded.

Public email servers like Gmail can't mostly be used directly due to their request for TSL/SSL encrypted connections. In this case a local email server should be used where connections from local clients on the LAN are allowed in plain.

3.4.2.2 SNMP setup

The RTU-810 can send logger messages and events to a SYSLOG server via the SNMP trap protocol. It does not support management by SNMP!

Messages are sent via UDP to port 162 of the given syslog server IP address.

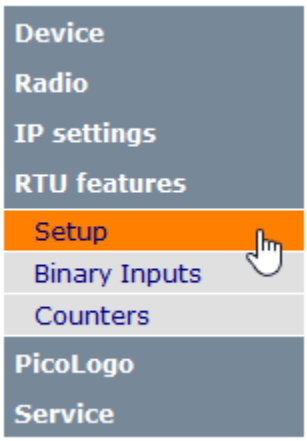


SNMP server address	The IP address of the syslog server where the RTU-810 sends the SNMP-trap messages to
SNMP options	• Cold start messages: Sends a startup message to the syslog server each time the RTU-810 boots • Logger events: Sends all events , which are entered into the Eventlogger additionally to the syslog server • Alive messages: Sends an "ALIVE" message every 15 minutes to the syslog server

Sending text alarm messages from PicoLogo via SNMP does not use this settings, the SNMP server IP address is passed via the PicoLogo TextSend block.

3.5 RTU features (not for TRM-810)

3.5.1 Setup



The "Setup" menu configures the basic features of the integrated inputs and outputs.

3.5.1.1 RTU information

RTU Information	
Binary inputs	16
Analog inputs	4
Binary outputs	8
Analog outputs	0
Counters	16

This is a summary about the physically available inputs, outputs and counter inputs of the RTU-810. This information includes connected expansion modules.

3.5.1.2 RTU options

RTU options		
Process ZZTR telegrams as MODBUS	○ On ◉ Off	
RTU Modbus Address	1	[1-239]

Process ZZTR telegrams as MODBUS	To access RTU-810 radio networks from a SCADA master, a TRM-810 or RTU-810 with the serial master mode ZZTR/ZZTRM can be used. If the RTU-810 receives a ZZTR telegram, it will check if this is a MODBUS-RTU datagram when this option is turned on. If the MODBUS address is matching the RTU-810's MODBUS address, it'll answer the data.
RTU MODBUS address	This is the MODBUS address of the RTU-810. This setting is the same as configured in "3.3.5 Access". Note that the MODBUS address and the radio slave address are different things. Although it's recommended to keep them the same, they can be configured different. In this case, the radio telegram must be sent to the radio slave address of the RTU-810, and the embedded MODBUS-RTU-telegram must be addressed to the MODBUS address of the RTU-810.

3.5.1.3 Virtual analog I/O

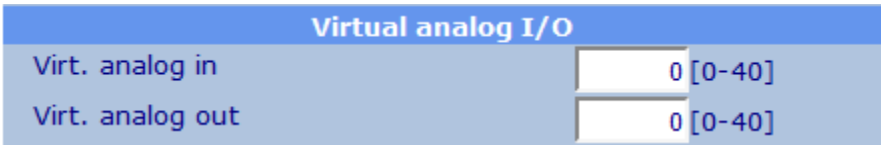
In some cases, some values , which are provided by locally connected equipment should be transmitted via the radio network to the control room.

The most protocols (MODBUS, IEC60870 and DNP3) know analog values. They are 16 bit wide in the RTU-810, and the RTU-810 supports analog inputs and analog outputs.

As the RTU-810 can be a MODBUS slave or can locally poll a MODBUS slave device like a meter via PicoLogo, the connected equipment can store data into up to 40 registers in the RTU-810. These registers are provided as analog inputs or outputs to the outside world and they can be accessed by the SCADA master like normal analog values.

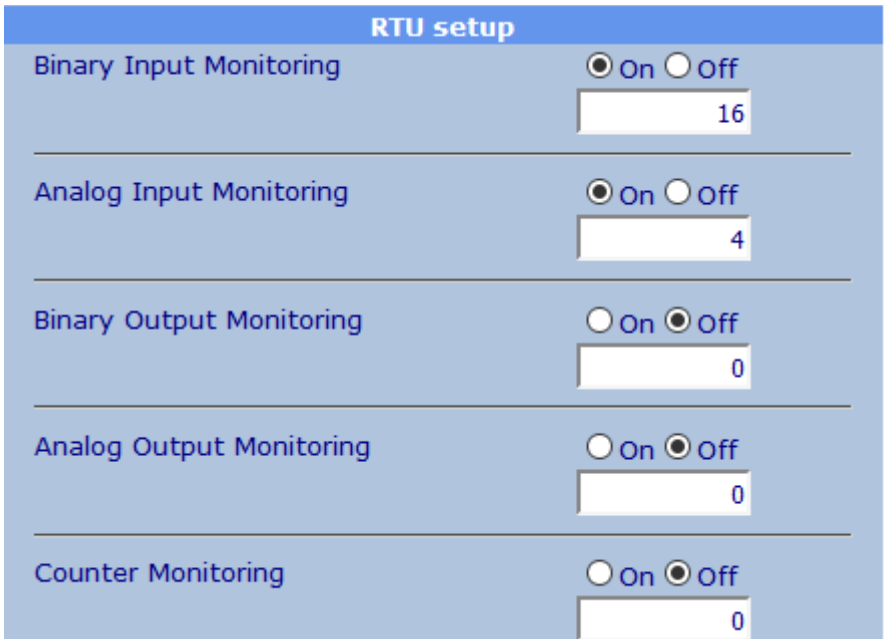
The **virtual analog inputs** are serving the communications direction **from the RTU-810 to the control room**. They are seen from the SCADA master as inputs. Locally connected equipment will write into these registers.
Local equipment can read/write these registers from MODBUS address 650-689. For remote equipment these will be seen as common analog inputs.

The **virtual analog outputs** are serving the communications direction **from the control room to the RTU-810**. They are seen from the SCADA master as outputs. Locally connected equipment will read from these registers.
Local equipment can read/write these registers from MODBUS address 700-739. For remote equipment these will be seen as common analog outputs.



Virt. Analog in	This is the number of virtual analog input registers , which are added to the number of RTU-810's analog inputs. These registers are appended behind the physical analog inputs and will increase the number of analog inputs shown in the RTU information.
Virt. Analog out	This is the number of virtual analog output registers , which are added to the number of RTU-810's analog outputs. These registers are appended behind the physical analog outputs and will increase the number of analog outputs shown in the RTU information.

3.5.1.4 RTU setup



This section allows the monitoring of the available I/O of the RTU-810.

Although the number of physical I/O is determined by the "DA"-type of the RTU-810 and is a fixed value, the I/O can be extended by expansion modules. In addition, the number of analog inputs and outputs can increase if virtual analog I/O are used.

The RTU-810 will scan the available expansion modules with each start and calculates the total number of inputs and outputs. The order of the I/O in the registers for access by the control room or a radio master like another RTU-810 or a MDP-310.200 can change if the number of available I/O is changing. I.e. the virtual analog outputs will shift back to a different register address if an expansion module with analog outputs is added.

Also, if i.e. an expansion module will fail, the total number of available I/O will also be changed.

If the monitoring is enabled for an I/O type, the number of available I/O in the RTU-810 will be continuously be compared with the number entered in this page.

If the configured number will not match the "real" number in this case, the following will happen:

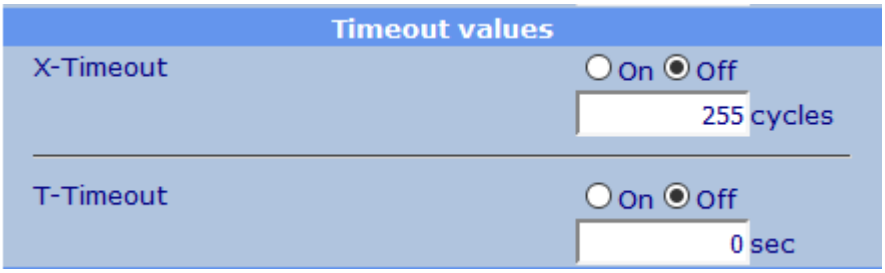
- The RTU-810 will force all its outputs into the secure state – this means they are all switched into off state.
- The RTU-810 will set the "I/O error" bit in the status register to show the master that an I/O error has occurred.
- The I/O error will be logged in the event logger

3.5.1.5 Timeout values

There are two different timeout types , which can force the outputs into the secure state when the radio communication to the master breaks or if the RTU-810 is not being polled anymore:

- **X-Timeout:** This applies only to data , which is received on the radio frequency. If this timeout is enabled, every radio telegram , which belongs to the same radio layer address and , which can be decoded by the RTU-810 decrements the given X-Timeout value by 1.
  Every received radio telegram , which is addressed to the RTU-810 itself will reset the X-Timeout counter to the configured value.
  The X-timeout will run off to 0 after the configured number of radio telegrams from a master addressed to other devices were seen by the RTU-810 on the radio channel without the RTU-810 itself being polled.

- **T-Timeout:** The T-Timeout is set to the configured value at the RTU-810's start and each time the RTU-810 is accessed by a master, regardless , which protocol the access is using (DNP3, IEC-60870, MODBUS) and regardless of the interface , which was used (radio, Ethernet, serial interface).
  Otherwise it's decremented by 1 each second.
  The T-Timeout will run off to 0 when the RTU-810 was not polled in any way within the given number of seconds.

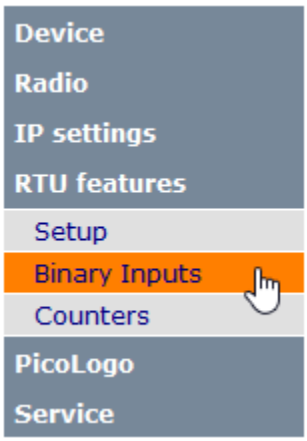


X-Timeout	The On/Off switch enables or disables the use of this timeout. The number of telegram receive cycles from , which it's counting down is given in the number field.
T-Timeout	The On/Off switch enables or disables the use of this timeout. The number of seconds from , which it's counting down is given in the number field.

Notes:

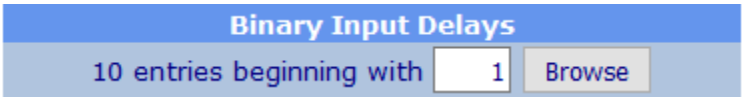
- A running PicoLogo application , which accesses the I/O of the RTU-810 will periodically reload the X- and T-Timeout values! They will not run off in this case
- A MDP-310.200 or RTU-810 polling the device in RTU master mode will configure the X-Timeout and T-Timeout cent rally. The locally configured values will be overwritten.
- When any of these timeouts is enabled and runs off, all outputs are forced to the secure OFF state.

3.5.2 Binary Inputs



This page configures the advanced functions of the internal binary inputs. Inputs provided by expansion modules do not provide advanced functions.

3.5.2.1 Browse page

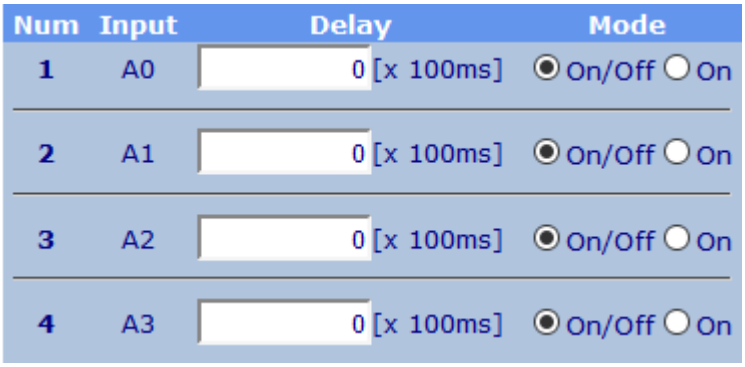


As only 10 entries are displayed per page, but the configuration table holds a maximum of 16 entries, the start index of each page can be entered here. A click on the "Browse" button will reload the page with the selected index number as first entry.

3.5.2.2 Delay configuration

Each of the internal binary inputs can be configured to use either an on-delay or a combined on/off-delay. The time unit is 100ms. This function can be used to suppress glitches.

Any delays configured here will be applied before any other processing functions like SCADA queries or PicoLogo.



Num	Number of the input
Input	Port name declaration of the input
Delay	Value of the delay in 100ms steps. Maximum value is 99.9 seconds. 0 turns the delay off.
Mode	Selects whether the delay is an on-delay of an on/off-delay.

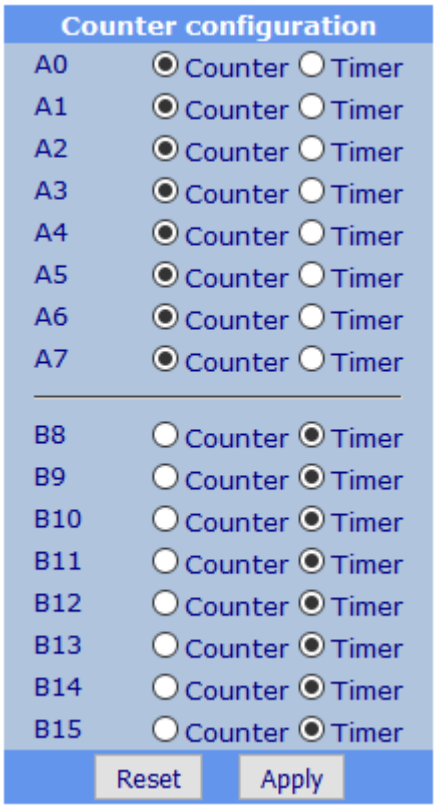
3.5.2.3 Counters

Any of the internal inputs of the RTU-810 has a double function and acts in parallel as an event counter or a time counter with a 16 bit register.

Configured as an event counter, each time the input changes from off to on, an event is counted. The maximum frequency is 10 Hz. This can be used as a pulse counter.

Configured as a time counter, the counting register will be increased by 1 in every second the input is active. Resolution is 100ms.

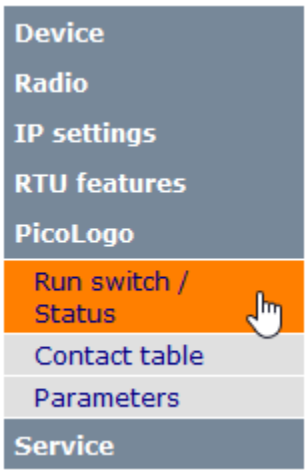
The 16 bit register will turn over from 65535 to 0, this must be taken into account in the SCADA when calculating the events or seconds from the delta between two pollings.



This page configures whether each input acts as an even counter or as a time counter. Default setting is port A as event counters, port B as time counters.

3.6 PicoLogo

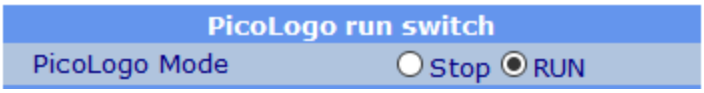
3.6.1 Run switch / status



This page can switch the PicoLogo application between "run" and "stop", it'll be also possible to delete the application loaded into the RTU-810.

Apart from this information about the loaded application is available as well as possible error messages.

3.6.1.1 PicoLogo run switch



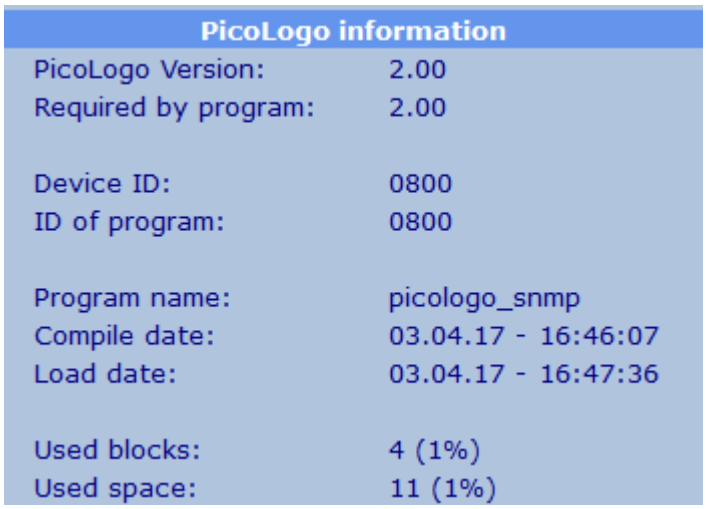
A loaded application can be switched between "Stop" and "RUN" mode. This switch is non-volatile and the setting is preserved during a power-down of the RTU-810.

3.6.1.2 PicoLogo status

This section indicates the status of the PicoLogo application:

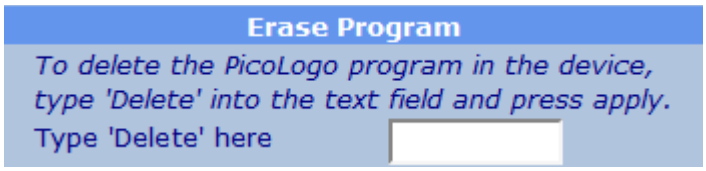
- STOPPED: No PicoLogo application is running. If an application is loaded into the RTU-810 it is stopped and will not work.
- RUNNING: The application is running and has no errors. The green "R"-LED in the front panel will indicate the running application.
- ON ERROR: An application is loaded but in error state. The red "E"-LED on the front panel will also indicate the error state.
  Further information about the error will be presented also in this section.

3.6.1.3 PicoLogo information



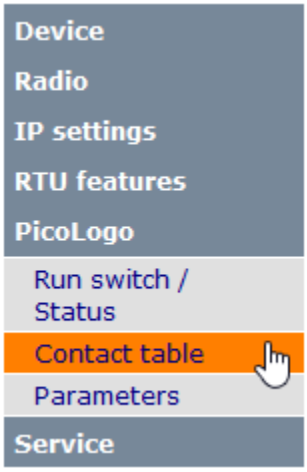
PicoLogo version	This is the version number of the internal PicoLogo run time. The PL run time code is part of the RTU-810 firmware and may be updated with a new firmware.
Required by program	This is the minimum PicoLogo version , which is required by the application. The required version must be less or equal the run time version in the RTU-810.
Device ID	The device ID of the RTU-810.
ID of program	The device ID for , which the application was compiled.
Program name	The name , which was used to store the application on disk in the PicoLogo editor.
Compile date	Date and time when the application was compiled
Load date	Date and time when the application was loaded into the RTU-810
Used blocks	A PicoLogo application can hold a maximum of 256 user function blocks. This shows the number of blocks the application uses.
Used space	This is the number of words the application needs in memory. Maximum is 1024.

3.6.1.4 Erase program



A loaded PicoLogo application can be erased here. For security reasons, the word "Delete" must be typed into the text field and "Apply" must be clicked to do this.

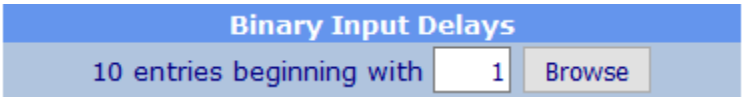
3.6.2 Contact table



Some PicoLogo blocks are sending messages to subscribers via IP, i.e. as SMTP trap messages.

The receiver telephone numbers or IP addresses are initially specified in the PicoLogo block. But in addition, these addresses can be viewed and edited in the contact table.

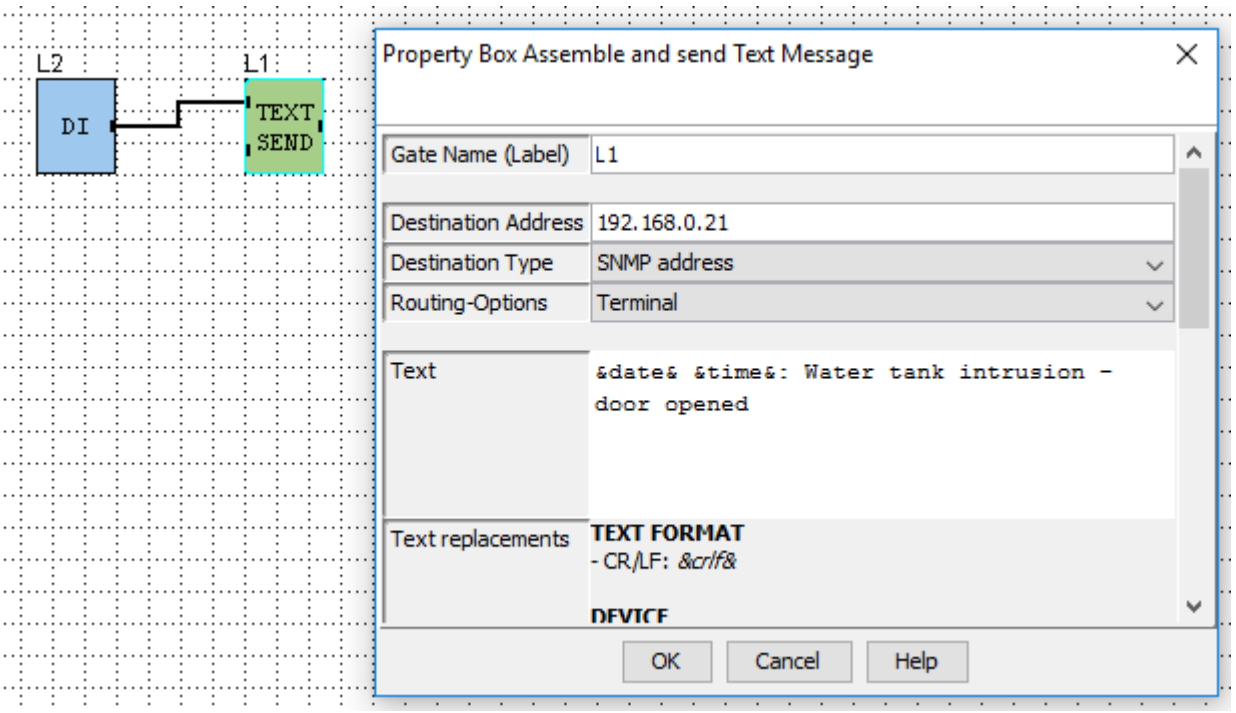
3.6.2.1 Browse page



As only 10 entries are displayed per page, but the contact holds a maximum of 50 entries, the start index of each page can be entered here. A click on the "Browse" button will reload the page with the selected index number as first entry.

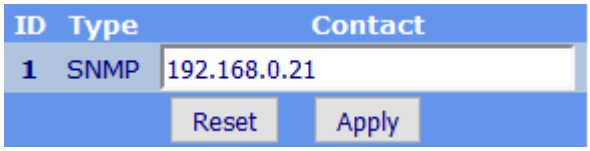
3.6.2.2 Contact entries

Each subscriber number or IP address where a message is sent i.e. via SMS or SMTP trap is initially defined in the PicoLogo block when the application is created:



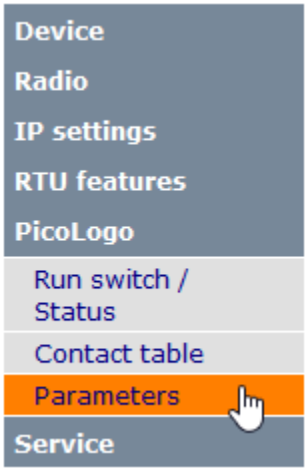
The example above sends a SMTP trap message to the IP address 192.168.0.21 when a door contact is opened.

To make destination addresses for messages easily changeable without recompiling the whole application in the PicoLogo editor, addresses are listed in the contact table. The application above creates this entry:



The contact can be edited and changed in this page on the fly without the need to recompile the PicoLogo application.

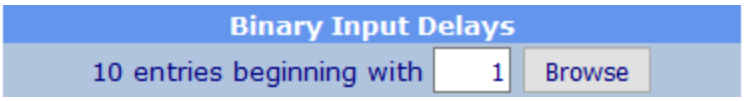
3.6.3 Parameters



Many function block in PicoLogo are using contant values , which are defined in the block like time delays etc.

Most of these constans can be put to the configuration web server for an easy on-the-fly change.

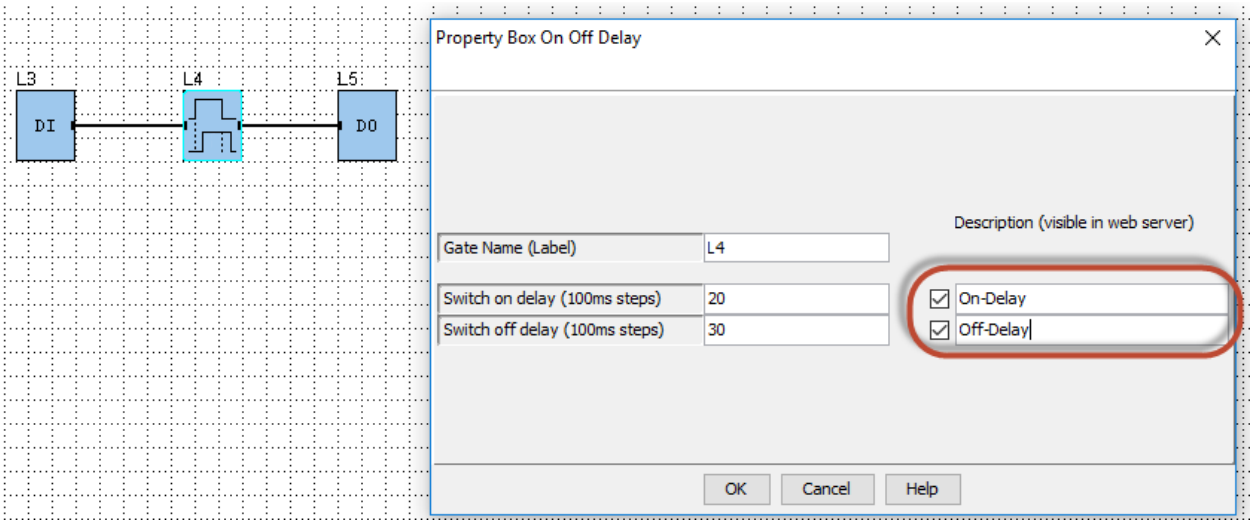
3.6.3.1 Browse page



As only 10 entries are displayed per page, but the parameters section can hold more entries, the start index of each page can be entered here. A click on the "Browse" button will reload the page with the selected index number as first entry.

3.6.3.2 Parameter entries

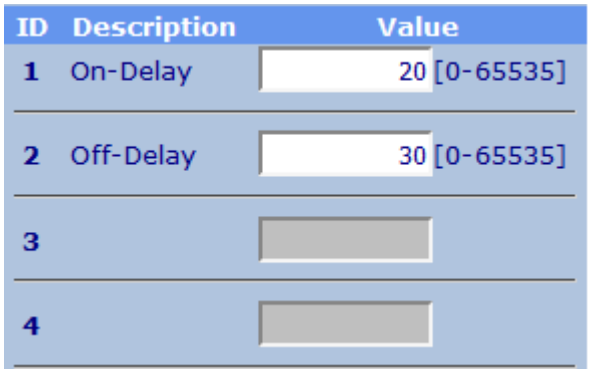
Many function blocks in PicoLogo are using constants, line the example below , which is an on/off delay from a binary input to a binary output:



This application defines an on-delay time of 2 seconds and an off-delay time of 3 seconds. Constants are stored in the function blocks and would normally need a change in the PicoLogo Editor and a recompilation of the application if they should be changed due to some reason.

But the constants of most blocks can be put into the web server for easy reviewing and changing. In this case, the checkbox "Description (visible in web server)" must be checked for the constants in the PicoLogo block, as done in the application above for both parameters. The description is a free text , which describes the parameter as it will appear in the web server.

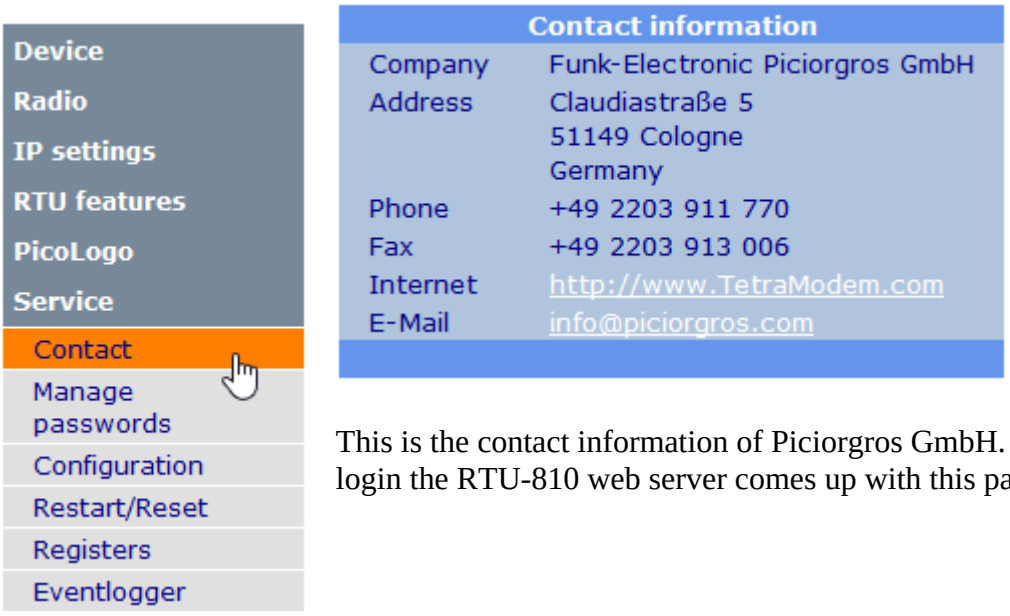
The parameters , which are tagged this way in the PicoLogo application can then be found as parameter entries on that page:



The values can be changed on-the-fly without the need of changing and recompiling the application in the PicoLogo editor.

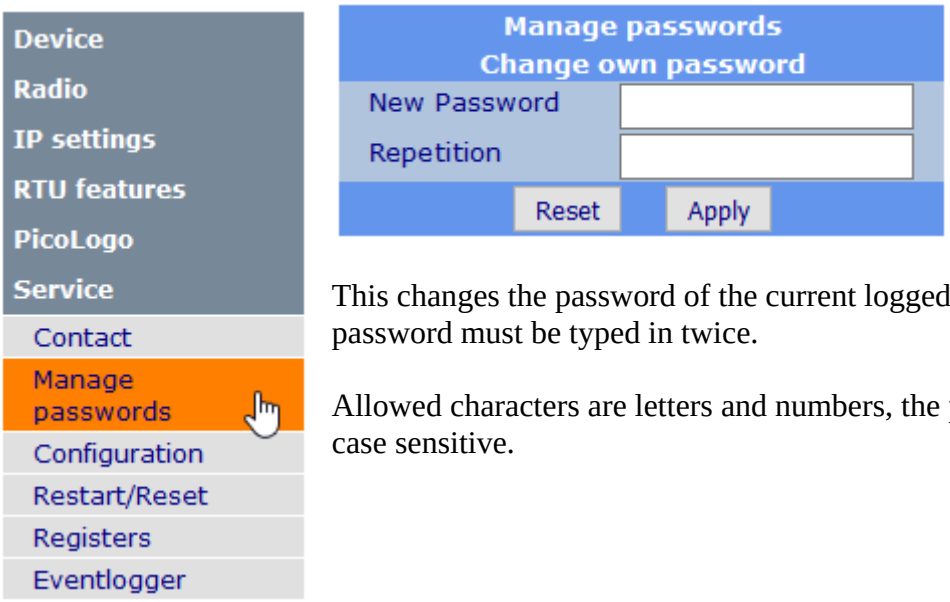
3.7 Service

3.7.1 Contact



This is the contact information of Piciorgros GmbH. After login the RTU-810 web server comes up with this page.

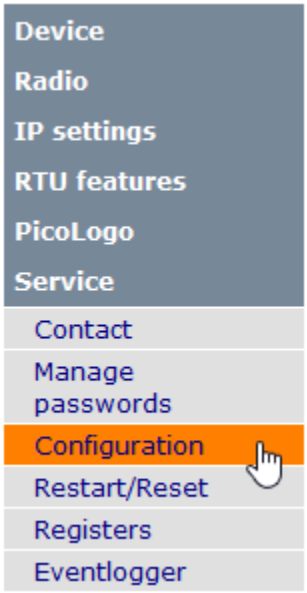
3.7.2 Manage password



This changes the password of the current logged in user. The password must be typed in twice.

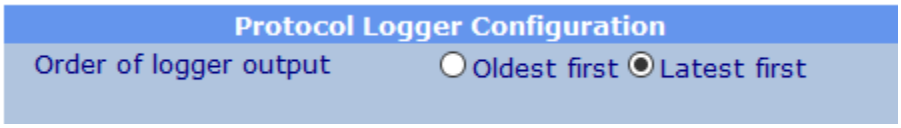
Allowed characters are letters and numbers, the password is case sensitive.

3.7.3 Configuration



This page allows some configuration for the Protocol Logger / Event logger, can show the status of the additional RTU-810 features and can be used to enter feature activation keys.

3.7.3.1 Protocol Logger Configuration



The RTU-810 has an embedded Event Logger , which stores up to 650 events like communications failures for the RTU master mode, device restarts and much more. The entries are nonvolatile and survive power-downs.

If 650 entries are stored, each new entry will delete the oldest one.

Order of logger output	Determines if the newest logger events should be on top of the page "Service"→"Eventlogger", or the oldest.

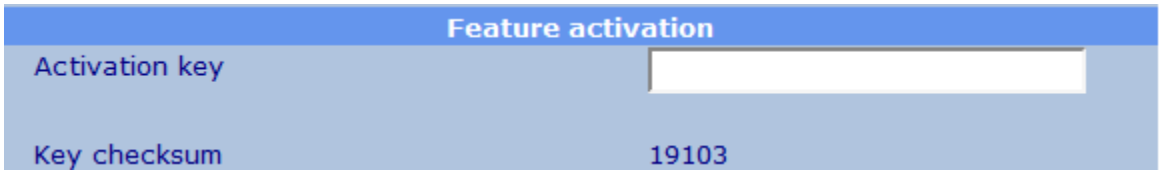
3.7.3.2 Feature activation

Feature activation	
IP Communication	----
I/O	Enabled
PicoLogo	Enabled
IEC 60870 / DNP3 access	----
RTU master mode	Enabled
Serial master modes	Enabled

This is a list of all additional features , which can be activated for a RTU-810. Features can have this status announced:

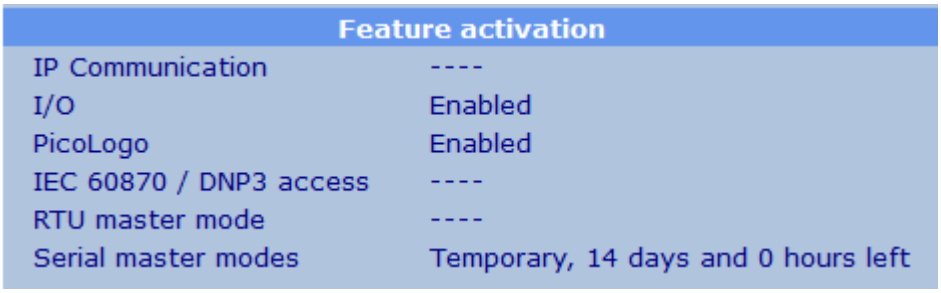
- Enabled: The feature is permanently enabled and can be used
- Temporary: This feature is in trial mode for a certain time. The remaining time is also listed. After the remaining time runs out, the feature will be
- Disabled: The feature is not activated
- Disabled (Temporary key used): The feature is not activated and a temporary trial key was previously used.

3.7.3.3 Applying activation keys

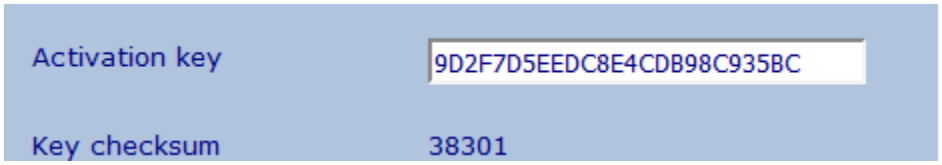


Software features can be purchased at any time. After the purchase the user will get an activation key , which is valid for the certain serial number for , which it was ordered.

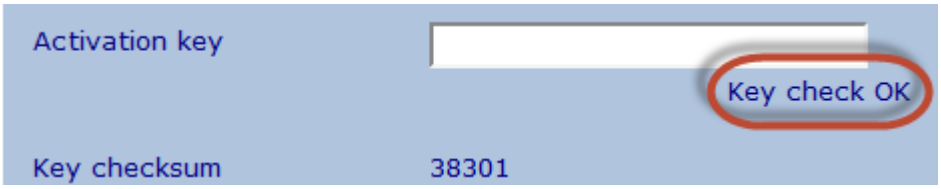
In this example an activation key is purchased to turn the serial master mode activation into a permanent one and to activate the RTU master mode feature. This is the situation before the activation:



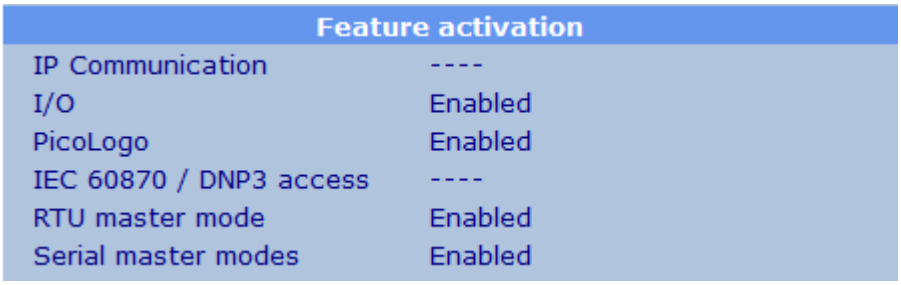
The activation key , which was received with the purchase must be pasted into the Activation key window:



Then "Apply" must be clicked …

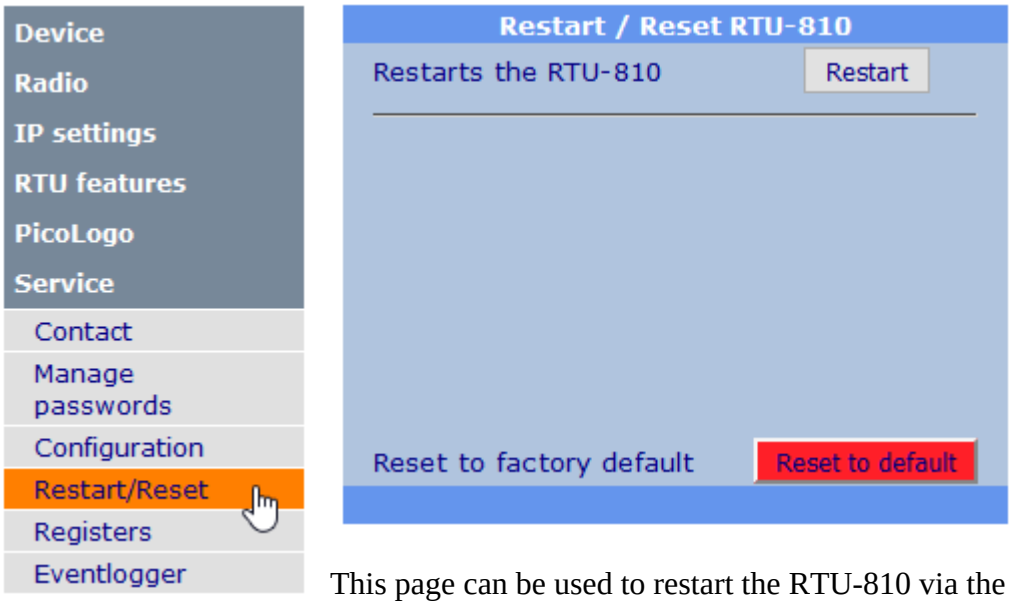


And the feature are activated:



A device restart is recommended after activating new features.

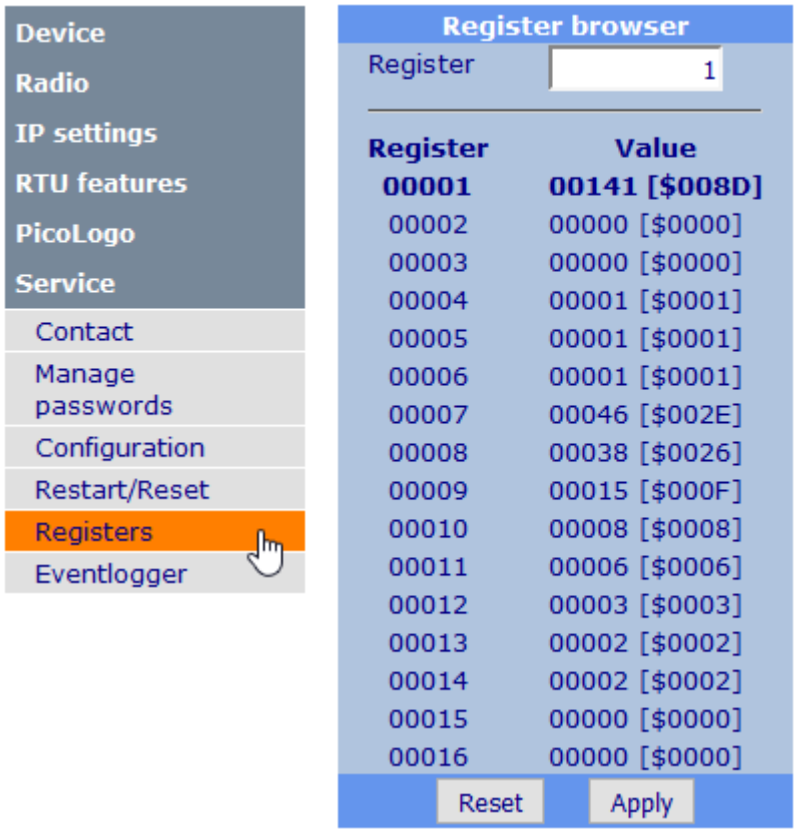
3.7.4 Restart/Reset



This page can be used to restart the RTU-810 via the web server or to reset if to factory default.

Restart	A click on this button restarts the RTU-810
Reset to factory default	A click on this button does a full reset to factory defaults. Any customer settings are erased, software features are of course maintained. This is the same as pressing the configuration button on the back of the device for at least 5 seconds.

3.7.5 Registers



The register browser can display register value. The register addresses are identical to the MODBUS register addresses, therefore it can be used to check the content of the I/O registers.

Also in a support case you might be asked to read out certain register values.

The starting address can be changed by entering the start register in the "Register" field and pressing "Apply".

3.7.6 Eventlogger

The event logger stores events and exceptions, communication failures in RTU master mode and much more. This is the first point to check if the communication to a RTU-810 was disturbed or if other problems should be analyzed.

It stores the last 650 events in a nonvolatile memory.

```
31.03.17 16:46:46 Master mode: Station addr. 2 was set to ONLINE
31.03.17 16:46:45 Master mode: Station addr. 2 station has unsufficient number of I/O
31.03.17 16:40:02 Master mode: Station addr. 2 set to OFFLINE with failcounter: 10
31.03.17 16:39:53 Master mode: Station addr. 2 was set to ONLINE
31.03.17 16:39:39 Master mode: I/O configuration error - suspending polling
31.03.17 15:26:03 Master mode: I/O configuration error - suspending polling
31.03.17 15:07:21 Master mode: Station addr. 2 set to OFFLINE with failcounter: 10
31.03.17 15:07:12 Master mode: Station addr. 2 was set to ONLINE
31.03.17 14:02:45 Ethernet link is up. Own IP:192.168.4.100
31.03.17 14:02:43 Local Bus: no modules recognized.
31.03.17 14:02:42 Device was 0 days, 20 hours, 54 minutes and 16 seconds down!
31.03.17 14:02:42 Displayboard found with SW-Version: 01.50.
31.03.17 14:02:42 I/O-Board found with SW-Version: 01.70.
31.03.17 14:02:42 Internal extension board found.
--.--.-- --:--:-- Restart cause: 00010401
--.--.-- --:--:-- ***Device started with SW-Version: 02.30***
```

4 Special Functions

4.1 IP Application Interface

The IP Application Interface is an additional data interface , which is very similar to the serial interfaces, but with IP as the communication method. It can be used to pass data, which are structurally protocols designed for serial interfaces, over an IP link.

The important thing is that the RTU-810 is the endpoint for the IP communication. SCADA protocols , which are designed as IP protocols uses different IP addresses to address each slave, therefore these protocols can't be used with the IP application interface.

Using the IP application interface the IP connection ends at the RTU-810 and the payload data inside the IP packet is extracted by the RTU-810 and used for further routing, similar as using the serial interfaces.

All IP data must be addressed to the RTU-810 as the IP connection is just locally. To route the data to different destinations in the radio network, the known mechanism of address detection and the use of the routing tables can be applied to the data , which was sent inside the IP packet.

To make it even easier to understand: The IP application interface works exactly like the two serial interfaces with the difference that the "physical" connection is not a simple wire but an IP connection.

4.1.1 Tunneling a serial connection with the SIG-810 (Q1 2018)

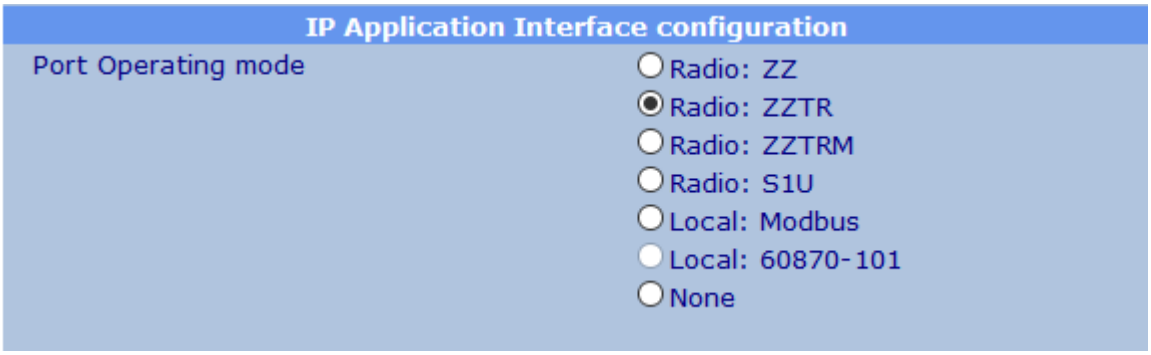
The Piciorgros SIG-810 is a Serial-to-IP gateway , which is designed for the use in conjunction with the IP application interface.

The SIG-810 has a configurable serial interface , which can be either RS-232 or RS-422/485. On the other side it connects to the IP Application interface of a RTU-810/TRM-810 or RTU-81.

This "extends" an existing serial connection basically to any possible length as long as an IP connection via LAN or WAN is available between the SIG-810 and the RTU-810.

4.1.2 Operating modes

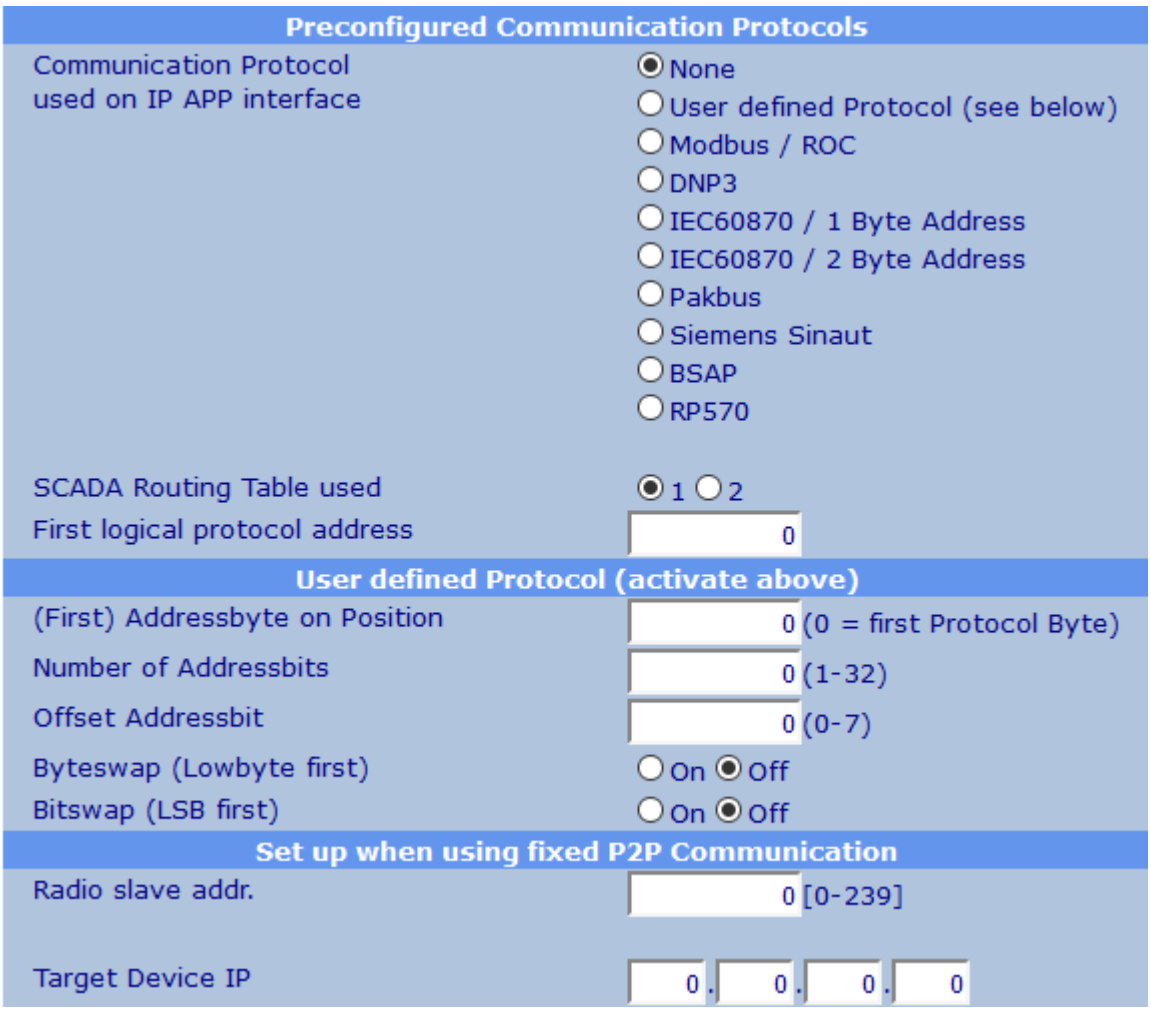
The operating modes available for the IP Application Interface are exactly the same as for the serial ports, they are not described here again in detail. For a detailed description please see "3.3.2.1 Interface configuration".



4.1.3 Address detection and routing

As on the serial interfaces, also on the IP Application Interface the RTU-810 can extract address bytes , which are inside the data payload received on the interface. This may be common and known serial SCADA protocols , which are embedded in the IP packet to the RTU-810, but also customer specific data containing an address information.

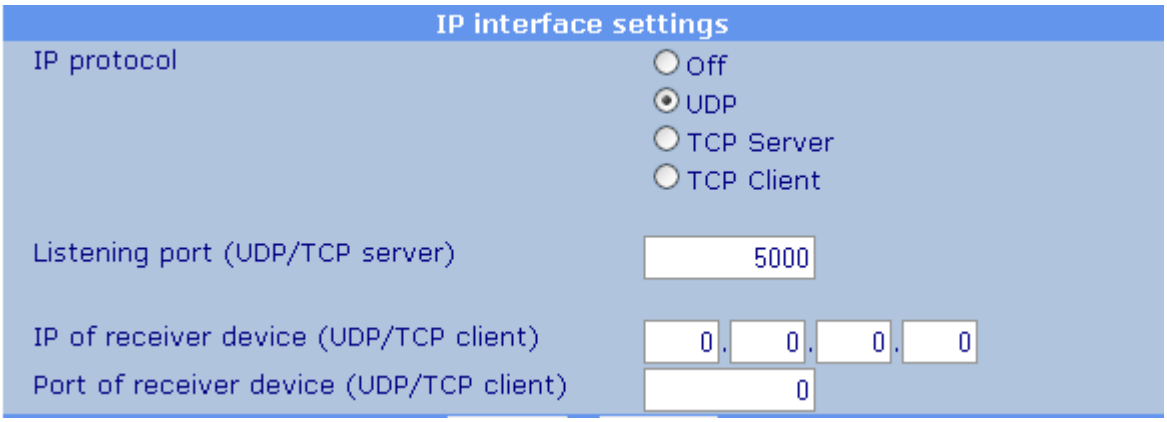
The extracted address is used as an index to one of the two SCADA routing tables, determining the radio slave address or IP address of the RTU-810 in the field, where the data should be sent to.



4.1.4 Configuring the IP link

The IP Application Interface can be configured to act in these three modes:

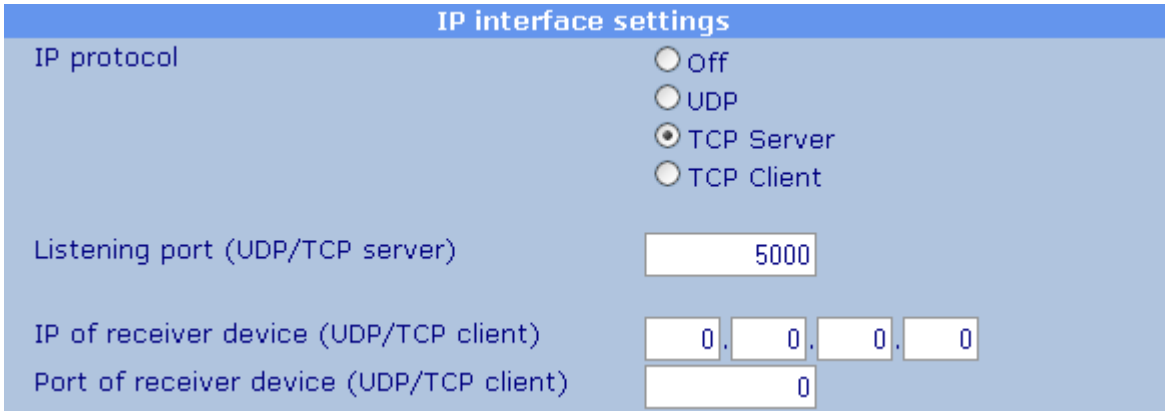
4.1.4.1 UDP operation



Configured for UDP operation, the RTU-810 will accept data sent by UDP to the port specified as "Listening port" (in this example port 5000).

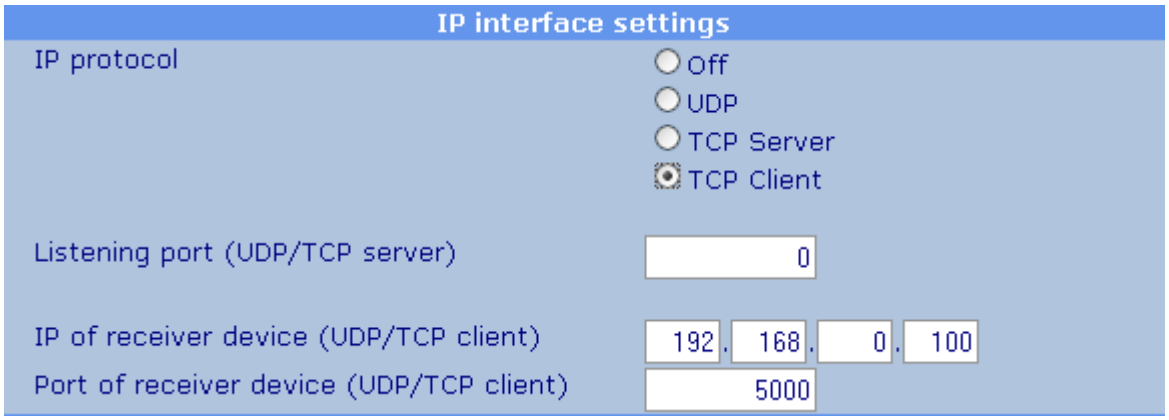
If no IP/Port is specified for a "receiver device", the RTU-810 will learn the source IP and port of the last packet , which was sent to it and send any received data for the IP Application Interface back to it. If an IP/port is configured here, received data from the radio side will always be sent to this destination.

4.1.4.2 TCP server operation



Configured as a TCP server the RTU-810 is waiting for an incoming TCP connection on the port configured as "Listening port". Once one device has established a TCP connection to this port data can be sent and received on this connection.

4.1.4.3 TCP client operation



Configured as TCP client the RTU-810 will periodically try to establish a TCP connection to the server specified as "Receiver device". IP address and port number must be specified in this case.

Once the RTU-810 could establish the TCP connection it can be used to send and receive data.

4.1.5 IP link status information

On top of the configuration page the status of the connection can be seen:

In TCP modes, the status is "CONNECTED" once a TCP link to the external equipment has been established.

In UDP mode the status is always "CONNECTED" as UDP is a connectionless protocol. In this case the IP address given in this line is the IP address of the external equipment , which has sent the last data to the RTU-810. If no data transfer has happened, the IP is shown as "0.0.0.0".

4.2 RGW-810 radio gateway (Q3/4 2018)

The RGW-810 is a gateway , which connects a radio network to an IP network.

With the ability of the RTU-810 to send and receive the data over IP networks in the same way as via radio networks, the RGW-810 can be used to separate parts of the radio networks into areas , which can't be directly accessed via radio connection from the radio master device.

The only requirement for this kind of setup is an IP link between the RGW-810 and the radio master, which can be a RTU-810, a TRM-810, a RTU-81 or by end of 2017 the MDP-810.

The RGW-810 includes a configurable radio part, similar to the RTU-810. Whenever it receives data from a master device via its IP interface, it'll transmit this data via the radio interface. Any data , which is received on its radio interface is sent back via to the master device.

The RGW-810 will learn the IP address and port of the master device once it has received the first data from the master. Therefore no configuration for the master IP address and port is needed. The communication between the RGW-810 and the master is using UDP port 4711.

4.3 Radio family address (Q2/3 2018)

In the first approach the RTU-810/TRM-810 series was designed to replace and enhance the 710 series with the focus on compatibility to existing radio networks using the 700 or 710 series.

To separate different customers , which are using Piciorgros devices, each customer has a fixed radio layer address , which is assigned by Piciorgros. Devices with different radio layer addresses respective different customers can't communicate with each other to eliminate unwanted effects and wrong data outputs in case networks from different customers on the same frequency are in a range where they can at least partially receive each other.

The need of some customers to even more separate their own networks from each other was covered by the ability to give a block of radio layer addresses to a customer where he can select one of the addresses inside this range by his own.

For the 810 series, we'll introduce the so called "family address". This is a secondary address layer , which gives the customer 65536 addresses he can fully select by his own. A switch in the radio configuration will be added to configure the RTU-810 to either use the "old" radio format with just the radio layer address to be compatible with 700/710 networks and the MDP-310.200, or to use the extended address format , which includes the family address.

Using the extended address format only devices with matching radio layer addresses **and** matching family addresses can communicate each other, where the family address can be fully configured by the customer.

The auto-configuration feature for the RTU master mode will also use the family address to easily set up small point-to-(multi)point networks , which are always separated from each other.

5 Supported Protocols

5.1 Layer one Protocols between RTU-810 and the external device, connected via serial interface

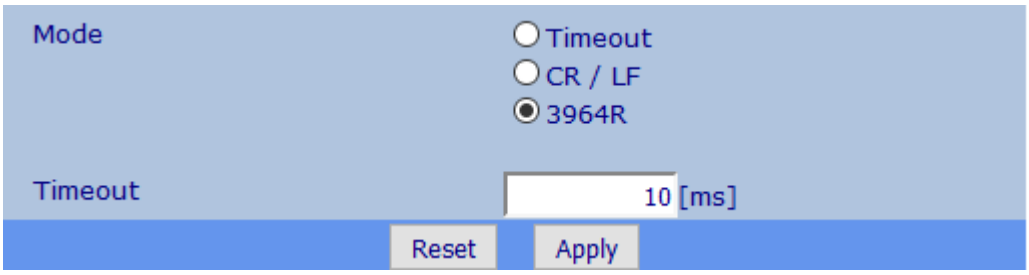
5.1.1 Timeout Protocol

Using the “Timeout Protocol”, the RTU-810 will accept any data received via its serial interface(s) accepting the whole character range from hex 00 to hex FF. The “Data End” criteria in that case are just the facts, that there is no more data received over a predefined period of time. (Factory default is set to 10 ms for 9600 bps. That reflects 10 characters in sequence).
Basically if used other baud rates, a timeout period of 10 character-times is suggested.

Using the COM interface, a timeout period between 3-1000 ms, using the AUX interface 10-1000 ms is programmable.

5.1.2 3964R Protocol

The 3964R often is used in the „Siemens World“, connecting PLC’s to PLS’s or PLS’s to SCADA systems. That “Layer One” protocol is fully supported by the RTU-810.
If possible, the priority settings of the device connected to the RTU-810 should be set to “low priority”.



Settings for the Serial Port(s)

5.2 Serial Protocols (RS-232 or RS-485/RS-422)

When setting the RTU-810 as a master unit, it is useful to know what communication protocol is used. Regarding that, the RTU-810 knows, where inside the data stream the address byte(s) is (are) located. With that knowledge, the RTU-810 extracts this “logical address byte” and using the predefined address translation (or routing) table, it can convert the protocol address into the destination slave address of the radio target device.

The outstation devices do not need any routing or address translation table. As these units usually are used as “polled devices”, they just answer to the polling request master address.

The following protocols are supported by the RTU-810.

5.2.1 Modbus RTU

As the Modbus RTU protocol can be used either for internal device access or for communication with an external device, care should be taken when sharing out the Modbus addresses. Under no circumstances, an external device should have the same Modbus address as the RTU-810.

5.2.2 ROC protocol

The ROC protocol has the same basic data structure for the address byte location and data packet detection than MODBUS, so MODBUS and ROC protocol shares the same protocol setting in the configuration.

5.2.3 DNP3

Using the Link Address out of the DNP3 protocol, the routing to the outstation can be calculated. In case of transmitting multiple DMP3-Data Segments while the timeout period is not considered, each data segment will be transmitted individually. Outstations can send data at any time to the master device (unsolicited messaging).

With the 60870/DNP3 option this protocol can also be used to access the RTU-810 itself (Future option)

5.2.4 IEC 60870-5-101

The routing for this protocol is similar to the DNP3. Also in this case it may be that multiple data segments are sent, to complete one message.

With the 60870/DNP3 option this protocol can also be used to access the RTU-810 itself (Future option)

5.2.5 PakBus

Pak bus is a rather unknown protocol, but it is fully supported by the RTU-810.

5.2.6 BSAP

The RTU-810 detects the difference between the normal and the extended messages in BSAP protocol and extracts the address information accordingly. The node address contains 7 bit, so the address range, which can be used for routing here is 0-127.

5.2.7 User-Protocol

Using user-defined protocols, the exact position of the address bytes and the address length can be pre defined.
If these parameters are unknown, the “Transparent Data Communication” can be used (see below).

5.2.8 Transparent Data Communication without Protocol Filter (User Defined)

Protocols, where the address byte cannot be extracted, or where the „Address Range“ exceeds the RTU-810 storage spectrum, (1024 entries, e.g.: sometimes that can happen using the HART protocol) can be transmitted in the broadcast mode "T1X".

In this mode, there is no dedicated radio master – every station can communicate with every station. On the con side, the use of radio relays is not possible in this mode as no addressing is used.

5.3 IP based Communication Protocols

In the future it is planned to have the ability to also transport IP based protocols using the TRM-810/RTU-810.

This will be a firmware update together with the feature option "IP communication".

6 Accessing the local I/O

6.1 Control and Measurement I/O (RTU Functionality)

(Note: in the following paragraph Ports A, B, C, & D are not to be confused with IP ports mentioned elsewhere in this document).

The RTU-810 includes control and measurement inputs and outputs integrated with the modem. The basic version (RTU-810/DA2)has 16 on-off inputs (Ports A and B) and 8 on-off outputs (Port C). A further optional version has Port D, which can be ordered as 8 additional on-off outputs, or as 4 analog inputs.

Each of the 16 on-off inputs of Port A have independent event counting functions that can be enabled by the user. Port A inputs counters operate as event counters, while Port B counters operate as time-totalizing counters. The maximum counting rate [input pulse rate] is 10 Hz.

The control and measurement I/O can be accessed using any of the following protocols:

- MoP / MoP2: This is the Piciorgros radio protocol , which is used to access the RTU-810 from a serial master device, OPC-server, MDP-310.200 and in Point-to-(Multi)point links with a RTU-810 master
- MODBUS/IP and -RTU through the serial interface or Ethernet interface.
- MODBUS-RTU via the radio network with a serial master in ZZTR mode
- IEC-60870-5-101 (unbalanced): through the serial interface or over the radio network using a ZZTR serial master. (Future option)
- IEC-60870-5-104 through Ethernet interface or over the radio network (Future option)
- DNP3 through Ethernet interface or over the radio network (Future option)

The protocols IEC-60870-5-104 and DNP3 do support unsolicited messages. In this case the RTU-810 will actively inform the SCADA master if any input value has changed. This saves a lot of traffic compared to polling protocols like MODBUS , which require continuous data transfer to get up-to-data values from the field.

6.1.1 Accessing I/Os using the MODBUS Protocol

The RTU-810 can be accessed remotely over the radio network or locally through its serial or Ethernet interface.

The Piciorgros radio protocol MoP/Mop2 is using exactly the same register addresses as MODBUS.

The I/O ports, the virtual PicoLogo input/output registers as well as other information of the TMO can be accessed using the following 16 Bit registers (e.g. using the Modbus RTU-Protocol)

Register	Assignment
0	Own Device ID (address)
1 (High Byte)	Field strength of latest received radio telegram in –dBm. A value of 90 means i.e. that the data was received with -90dBm by the radio transmitting device. If the RTU-810 is set to RTU-710 compatibility mode, this will be an uncalibrated percentage value instead where 100% is the best reception.
1 (Low Byte)	Device status as follows: • Bit 7: Will be set on device restart. The bit can be cleared if the status register is written with a value where bit 7 is set. • Bit 6: A set bit means that an I/O-error has occurred. This can be a failed or removed expansion module or a non-matching I/O monitoring value. The bit can be cleared if the status register is written with a value where bit 6 is set. If the error still exists, the bit will not reset to 0 • Bit 5: The time sync source for timeslot operation mode is working when this bit is set • Bit 3: Is set if the system time is valid • Bit 2: Is set on a properly operating device • Bit 1: If this bit is set, the radio module is on error
2- ….	All Input Data and Timer / Counter in packed format: • Digital Inputs • Analog Inputs as 12 Bit values • Counter and Timer • 32 virtual PicoLogo input registers
200-231	32 virtual PicoLogo input registers. These can be written by a PicoLogo application and be read out from the RTU-810 (i.e. by a SCADA).
232-263	32 virtual PicoLogo output registers. These can be written to the RTU-810 (i.e. by a SCADA) and can be read out by a PicoLogo application.

300-399	All Output Data in packed format and in this order: • Digital Output • Analog Output as 12 Bit values • 32 virtual PicoLogo output registers
400-449	Digital Inputs, I/O port A0 is the LSB
450-599	Analog Inputs (if present) as 12 Bit values The upper four Bit are set to 0000 (can be changed in future) Virtual analog inputs are added after the physical inputs.
600-649	Counter, 16 Bit each. The first counter relates to I/O port A0, followed by A1, … , B15 As factory default, the port A counters are set as event counter, the port B counters are set to Time-Counters (resolution is one second)
650-689	Virtual analog inputs These can be written by a locally connected equipment and will be shown to the SCADA as normal analog input. Refer to "3.5.1.3 Virtual analog I/O"
700-739	Virtual analog inputs These can be written by a SCADA master as analog outputs and will be shown to locally connected equipment. Refer to "3.5.1.3 Virtual analog I/O"
750-799	Digital outputs, I/O-port C0 is the LSB
800-899	Analog Outputs (if present), first value reflects input D0. Virtual analog outputs are added after the physical outputs.

6.1.2 Special Registers

This is a summary of special MODBUS registers , which can be useful to gather status information from the RTU-810.

Red values are alternative register addresses for the same register , which are mapped for the best compatibility with the RTU/TRM-710 series.

Register	Function	Comment
1000 906	Device ID	0x0800 for RTU-810, 0x0810 for RTU-81 Will be set to RTU-710/TRM-710 device ID in the RTU-810 if the RTU-710 compatibility mode is selected
1001 907	Main software version	Firmware version of the RTU-810
1004	Hardware revision	
1008 908	Serial number	The serial number of the device
1010 1	Status/RSSI	Same as MODBUS register 1 described before
1030 953	Time YYMM	Current time: Year (High.Byte) and Month (Low Byte) In compatibility mode 953 will be BCD coded like in the TRM-710 devices!
1031 954	Time DDhh	Current time: Day (High.Byte) and Hour (Low Byte) In compatibility mode 953 will be BCD coded like in the TRM-710 devices!
1032 955	Time mmss	Current time: Minute (High.Byte) and Second (Low Byte) In compatibility mode 953 will be BCD coded like in the TRM-710 devices!
1033	Time Weekday	Day of the week, 1(Monday) – 7 (Sunday)
1034	Time Status	• Bit 0: 1=DST active • Bit 7: 1=Time not valid, clock set required
1035	Set time YYMM	New value for Year/Month to set the time
1036	Set time DDhh	New value for Day/Hour to set the time
1037	Set time mmss	New value for Minute/Second to set the time
1038	Set time	When bit 0 is set to "1", the values in the set registers are written into the real time clock
1428 944	Time slot configuration	Configuration for time slot operation. • Bits 0-9: Time slot 1-10 are used for communication, each set bit represents the timeslot being used. • Bit 12-15: Value 0 means no sub timeslot use Value 1-4 means sub timeslot 1-4 is used.
1429	Time to next time slot in ms	Remaining time to the next following active time slot in ms
1430	Remaining time in time slot	Remaining time in the current active time slot in ms. Returns 0 if no time slot is currently active.
1431 945	Number of current active time slot	Returns the number of the current active time slot (1-10). Returns 0 when currently no time slot is active.

1433	Time sync status	• Bit 0: 1=SNTP sync OK • Bit 1: 1=DCF sync OK • Bit 2: 1=Time base synchronized, time slot use possible • Bit 3: 1=Time base currently not synchronized, but in reserve mode. Time slot use is still possible. • Bit 4: 1=DCF is in decoding process
1434	Remaining sync reserve	Number of remaining minutes in , which time slot use still will be possible in case the time synchronization is lost
1435 947	Number of the next timeslot	This is the number of the next following active time slot
1436 950	Minutes since last time sync	Number of minutes since the last time base synchronization. If DCF is used, the sync will happen each minute , which was successfully decoded. SNTP will synchronize two times after one minute each, then once after 10 minutes and then every 30 minutes. The tighter interval after a device restart is used to calibrate the internal time base as fast as possible.
1437 949	Time sync status (old format)	Sync status for the internal time base in the old format used in the 710 series: • 0: Not synchronized • 1: DCF-77 has started to decode the sync pulses • 2: Time base is synchronized • 3: Time base is not synchronized by still running in reserve. Time slot use is still possible
1438 946	Remaining time in time slot (old format)	Same as 1430, but the value is in a number of 25ms ticks instead of raw ms.
1439 948	Time to next time slot (old format)	Same as 1429, but the value is in a number of 25ms ticks instead of raw ms.

1506 902	X-Timeout value	Value of the X-Timeout
1507 901	T-Timeout value	Value of the T-Timeout
1510 930	Number of BI	Number of total available binary inputs on the RTU-810
1511 931	Number of AI	Number of total available analog inputs on the RTU-810 (including virtual analog inputs)
1512 932	Number of BO	Number of total available binary outputs on the RTU-810
1513 933	Number of AO	Number of total available analog outputs on the RTU-810 (including virtual analog outputs)
1514 934	Number of CNT	Number of total available 16 bit counters
1721 0 916	Radio slave address	Configured radio slave address
1737 914	Last data RSSI	RSSI of the latest received data in –dBm. Will be an uncalibrated % value in RTU-710 compatibility mode.
1745 900	Radio Layer address	Currently configured radio layer address

6.1.3 MODBUS Coil- and Input commands

The binary in- and outputs as well as 32 virtual in- and outputs from PicoLogo can be accessed with the coil / input commands by using the MODBUS protocol.

Coil / Input	**Assignment**
Coils 0-299	Binary hardware outputs of the RTU-810
Coils 1000-1031	32 virtual coils , which can be written by the SCADA or master and be read/written by a PicoLogo application
Inputs 0-299	Binary hardware inputs of the RTU-810
Inputs 1000-1031	32 virtual inputs , which can be written by a PicoLogo application and can be read out by the SCADA or master

6.1.4 Supported MODBUS commands

These MODBUS commands are supported by the RTU-810:

Command	Description
1 (0x01)	Read Coil Status
2 (0x02)	Read Input Status
3 (0x03)	Read Holding Registers
4 (0x04)	Read Input Registers
5 (0x05)	Force Single Coil
6 (0x06)	Write Single Register
8 (0x08)	Sub 0: Local Loopback
15 (0x0F)	Force Multiple Coils
16 (0x10)	Preset Multiple Registers
23 (0x17)	Read/Write Multiple Registers

6.2 Accessing the I/O using the IEC60870 protocol (Future option)

The RTU-810 offers the access to the hardware I/O and virtual PicoLogo I/O by the use of the IEC60870 protocol. Currently the IEC60870-5-101 protocol is supported, the IEC60870-5-104 protocol is in preparation.

The IEC60870 protocol is a commercial option , which must be purchased on a per-device basis. An activation for an existing device is possible on any time with an activation key. IEC60870 support is officially starting with firmware version 2.80.

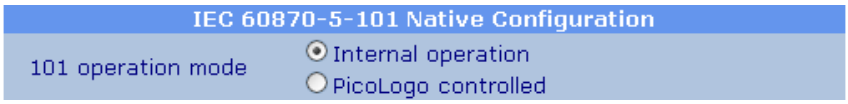
There are two ways of accessing the I/O of the RTU-810 with IEC60870 protocol , which are explained below.

6.2.1 Native (internal) IEC60870 support for the basic I/O's

If the IEC60870 option is active in a RTU-810, the internal inputs and outputs of the RTU-810 are directly supported and can be easily configured with the webserver. This implementation supports up to 16 binary inputs, up to 16 binary outputs and up to 4 analog inputs. The native support will match most applications in small substation environments.

For each I/O the object address and the data class (class 1 or class 2) can be configured individually.

To activate the internal IEC60870 support it must be set on IEC60870 → Native Configuration:



7 PicoLogo® User Application Interface

PicoLogo® is an easy to use Application Platform that can be used to:

- combine the embedded I/O with logic functions
- set up M2M communication
- send text- and alarm-messages as SNTP trap messages to syslog servers
- for versatile control- and supervision functions

For creating the logical functions for PicoLogo®, an graphical editor is optional available

8 Troubleshooting and Administration

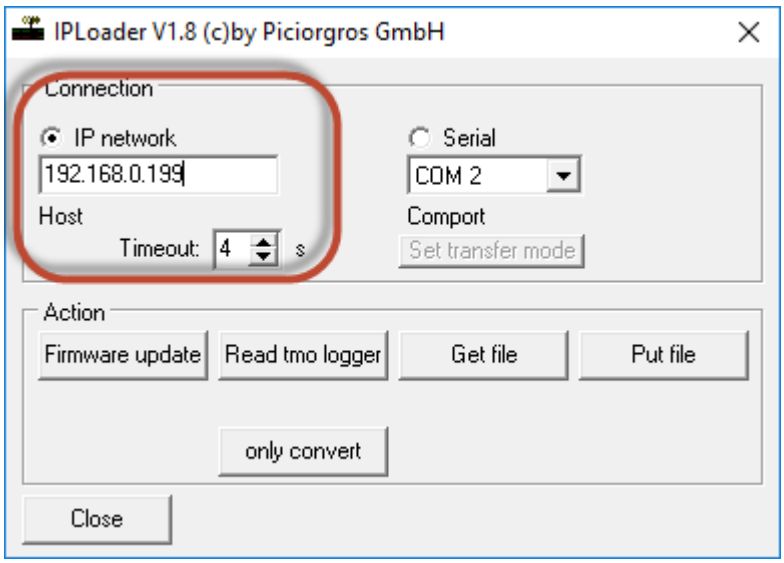
8.1 Save and restore the configuration

The configuration of the RTU-810 can be saved to the computer to have a configuration backup or to "clone" the configuration to other RTU-810.

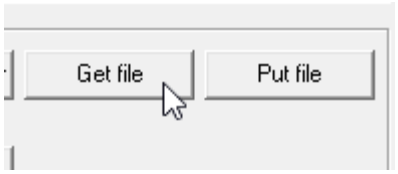
The "IPLoader" software is needed for this.

8.1.1 Configuration download from the RTU-810

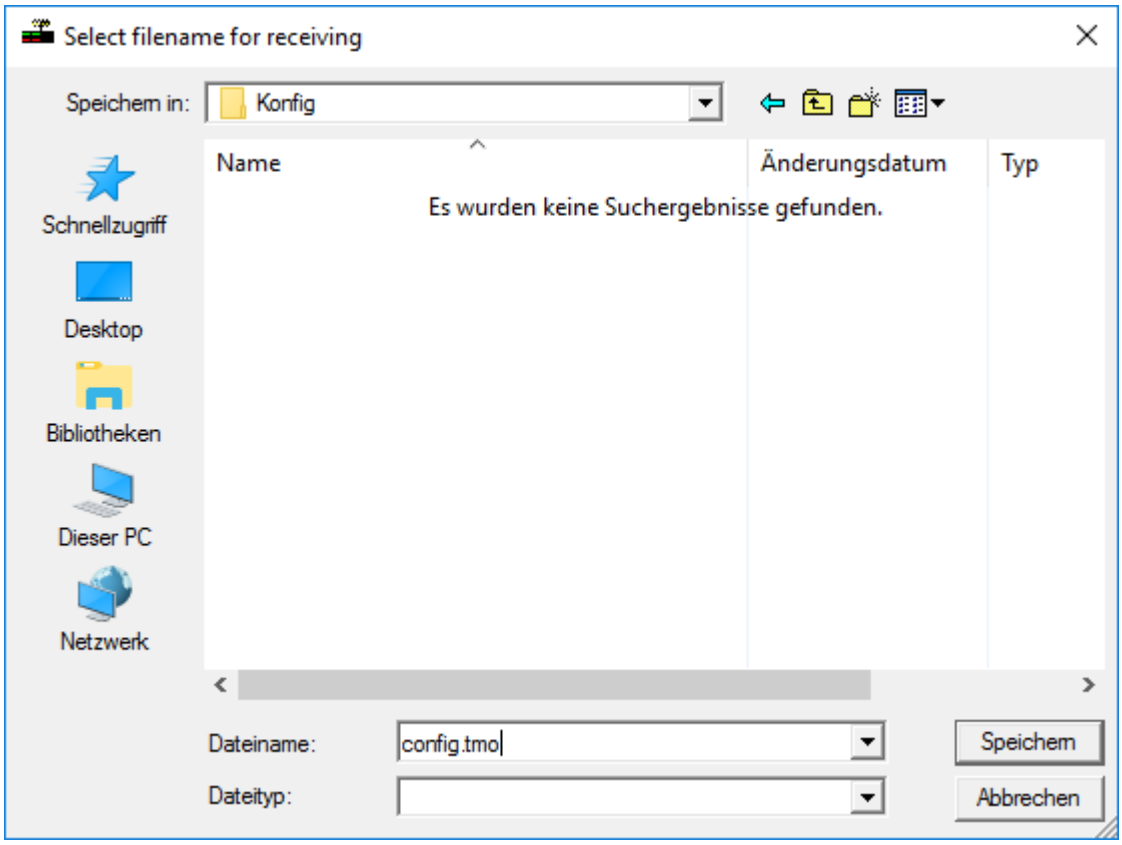
To read the configuration out of the RTU-810, start the IP Loader and ensure it is set to "IP network" with the IP address of the RTU-810 entered:



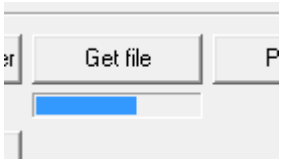
Click on "Get file"



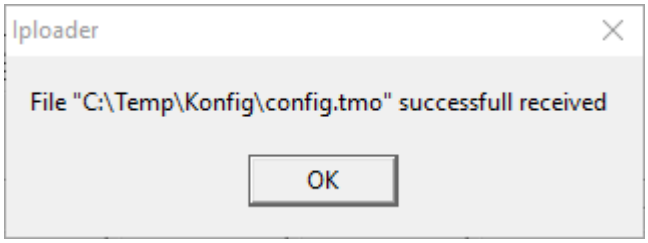
Specify a location on the computer where the file should be stored and give "config.tmo" as the file name:



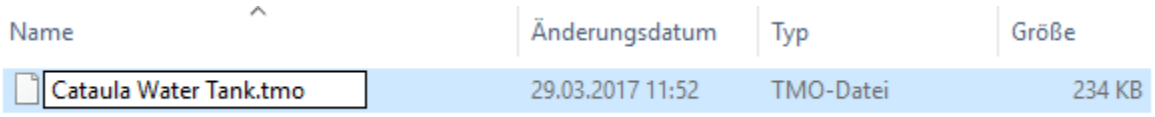
A progress bar will be shown below the button to indicate the transfer is ongoing:



Finally followed by a success message:



The file name can afterwards be edited to any convenient name to keep record from , which device the configuration was read. It is important to keep the file extension ".tmo"!

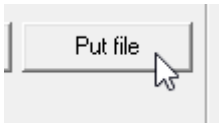


8.1.2 Restoring the configuration

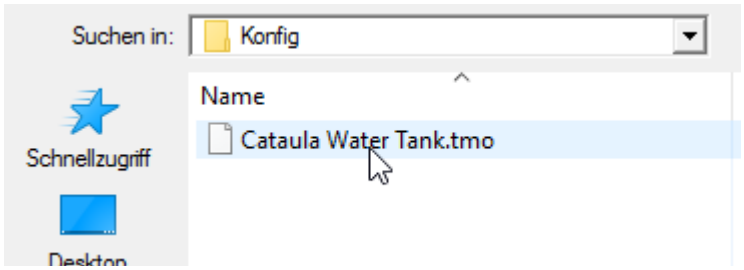
A configuration file can be uploaded in any RTU-810 , which has the same hardware version. This gives you the ability to restore the exact configuration if someone has broken it, or to upload the configuration into other RTU-810's to "clone" a certain configuration.

If the configuration should be "cloned" or being used as a template, make sure to alter unique parameters like the radio slave address to avoid double addresses in the network , which would cause trouble.

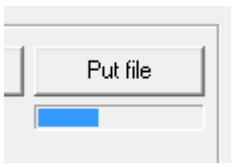
To restore the configuration, click on "Put file":



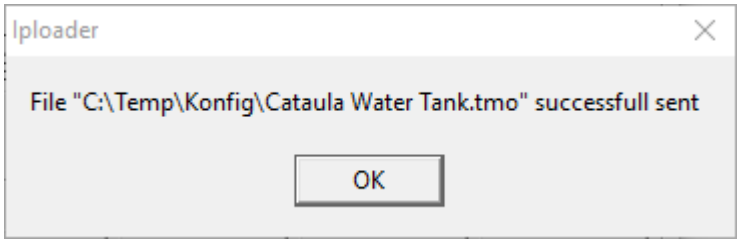
And specify the configuration file:



A progress bar below the button will indicate the transfer progress:



Followed by a success message:



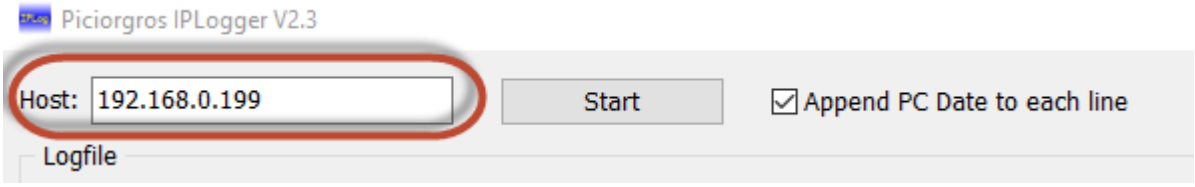
The RTU-810 will automatically restart to apply the new configuration.

8.2 Getting support logs

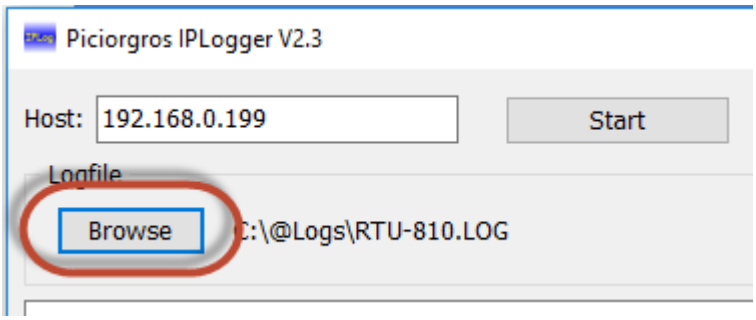
When problems are experienced the support might ask for a so called "IPLog". This log shows many internal processes and data flows of the RTU-810.

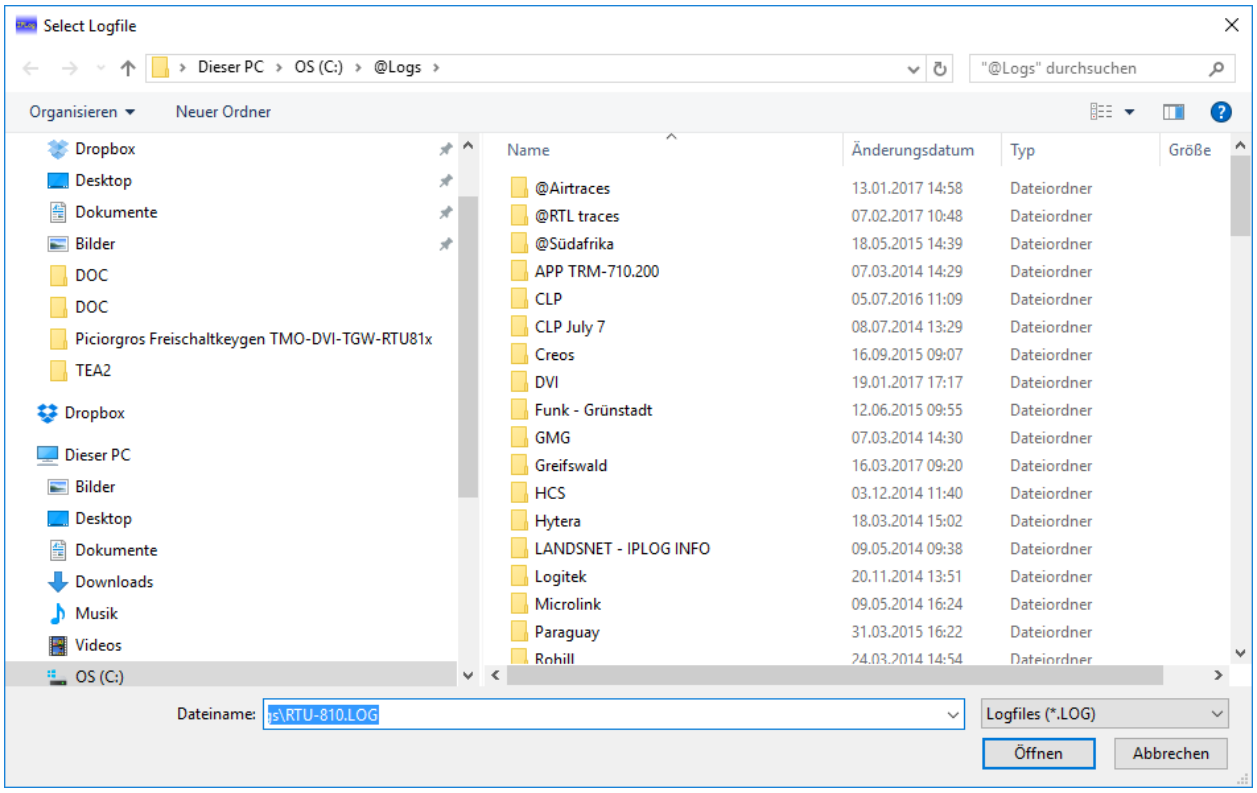
IPLogs can be taken with the software "IPlogger". This requires an Ethernet connection of the PC running the IPLogger to the RTU-810. **Due to the high amount of data don't use the IPLogger over a WAN or internet connection as this might overflow the TCP buffer of the RTU-810!**

When the IPLogger is started, ensure that the IP address is set to the IP address of the RTU-810:



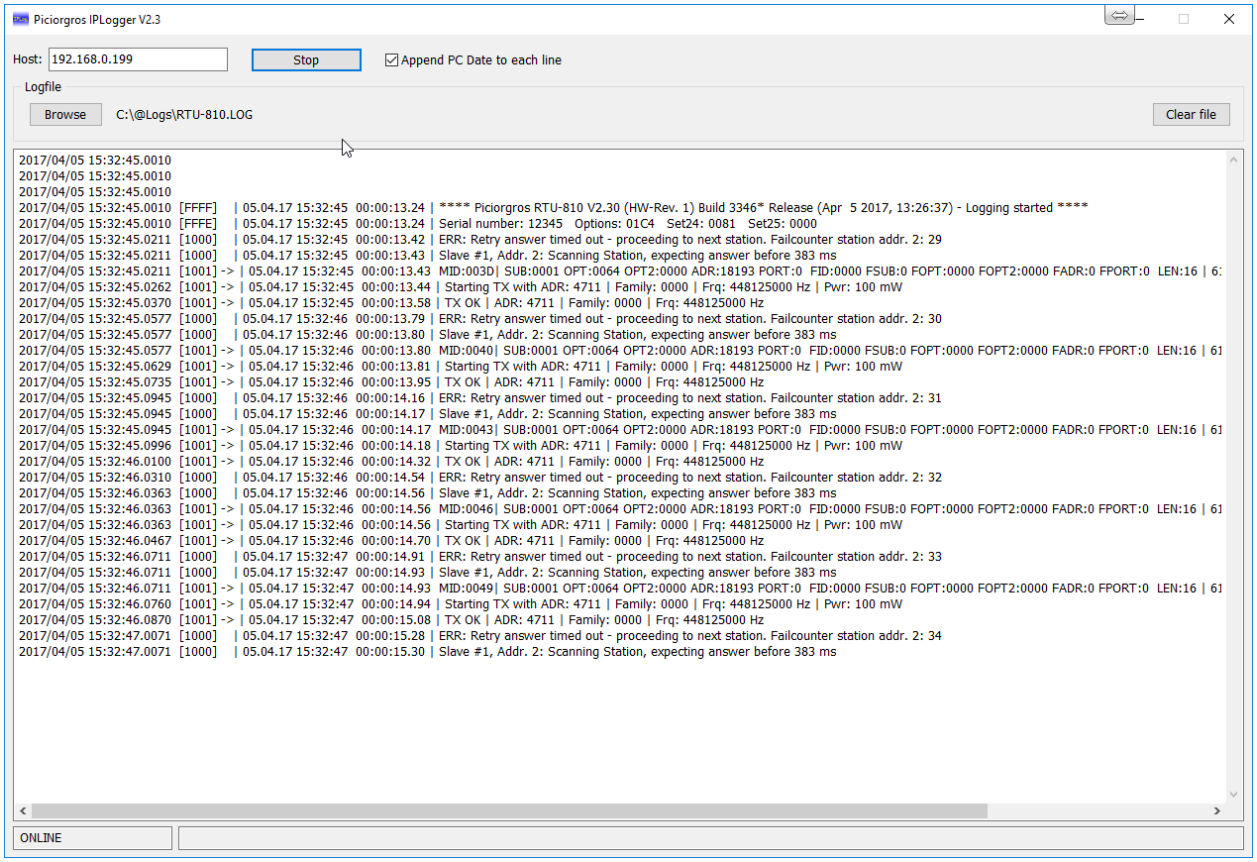
In the next step specify the folder and file name on the PC where the log should be written to:





The extension ".log" should be kept!

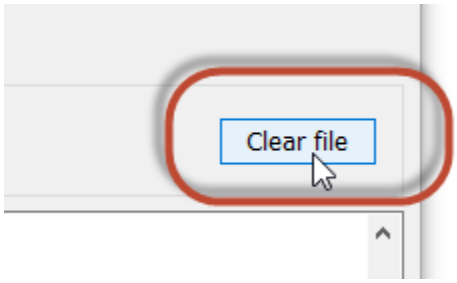
Finally the button "Start" can be pressed to start the logging:



The logged messages will appear in the window of the IPLogger to give a control that the logging is active and running.

The logging can be stopped and continued at any time. Even when the IPLogger is restarted, it will always append its data at the end of the specified log file.

If the log file should be cleared, press the button "Clear file".



This can be done during a running log "on the fly" without the need to stop the IPLogging.

Finally, when sending the log file to the support, it is highly recommended to put it into a ZIP archive as this will dramatically shrink the size of the file , which needs to be sent.

9 RTU-810 Firmware update procedure

9.1 Preparation and setup

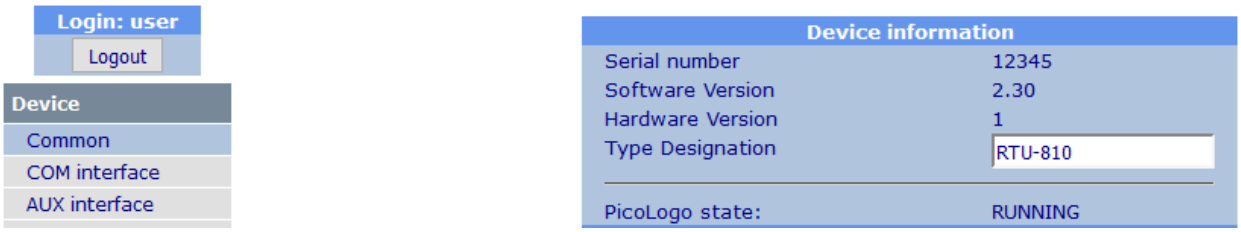
For updating the RTU-810 you will need the following equipment:

- PC with Ethernet connection
- IPLoader software
- Firmware file for the RTU-810

9.2 Update procedure for the RTU-810 main firmware

The RTU-810 firmware is a single file with the extension ".pfo". To update the RTU-810, it must be connected to the PC via Ethernet.

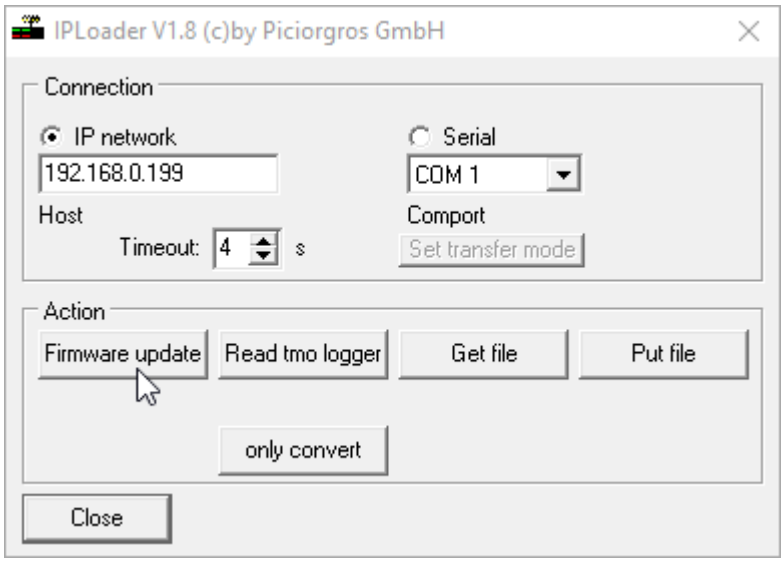
The software version in the RTU-810 can be checked in the webserver on the page "Device" → "Common". In this screenshot it's version 2.30:



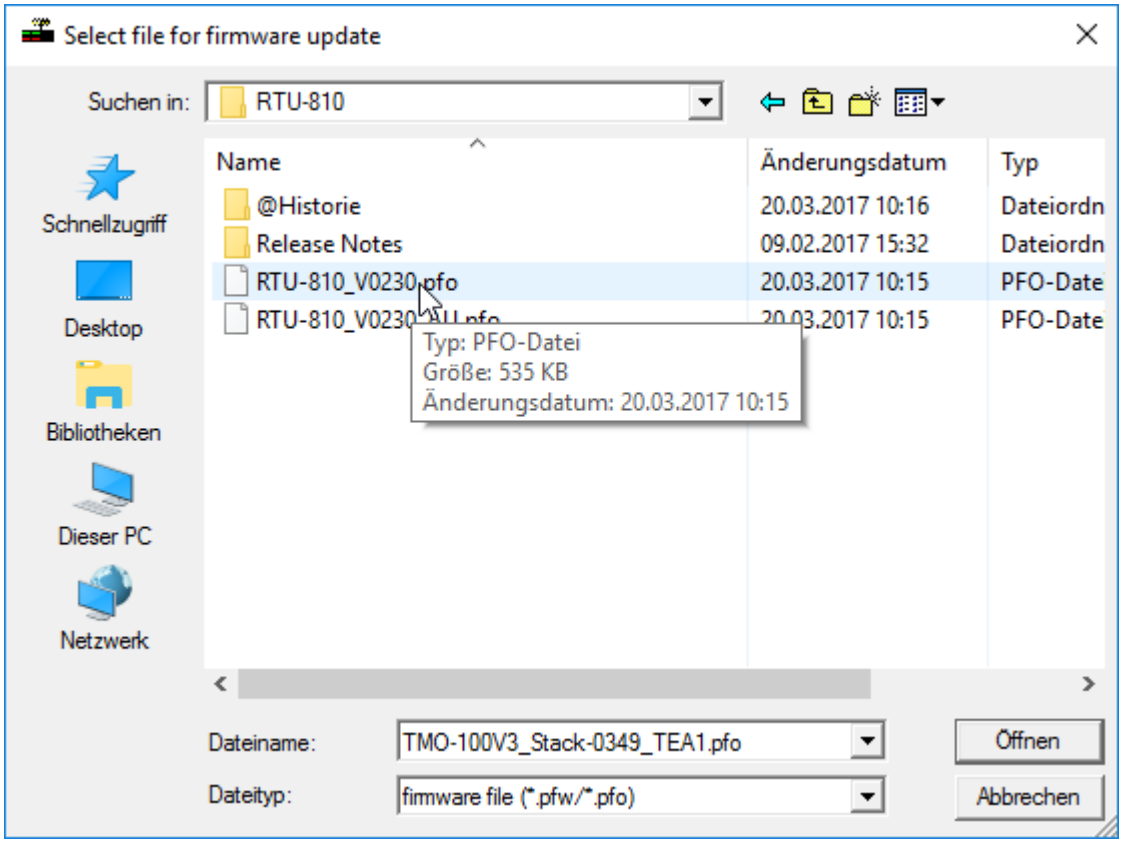
A firmware with the same version can be loaded into the RTU-810 but will not trigger the internal update process.

All configurations and feature activations are preserved and will not be lost after the update!

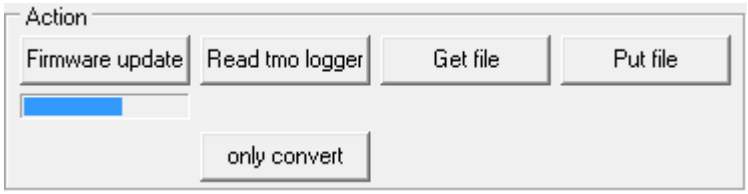
In the IPLoader the IP address of the RTU-810 must be entered in the "IP network" field. Then the button "Firmware update" must be clicked.



Select the RTU-810 firmware file and open it:



The IPLoader will immediately start the transfer to the RTU-810, indicating the transfer progress:



The internal update process will start immediately after the file has been transferred. The "Link"-LED on the Ethernet plug will go off for approximately around 30 seconds.

Important!
During the internal reprogramming, **DO NOT INTERRUPT THE POWER TO THE RTU-810.** Otherwise the RTU-810 will not start and must be restored by Piciorgros or a service partner.

After the update process has ended, the device will start immediately with the new firmware. The Link-LED on the Ethernet plug will light up again, along with the OK-LED. The firmware has been updated.

10 Specifications

Functions:	• Radio modem for serial data communication • Radio RTU with embedded I/O and optional PicoLogo Micro-PLC • Radio modem for direct I/O links (Point to (Multi)point) • Remote Alarm Monitor • Ethernet and serial RTU
RF output power:	• Up to 6W (High Power "H" version RTU-810) • Up to 500mW (Medium Power "M" version RTU-810) • Up to 10mW (RTU-81)
RF power adjustment:	Adjustable via embedded web server
Frequency range:	RTU-810/TRM-810: • 410-430 MHz (H/M) * • 430-450 MHz (H/M) * • 450-470 MHz (H/M) * • 860-870 MHz (M) * RTU-81: • 433-434 MHz
Receiver Sensitivity:	SN: typ.<1.2uV für 20dB, dF= 2.4kHz, fm= 1kHz SINAD: typ.<0.5uV, CCITT- Filter, dF= 2.4kHz, fm= 1kHz
Interfaces:	
COM:	• RS-232 or RS-485/422, Sub-D (F)
AUX:	• RS-232, Sub-D (F)
Ethernet:	• 10/100 Mbit/s
RTU I/O (optional):	All I/Os have separate electrical isolation
On-off inputs:	RTU-810:
On-off outputs:	• 16 on-off inputs (V_{in} = 12 to 24 VDC +/-20%)
Analog inputs:	• 8 (optional 16) outputs PNP (max. 500 mA per output)
Analog outputs:	• 4 analog inputs, 0-20mA / 4-20mA, 12-bit A/D resolution • 2 analog outputs, 0-20mA, 12-bit resolution, active source • Expansion port for PEM modules RTU-81: • 8 on-off inputs (V_{in} = 12 to 24 VDC +/-20%) • 8 outputs PNP (max. 500 mA per output) • 2 analog inputs, 0-20mA / 4-20mA, 12-bit A/D resolution • Expansion port for PEM modules

Special functions:	
Radio routing:	Up to 31 radio relays for each slave
RSSI display:	Calibrated front panel LED bar with Peak Hold
Network monitor:	RSSI monitor stores time and RSSI value of any station in the network
Configuration:	Fully via embedded web server
IP connectivity:	Supports IP SCADA protocols Supports partial network routing via IP Tunneling of serial protocols via IP (with SIG-810) Remotely offset parts of the radio network via RGW-810 radio gateway.
Protocols:	• MODBUS-RTU, MODBUS/TCP • IEC-60870-5-101, IEC-60870-5-104 • DNP3, DNP3/IP • PakBus, Siemens Sinaut, BSAP, RP570 • Custom protocols: serial or IP based
RF Conformance	EN 300 113 (H versions) * EN 300 220 (M versions and RTU-81) *
EMC Conformance	EN 301 489-1/-3/-5
ESD Conformance	61000-4-2 from 1998
Power supply voltage:	12-24 VDC +/- 20%
Power consumption (av.)	
Receive:	~140mA @12V, ~75mA @12V
Transmit:	~660mA @12V, ~325mA @24V during 500mW Tx ~2.4A @12V, ~1.2A @24V during 6W Tx
Enclosure:	Extruded aluminum body; plastic end caps
Operating Temperature:	-20°C to +55°C
Mounting:	35 mm DIN rail, symmetrical
Dimensions:	80mm x 162mm x 62 mm (excluding antenna and power connectors)

Remarks with (*)

M Version:
RF module with 10mW – 500 mW RF Output power

H Version:
RF module with 10=mW – 6 W RF Output power