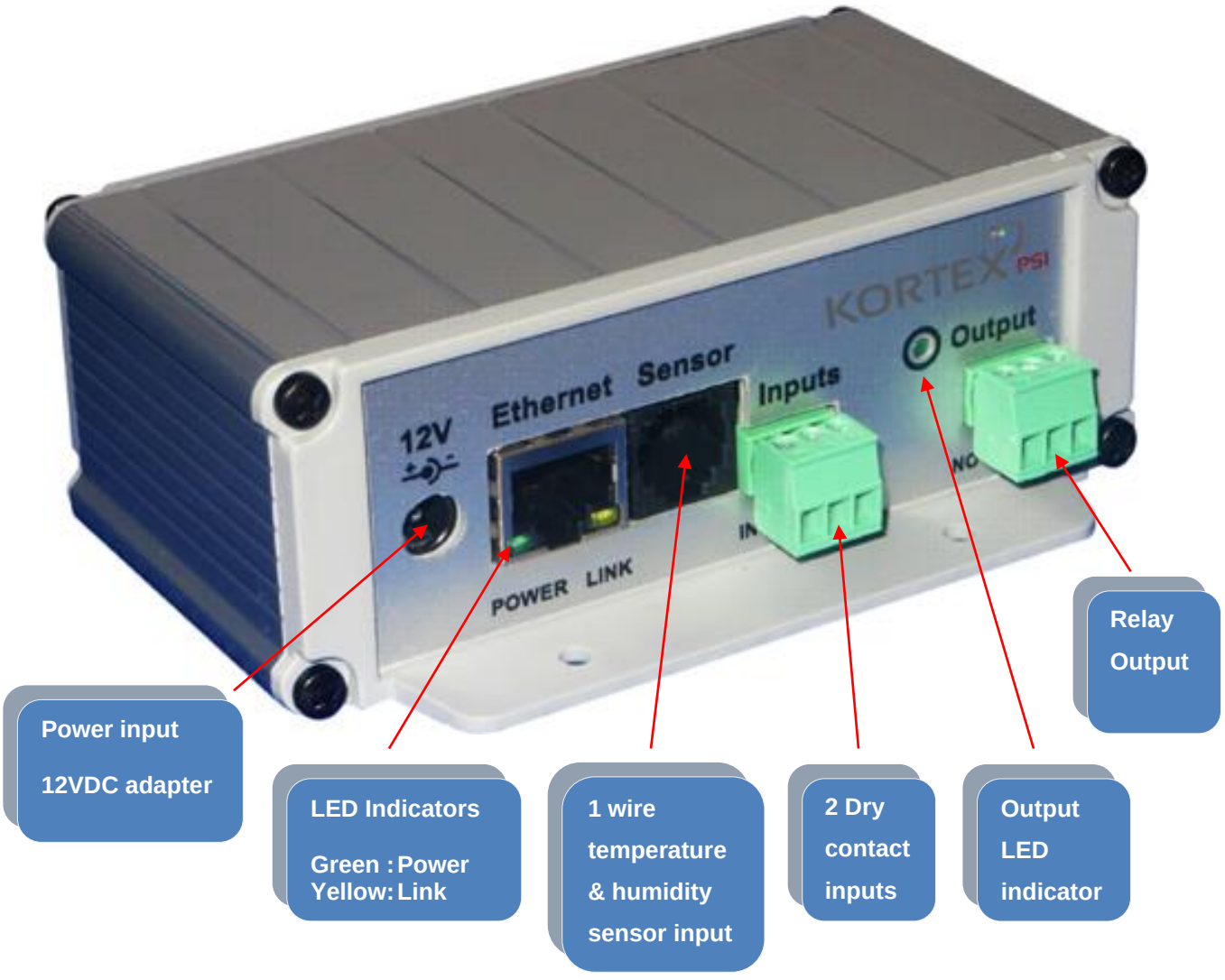
QUATTRO IOT

Manual



Benefits and Features.

KORTEX QUATTRO IOT is a device with build on Ethernet web server with 2 dry contact inputs, relay output, temperature & humidity sensor input and ICMP ping watchdog. Controller enables monitoring of digital inputs as well as control of outputs remotely over Ethernet. Configuration and control over TCP and SNMP v1 and v2.

Benefits

- Easy and fast configuration – full configuration can be performed through the Web using a web browser
- Control and configuration over TCP/IP and SNMP v1 and v2
- IP security – access to the QUATTRO IOT is protected by name and password
- 2 dry contact inputs
- 1 Relay Output for resetting monitored device
- Temperature & humidity sensor input for **up to 5 sensors**
- Monitoring of one IP device (server, router, camera) with automatic reboots in terms of monitored actions
- Methods of monitoring IP devices:
  - Incoming ping
  - Outgoing ping
- Alarm alert – SNMP trap and E-mail
- XML (over HTTP)
- SNTP synchronization protocol
- Firmware update over IP

Technical Features

- Power supply 12VDC / 0.5 A (adapter) – Jack Ø 5.5 x 2.0 / 10mm
- Network standards: IEEE 802.3, 10 Mbit/s
- Network connection: RJ 45 – 10BASE-T / 10 Mbps
- Network protocol – TCP/IP and HTTP
- Operating conditions: -20 to +70 °C
- 2 dry contact inputs (volt free)
- 1 Relay Output – 7A / 30VDC / 220VAC
- 1-Wire sensor input – RJ11 for connecting for **up to 5** temperature sensors or temperature & humidity sensors
- Aluminum enclosure
- Dimensions 112 x 82 x 40 [mm] / wall mount

Device Connection

Connect the QUATTRO IOT to the 10 Mbit or 10/100 Mbit network. Use a classic TP patch cable for connection to switch, router or to PC.

If the Ethernet connection is OK, the POWER and LINK indicator should light up and then LINK indicator blink according to network data transfer.

Restoring defaults factory configuration

Disconnect the power supply of the controller. Press and hold the Reset button located on the back of the box. When the button is held connect the power supply of the controller. Hold the button pressed for 10 seconds or more.

Default network parameters configuration:

IP address: 192.168.8.2

Network mask: 255.255.255.0

Default gateway: 192.168.8.1

IP configuration via DHCP: Disabled

User name: admin

Password: quattro75002

Installation Guidelines

QUATTRO IOT controller must be installed by qualified personnel.

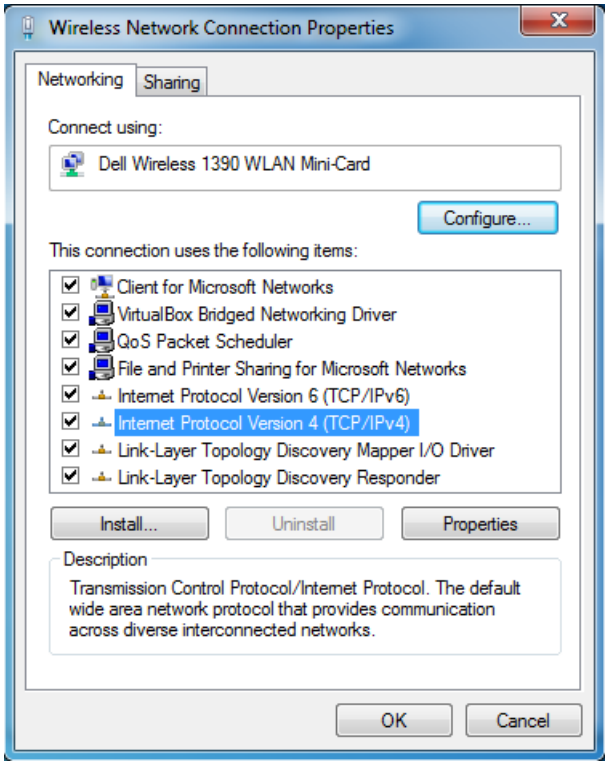
Controller must not be installed directly outdoors.

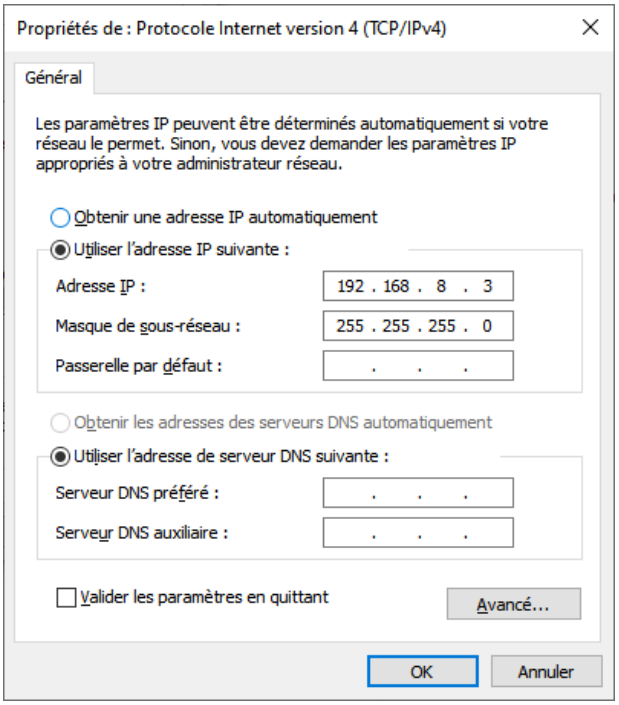
QUATTRO IOT must not be used for medical, life saving purposes, or for any purpose where its failure could cause serious injury or the loss of life.

This unit must not be used in any way where it's function or failure could cause significant loss or property damage.

The first step is to establish communication between computer and QUATTRO IOT controller. This can be done by assigning a temporary IP address to the computer.

For computers with Windows OS assigning of IP address is made in “Local area connection properties” or “Wireless Network Connection Properties”.





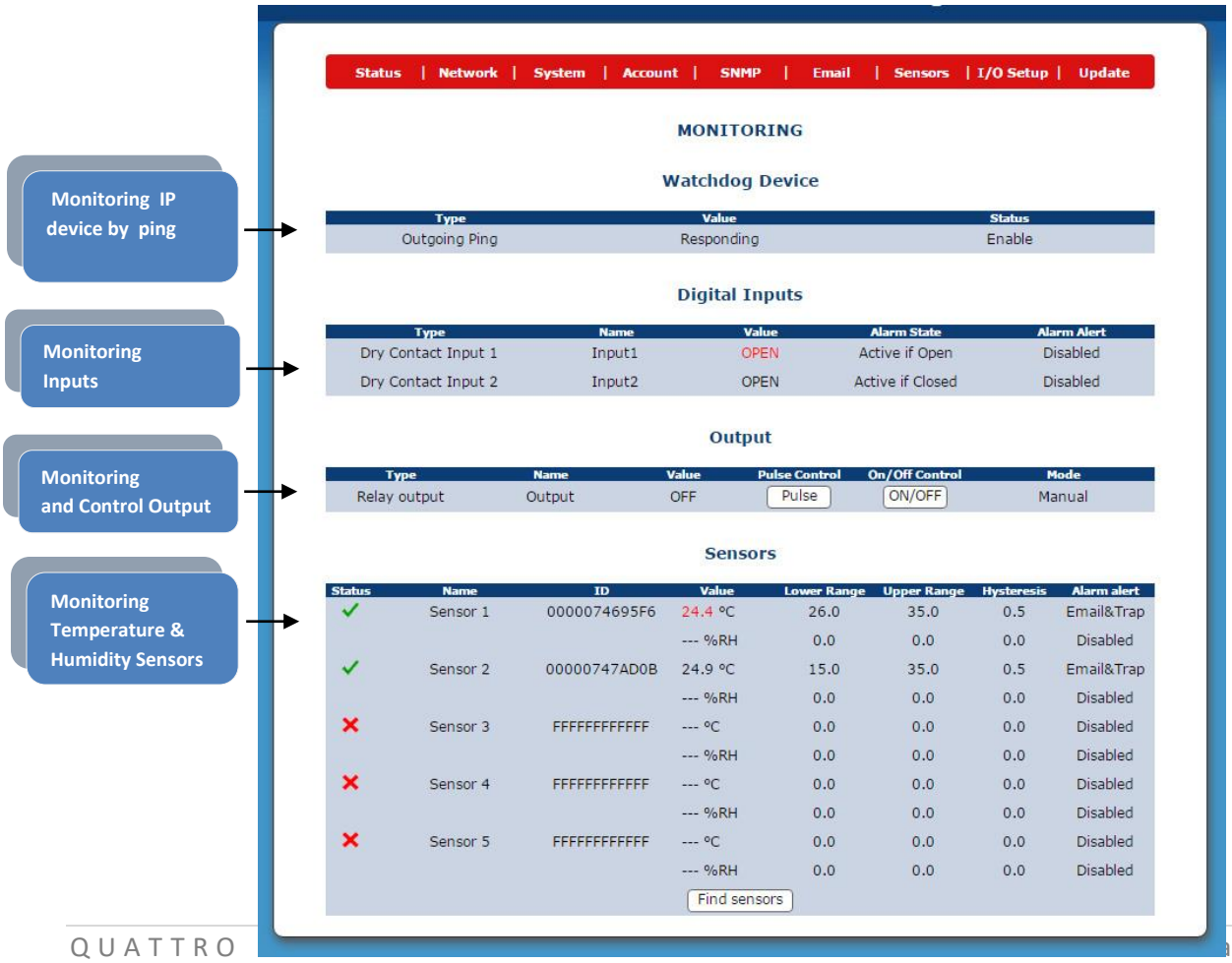
IP Address of the computer should be at the same network.

Into the browser you should type http://192.168.8.2. If network settings are correct the login page will appear.

Web Pages

❖ Login Page - access to the QUATTRO IOT is protected by name and password

❖ Status Page - Monitoring parameters



Status page has 4 sections – “Watchdog”, “Digital Inputs”, “Output” and “Sensors”.

- **Watchdog section** – monitoring “watchdog” function of controller
- **Digital Inputs section** can be used for monitoring sensors in “dry contact” (volt free) mode - motion PIR sensors, Water level sensors, Door/window sensors, Smoke detectors.

  One side of the sensor is connected to IN1 or IN2 terminal and the other is connected to GND terminal.

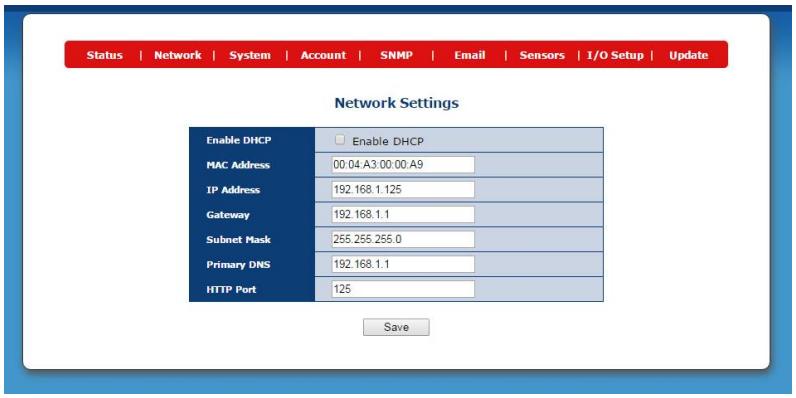
  Please note that Digital inputs are not galvanic isolated!
- **Output section** can be used to open/close garage, control motors, pumps, valves, lights …

  For loads greater than 7A / 30VDC / 220VAC an external relay should be used.
- **Sensors section** can be used for control 5x1-wire sensors for temperature and humidity

  Maximum cable length when is used UTP cable is 50m.

Network Settings – configuration of network parameters

- **MAC address** – hardware address of the device. After a change and click" Save" button, device will be rebooted automatically and redirect to the index page.



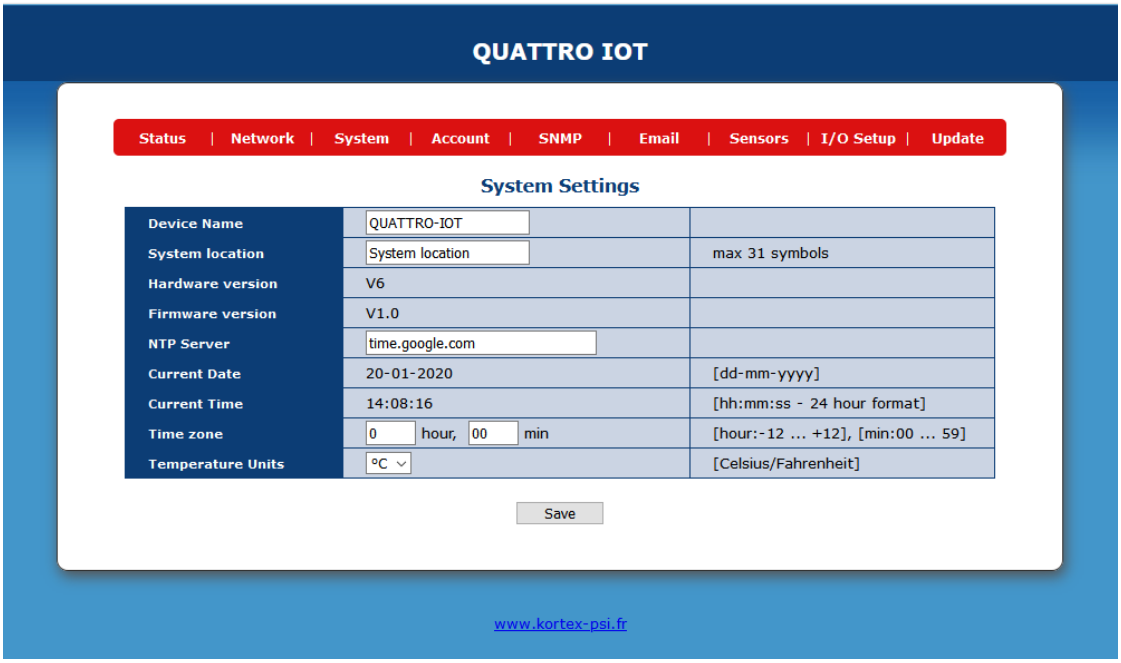
- **IP address** – software address of the device. After a change, device will be reboot automatically and you must search it on the new IP address.
- **Enable DHCP** - allows device automatically obtain a valid IP address
- **Gateway** – gateway IP address
- **Subnet mask** – local network mask
- **Primary DNS** – needs to be set properly for correct operation.
- **HTTP Port** – The TCP port used for HTTP communications with QUATTRO IOT. Default settings for this port is 80. Value of HTTP port can be changed from 80 to 65500.

When an HTTP request is without specific port then http://192.168.8.2 is exactly the same as http://192.168.8.2:80. For example, if port is changed to 1560 then HTTP request must be – http://192.168.8.2:1560.

DNS – Domain Name Service associates host names with IP addresses. Without a correctly configured DNS server, the following functions will not work:

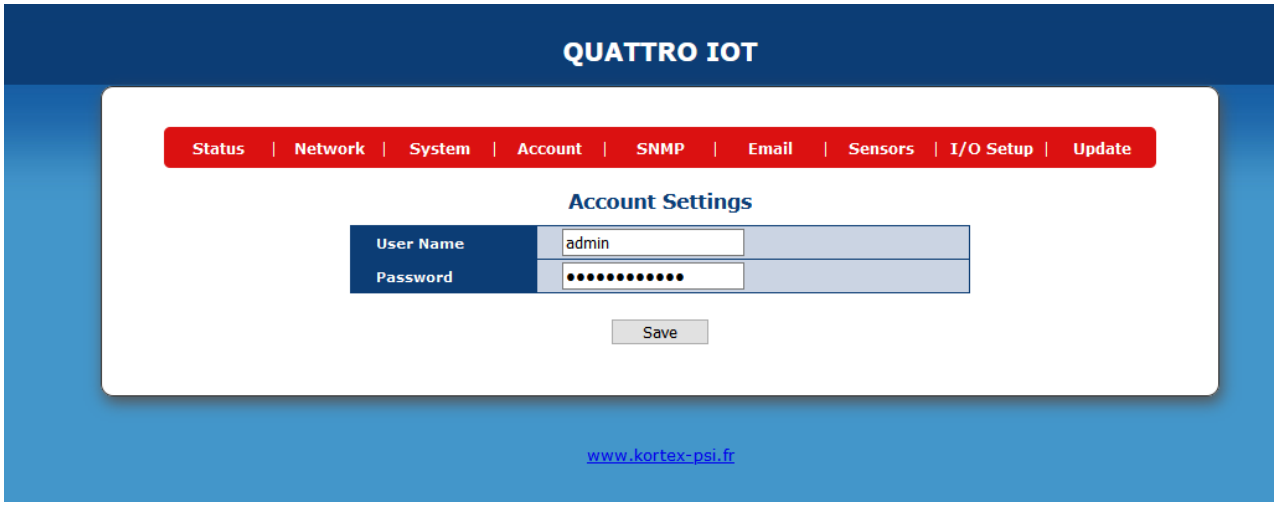
- Time synchronization (SNTP), used in e-mails and SNMP traps
- E- mailing (SMTP)

❖ System Settings

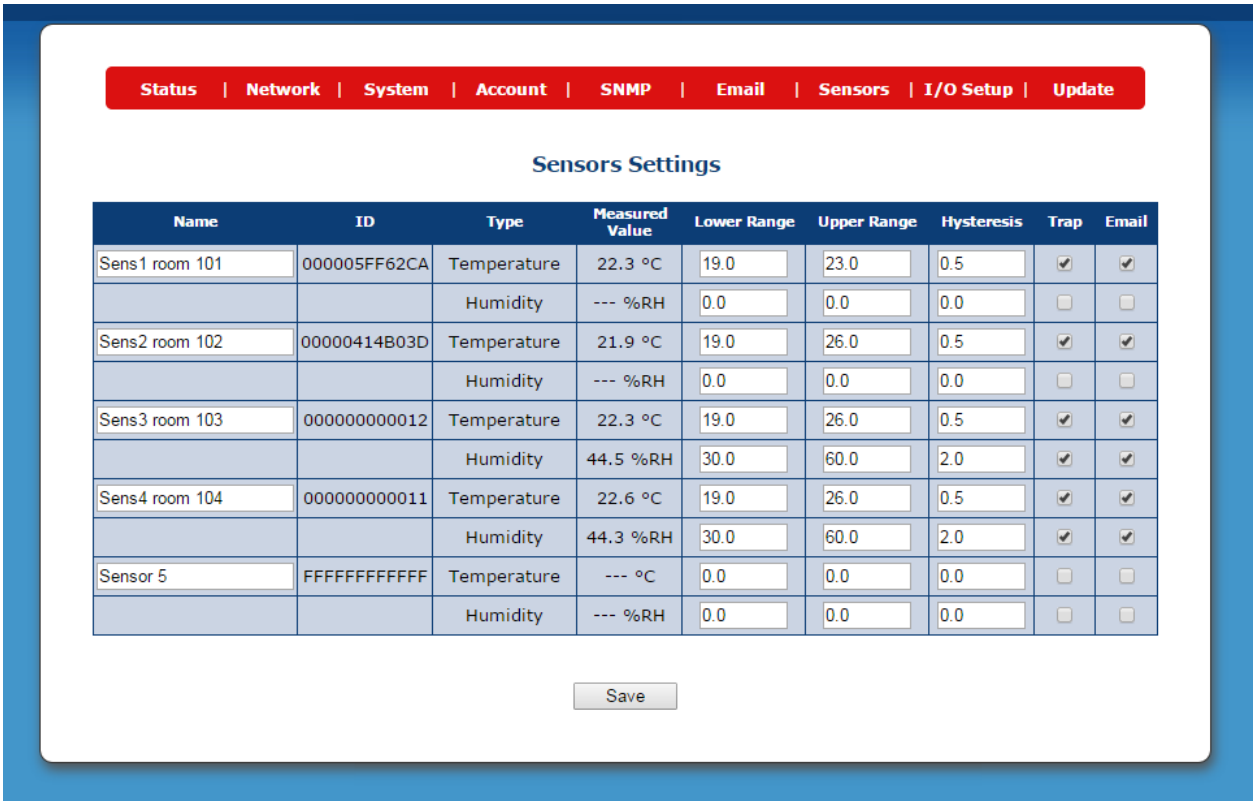


- **Device name** – is used to identify device on the network
- **NTP server** – host name of the NTP server to synchronize the time with.
- **Time zone** – offset of your time zone with respect to that of the NTP Server. If the time is not set the device attempts to synchronize the time approximately once per hour until successful.
- **Current Date -** it is necessary to send e-mails.
- **Current Time** – it is necessary to send e-mails. The time updates on every minute. To refresh the time, it is necessary to click the System page.
- **Temperature units** - Specifies the unit of temperature (C – Celsius / centigrade, F – Fahrenheit)

❖ **Account Settings** - One user account with username and password can be configured for HTTP access.

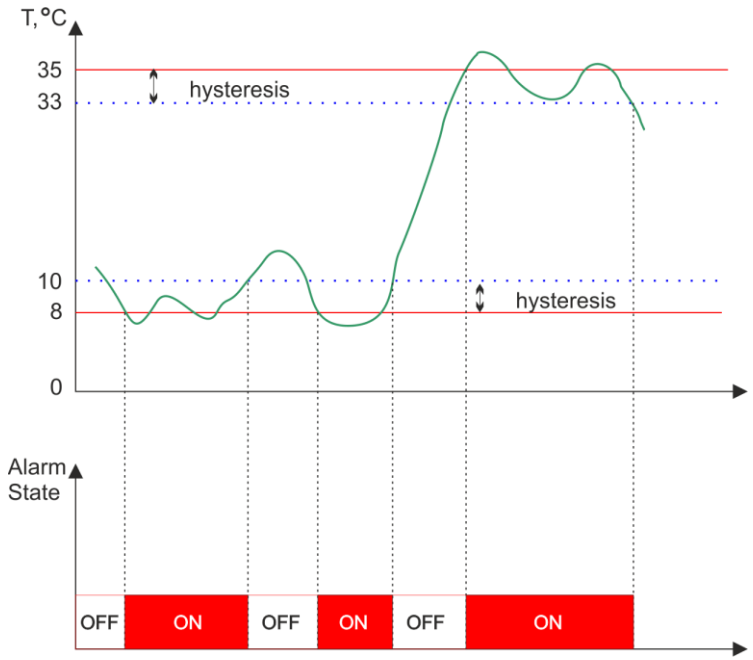


❖ **Sensors Settings**



Can be connected up to 5 temperature sensors or temperature / humidity sensors to the RJ11 input of the controller.

- **Name** – name of the sensor up to 17 chars (e.g. like “Indoor”)
- **Measured value** – the value that sensor is reading. Sensors that not working read as “---“
- **Lower range** - value below this range are considered to be “incorrect” and alarm event occurs
- **Upper range** - value above this range are considered to be “incorrect” and alarm event occurs
- **Hysteresis** – hysteresis is used to avoid multiple alarms when the measured quantity fluctuates.

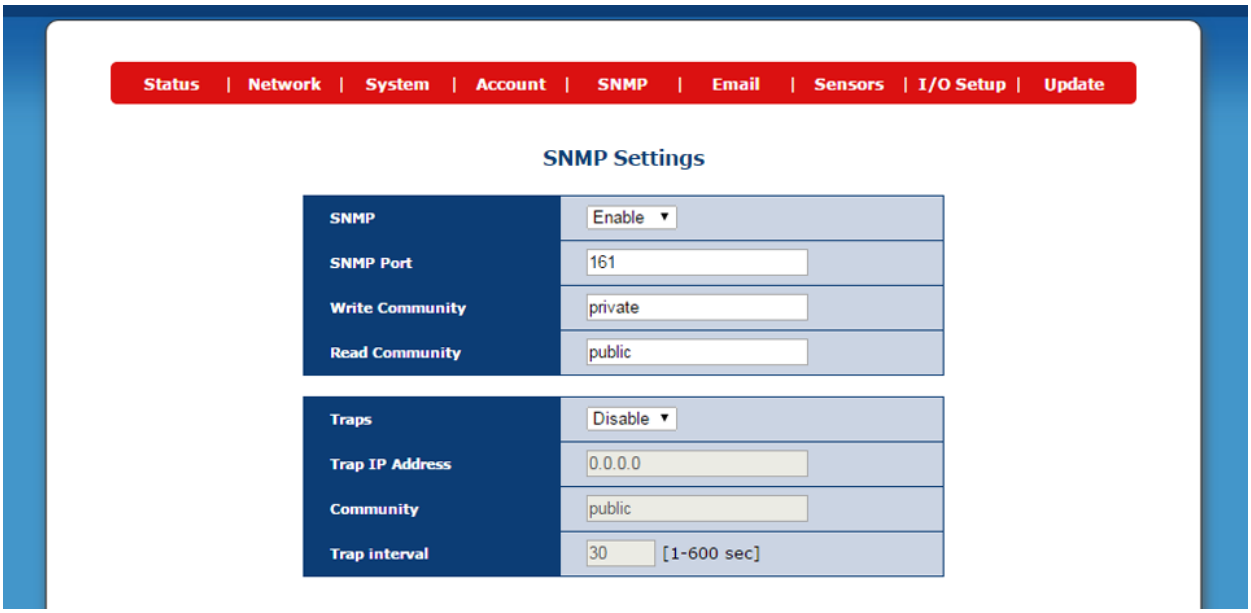


- **Trap** – device send **SNMP Trap**, when alarm event occurs and this option is checked
- **E-mail** - device send **E-mail**, when alrm event occurs and this option is checked

After connecting new sensors or changing RJ11 connections, detect sensors again by click on the button “Find sensors” on the “Status” web page.

❖ SNMP Settings

The Simple Network Management Protocol, SNMP, is a commonly used service that provides network management and monitoring capabilities. SNMP offers the capability to poll network device and monitor data. SNMP is also capable changing the configurations on the host, allowing the remote management of the network device. The protocol uses a community string for authentication from the SNMP client to the SNMP agent on the managed device.



The **SNMP Settings** page allows you to configure the settings for communication with the device using the SNMP protocol v2.0.

SNMP settings

- **SNMP** - Enable / Disable communication by SNMP
- **SNMP Port** – communication port for SNMP Protocol. Default port is 161.
- **Write Community** - The community is authorized to write values to variables over SNMP. The default write community string is “private”.
- **Read Community** - The community is authorized to read values to variables over SNMP. The default read community string is “public”.

Trap settings

- **Trap -** Enable / Disable sending trap messages
- **Trap IP address** — IP address of the trap destination
- **Trap community** – default trap community is “public”
- **Trap interval** – time interval for trap messages in seconds

Traps are sent in following conditions:

- dry contact inputs change their status
- measured temperature or humidity goes outside the range

For management and control by SNMP is necessary to use MIB file. MIB file you can download from our website www.kortex-psi.fr.

QUATTRO IOT SNMP parameters can be controlled with any SNMP compatible program. To obtain a valid OID number it is necessary to replace the “X” symbol with ”1.3.6.1.4.1.17095”.

product

OID	Name	Access	Description	Syntax
X.1.1.0	name	read-only	Device name	String
X.1.2.0	version	read-only	Firmware version	String

setup -> sensors -> sensor1

OID	Name	Access	Description	Syntax
X.2.1.1.1.0	sens1ID	read-only	Sensor 1 ID	String
X.2.1.1.2.0	sensor1Name	read-only	Sensor 1 description	String
X.2.1.1.3.0	temp1Lower	read-write	Temperature lower range	Integer
X.2.1.1.4.0	temp1Upper	read-write	Temperature upper range	Integer
X.2.1.1.5.0	temp1Hyst	read-write	Temperature hysteresis	Integer
X.2.1.1.6.0	temp1Email	read-write	Temperature email enable	Integer{no(0), yes(1)}
X.2.1.1.7.0	temp1Trap	read-write	Temperature trap enable	Integer{no(0), yes(1)}
X.2.1.1.8.0	humi1Lower	read-write	Humidity lower range	Integer

X.2.1.1.9.0	humi1Upper	read-write	Humidity lower range	Integer
X.2.1.1.10.0	humi1Hyst	read-write	Humidity hysteresis	Integer
X.2.1.1.11.0	humi1Email	read-write	Humidity email enable	Integer{no(0), yes(1)}
X.2.1.1.12.0	humi1Trap	read-write	Humidity trap enable	Integer{no(0), yes(1)}

setup -> sensors -> sensor2

OID	Name	Access	Description	Syntax
X.2.1.2.1.0	sens2ID	read-only	Sensor 2 ID	String
X.2.1.2.2.0	sensor2Name	read-only	Sensor 2 description	String
X.2.1.2.3.0	temp2Lower	read-write	Temperature lower range	Integer
X.2.1.2.4.0	temp2Upper	read-write	Temperature upper range	Integer
X.2.1.2.5.0	temp2Hyst	read-write	Temperature hysteresis	Integer
X.2.1.2.6.0	temp2Email	read-write	Temperature email enable	Integer{no(0), yes(1)}
X.2.1.2.7.0	temp2Trap	read-write	Temperature trap enable	Integer{no(0), yes(1)}
X.2.1.2.8.0	humi2Lower	read-write	Humidity lower range	Integer
X.2.1.2.9.0	humi2Upper	read-write	Humidity lower range	Integer
X.2.1.2.10.0	humi2Hyst	read-write	Humidity hysteresis	Integer
X.2.1.2.11.0	humi2Email	read-write	Humidity email enable	Integer{no(0), yes(1)}
X.2.1.2.12.0	humi2Trap	read-write	Humidity trap enable	Integer{no(0), yes(1)}

setup -> sensors -> sensor3

OID	Name	Access	Description	Syntax
X.2.1.3.1.0	sens3ID	read-only	Sensor 3 ID	String
X.2.1.3.2.0	sensor3Name	read-only	Sensor 3 description	String
X.2.1.3.3.0	temp3Lower	read-write	Temperature lower range	Integer
X.2.1.3.4.0	temp3Upper	read-write	Temperature upper range	Integer
X.2.1.3.5.0	temp3Hyst	read-write	Temperature hysteresis	Integer
X.2.1.3.6.0	temp3Email	read-write	Temperature email enable	Integer{no(0), yes(1)}

X.2.1.3.7.0	temp3Trap	read-write	Temperature trap enable	Integer{no(0), yes(1)}
X.2.1.3.8.0	humi3Lower	read-write	Humidity lower range	Integer
X.2.1.3.9.0	humi3Upper	read-write	Humidity lower range	Integer
X.2.1.3.10.0	humi3Hyst	read-write	Humidity hysteresis	Integer
X.2.1.3.11.0	humi3Email	read-write	Humidity email enable	Integer{no(0), yes(1)}
X.2.1.3.12.0	humi3Trap	read-write	Humidity trap enable	Integer{no(0), yes(1)}

setup -> sensors -> sensor4

OID	Name	Access	Description	Syntax
X.2.1.4.1.0	sens4ID	read-only	Sensor 4 ID	String
X.2.1.4.2.0	sensor4Name	read-only	Sensor 4 description	String
X.2.1.4.3.0	temp4Lower	read-write	Temperature lower range	Integer
X.2.1.4.4.0	temp4Upper	read-write	Temperature upper range	Integer
X.2.1.4.5.0	temp4Hyst	read-write	Temperature hysteresis	Integer
X.2.1.4.6.0	temp4Email	read-write	Temperature email enable	Integer{no(0), yes(1)}
X.2.1.4.7.0	temp4Trap	read-write	Temperature trap enable	Integer{no(0), yes(1)}
X.2.1.4.8.0	humi4Lower	read-write	Humidity lower range	Integer
X.2.1.4.9.0	humi4Upper	read-write	Humidity lower range	Integer
X.2.1.4.10.0	humi4Hyst	read-write	Humidity hysteresis	Integer
X.2.1.4.11.0	humi4Email	read-write	Humidity email enable	Integer{no(0), yes(1)}
X.2.1.4.12.0	humi4Trap	read-write	Humidity trap enable	Integer{no(0), yes(1)}

setup -> sensors -> sensor5

OID	Name	Access	Description	Syntax
X.2.1.5.1.0	sens5ID	read-only	Sensor 5 ID	String
X.2.1.5.2.0	sensor5Name	read-only	Sensor 5 description	String
X.2.1.5.3.0	temp5Lower	read-write	Temperature lower range	Integer
X.2.1.5.4.0	temp5Upper	read-write	Temperature upper range	Integer
X.2.1.5.5.0	temp5Hyst	read-write	Temperature hysteresis	Integer
X.2.1.5.6.0	temp5Email	read-write	Temperature email enable	Integer{no(0), yes(1)}
X.2.1.5.7.0	temp5Trap	read-write	Temperature trap enable	Integer{no(0), yes(1)}
X.2.1.5.8.0	humi5Lower	read-write	Humidity lower range	Integer
X.2.1.5.9.0	humi5Upper	read-write	Humidity lower range	Integer
X.2.1.5.10.0	humi5Hyst	read-write	Humidity hysteresis	Integer
X.2.1.5.11.0	humi5Email	read-write	Humidity email enable	Integer{no(0), yes(1)}
X.2.1.5.12.0	humi5Trap	read-write	Humidity trap enable	Integer{no(0), yes(1)}

setup -> sensors

OID	Name	Access	Description	Syntax
X.2.1.6.0	tempUnit	read-write	Temperature sensors unit	Integer{ celcius(0), fahrenheit(1)}

setup -> ping

OID	Name	Access	Description	Syntax
X.2.2.1.0	pingEnabled	read-write	Ping Enable	Integer{no(0), yes(1)}
X.2.2.2.0	pingIncomming	read-write	Incomming Ping Enable	Integer{no(0), yes(1)}
X.2.2.3.0	pingOutgoing	read-write	Outgoing Ping Enable	Integer{no(0), yes(1)}
X.2.2.4.0	pingRestartPulse	read-write	Restart pulse time	Integer
X.2.2.5.0	pingTimeoutForRestart	read-write	Timeout for restart	Integer
X.2.2.6.0	pingTimeoutAfterRestart	read-write	Timeout after restart	Integer
X.2.2.7.0	pingNumOfRestart	read-write	Numbers of restarts	Integer
X.2.2.8.0	pingInterval	read-write	Time interval between pings	Integer

setup -> inputs -> input1

OID	Name	Access	Description	Syntax
X.2.3.1.1.0	input1Name	read-write	Digital input 1 name	String
X.2.3.1.2.0	input1AlarmState	read-write	Digital input 1 alarm state	Integer
X.2.3.1.3.0	input1TrapEnable	read-write	Digital input 1 trap enable	Integer{no(0), yes(1)}
X.2.3.1.4.0	input1MailEnable	read-write	Digital input 1 email enable	Integer{no(0), yes(1)}

setup -> digitals -> diginput2

OID	Name	Access	Description	Syntax
X.2.3.2.1.0	Input2Name	read-write	Digital input 2 name	String
X.2.3.2.2.0	Input2AlarmState	read-write	Digital input 2 alarm state	Integer
X.2.3.2.3.0	Input2TrapEnable	read-write	Digital input 2 trap enable	Integer{no(0), yes(1)}
X.2.3.2.4.0	Input2MailEnable	read-write	Digital input 2 email enable	Integer{no(0), yes(1)}

setup -> outputs

OID	Name	Access	Description	Syntax
X.2.4.1.0	out1Name	read-write	Output 1 name	String
X.2.4.2.0	out1Control	read-write	Output 1 setup control	Integer { manual(0), dryContactInput(1), dryContactInput(2), watchdog(3), temperature1(4), humidity1(5), temperature2(6), humidity2(7), temperature3(8), humidity3(9), temperature4(10), humidity4(11), temperature5(12), humidity5(13), allSensors(14)}

setup

OID	Name	Access	Description	Syntax
X.2.5.0	configSaved	read-write	Configuration save	Integer

control

OID	Name	Access	Description	Syntax
X.3.1.0	output1	read-write	Relay 1 output	Integer
X.3.2.0	dryContactInput1	read-only	Dry contact input 1	Integer{closed(0), open(1)}
X.3.3.0	dryContactInput2	read-only	Dry contact input 2	Integer{closed(0), open(1)}
X.3.4.0	temperature1	read-only	Sensor 1 temperature	Integer(-400-:-1250)
X.3.5.0	humidity1	read-only	Sensor 1 humidity	Integer(0 -:-1000)
X.3.6.0	temperature2	read-only	Sensor 2 temperature	Integer(-400-:-1250)
X.3.7.0	humidity2	read-only	Sensor 2 humidity	Integer(0 -:-1000)
X.3.8.0	temperature3	read-only	Sensor 3 temperature	Integer(-400-:-1250)
X.3.9.0	humidity3	read-only	Sensor 3 humidity	Integer(0 -:-1000)
X.3.10.0	temperature4	read-only	Sensor 4 temperature	Integer(-400-:-1250)
X.3.11.0	humidity4	read-only	Sensor 4 humidity	Integer(0 -:-1000)
X.3.12.0	temperature5	read-only	Sensor 5 temperature	Integer(-400-:-1250)
X.3.13.0	humidity5	read-only	Sensor 5 humidity	Integer(0 -:-1000)
X.3.14.0	rstContr	read-write	Restart controller	Integer{cancel(0), restart(1)}

deviceinfo

OID	Name	Access	Description	Syntax
X.4.1.0	deviceMacAddress	read-only	Device MAC Address	String

❖ Watchdog settings

- **Ping** – Enable / Disable watchdog function
- **Ping mode** – Incoming or Outgoing ping mode
- **IP address** – of monitoring devices
- **Restart pulse** – time of relay restart pulse in seconds
- **Ping interval** – interval for outgoing pings
- **Ping timeout for restart** – timeout during which the controller waits for incoming ping or answer from outgoing ping before restart controlled device
- **Timeout after restart** – in seconds
- **Numbers of restarts** – number of restarts of the controlled device

Watchdog Settings

Ping	Enable
Ping Mode	Incomming
IP address	0.0.0.0
Device Name	Device
Restart Pulse	0 [1-200 sec]
Ping Interval	5 [1-200 sec]
Ping Timeout for Restart	50 [1-2000 sec]
Timeout after Restart	50 [1-2000 sec]
Numbers of Restarts	5 [1-200]
Email when Restart	No

You need before to select “Watchdog” on the “Relay Control” section, from the Inputs/Outputs Settings.

Inputs/Outputs Settings

Digital Inputs

Name	Current value	Low level	High level	Alarm state	Trap	Email
Input1	OPEN	CLOSED	OPEN	CLOSED	No	No
Input2	OPEN	CLOSED	OPEN	CLOSED	No	No

Relay Output

Name	Current value	Relay Control	Pulse duration, sec.
Output	OFF	Watchdog	10

Save

Example of using Watchdog:

When we want watchdog to reboot device after 5 unsuccessful pings, so we can set the value of parameter “Ping timeout for restart” 50 sec. and set ping interval to 5 sec.

After each reply of ping before 50 seconds have elapsed, the parameter “Ping timeout for restart” is loaded automatically with 50 sec - so the watchdog will not restarted device.

If controller not received answer during the time of 50 sec. then the watchdog restarted controlled device. When the controller reaches the number of "Number of Restart", then stops restart function.

"Timeout after Restart" is the time that the controller waits before it starts counting down again "Ping timeout for Restart".

❖ HTTP API commands and XML status

The relays can be controled by HTTP commands. Before that is required basic authentication .

HTTP command	Description
http:// device_ip_address /status.xml?r1=1	Relay 1 = ON
http:// device_ip_address /status.xml?r1=0	Relay 1 = OFF
http:// device_ip_address /status.xml?tg1=1	Toggle Relay 1 state
http:// device_ip_address /status.xml?p1=1	Pulse Relay 1

Any application can easily read current values from “status.xml” tags.

File is located at http://device_ip_address/status.xml

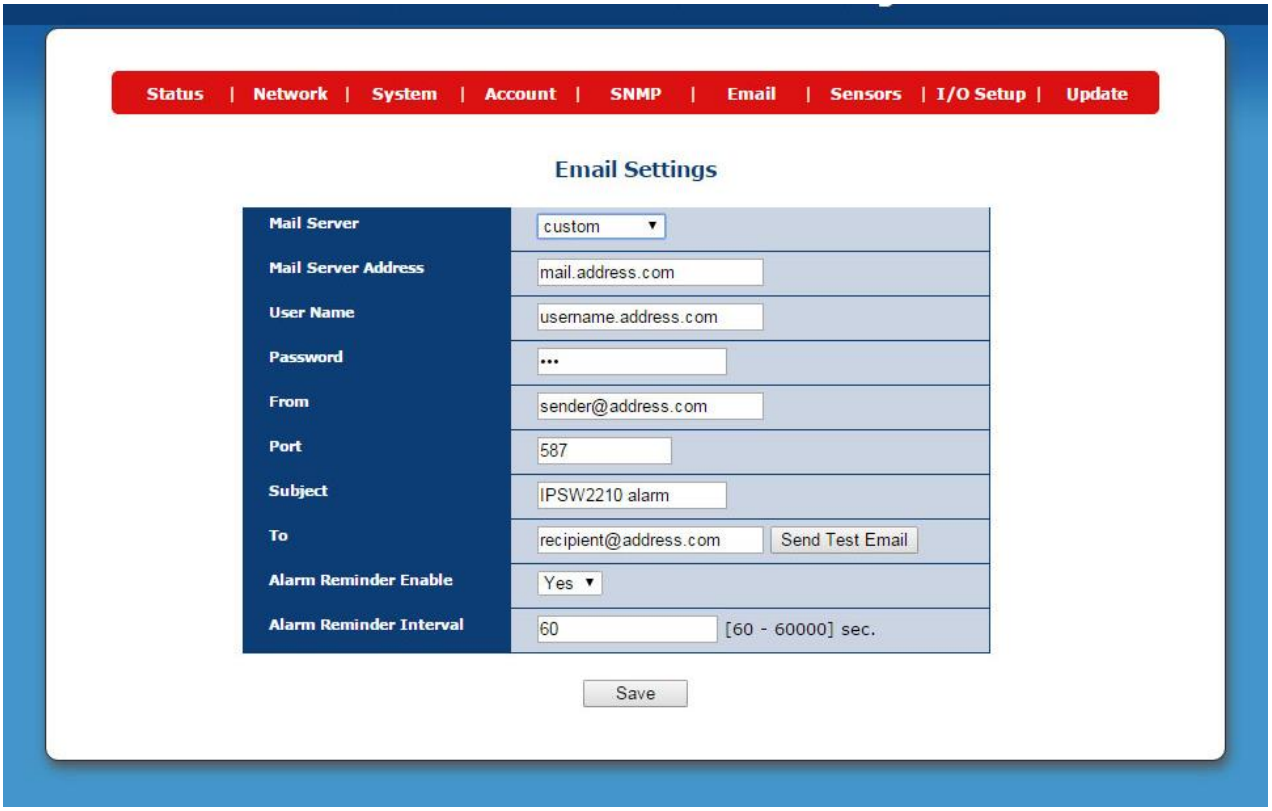
```
<response>
      <Device>QUATTRO IOT</Device>
      <HostName>QUATTRO IOT</HostName>
      <MAC>00:04:A3:00:00:A9</MAC>
      <Location>Office</Location>
      <ping>
            <Name>Ping</Name>
            <Ping_Value>Responding</Ping_Value>
            <Ping_Status>Enable</Ping_Status>
      </ping>
      <dIn1>
            <In1_name>Input1</In1_name>
            <In1_val>OPEN</In1_val>
            <In1_alarm>1</In1_alarm>
      </dIn1>
      <dIn2>
```

```
	<In2_name>Input2</In2_name>
	<In2_val>OPEN</In2_val>
	<In2_alarm>0</In2_alarm>
</dIn2>
<Out1>
	<Out1_name>Output</Out1_name>
	<Out1_val>OFF</Out1_val>
	<Out1_pw>1</Out1_pw>
	<Out1_contr>Manual</Out1_contr>
</Out1>
<S1>
	<S1_name>Sensor 1</S1_name>
	<S1_id>0000074695F6</S1_id>
	<S1T_val>22.8</S1T_val>
	<S1T_unit>°C</S1T_unit>
	<S1T_alarm>1</S1T_alarm>
	<S1H_val>---</S1H_val>
	<S1H_alarm>0</S1H_alarm>
	<S1_st>1</S1_st>
</S1>
<S2>
	<S2_name>Sensor 2</S2_name>
	<S2_id>00000747AD0B</S2_id>
	<S2T_val>23.4</S2T_val>
	<S2T_unit>°C</S2T_unit>
	<S2T_alarm>0</S2T_alarm>
	<S2H_val>---</S2H_val>
	<S2H_alarm>0</S2H_alarm>
	<S2_st>1</S2_st>
</S2>
<S3>
	<S3_name>Sensor 3</S3_name>
	<S3_id>FFFFFFFFFFFF</S3_id>
	<S3T_val>---</S3T_val>
	<S3T_unit>°C</S3T_unit>
	<S3T_alarm>0</S3T_alarm>
	<S3H_val>---</S3H_val>
	<S3H_alarm>0</S3H_alarm>
	<S3_st>0</S3_st>
</S3>
<S4>
	<S4_name>Sensor 4</S4_name>
	<S4_id>FFFFFFFFFFFF</S4_id>
	<S4T_val>---</S4T_val>
	<S4T_unit>°C</S4T_unit>
	<S4T_alarm>0</S4T_alarm>
	<S4H_val>---</S4H_val>
	<S4H_alarm>0</S4H_alarm>
	<S4_st>0</S4_st>
</S4>
<S5>
	<S5_name>Sensor 5</S5_name>
	<S5_id>FFFFFFFFFFFF</S5_id>
	<S5T_val>---</S5T_val>
	<S5T_unit>°C</S5T_unit>
	<S5T_alarm>0</S5T_alarm>
	<S5H_val>---</S5H_val>
```

```
            <S5H_alarm>0</S5H_alarm>
            <S5_st>0</S5_st>
      </S5>
</response>
```

❖ Email Settings

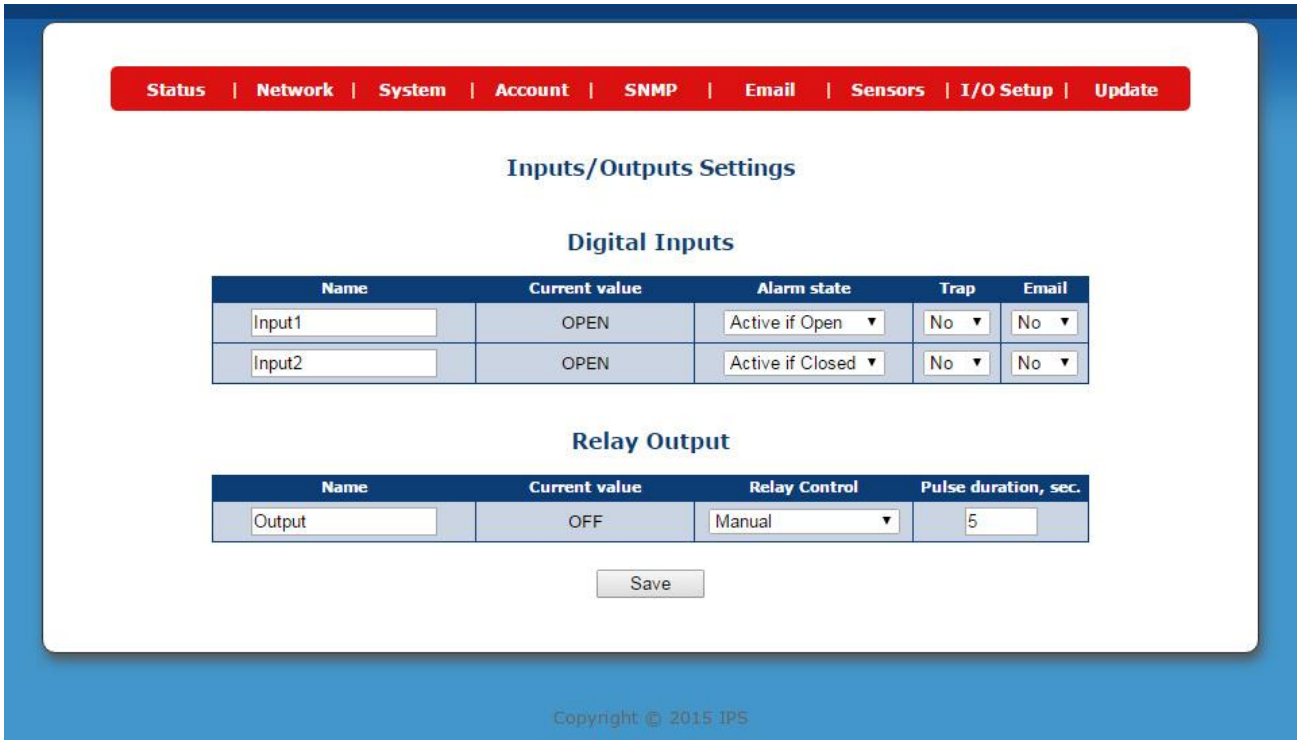
- **Mail server** – **custom**.
- **Mail Server Address** – Host name or IP address of the SMTP server
- **Authentication** – if necessary, use name and password for Mail server
- **Port** – This controls the port for which outgoing mail connections will be accepted. SMTP uses TCP port 25 or 587. **SSL is not supported.**
- **Alarm Reminder Enable –** enable alarm reminder option
- **Alarm Reminder Interval –** time interval in sec.



Once the parameter cross threshold level, controller send alarm email.

With Alarm Reminder you can get emails with current value of alarmed parameter during Alarm Reminder Interval of time.

❖ Inputs / Outputs Settings



- **Name** - In this field you can enter the name of the corresponding parameter as it should be spelled in the Status page
- **Dry contact inputs –** used for direct connection of volt-free contacts. One side of the contact is connected to Input1 or Input2 and the other side is connected to terminal in the middle – GND. When alarm state is “Active if Close” then default value is “Open” and the alarm is activated when active value become “Close”. **Digital inputs are not galvanic isolated!**

  When alarm state is “Active if Open” then default value is “Closed” and alarm is activated when active value become “Open”.

  Choose which type of alarm to be activated – Trap or Email or both alarms**.**

- **Relay control** - select whether the relay to be activated manually or automatic by any of the following parameters or by all sensors:

  - Dry contact input 1

- Dry contact input 2
- Watchdog
- S1 Temp
- S1 Hum
- S2 Temp
- S2 Hum
- S3 Temp
- S3 Hum
- S4 Temp
- S4 Hum
- S5 Temp
- S5 Hum
- All Sensors

For loads greater than 7A an external relay should be used.

For Relay control can be set different time for Pulse duration. The resolution is 1 second. By default, relay are activated manualy by web interface of Status page.

❖ SMS commands to control QUATTRO IOT relay output with 4G KORTEX router

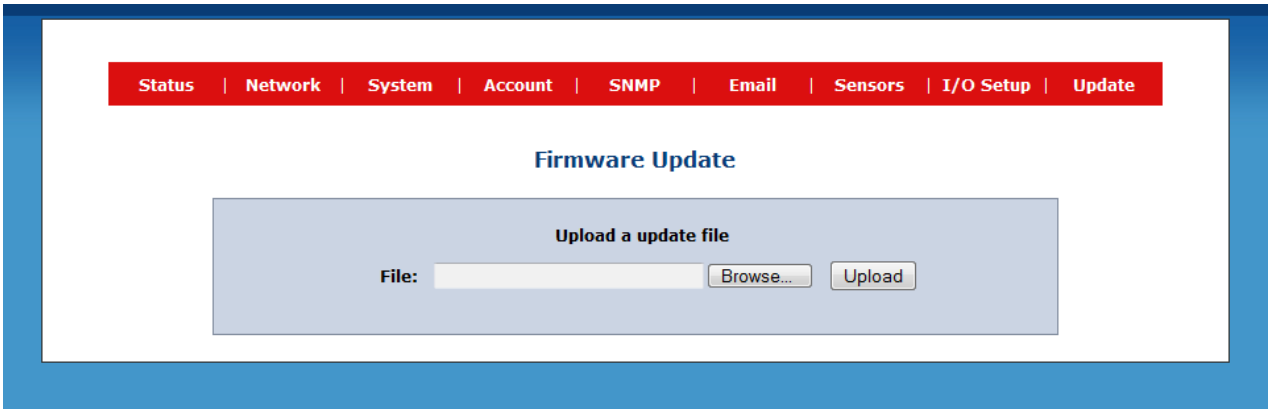
SMS commands for controlling the QUATTRO IOT

QR1=ON command activates the relay.
QR1=OFF command deactivates the relay.
QR1=PULSE command pulse with the delay value set from QUATTRO IOT.
QR1? command allows to obtain the relay status.
QR1 = xxx (in seconds) command, switches the relay in the opposite direction for xxx seconds and reposition it in its original state (ON or OFF).

❖ Firmware update

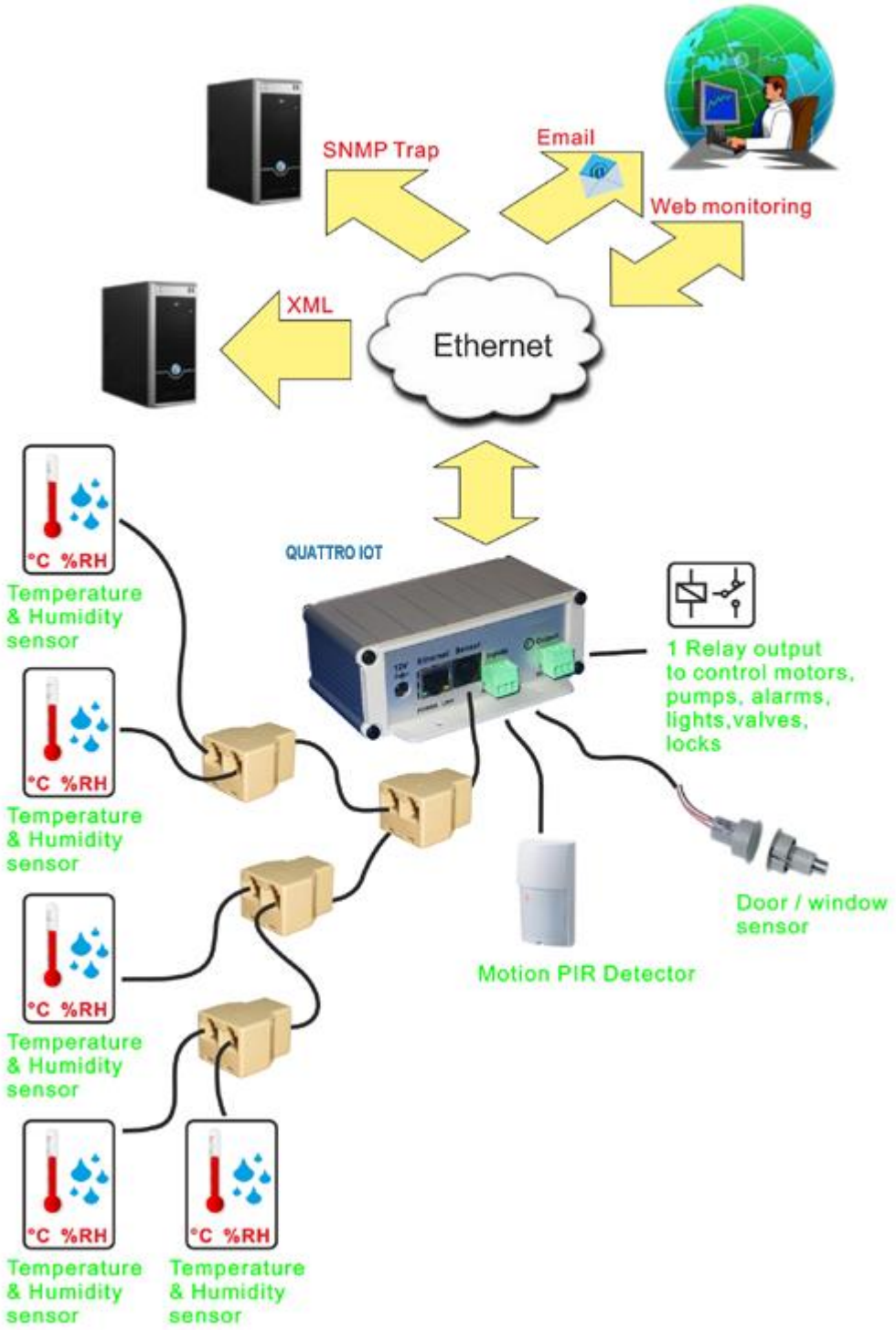
In this page you can update QUATTRO IOT with the new firmware.

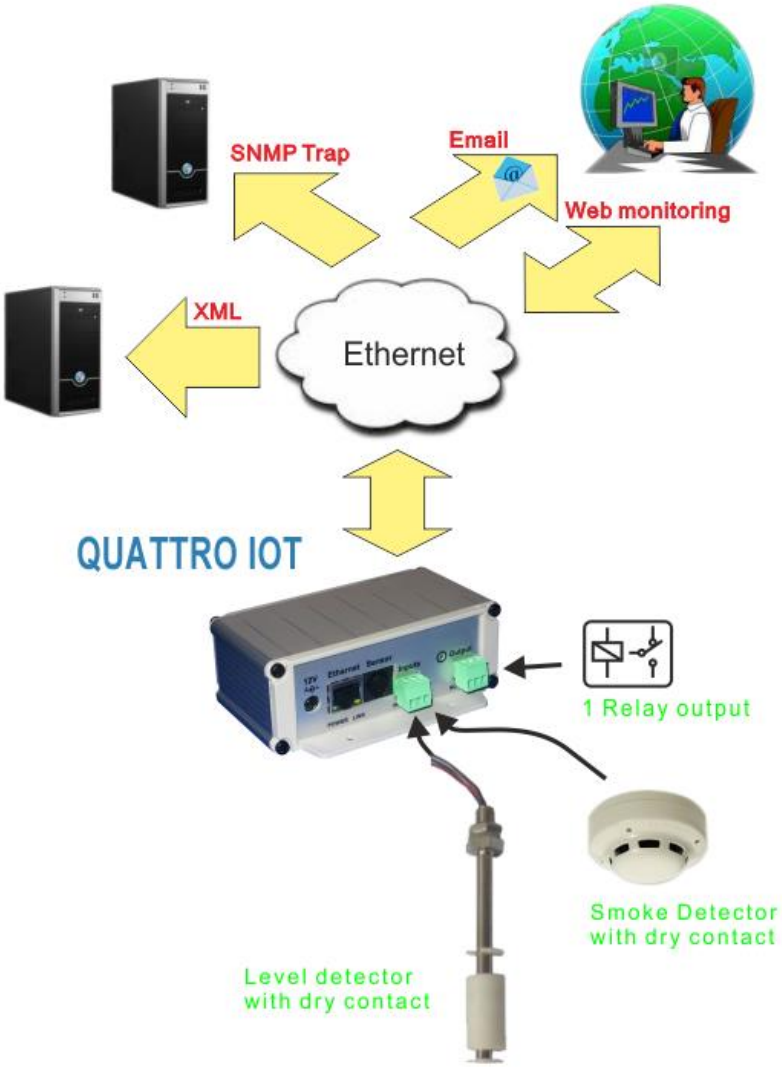
Please keep in mind not to power off the device during the update! It can damage the device!



Application Examples

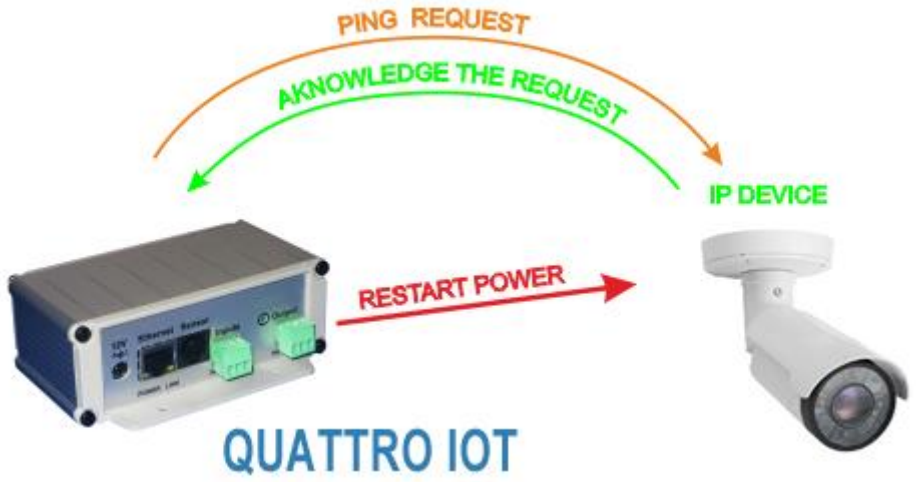
- **Remote monitoring applications**



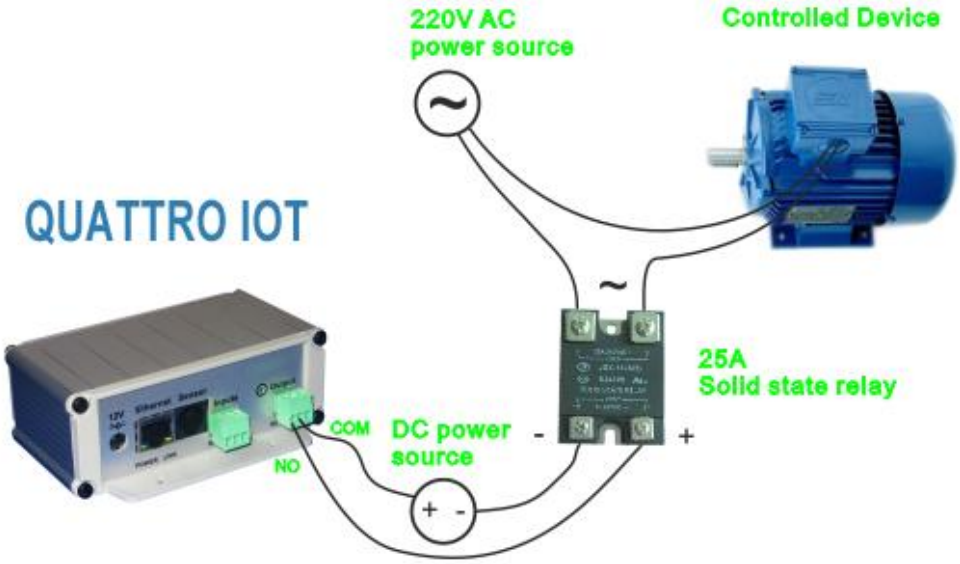

SNMP Trap
Email
Web monitoring
XML
Ethernet
QUATTRO IOT
1 Relay output
Smoke Detector
with dry contact
Level detector
with dry contact

- **Watchdog application**

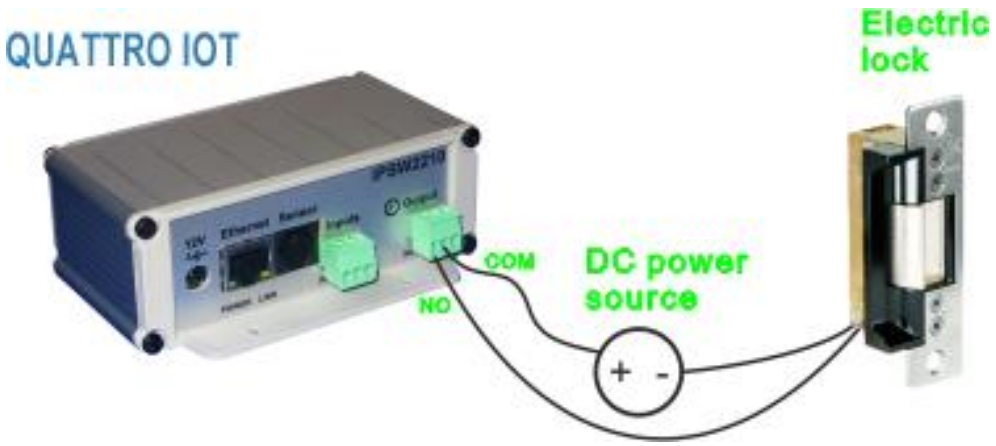
QUATTRO IOT send a ping request to monitored device. The monitored device acknowledges the request. In case of lock-up it will be automatically rebootes.

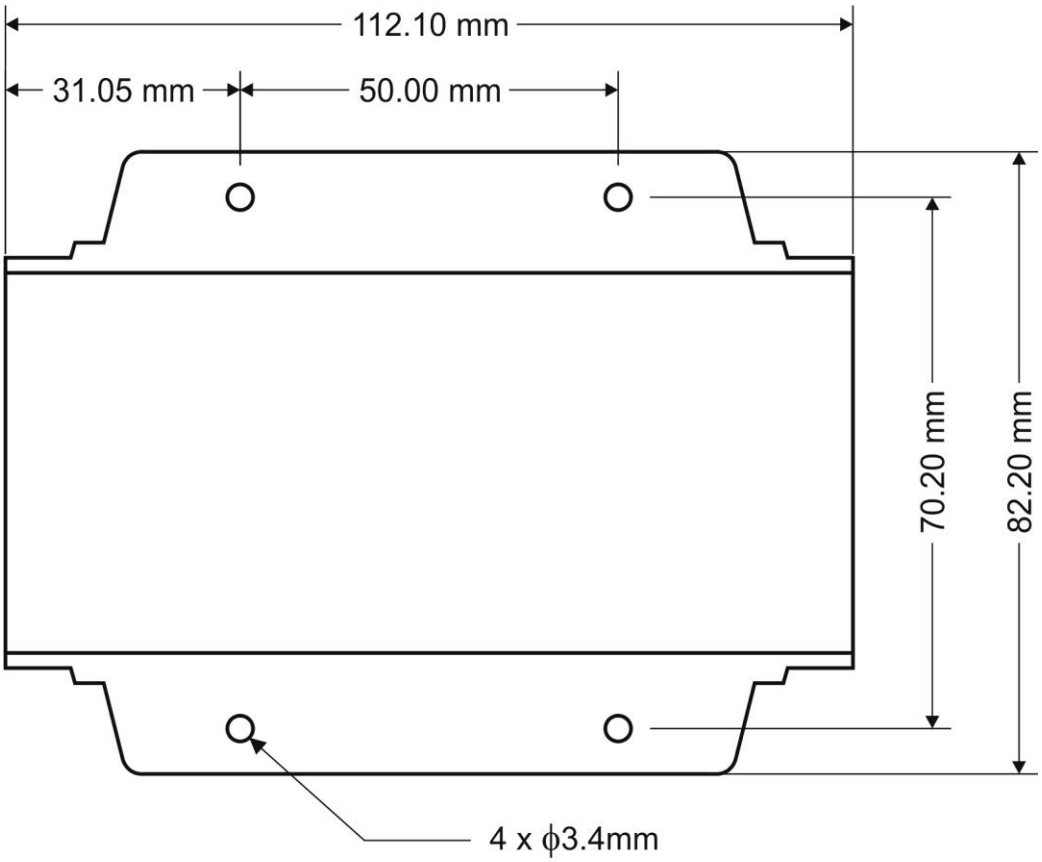


- **Control of 220V AC / 20 A motor using external relay**



- **Control of DC powered Electric Lock**



❖ Mechanical dimmensions

Bottom view