MANUAL FOR PERSEUS PLATFORM

For firmware version 1.0.1

Safety warning

The device has been tested and is in good working condition, meeting the standards required in the Czech Republic.

To keep it working properly, it's important to follow the safety and maintenance guidelines listed below. If the device is used improperly, it may not work safely.

Also, make sure that the power socket or the point where the device can be unplugged from the power supply is easy to reach.

Do not use the device if:

- It looks damaged
- It's not working properly
- There are loose parts inside
- It was exposed to moisture or water for a long time
- Someone who is not authorized tried to repair it
- The power adapter or its cable looks damaged
- You're using the device in a way that's not recommended, which may compromise its safety features
- The switch, fuse, and other power surge protection features must be part of the device's overall construction.

The manufacturer is only responsible for the device if it's being powered by an approved or supplied power source.

If you experience any difficulties with installing or starting up the device, please don't hesitate to reach out to our technical support team for assistance.

HW group s.r.o.

http://www.hw-group.com

email: support@HWg.cz

Rumunská 26/122

Prague, 120 00

Phone: +420 222 511 918

Before contacting technical support, prepare a precise model of your device (on the manufacturing label and the firmware version (see below), if you know it.

Contents

Perseus Monitoring 150 & 155

Perseus Monitoring 150 & 155 are a complex 1U solutions for a 19” rack remote monitoring system (data centers, AV installations).

Connect external physical sensors (°C, %RH, A, V, …), Detectors (relay / dry contact) or WLD (Water Leak Detection) sensing cable. 3rd party meters (sensors) can also be connected via LAN or Modbus/RTU. LTE backup connectivity allows the device to stay operational in case of an Ethernet outage for models with modem.

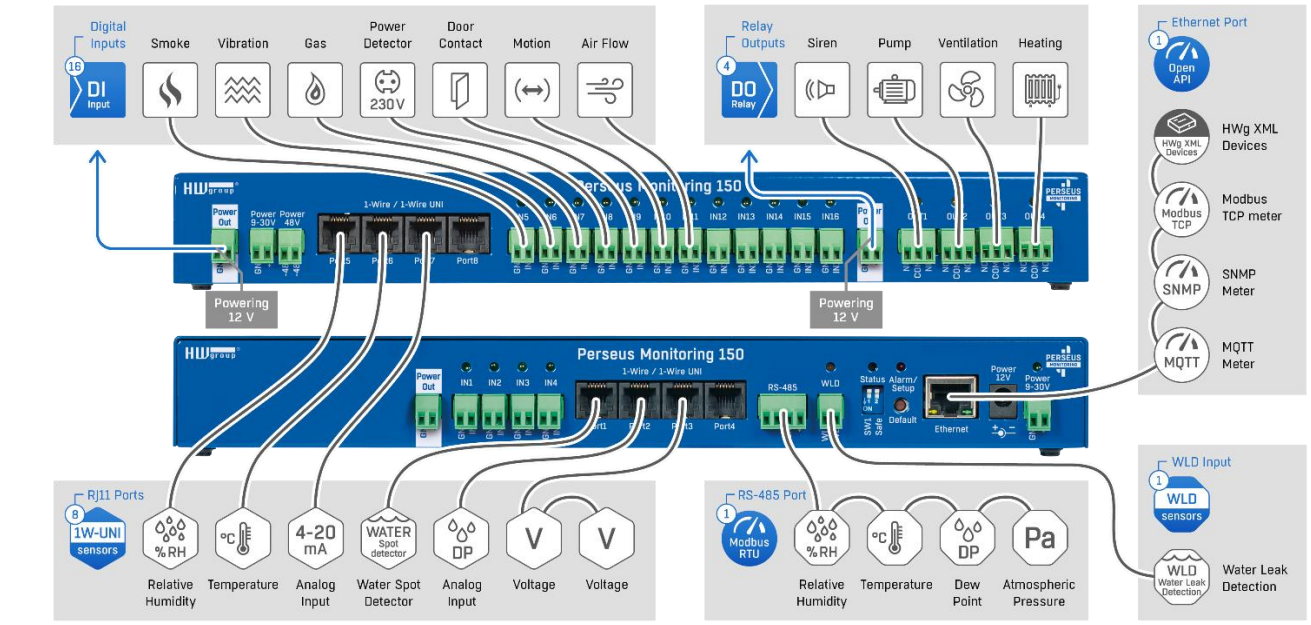
Perseus 150 combines data from external physical sensors / detectors with data from other HWg devices connected via LAN. Local conditions, Lua scripting, calendar or periodic scheduler help to analyze the measured data.

Email or SNMP Trap alert can be sent from the Perseus device. Measured values are available via Open API (XML, Modbus/TCP, MQTT).

Perseus Monitoring 150 can be connected to the Portal (SensDesk technology). Together with the Portal it’s a powerful solution for professional applications.

A unique feature of Perseus products is the Meters API. Perseus can read over LAN sensor values from other devices (e.g. STE2 R2) and use this data for analysis or alerting.

Perseus Monitoring 150: Interfaces



Perseus Monitoring 140 & 145

Perseus Monitoring 140 & 145 is a solution for remote monitoring on site with LAN and LTE (only Perseus Monitoring 145).

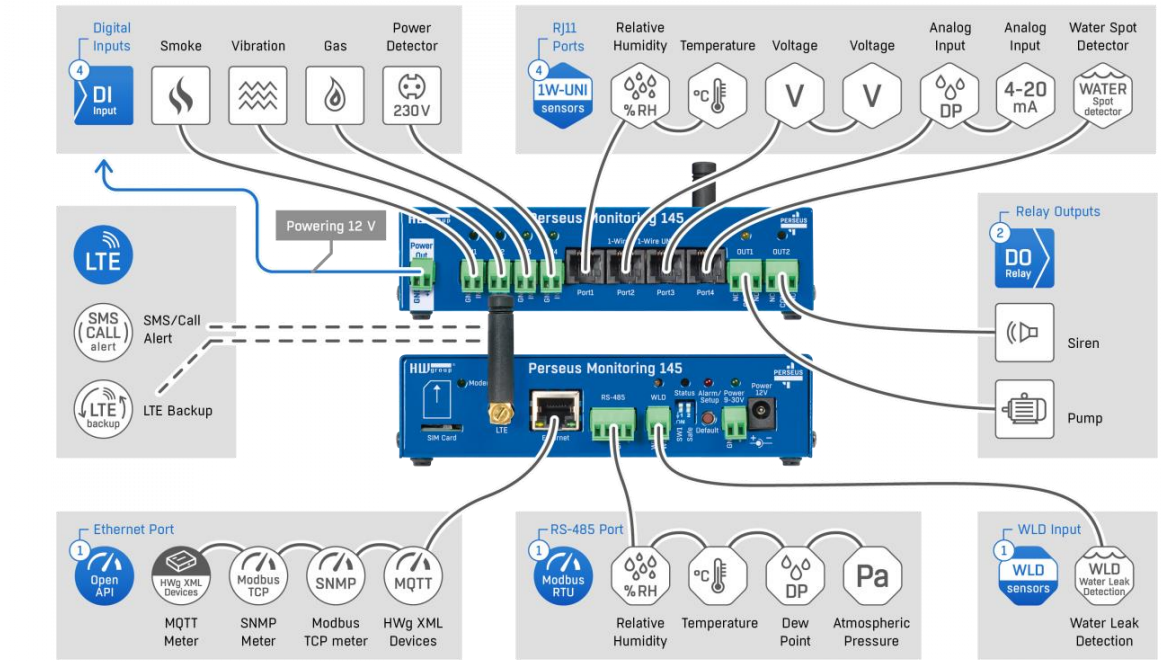
External sensors (up to 1000 variables) - RJ11 ports (1W-UNI) or RS-485 (Modbus/RTU). Multiple DI / DO (relay) and a WLD port.

Perseus Monitoring 140 & 145 is a remote environmental monitoring device with advanced alarming, local conditions and actions. It can be used to integrate all sensors at the remote site (telco room, industrial machine or warehouse).

The alert situation can be indicated locally by switching DO (relay outputs), sent from the device as standalone alerts (SMS, Call, Email) or analyzed & alerted from the Portal (SensDesk Technology) with Graphs, Virtual sensors, Device Invalid alerts, PDF reports or Portal account open API (SNMP / XML). Perseus device can also individually share all measured values via the Open API (SNMP, MQTT, Modbus/TCP, XML, ..)

Internal LTE modem in Perseus Monitoring 145 allows for backup connectivity to the Portal, as well as SMS and Call alerts. Alerting from the Portal is recommended, it's reliable for LAN even with the LTE device connectivity, the user can be alerted if the device is disconnected (Device Invalid Alert). The Perseus 145 internal LTE modem can also be configured as an SMS gateway for other HW group devices.

Perseus Monitoring 145: Interfaces



Perseus Energy 285

Perseus Energy 285 is DIN rail mount device for environmental monitoring of a site of any size and complexity. Up to 100 external meters (and sensors) with 1000 values (variables) can be connected.

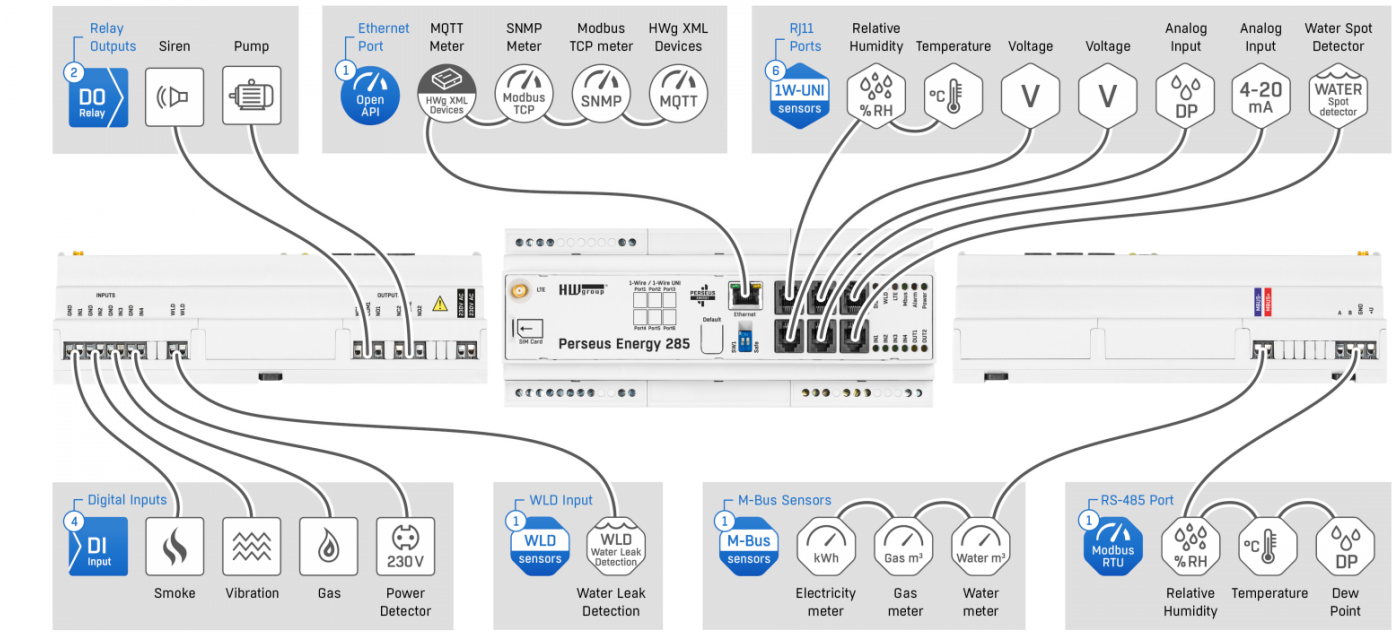
Within one DIN rail unit there are 6 1W-UNI (RJ11) ports, 4 DI (Digital Inputs) with S0 pulse counters, RS-485 (Modbus/RTU), one WLD (Water Leak Detection) zone input, an M-Bus Master input, and 2 DO (Relay Outputs).

Perseus Energy 285 can only be powered from 230V.

The combination of the RS-485 physical interface on the Perseus Energy 285, M-Bus Master and Modbus-RTU Meters API allows data to be read from third party electricity/gas/water meters, UPS or air conditioning products.

Perseus Energy 285 has a WLD zone input and 2 relay outputs that can be controlled via the device's interface or according to the value of a connected sensor or detector, for example to control air conditioning, ventilation, automatic doors, etc.

Perseus Energy 285: Interfaces



Perseus Energy 242

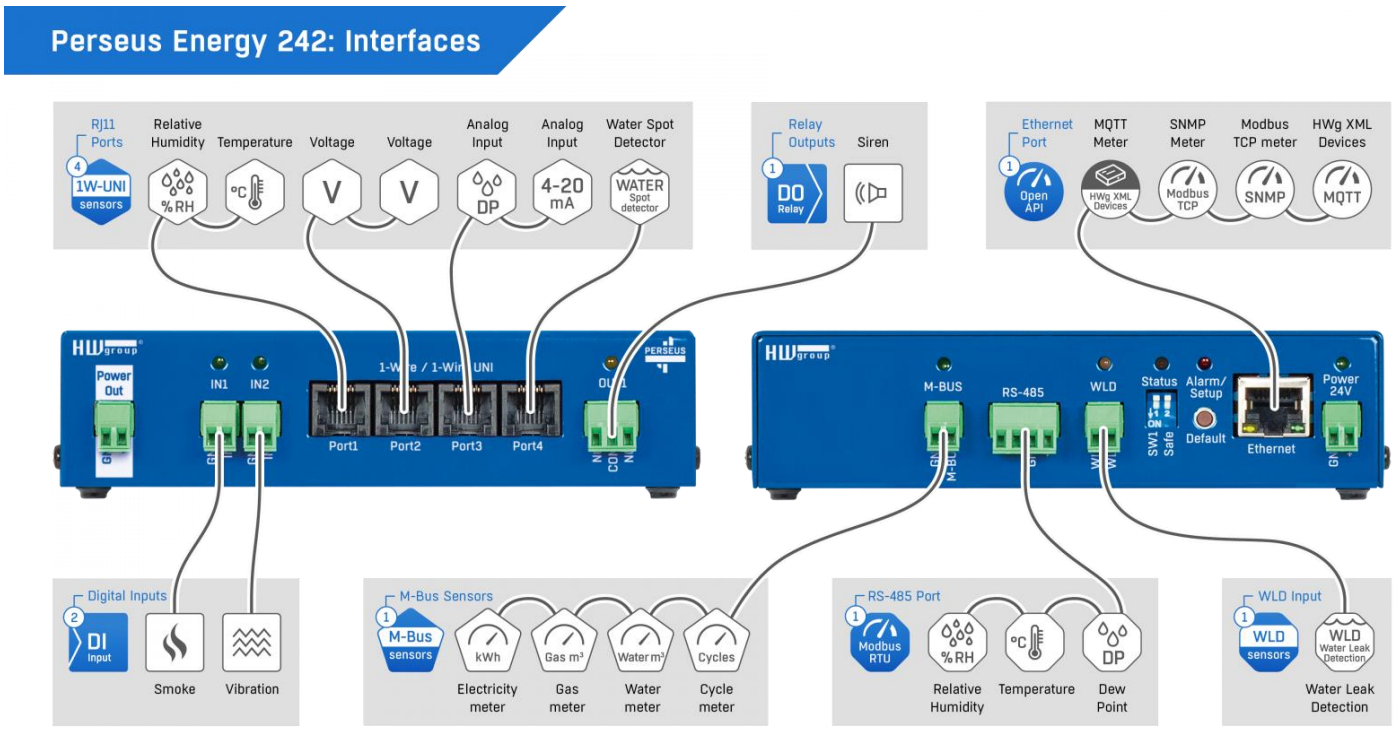
Perseus Energy 242 is a LAN device in metal case for environmental monitoring of a site of any size and complexity. Up to 100 external meters (and sensors) with 1000 values (variables) can be connected.

Within one DIN rail unit there are 4 1W-UNI (RJ11) ports, 2 DI (Digital Inputs) with S0 pulse counters, RS-485 (Modbus/RTU), one WLD (Water Leak Detection) zone input, an M-Bus Master input, and 1 DO (Relay Outputs).

Perseus Energy 242 can only be powered from 24V. Therefore, power out terminal block provides 24V.

The combination of the RS-485 physical interface on the Perseus Energy 242, M-Bus Master and Modbus-RTU Meters API allows data to be read from third party electricity/gas/water meters, UPS or air conditioning products.

Perseus Energy 242 has a WLD zone input and 1 relay output that can be controlled via the device's interface or according to the value of a connected sensor or detector, for example to control air conditioning, ventilation, automatic doors, etc.



Perseus Energy 240

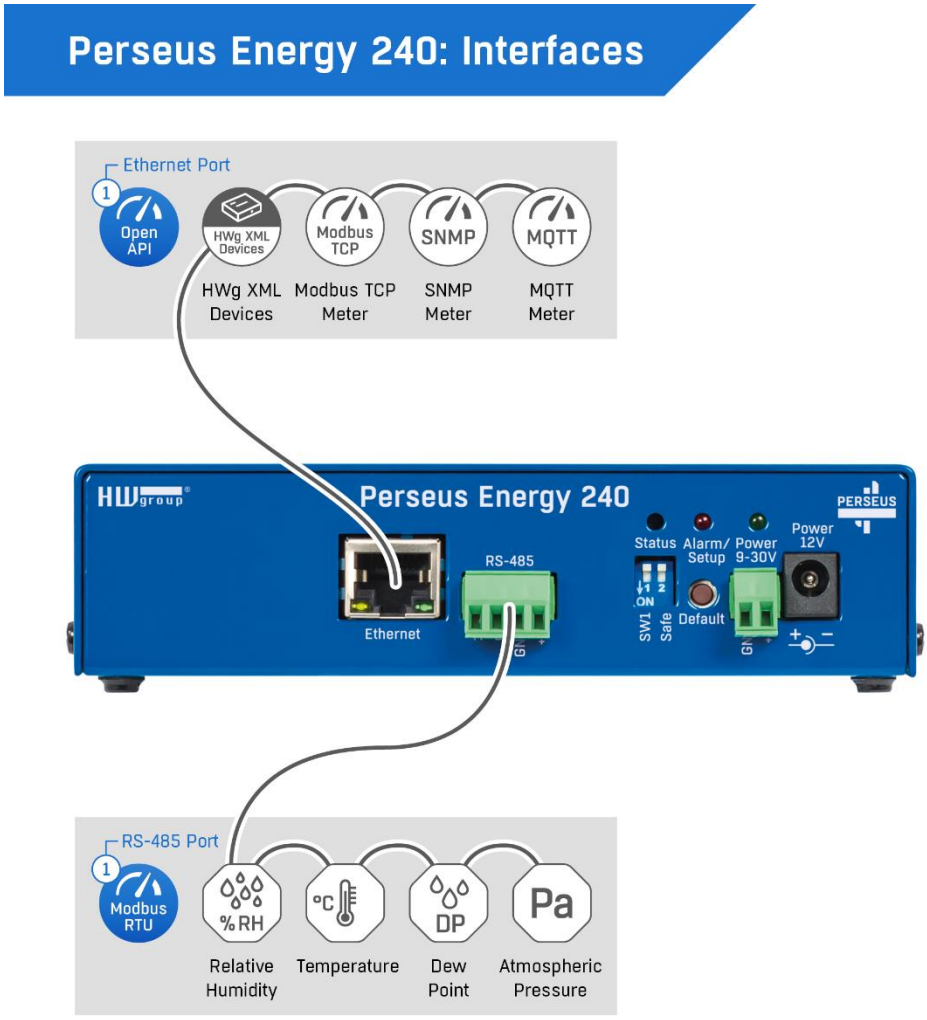
Perseus Energy 240 is the simplest and most affordable Perseus LAN monitoring device, designed to connect HW group's and third-party RS-485 Modbus/RTU sensors to the SensDesk Technology-based Portal. It can also be used as a Modbus/RTU to Modbus/TCP gateway.

Perseus Energy 240 supports up to 100 Meters with 1000 Variables, connected via RS-485 (Modbus/RTU).

It can read values from other network devices via generic SNMP, Modbus/TCP, MQTT and XML (for HWg devices).

It can analyze metered data from RTU / LAN devices with using local alarms, conditions and actions, and LUA scripts executed in the Perseus device.

Perseus Energy 240 is a gateway, it has only RS-485 interface, no RJ11 ports and no DI (Digital Inputs).



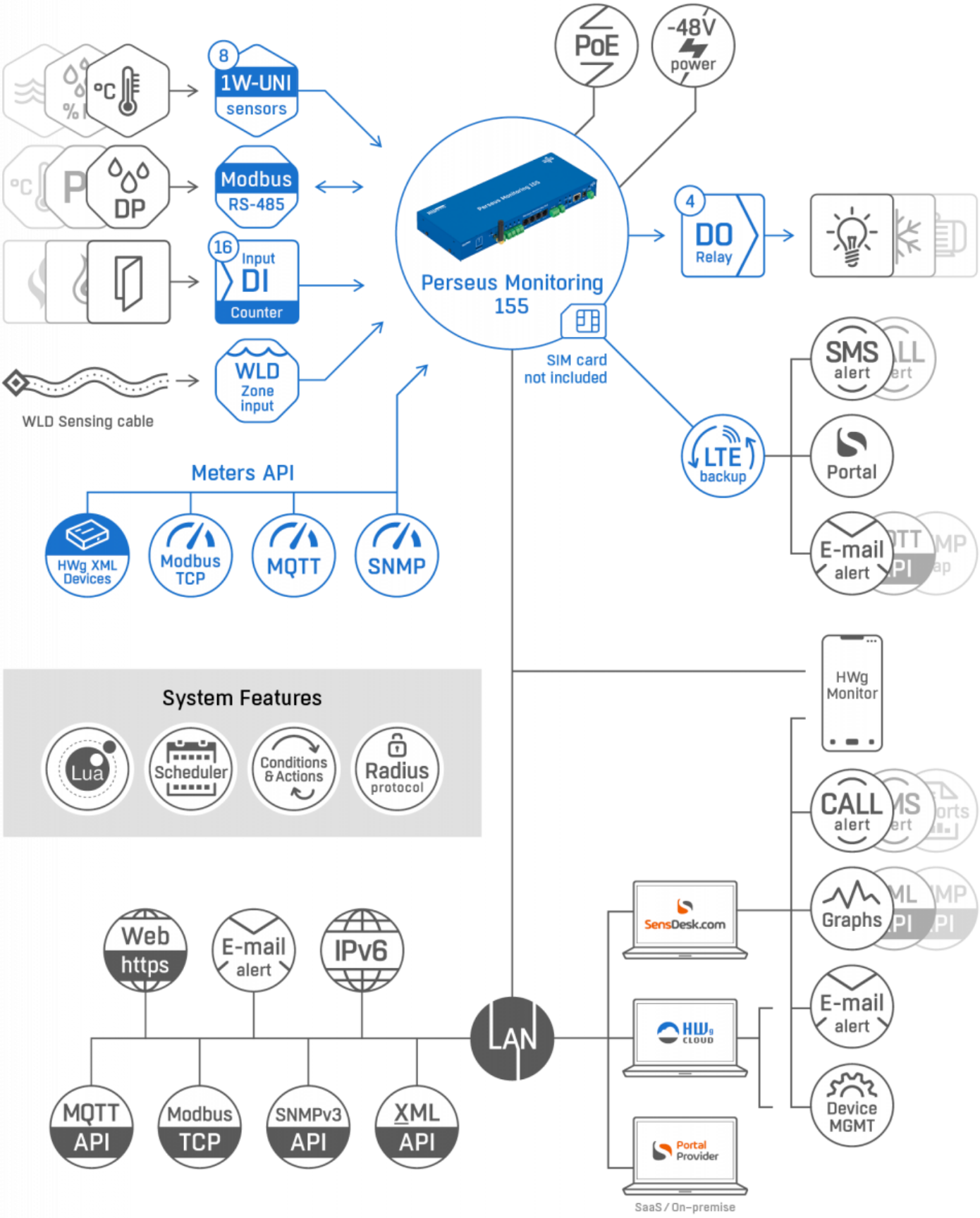
Perseus products comparison

	Perseus Energy 240	Perseus Energy 242	Perseus Energy 285	Perseus Monitoring 140	Perseus Monitoring 145	Perseus Monitoring 150	Perseus Monitoring 155
IPv6	✔	✔	✔	✔	✔	✔	✔
SNMP	SNMP v1 + SNMP v3	SNMP v1 + SNMP v3	SNMP v1 + SNMP v3	SNMP v1 + SNMP v3	SNMP v1 + SNMP v3	SNMP v1 + SNMP v3	SNMP v1 + SNMP v3
SNMP Trap	✔	✔	✔	✔	✔	✔	✔
COAP	✘	✘	✘	✘	✘	✘	✘
DHCP	✔	✔	✔	✔	✔	✔	✔
HTTP	✔	✔	✔	✔	✔	✔	✔
HTTPS	✔	✔	✔	✔	✔	✔	✔
XML	✔	✔	✔	✔	✔	✔	✔
SMTP	✔	✔	✔	✔	✔	✔	✔
SMTP TLS	✔	✔	✔	✔	✔	✔	✔
HWg-Push (SensDesk)	✔	✔	✔	✔	✔	✔	✔
MQTT	✔	✔	✔	✔	✔	✔	✔
Modbus/TCP	✔	✔	✔	✔	✔	✔	✔
Meters limit	100	100	100	100	100	100	100
Meters API: 1-Wire	✘	✔	✔	✔	✔	✔	✔
Meters API: RS-485/Hwg	✘	✘	✘	✘	✘	✘	✘
Meters API: Modbus/RTU	✔	✔	✔	✔	✔	✔	✔

Meters API: Modbus/TCP	✔	✔	✔	✔	✔	✔	✔
Meters API: HWg XML	✔	✔	✔	✔	✔	✔	✔
Meters API: SNMP	✔	✔	✔	✔	✔	✔	✔
Meters API: MQTT	✔	✔	✔	✔	✔	✔	✔
Meters API: M-Bus	✖	✔	✔	✖	✖	✖	✖
Variables limit	1000	1000	1000	1000	1000	1000	1000
1-Wire values *	0	100	100	100	100	100	100
RS-485 values *	100	100	100	100	100	100	100
Number of 1-Wire ports	0	4	6	4	4	8	8
Number of RS-485 ports	1	1	1	1	1	1	1
Inputs	0	2	4	DI: 4 WLD: 1	DI: 4 WLD: 1	DI: 16 WLD: 1	DI: 16 WLD: 1
Outputs	0	Relays: 2	Relays: 2	Relays: 2	Relays: 2	Relays: 4	Relays: 4
Power supply	9-30V / 500mA	24V	230V / 130mA	9-30V / 500mA	9-30V / 500mA	9-30V / 500mA 48V / 250mA PoE / 250mA	9-30V / 500mA 48V / 250mA PoE / 250mA
PoE/48V	✖	✖	✖	✖	✖	✔	✔
Operating	-30°C ÷ 85°C	-30°C ÷ 85°C	-30°C ÷ 65°C	-30°C ÷ 85°C	-30°C ÷ 65°C	-30°C ÷ 85°C	-30°C ÷ 65°C

environment	/ 5%RH ÷ 95%RH	/ 5%RH ÷ 95%RH	/ 5%RH ÷ 95%RH	/ 5%RH ÷ 95%RH	/ 5%RH ÷ 95%RH	/ 5%RH ÷ 95%RH	/ 5%RH ÷ 95%RH
Storage environment	-30°C ÷ 65°C / 5%RH ÷ 95%RH	-30°C ÷ 85°C / 5%RH ÷ 95%RH	-30°C ÷ 85°C / 5%RH ÷ 95%RH	-30°C ÷ 65°C / 5%RH ÷ 95%RH	-30°C ÷ 65°C / 5%RH ÷ 95%RH	-30°C ÷ 65°C / 5%RH ÷ 95%RH	-30°C ÷ 65°C / 5%RH ÷ 95%RH
Ethernet	10/100 Mbit	10/100 Mbit	10/100 Mbit	10/100 Mbit	10/100 Mbit	10/100 Mbit	10/100 Mbit
WiFi	NO	NO	NO	NO	NO	NO	NO
Mobile networks	NO	NO	LTE	NO	LTE	NO	LTE
RS-232	NO	NO	NO	NO	NO	NO	NO
RS-485	YES	YES	YES	YES	YES	YES	YES

Perseus 155 usage scheme (example)



Basic features

- **Ethernet**: RJ45 (100BASE-T); WEB: Built-in web server
- **LTE**: Global (for models with modem)
- **Sensors**: RJ11 port - max. 100 external sensors (1-Wire / 1-Wire UNI)
- **RJ11 sensors**: Temperature, Relative Humidity (Indoor / Outdoor), Voltage, Current (0-30A), Current Loop (4-20mA) and much more
- **RS-485 (Modbus/RTU)**: Terminal block
- **WLD**: Zone input for external Water Leak Detection sensing cable
- **DO (Digital Outputs)**: DO (relays)
- **DI (Digital Inputs)**: DI (dry contact)
- **Detectors**: Smoke, door contact, fan failure detector, power presence, etc
- **Device Alerts**: SNMP trap, email, (SMS / Calls with internal LTE modem or external SMS GW)
- **Portal Alerts**: Email, SMS / Calls, Device Invalid Alert
- **Open API**: SNMP, XML, Modbus/TCP, MQTT, BACnet
- **Professional product**: IPv6, SNMP Traps, SNMP v3
- **Security**: Radius - IEEE 802.1X, Admin and user accounts, HTTPs (customer certificate)

Application and usage

- Remote site data concentrators from all local sensors / meters
- Remote environmental monitoring inside the cabinet or telco site
- Data Center / Single rack monitoring
- Remote technical room monitoring
- Telecommunication installations
- Remote infrastructure monitoring
- Remote machine monitoring
- Remote environmental monitoring of warehouse / storage spaces
- Pharmaceutical / warehouse environment monitoring
- Remote base transceiver station monitoring
- Greenhouse monitoring

Technical parameters (shared)

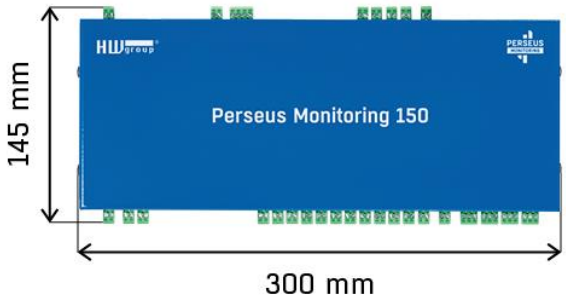
Ethernet	
Interface	RJ45 (10/100BASE-T)
Supported protocols	IP: ARP, TCP/IP (HTTP, HTTPS, SNTP, SMTP, HWg-Push, netGSM, TLS, Modbus/RTU, MQTT, 802.1x), UDP/IP (SNMP, Syslog), BACnet
SNMP	Version 1 + Version 3
RJ11 1-Wire sensors	
Port/connector	RJ11 (Port1 – Port8*) *can vary by model type
Meter limit	Max. 2 Meters (physical sensors) per port (*for details see sensor's manual)
Sensor type	Only HW group sensors can be connected to 1-Wire RJ11 ports
Sensors/distance	Max. 60 meters total length (per each RJ11 port) (*for details see sensor's manual)
Powering sensors	Each RJ11 can power 2 or more physical sensors. Details in manual for each sensor.
RS-485 sensors	
Port/connector	Terminal block
Meter limit	Up to 100 Meters (*up to 5 directly powered)
Sensor type	Modbus/RTU (Original from HW group or generic Modbus/RTU)
Sensors/distance	1× Max. 1000 meters total length (per each RS-485 port)
M-bus sensors	
Port/connector	Terminal block
Meter limit	Up to 100 Meters
Sensor type	M-Bus (EN 13757-2, EN 13757-3)
Sensors/distance	Max. 1200 meters total length
DI (Digital Inputs)	
Type	Dry contact input (mechanical switch or relay outputs)
Port/connector	IN1 – IN16* / ø2 mm terminal block *can vary by model type
Type	Digital Input (supports S0 pulse counter)
Sensitivity	Sensitivity 1 (On) = 0-500 Ohm, 10-12V/ ~ 3mA
Max. distance	Up to 50m
DO (Digital Outputs)	
Type	Relays standard
Load voltage	50V / 1000mA
Load voltage (Perseus Energy 285 only)	250V AC / 10A
WLD zone (Water Leak Detection)	
Port/connector	WLD/ Terminal block
Type	WLD water sensing (detection) cable (Type A)
Sensor states	0 = OK, 1 = Flooding, 2 = Cable disconnected
Sensing cable length	Total length max. 185 m (non-sensing extension cables included)
Datalog	
Memory for datalog	32 Mbyte – approx. 800 000 records
Power supply	
Power voltage	9-30V / 500 mA, 48V (PoE version only)

Power voltage (Perseus 285 only)	230V ±10% / 40VA
Connector	Connector Jack Ø5.5 x 2.1 / 10 [mm], Terminal Block
PoE (Power over Ethernet)	PoE (Power over Ethernet) RJ45 - IEEE 802.3af Class 2 (where applicable)
LED	
Link	Yellow - Ethernet connection state
Activity	Green - Ethernet activity
Alarm / Setup	Red – Solid – A meter, Variable or Condition is in the Alarm state Red – Lit after startup – The Default button is pressed and the configuration is waiting for the configuration to be restored Blinking – Device is upgrading
Status	Lit while waiting for a factory reset
DI (Digital Input)	Green = Input ON (1)
WLD zone	Green = Detection Zone OK, Red = Detection Zone Flooded, Yellow = Detection Cable Disconnected/Damaged
DO output	Yellow = Output ON (1)
Modem	Green = Solid LTE connection Green flashing = during connection and network search
Button	
Reset	Bootloader recall (safe mode): Press the button, turn on the power within 2 seconds, press the button 3 times Factory reset: Press the button, connect the power and wait 10 seconds for the Status LED to turn off
LTE modem (modem models only)	
Interface	FDD LTE bands: B1/ 2/ 3/ 4/ 5/ 7/ 8/ 12/ 13/ 17/ 18/ 19/ 20/ 25/ 26/ 28/ 66 TDD LTE bands: B34/ 38/ 39/ 40/ 41 GSM bands: B2/ 3/ 5/ 8
Antenna connector	SMA small
Other parameters	
Elmag. radiation	CE / FCC Part 15, Class B
Elmag. compatibility	EN 61326-1:2013, EN 61010-1:2010, EN 55011:2009, EN 62311:2008
Op. Environment	Designed for indoor use only
Altitude	Designed for use up to 4000m
Overvoltage category	Overvoltage category II.
Pollution Degree	Pollution Degree Rating II.
IK Rating	IK08
IP Rating	IP40

The equipment requires professional installation. Devices with outlets for voltages higher than 48V (typically Perseus Energy 285) may only be installed by a person properly trained and licensed for this activity according to the requirements of the region where

the device is installed and used. The devices do not require any additional protection (over-current, over-voltage, insulation, etc.) and their use is therefore only determined by local legislation.

Perseus Monitoring 150

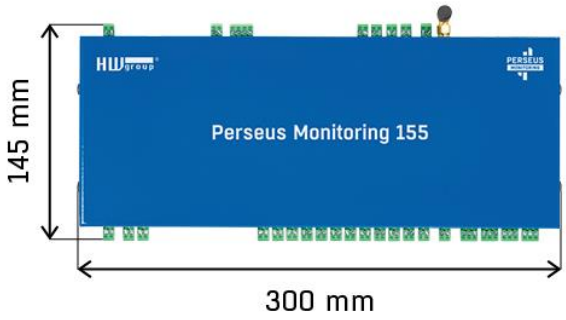


Operating environment:
-30 °C to 85 °C (-22 °F to 185 °F), 5 % RH to 95 % RH

Storage environment:
-30 °C to 65 °C (-22 °F to 149 °F), 5 % RH to 95 % RH

Weight:
1120 g

Perseus Monitoring 155

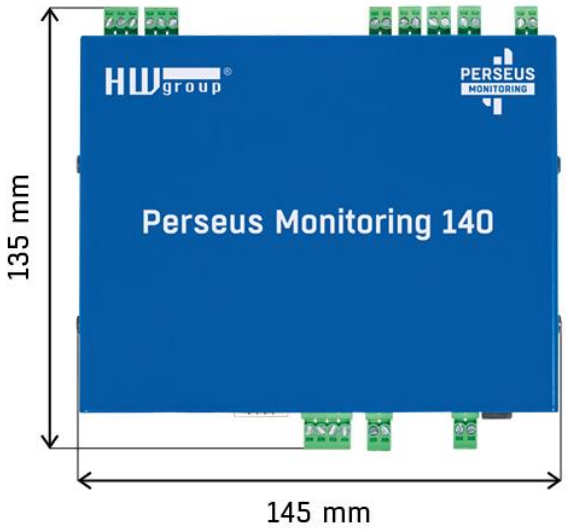


Operating environment:
-30 °C to 65 °C (-22 °F to 149 °F), 5 % RH to 95 % RH

Storage environment:
-30 °C to 65 °C (-22 °F to 149 °F), 5 % RH to 95 % RH

Weight:
1120 g

Perseus Monitoring 140

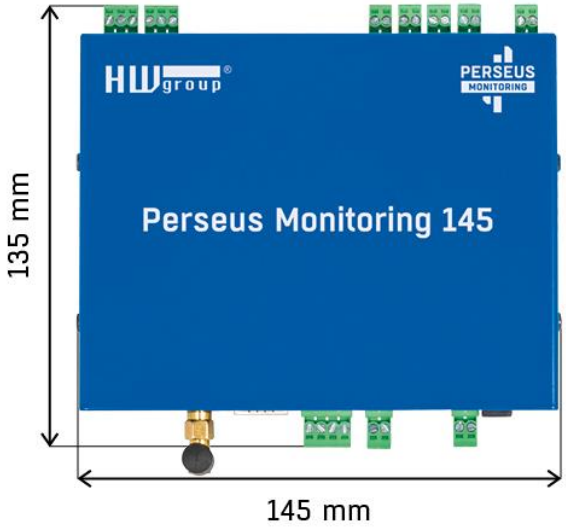


Operating environment:
-30 °C to 85 °C (-22 °F to 185 °F), 5 % RH to 95 % RH

Storage environment:
-30 °C to 65 °C (-22 °F to 149 °F), 5 % RH to 95 % RH

Weight:
550 g

Perseus Monitoring 145

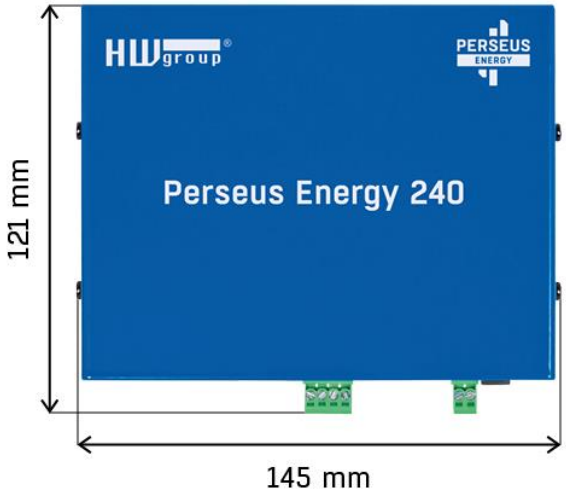


Operating environment:
-30 °C to 65 °C (-22 °F to 149 °F), 5 % RH to 95 % RH

Storage environment:
-30 °C to 65 °C (-22 °F to 149 °F), 5 % RH to 95 % RH

Weight:
550 g

Perseus Energy 240

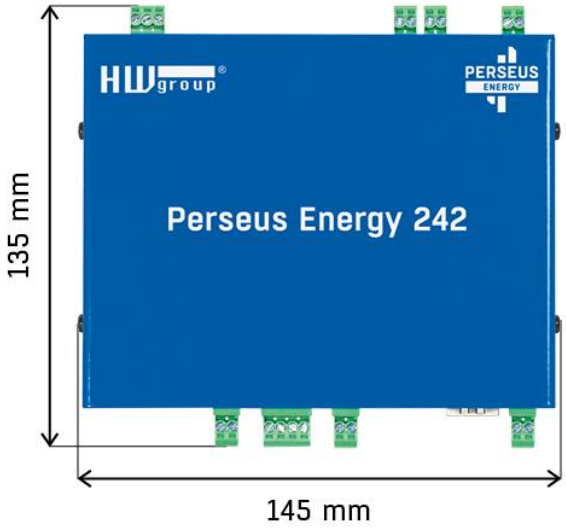


Operating environment:
-30 °C to 85 °C (-22 °F to 185 °F), 5 % RH to 95 % RH

Storage environment:
-30 °C to 65 °C (-22 °F to 149 °F), 5 % RH to 95 % RH

Weight:
488 g

Perseus Energy 242

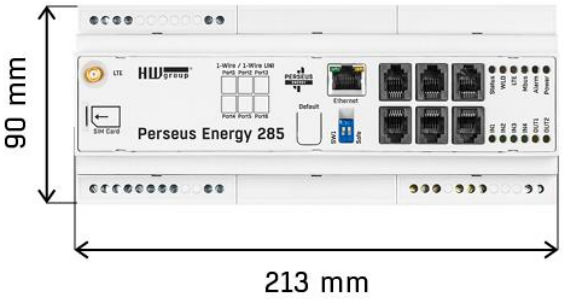


Operating environment:
-30 °C to 85 °C (-22 °F to 185 °F), 5 % RH to 95 % RH

Storage environment:
-30 °C to 65 °C (-22 °F to 149 °F), 5 % RH to 95 % RH

Weight:
505 g

Perseus Energy 285



Operating environment:
-30 °C to 65 °C (-22 °F to 149 °F), 5 % RH to 95 % RH

Storage environment:
-30 °C to 65 °C (-22 °F to 149 °F), 5 % RH to 95 % RH

Weight:
529 g

First start-up

1) Cable connection

Connect the device to the **Ethernet** (direct cable to the switch, crossed to PC).

Connect the power adapter to the power grid and to the device.

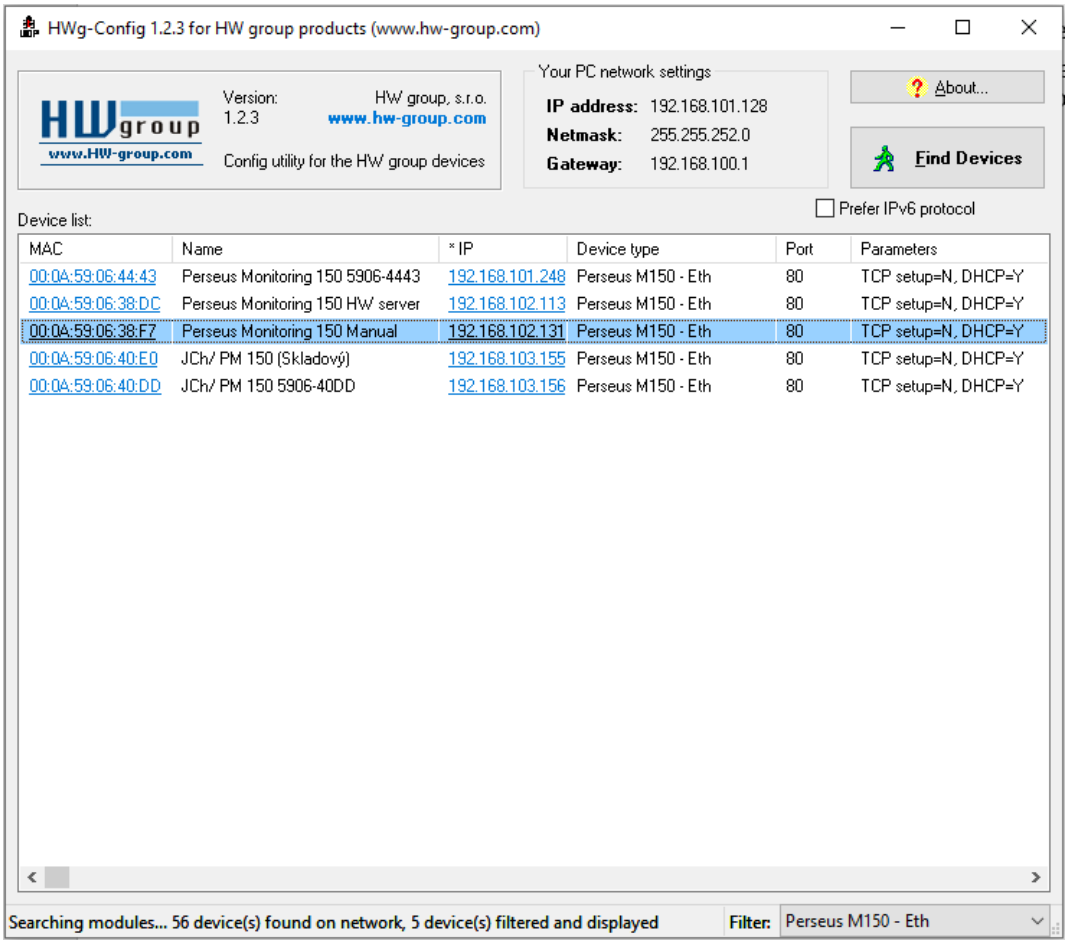
If the Ethernet connection is OK, the **LINK (yellow)** light should come up a moment later. The **ACTIVITY light (green)** indicates Ethernet activity.

2) The setting of the IP address - HWg-Config

The **HWg-Config** program for MS Windows can be downloaded at https://www.hw-group.com -> Products -> Software -> HWg-Config.

Launch **HWg-Config,** program automatically searches for LAN connected devices.

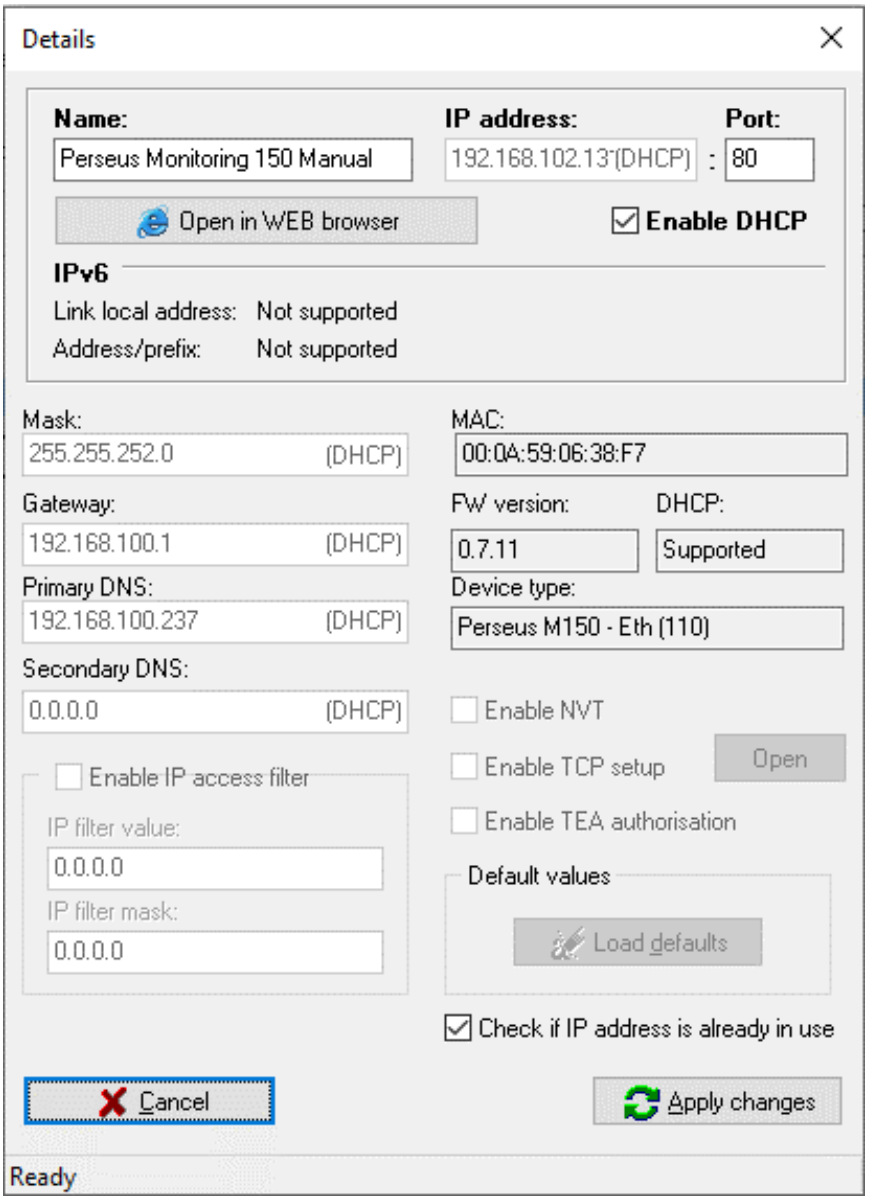
If the device is connected later, click the **Find Devices** button. Local network HWg devices will be listed. Click on the MAC address of the device to open the dialogue window for device settings.



Set device network parameters:

- IP address / HTTP port (80 by standard)
- Your network mask
- IP address of your network gateway
- You can also use DHCP (if available in your network)
- Device name (optional parameter)

Save the settings by clicking on **Apply Changes.**

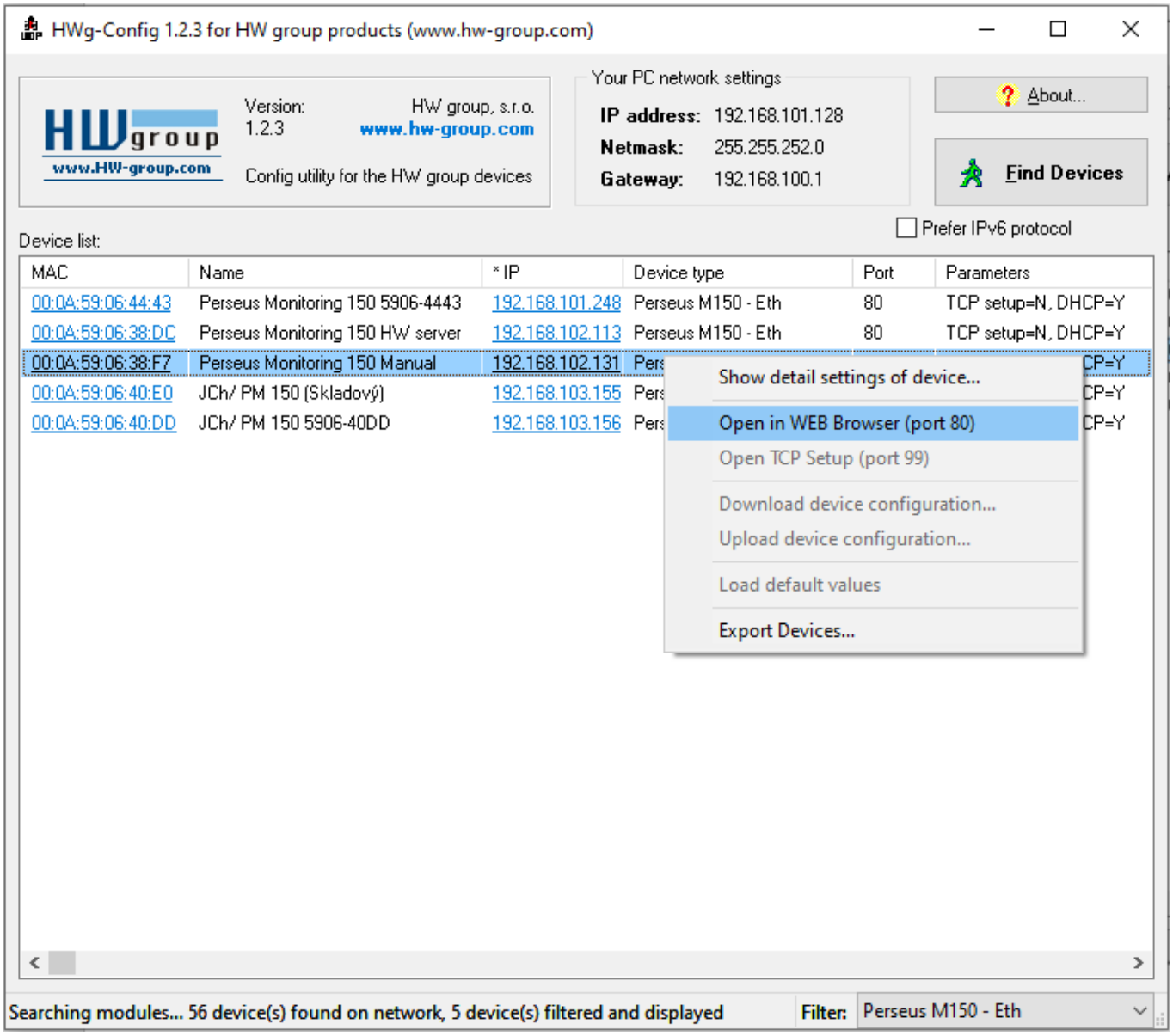


Note: The device provides 2 options how to restore its default settings:

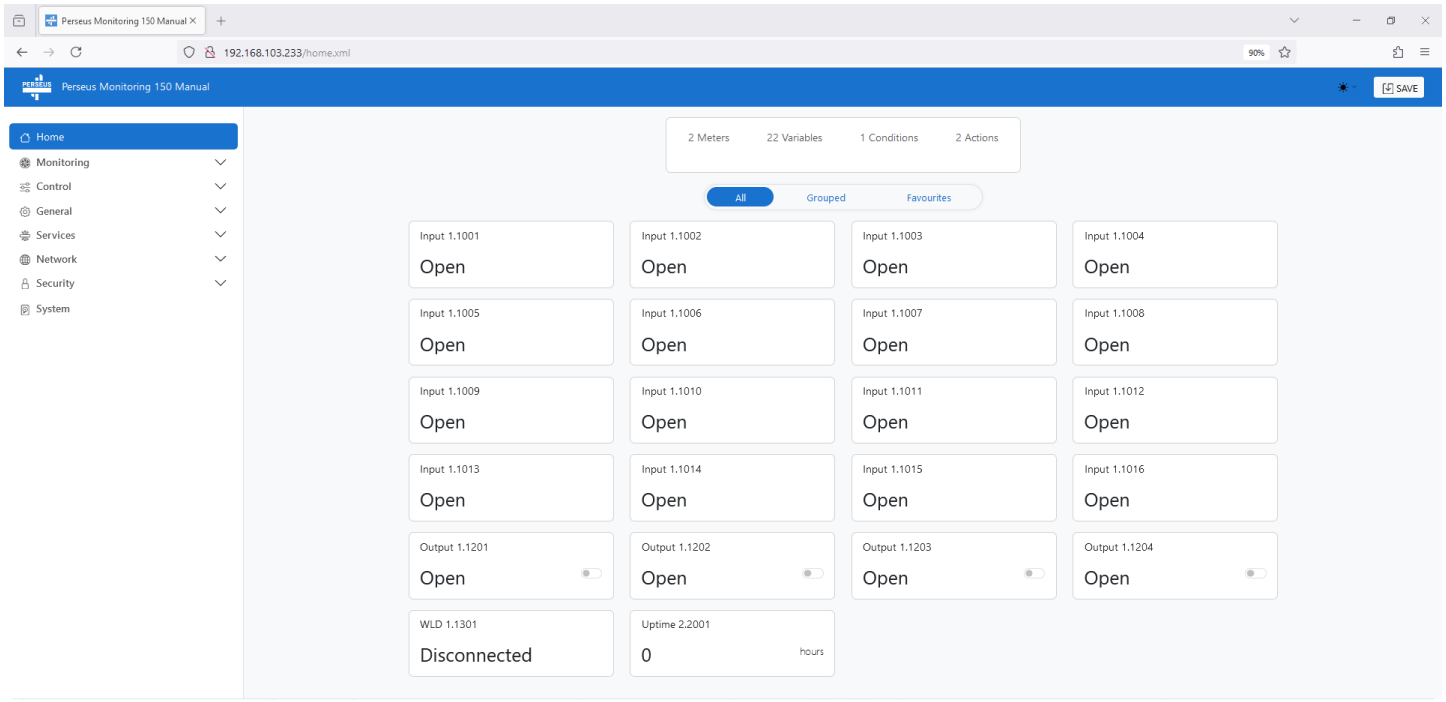
1) Right-click on the device's MAC address. Click on the **Load default values** item. Note: Device default values can be restored from the HWg-Config program only during the first 60 seconds after the device is powered up.
2) Switch off the device. Press the **RESET** button on the device, **hold it down and connect the device** power source (power adaptor). Hold the button down for another 5 seconds until all the LEDs light up.

3) How to open the device website

1) Enter the device IP address in your web browser if you know it.
2) Use right-click on device in the HWg-Config program. Select **Open in WEB Browser**.
3) Click on the underlined IP address in the HWg-Config program.



4) Perseus device GUI will open on the Home page.



Introduction into Perseus monitoring units

The Perseus family of products is a modern monitoring control panel designed to read, record, analyze and send values from connected sensors and detectors to a remote portal, as well as to perform simple and more complex actions based on the received data. Of course, it is also possible to send alarm messages, control external devices and inform about sensor, device and operator failures. There is a new possibility of customer scripts depending on the product line, from simple conditions and calculations to a complete application defined by a script (PLC like).

The Perseus family contains the same basic features, but with regard to the expected use in the target application, its products are divided into 4 product lines:

Perseus Monitoring - a base line designed for environmental monitoring in IT, pharmaceutical, food and other industries, including support for independent audits for quality management systems (ISO) and specific professional certifications.

Perseus Energy - a range of products primarily designed to measure energy consumption, energy savings and design methods to increase the efficiency of heating and cooling systems, including the preparation of independent reports for subsidy schemes.

Perseus Industrial - a series of customer products with a high degree of durability, designed for project deployment not only in the engineering environment, focused mainly on the support of industrial standards. The series is characterized by the possibility of defining complete customer applications in a program in LUA (PLC like).

Perseus Concept - a series of products for sampling in projects, tenders, public procurement, etc. These units are functionally and communicatively (API) identical to the entire Perseus family, but without a guarantee of continuity of production of specific models.

Features common to all products in the Perseus family

- **Modular system** - all units share a common interface (API) and differ only in HW design, number of connectors and types of buses or presence of the LTE modem.
- **Wide range of connectable sensors and devices** (varies by model):
  - **1-Wire** - sensors designed for simple applications for easy installation and configuration for specific bus extensions;
  - **Dry contact** - potential-free two-state input for connecting external relays or open collectors;
  - **RS-485 (Modbus/RTU)** – sensors designed for harsh industrial environments, long distances and the ability to connect third-party sensors. The Perseus family can automatically search for HW group sensors of the Modbus/RTU standard on the bus;
  - **M-BUS** – energy meters such as electricity meters, calorimeters, water meters and gas meters and other sensors equipped with this interface, designed for remote energy readings in residential buildings and industrial plants;
  - **SNMP** - devices equipped with SNMP protocol, from which it is necessary to read and process the obtained values. Within the Perseus exchanges, the data obtained appear to be connected to a physical bus;
  - **Modbus/TCP** - devices using the Modbus/TCP protocol, from which it is necessary to read and process the obtained values. Within the Perseus exchanges, the data obtained appear to be connected to a physical bus.
  - **MQTT (Subscriber)** - devices equipped with a MQTT broker, from which it is necessary to read and process the received values. Within the Perseus exchanges, the received data appear to be connected to a physical bus.
  - **HW group XML** - older devices produced by HW group, such as HWg-STE, STE2, Poseidon2, Damocles2 and others, from which it is necessary to read and process the received values. Within the Perseus exchanges, the received data seem to be connected to a physical bus.
- **Wide range of options for concentration, serialization and publishing of acquired data (API):**
  - **XML** - the obtained values are provided via an XML file that can be downloaded using the standard http protocol;
  - **SNMP** - the obtained values are provided via the UDP SNMP protocol. We support SNMPv1 and SNMPv3;
  - **MQTT** - the obtained values are provided via MQTT TCP protocol and are periodically sent to the MQTT broker (MQTT publisher);
  - **Modbus/TCP** - the obtained values are provided via TCP protocol Modbus/TCP.
  - **BACnet** - the obtained values are provided via TCP protocol to BACnet client.
- **High level of security** - Perseus control panels can be secured against misuse not only with a local user list, but also centrally via a RADIUS server. Communication via HTTPS with the option to upload your own SSL certificates is a matter of course.
- **Multiple safe ranges support** - Multiple safe ranges can be defined for each monitored variable, each defined by the range of MIN, MAX, Hysteresis and Delay for alarm escalation and reminder capabilities.
- **Almost unlimited target directory for alarms** - Alarm message recipients, such as email or SMS, are defined using the address book.

- **User-defined alarm message templates** - All user messages can be defined using a set of macros to customize them to the user's needs.
- **Sophisticated system of calendars (planners)** - it is now possible to schedule different events or their recipients during working hours and outside them, to take into account working days and weekends or holidays, it is possible to launch specific events on specific days, at the exact time or, for example, with a clearly defined recurrence period (e.g. the 1st Wednesday of the month at 12:00, if it is not the 24th).
- **Sophisticated system of triggering actions** - user-defined actions can now be triggered by conditions or based on a schedule (scheduler), it is possible to suppress their execution at the start of the device, or to define whether they should be executed only at the beginning of the triggered condition, at its end, in both cases.
- **Support for disabling triggered actions** - Each meter, value, condition and action can be individually enabled or disabled. This feature can be used, for example, to suppress the execution of actions in case of a planned shutdown of the monitored technology, long-term failure, etc. Supported types of actions:
  - E-mail - sends a message via e-mail;
  - SMS - sends a message by SMS;
  - SNMP Trap - sends a message via SNMP;
  - Set Output - sets the output to the desired value;
  - Script - Executes the defined LUA script (for future use, probably only Perseus Industrial models).
- **LUA scripting support** - All products in the Perseus family allow you to recalculate values and perform simple logical operations using LUA language scripts defined for each value. In addition, the Industrial series offers the possibility of defining the entire application through scripts.
- **Support for variable recalculations** - Each variable can now be recalculated using simple mathematics (value shift, multiplication or division) before processing, or processed using a LUA script containing a value recalculation or a simple condition.
- **Support for virtual values and meters** - It is possible to create virtual values and apply recalculations of quantities (from mA to kPa, etc.), conditions, cycles and other operations definable by LUA scripts.
- **Support for connecting sensors from external HW group devices** - The Perseus family is able to connect older devices from the HW group family of monitoring control panels and work with its sensors as if they were from the Perseus family.
- **Support for connecting sensors from external SNMP devices** - Perseus family is able to connect any SNMP devices and work with the received values as if they were Perseus sensors.
  - The Perseus family is equipped with the ability to scroll through the available values (SNMP walk).
- **Support for different reading periods for different meters** - it is possible to set an individual reading period for each meter, thus increasing the number of sensors on the bus or saving meter batteries.
- **Ability to use an LTE modem as a backup connection** - devices equipped with LTE modems can, in addition to sending SMS and ringing, also use the LTE connection as a backup in case of Ethernet connection failure, and in such a case the modem is also able to send data to remote servers (HWg Push for SensDesk

technology, MQTT publish/subscribe for MQTT brokers, SNMP traps) or e-mails. The modem has no router function.

- **Brand new output APIs (XML, HWg-PUSH, SNMP, Modbus/TCP, BACnet)** - All output APIs have a completely new form that reflects all technological innovations, even at the cost of losing backward compatibility with previous HWg product families.

Basic differences between Poseidon2, Damocles2, HWg-PWR, Ares vs. Perseus platform

Poseidon2, Damocles2, HWg-PWR, Ares

+ Simplicity of adding sensors

+ Ease of setting simple alarms

+ Clear GUI

+ Simple licensing per monitored variable

+ Clear division of sensors according to function into sensors, inputs (DI) and outputs (DO)

- Inability to group quantities of the same sensors

- Inability to create logical conditions

- Inability to connect input expanders and classify them as DI

- Absence of an event planner

- Does not support 3rd party sensors

Perseus

+ Wide range of settings and functions

+ Ability to set different behavior at different times

+ Modern graphical interface with day and night mode

+ Ability to create user templates

+ Support for virtual variables

+ LTE modem as backup data line

+ All quantities are variables, only some are of sensor type, others are input, output or even analog output. The variable has its parameters such as: value, unit, state and others

+ Support for 3rd party Modbus/RTU sensors

- More complex alarm creation

- New licensing on Meter/Variable

- Lack of backward compatibility with Poseidon2, Damocles2, HWg-PWR, Ares

1st Generation and 2nd Generation Sensors

With regard to the capabilities of the Perseus family, HW group has developed a completely new range of sensors with new protocols.

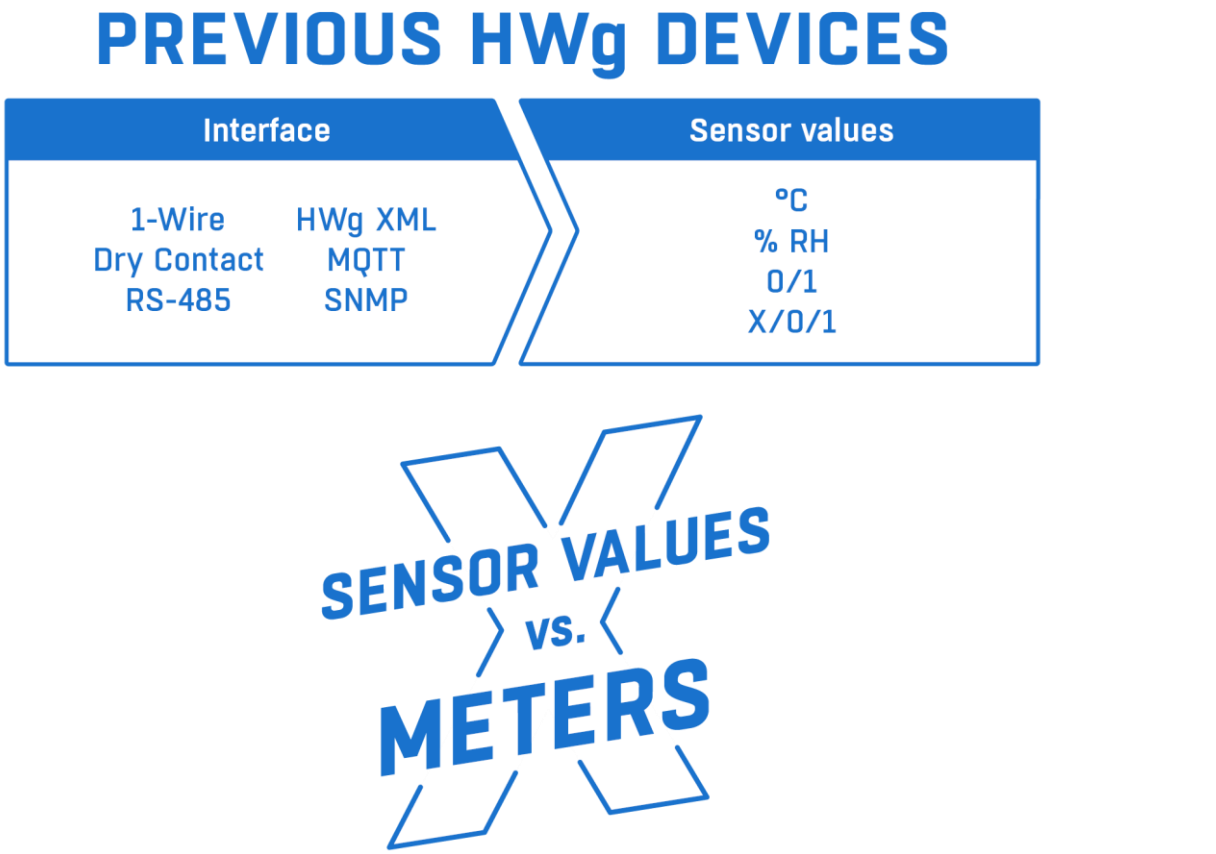
Protocols	*1st generation*	*2.generation*	*Backward compatibility with the 1st generation*
1-Wire	1-Wire/1-Wire UNI	1-Wire UNI v3	**YES**
RS-485	HW group ASCII	Modbus/RTU	**NO**

In both cases, the goal is to provide multiple monitored values under one physical address so that these values can be logically grouped and worked with as a whole. If you are using 3 Htemp. 1st generation in Poseidon or Ares units, you will see 3 temperature values and 3 humidity values without specifying which two values belong to which physical sensor.

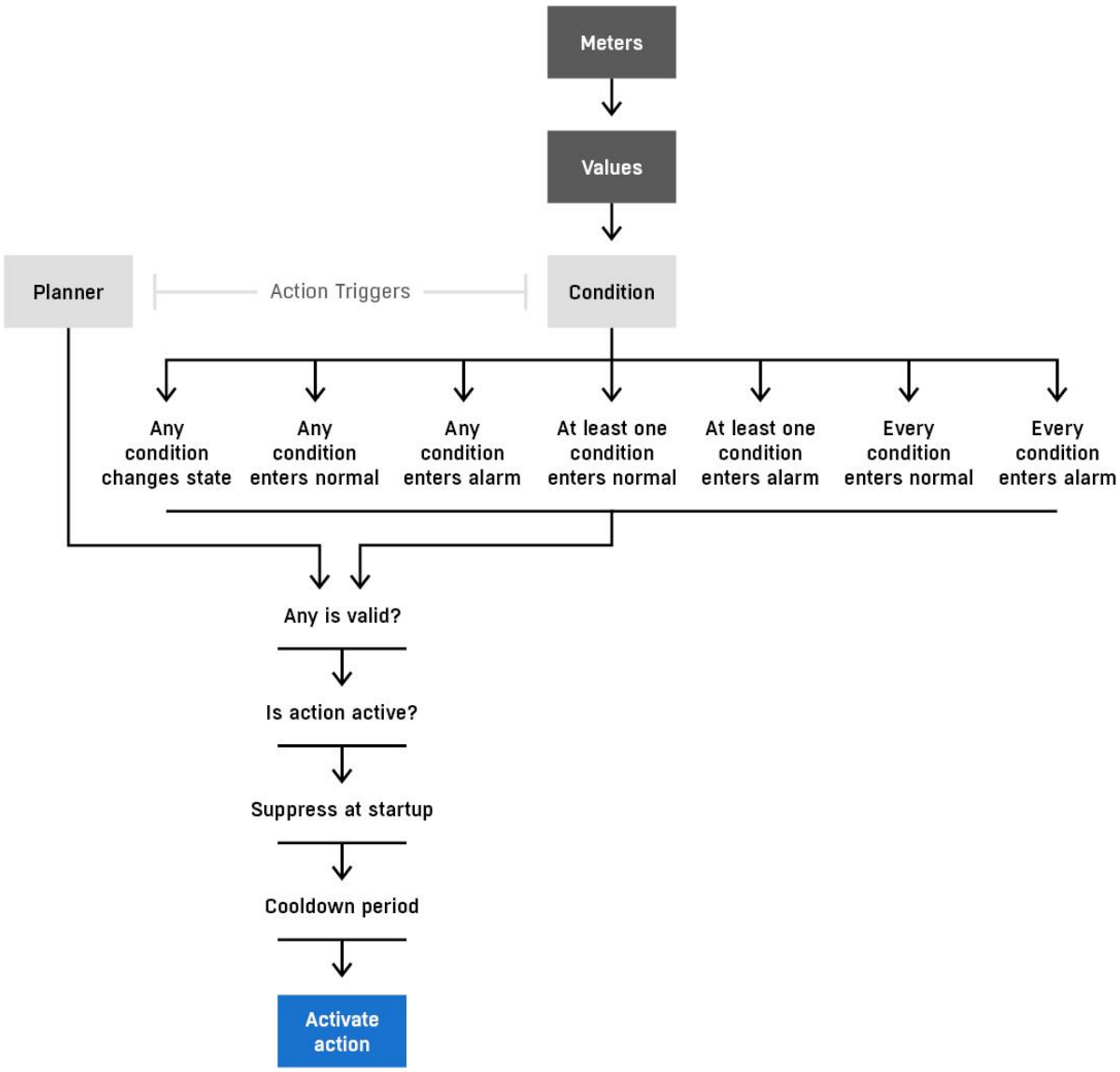
If you use 3 2nd generation sensors, they will be interpreted as 3 meters in the Perseus unit, each with 2 variables.

At the same time, the HW group is trying to maintain an identical set of sensors with both 1-Wire and RS-485 interfaces (and in the future, other interfaces such as wireless radio interfaces). All HW group sensors (1-Wire and RS-485 Modbus/RTU) are capable of automatically detecting Perseus units on the network, despite the fact that the RS-485 standard does not support anything like search broadcast polling.

Meter vs variable vs value



With the introduction of a new merging value structure, the nomenclature has been changed to better reflect the new structure.



- **Interface** - represents the HW or SW interface through which the data is read.

- **Meter** - represents the physical Meter (sensor) itself. We use the term meter, taken from HWg-PWR, to distinguish it from sensor, which could be interpreted differently in older devices. In addition, the meter can be a Poseidon unit, UPS or PLC, so the term sensor is inappropriate.

- **Variable** - represents the actual measured quantity and is equivalent to value in older devices. However, the term Variable is more accurate because of the options, as "variable has value" while "value has value" is misleading.

- **Parameters** - the actual definition of the variable, including the value, unit, exponent, and other properties.

Licensing change

While the Poseidon and Ares products perceived any connected variable as a "Sensor" and each device had a limited number of sensor values, the Perseus family perceives each connected 2nd generation sensor as a "Meter".

	Htemp - number of "licenses"		**THPVoc 1W-UNI - number of "licenses"**	
	Sensor 1st generation	*Sensor 2nd generation*	*Sensor 1st generation*	*Sensor 2nd generation*
Poseidon	2	2	4	4
Perseus	2	1	4	1

The Perseus family supports the connection of up to 102 meters and the monitoring of up to 1200 variables. 2 meters are "reserved" for a physical device (meter of type Local) and for 1 virtual meter (for the creation of virtual values - "virtual sensors"). This leaves 100 meters for external value sources.

Along with the licensing change, there is also a simplification of the limits on the number of connected sensors (meters), where we no longer distinguish between restrictions on a specific interface, but only an overall limit on the number of meters, regardless of their type. Perseus units have internal limits of 102 (100 for external meters + 2 system - motherboard and operating parameters) meters and 1200 variables (up to 200 can be reserved for system variables - motherboard and operating parameters).

In practice, this means that there is no specific SW limit on the number of 1-Wire sensors, RS-485 (Modbus/RTU) or M-Bus meters, but the limitation is based on the nature and speed of the bus (see chapter Description and limitations of buses). For example, if we use 8pcs of 1-Wire meters, 32pcs of RS-485 meters and 12pcs of M-Bus electricity meters, we have 52 external meters out of a total limit of 100. On the other hand, it is not possible to require the possibility of connecting 100 pcs of 1-Wire sensors if the device has only 4 RJ-11 connectors, because in all probability it would not be possible to maintain the topology of the 1-Wire buses (see chapter Description and limitations of buses).

Third-party templates and sensors (meters)

The Perseus family supports a wide range of third-party meters (sensors). Regardless of the manufacturer, they allow you to connect any device that complies with the protocol standards supported by the Perseus family - currently Modbus/RTU, Modbus/TCP, SNMP, MQTT and partially M-Bus (only models equipped with this interface). For these devices, it is not possible to use the variable auto-detection function similar to 1-Wire or M-Bus, and it is necessary to define the individual parameters manually. Due to the fact that there are dozens and in some cases hundreds of parameters that even a layman has to find out and fill in, the Perseus series is equipped with template support for these meters. Templates are predefined sets of parameters for specific sensors (meters) from specific manufacturers. They work in such a way that the necessary template of meters is uploaded to Perseus, and then when adding a device, a template is used for pre-setting (in fact, it is enough to just fill in the address and possibly login details) and add the meter and variables according to the template. There is no need to know data types, conversion formulas, specific addresses or SNMP OID values, or to find out their units. However, templates cannot work for dynamic structures (sensors with a variable number of tracked values)!

Template repository

Since Pereus can support literally thousands of third-party meters, and it would be very difficult to work with them in the device itself, Perseus units do not contain all available templates, but only a very small selection.

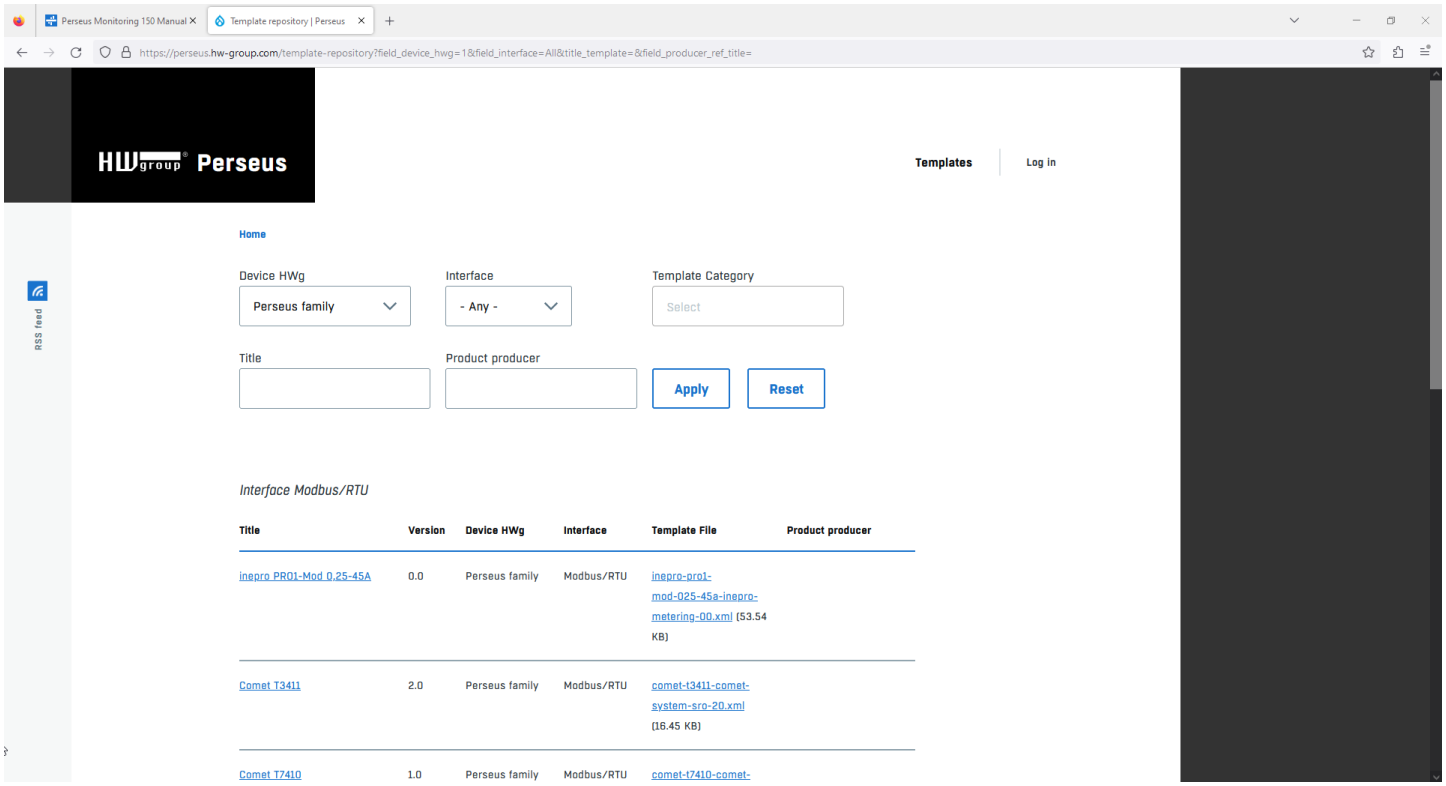
It is unlikely that the user will need more than say 10 different templates in one device (their number is not limited), it is useless to go through lists of hundreds to thousands of templates, even if they are sorted parametrically, not to mention that adding a single template would require a firmware release.

Therefore, a public portal (https://perseus.hw-group.com) was created as a repository for templates, LUA scripts and customer configurations (only project customers for medium sized projects). Here the customer searches for the required template (if he does not find it, the HW group will help him to create it and then it will be added to the repository), downloads it and uploads it to the device.

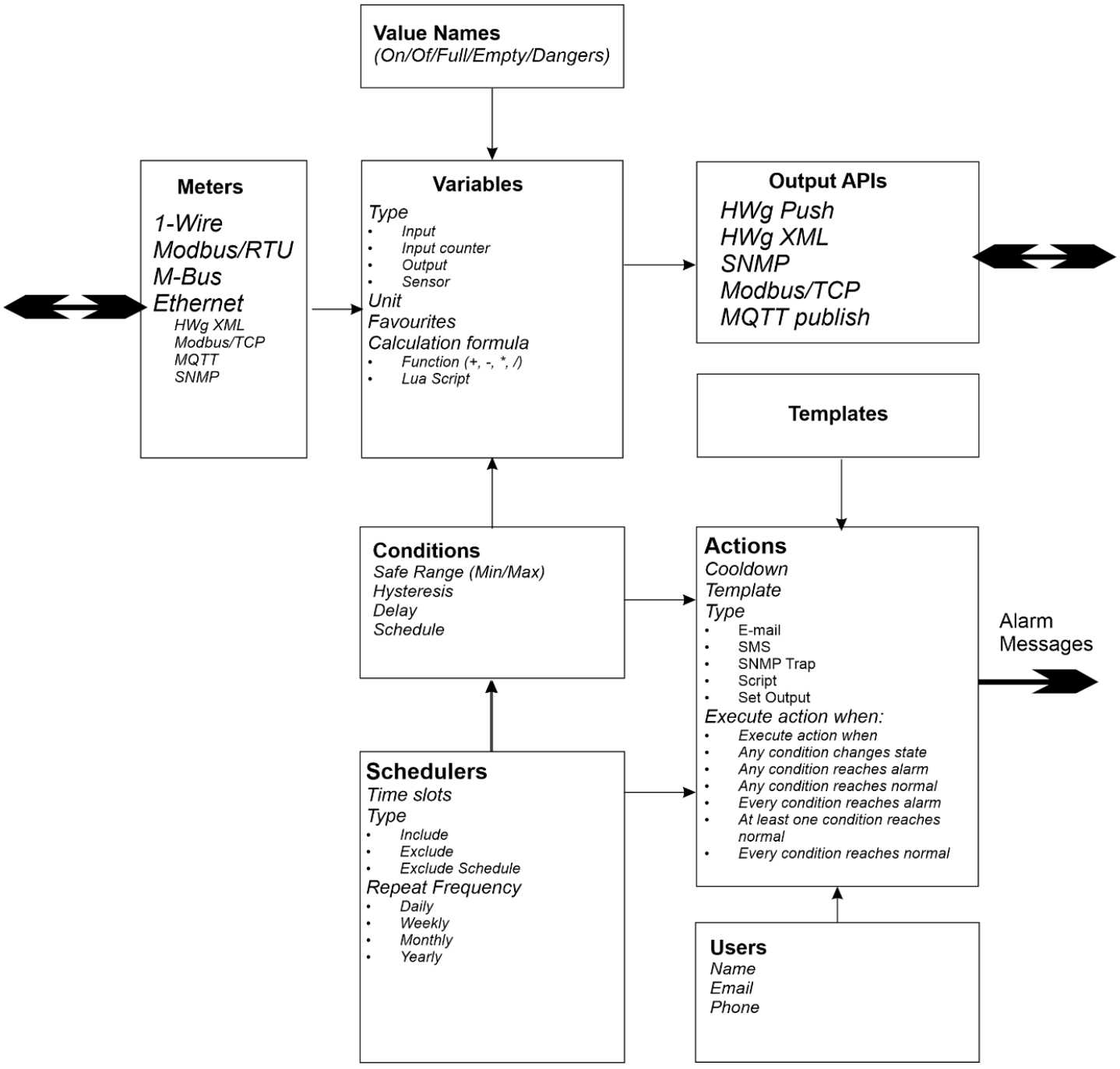
Over time, there will probably be a functional communication channel for downloading directly from the device, but primarily devices in closed networks should be considered.

It is not in the power of the HW group to create all templates in advance, but there is a prepared process for their remote creation in cooperation with the customer. Contact us at sales@hw-group.com if you need help with third-party sensor templates, custom LUA scripts or custom configuration.

Templates are available at a special website: https://perseus.hw-group.com



Architecture of the Perseus platform



The Perseus platform products are identical in all series and built on a common foundation. In contrast to the Poseidon units, where sensors are recognized, the Perseus units recognize Meters, similar to the HWg-PWR units.

Meters have their own physical interface (1-Wire, RS-485, M-Bus, Ethernet (XML or SNMP), motherboard, etc.) and can be set to Measure period, name, etc. The meter is an envelope for variables (values) and is supposed to represent the physical sensor, its connector and supply cable, regardless of what quantities it measures, how many detectors it contains, and so on. Meter is a HTemp 1-wire outdoor sensor, as well as an externally connected STE2 r2 unit or even the entire Poseidon2 4002.

Variable is a monitored variable, no matter if it is an analog value, a digital input, a relay output or a WLD detector.

Each Meter can have one or more (n) variables - i.e. the monitored variable, whether it is a sensor, a detector (DI) or an output (DO or analogue). Variables can be physical (data coming from the meter after it has been created) or virtual (for example, a calculated dew point, or a simple condition). A variable has its value, unit, exponent, and other parameters.

Each variable can have one or more (n) Conditions. The Conditions are formed by defining the Saferange, i.e. Minimum and Maximum Allowed Value, Hysteresis (Inertia Area) and Delay. Based on the conditions, the state of each Variable is defined. If a Variable has no condition defined, its state can be only OK or Invalid, but the Alarm state (Alarm Hi, Alarm Lo, etc.) can never occur. A variable (value) has a status of OK (read OK), invalid (read error) or disabled. Variables never take the state Alarm, which is only available for individual Conditions. Also, Actions handle the state of the Condition, not the state of the Variable.

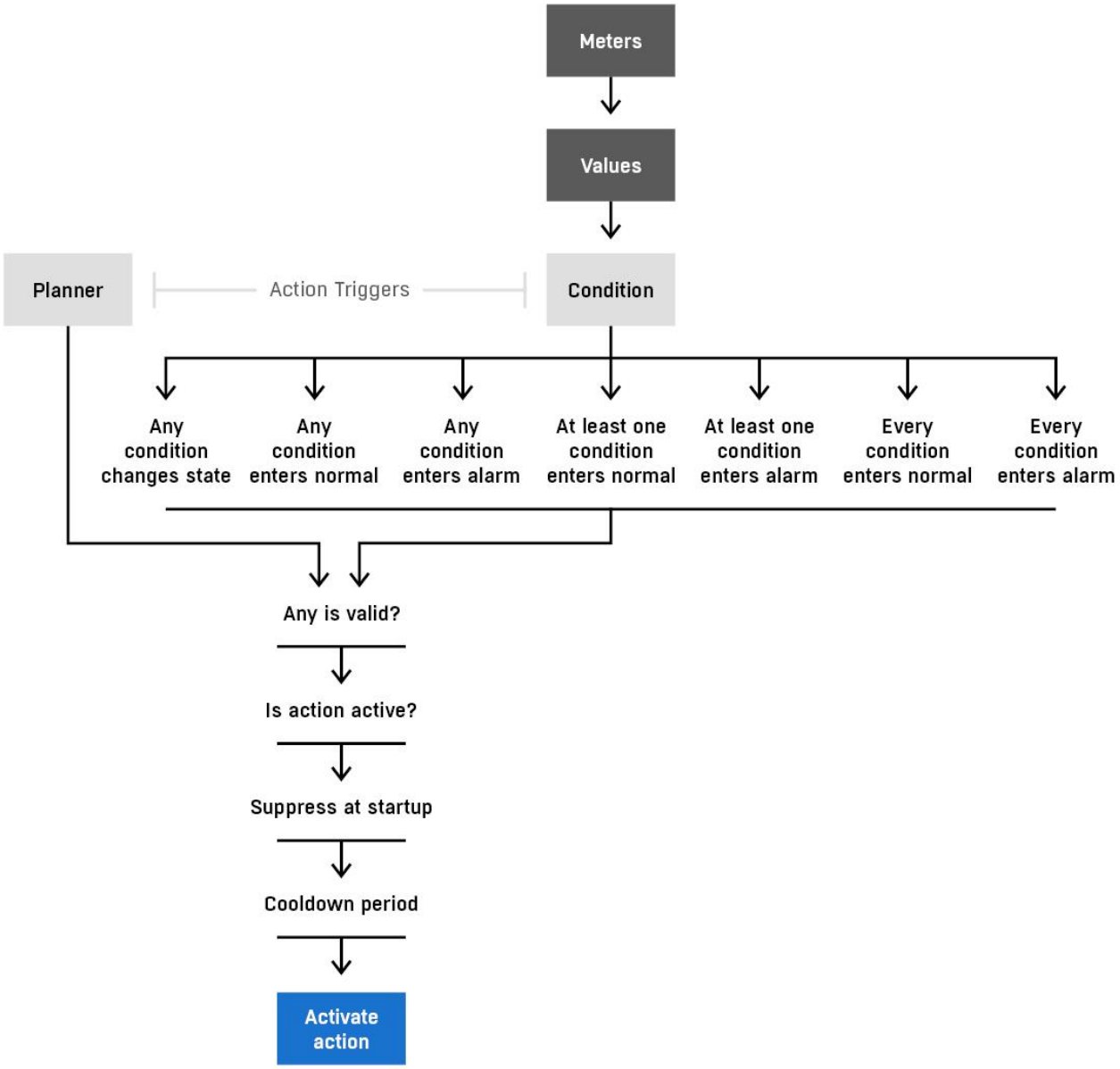
In addition, the Condition can be supplemented with a Scheduler, which can be used to define when the condition will be evaluated. If no calendar is selected, the Condition is evaluated each time the variable or meter is read, whereas if a calendar defining the working time period is selected, the Condition outside this period will be ignored and no follow-up actions will be performed.

In addition, one or more Actions can be assigned to each Condition, such as sending an email, sending an SMS, sending an SNMP trap or switching the output. In addition, the Action can be suppressed on device startup, or it can be performed only at the beginning or end of the alarm state, at the beginning or end of the OK state, and so on. So the setting options are really detailed.

An Action is an action element that is triggered either in real time by a scheduler, by changing the state assigned to a Condition, or by a combination of both.

In the case of actions such as sending an SMS or e-mail, it is possible to assign one or more destinations in the form of SMS or e-mail message recipients, which are defined separately and can be freely assigned to actions

An action can be assigned to a calendar (planner) and thus regularly trigger actions such as periodic e-mail, SMS or output switching, i.e. similar to the cron function.


Meters
Values
Planner
Action Triggers
Condition
Any condition changes state
Any condition enters normal
Any condition enters alarm
At least one condition enters normal
At least one condition enters alarm
Every condition enters normal
Every condition enters alarm
Any is valid?
Is action active?
Suppress at startup
Cooldown period
Activate action

Alarms

The procedure for creating alarm messages is therefore very different from simpler systems such as Poseidon2 or STE2, because to send an e-mail for example, in addition to defining an SMTP server, it is also necessary to define a destination (recipient), a message template (if the default one is not sufficient), a condition and to connect these elements at the level of each Value. On the other hand, it is not a problem to always send Alert messages to those employees who are currently on duty, or to suppress operations performed on weekends and holidays.

Types of meters of the Perseus platform

Each interface has its own set of IDs by which the meters can be sorted:

1	Local
1xx	1-Wire
3xx	M-BUS
4xx	Modbus/RTU
5xx	Modbus/TCP
6xx	Network
7xx	Virtual
8xx	MQTT
9xx	SNMP
10xx	LocalGPS* Perseus Industrial

Types of Perseus platform variables

As already mentioned, within the Perseus family there is no distinction between sensors, digital inputs (DI) and relay outputs (DO), but all variables are Variables and have their type as a parameter:

0	Unknown
1	Input
2	Input counter
3	Output
4	Sensor

In the future, variable types will expand based on their parameters, such as:

- Input
  - Binary
  - Status
- Output
  - Analog
  - Two-state
  - Multi-state

Meter States, Variables, and Conditions

Unlike previous generations of measuring data panels from HW group, the Perseus family has the ability to work with different states of its components. Therefore, not only the Variable, but also the Meter and the Conditions have their state. Variables or meters can never be in the Alarm state, that is a matter of Condition.

ID	Meter	Variables	Condition	Scheduler	*Note*
0	Unknown	Unknown	Unknown	Inactive	
1	OK	OK	OK	Active	
2		VariableInvalid	VariableInvalid		
3	MeterInvalid	MeterInvalid	MeterInvalid		
4			AlarmLow		
5			AlarmHigh		
6			Inactive		Deactivate by scheduler
7	Disabled	Disabled	Disabled		

Measured vs. Monitored Variable - Just a note for the dictionary.

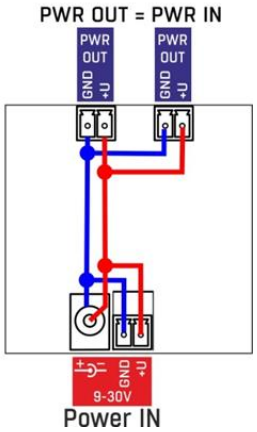
Hardware Specifications

Ethernet

Ethernet 100Base-T (10/100Mbps). Once connected, the green "Link" LED on the connector will illuminate to indicate that the active element is properly connected, and the yellow "Activity" LED will blink during communication.

Power

The power supply of the device is indicated by a green LED. The power supply is always in the range of 9-30V, and 48V, PoE (Power over Ethernet).



PowerOut

It is used to power the end sensors and accessories. The PowerOut output is directly coupled to the 9-30V input. For units with 48V or PoE input, the output can be used as a 12V/120mA power supply.

1-Wire/1-Wire UNI

Connection of sensors produced by HW group with 1-Wire/1-Wire UNI interface. A bus of up to 60 meters and a maximum of 2 1-Wire UNI sensors can be directly connected to each port. For more information, see chapter Connecting sensors.

RS-485

It is used to connect RS-485 sensors and Modbus/RTU protocol.

M-Bus

It is used to connect M-bus meters. The industrial data bus M-BUS or Meter-BUS (EN 13757-2 for the physical layer, EN 13757-3 for the application layer) is designed for remote data reading and control in the field of metering and control of heating systems, gas, water and electricity consumption.

Inputs

INx - Inputs for connecting dry contacts. Inputs always have a common GND potential. The switching of the input is signaled by a green LED. The inputs are equipped with pulse counters.

Outputs

OUTx – relay outputs with changeover contact. In idle state, the terminals NCx (Normally Close) + COMx (Common) are closed, in closed NOx (Normally Open) + COMx. The switched output is indicated by a yellow LED.

Alarm/Setup LED

A red LED indicates the status of the device

- Solid On – A meter, variable or condition is in the Alarm state
- Blinking – Device is upgrading

LED Status

While waiting for the configuration to be restored, the status LED lights up. After the light goes out, a recovery is performed.

LED Modem

Indicates the status of the modem.

- Lights up during LTE connection
- Flashes while connecting and searching for a network

DIP1/DIP2 System Switches

DIP1 - SW1 - Reserved for future use.

DIP2 - Safe Mode - Activation activates HW protection of the setting. No parameters can be changed.

WLD

It is used to connect the flood detection cable. LED indicates WLD state:

Green – OK

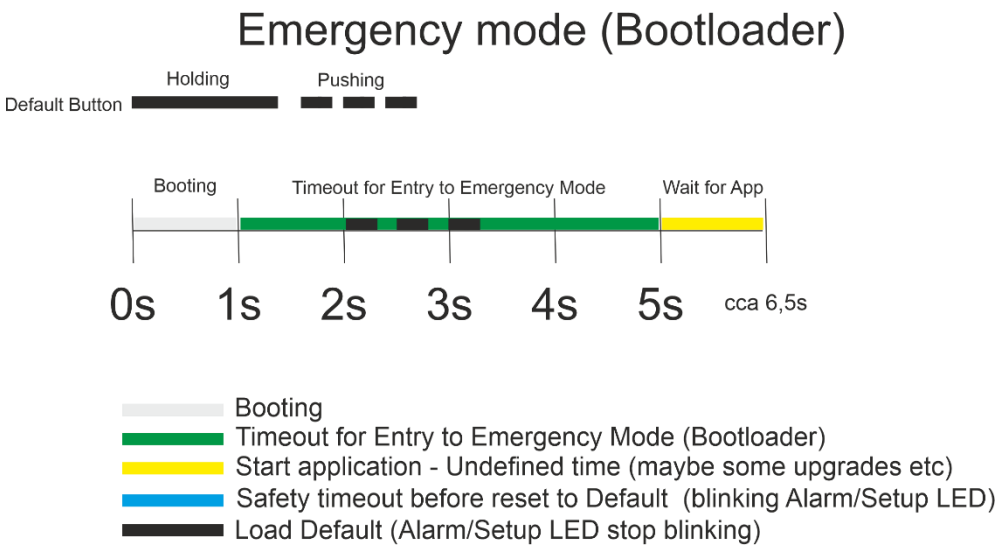
Red – Flooded

Yellow - Disconnected

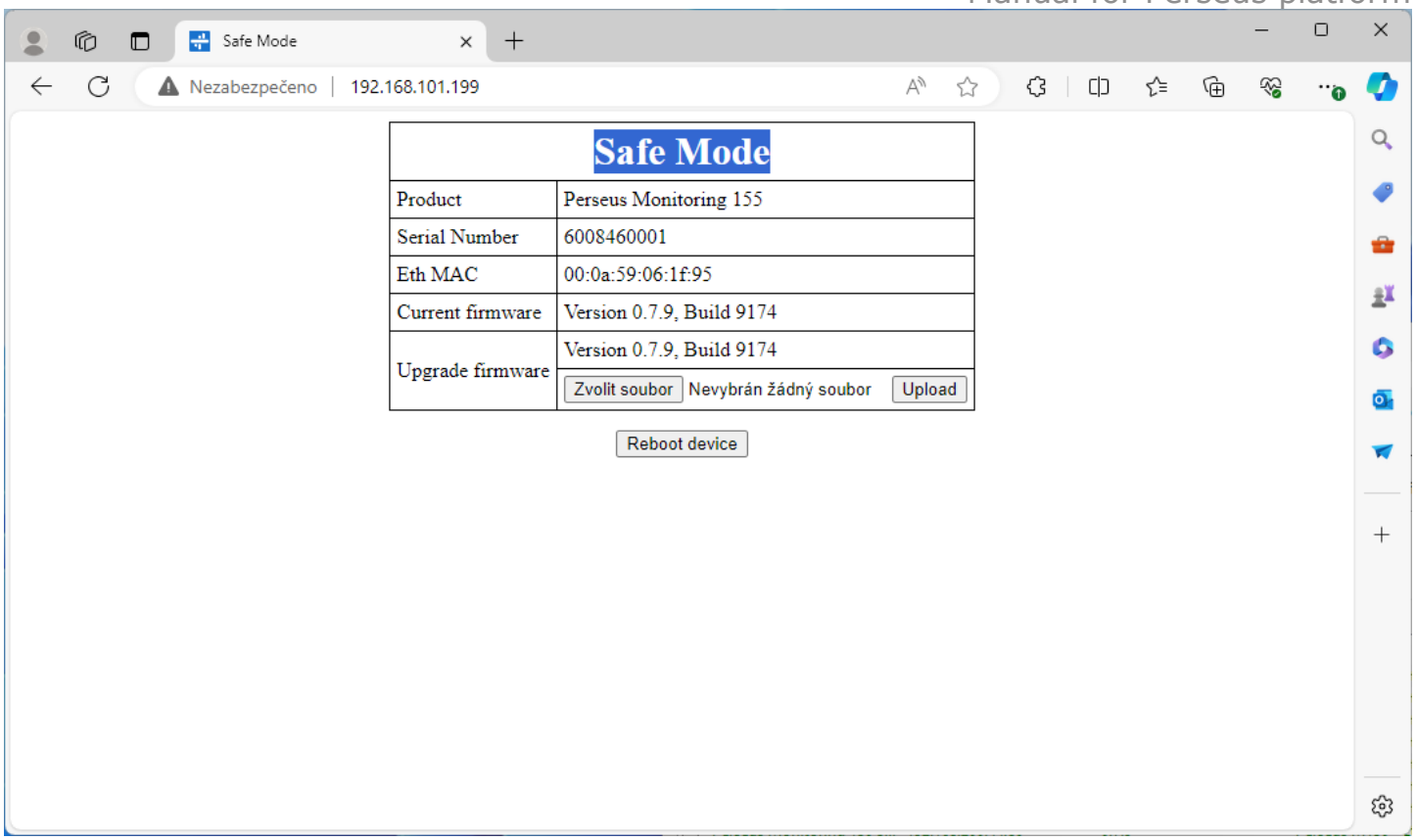
Default button

Multi-function button for factory reset and emergency boot loader recall.

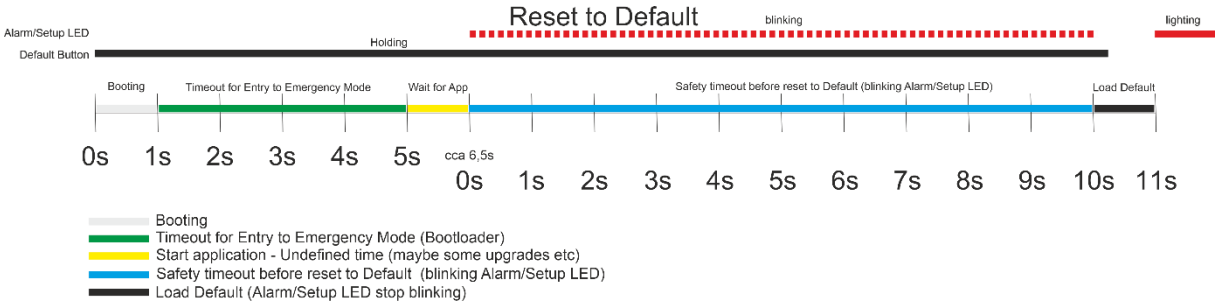
Bootloader Recall (Safe Mode) - Press the button, turn on the power within 3 seconds, press the button 3 times.



This will take you to the Safe Mode web page where you can upload the new firmware:

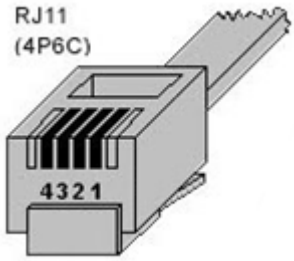


Factory Reset - Press the button, apply the power and wait for the Alam/Status LED to turn off. Releasing the button while the LED is flashing will ensure that the application is booted up with the current configuration.



Connecting Sensors

1-Wire Bus (RJ11) sensors



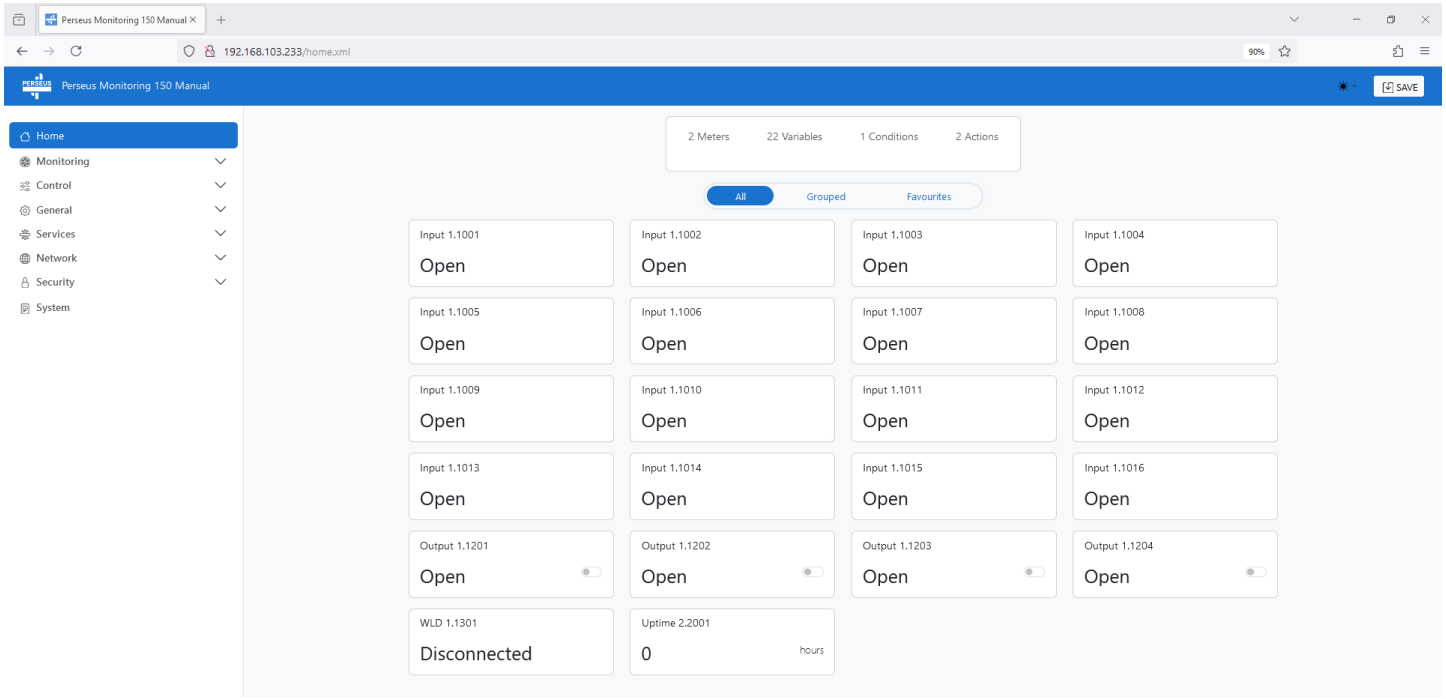
- Connect the sensor to the Poseidon before turning on the power - **the connector must click;**
- **Max distance per active port is 60m**;
- Sensors can be daisy-chained to connect them one after the other;
- After a change in the connected sensors, you must perform sensor auto-detection again. (WWW interface > Sensors > **Autodetect sensors**)
- for details see sensor's manual)

WWW interface of Perseus platform

Menu structure

- **Home** - Front Page
- **Monitoring** - a group of pages for working with sensors and their safe range
  - **Meters** - a page for working with meters
  - **Variables** - a page for working with monitored variables
  - **Conditions** - overview of existing conditions (SafeRange)
- **Control** - device settings in terms of action and reaction to the data obtained;
  - **Actions** - overview of existing actions and the possibility of creating new ones
  - **Targets** - overview of existing alarm targets and the possibility of creating new ones
  - **Scheduler** - overview of existing calendars and schedulers and the ability to create new ones
  - **SMS** - setting parameters for sending SMS, template definitions and queue preview
  - **E-mails** - setting parameters for sending SMS, define templates and preview the queue
  - **Scripts** - overview of existing variables that have been set to be recalculated by a script and edited by LUA
- **General** - General device settings
  - **Identification** - Identification of devices on the network and in the API
  - **Web Server** - web server parameters (http and HTTPS ports) and the ability to upload SSL certificates
  - **Time** - set the system time and its synchronization
  - **Meter templates** - Manage meter templates, variables, and LUA Scripts
  - **Log** - It is used for switching on the datalog, setting the logging period, selecting logged values and simple downloading of measured data
- **Services** - Additional device services
  - **Portal** - set up the target Sensdesk Technology portal
  - **SNMP** - set parameters for SNMP communication
  - **MQTT** - MQTT publisher settings
  - **Modbus TCP** - Modbus/TCP settings
  - **BACnet –** BACnet server settings
  - **Syslog –** Syslog settings
- **Network** - Setting the network parameters
  - **Ethernet** - Basic network parameters
  - **WiFi** - setting WiFi parameters (where applicable)
- **Security** - System Security
  - **Radius** - set the Radius server parameters
  - **Users** - Add users authorized to work with the device
- **Modem** - set the properties of the LTE modem (where applicable)
- **System** - Basic information about the device, FW upgrade and configuration

Home



On the front page of Perseus products, you will find a navigation menu in the left column and an overview of all monitored variables in the main window.

The Summary section lists the number of monitored Meters, Variables, Conditions and Actions, with the numbers in the Invalid and Alarm states. Individual elements include a link to the list to make it easier to identify the problem.

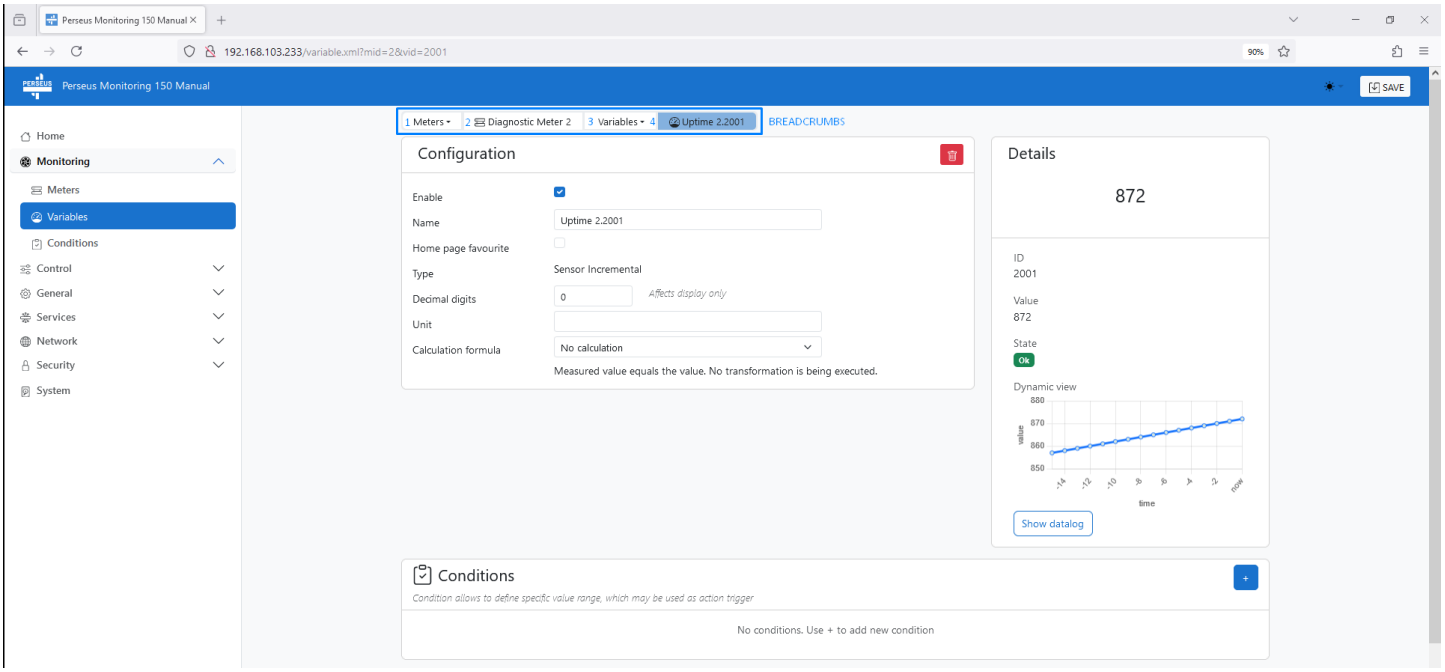
The All/Grouped/Favourites switch allows you to list all variables, list them merged by meters, or filter the overview of monitored variables to Favourites only. Which variables are preferred is set by a checkbox in the variable detail.

The variable overview allows you to sort by ID, name, or status. Variables in the Invalid state are highlighted in yellow, in the red circle they have the number of conditions that are in the Alarm state and preferred ones have an asterisk displayed in the upper right corner.

If it is an output type variable, the slider on the home page allows you to switch the output (change the output value). The exception is an output to which an internal condition is assigned.

Monitoring

Monitoring is a group of functions for working with Sensors, Meters, Variables and Conditions, the result of which is the definition of the status of individual elements of the chain, to which it will be possible to apply other functions and actions (alarm messages, etc.).



Within a group of pages, it is possible to move between objects not only by using the main menu and the overview pages Meters, Variables and Conditions, but also by using the "breadcrumbs".

Meters => name of Meter => Variables => name of Variables => Conditions => Name of condition.

1 – Overview of individual meters. Their overview will expand and it is possible to switch to another one. Of course, switching to another meter shortens the menu accordingly.

2 – Current meter – by clicking on it, you can switch to its properties and the menu will be shortened accordingly.

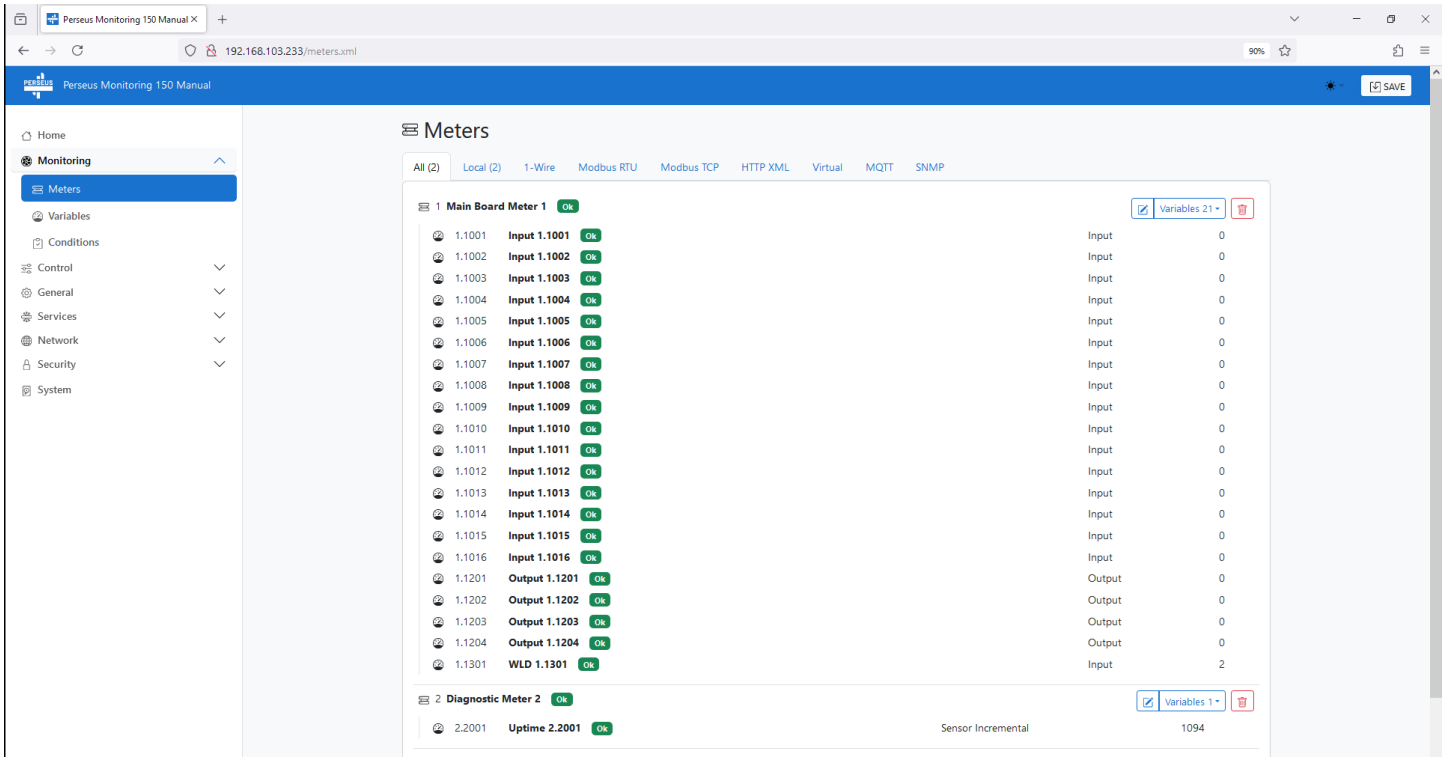
3 – Overview of individual variables of the meter. Their overview will expand and it is possible to switch to another one.

4 - The current variable, whose parameters are displayed on the page.

Of course, switching to another meter shortens the menu accordingly.

Menu Monitoring / Meters

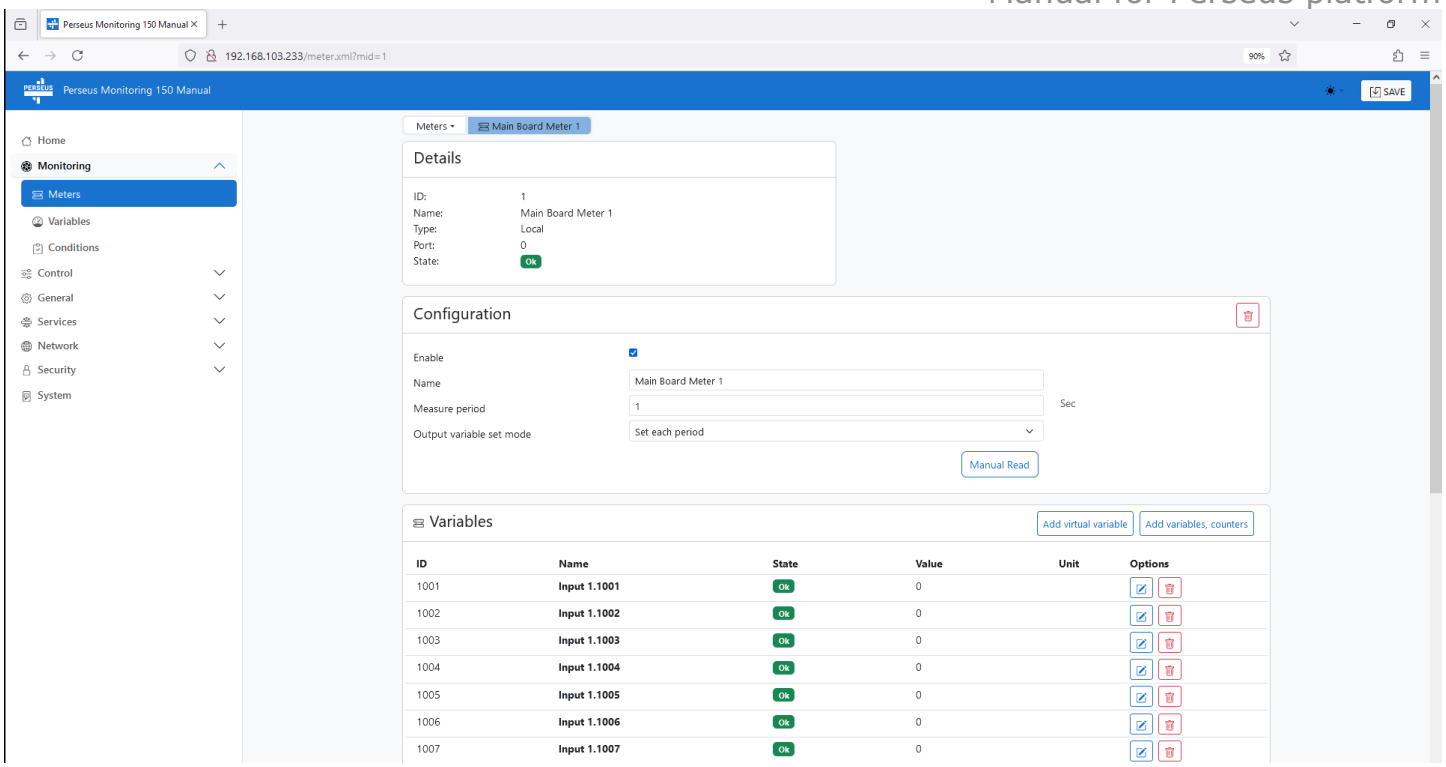
The site is used to work with meters, search, add, delete and edit them. It allows you to work with all the interfaces that the Perseus can handle. *The availability of individual items (interfaces) may vary in different models.*



The All tab provides an overview of all meters, their names, statuses and IDs. It can also be used to find out the total number of connected meters (number in brackets).

Local

Local contains local and diagnostic meters, typically local sensors, digital inputs, outputs and pods contained directly on the motherboard of the device. These are not meters connected via external buses.



Details section

- **ID** - ID of the meter – see chapter Monitoring / Meters
- **Name** - The name of the meter. The name can be changed in the Configuration section
- **Type** - type of meter – see chapter Monitoring / Meters
- **Port** - The port number to which the meter is connected. For local or Ethernet sensors, this is an internal designation.
- **State** - meter status (Unknown/OK/Device Invalid)

Configuration section

- **Enable** - Turns the meter on or off
- **Name** - The name of the meter for easier orientation
- **Measure period** - the minimum period is 1s, but it may vary depending on the unit configuration, numbers and types of sensors in order to comply with the necessary timeouts
- **Output variable set mode –** Option for sending the value in set measure period or when the value has changed
- **Manual** Read - Button for manual reading of the meter

Variables section

A well-arranged table of connected variables, their states and values. Depending on the type of interface, it is possible to detect meter variables in this section (**Scan Variable** button) or manually add them (**+ button**).

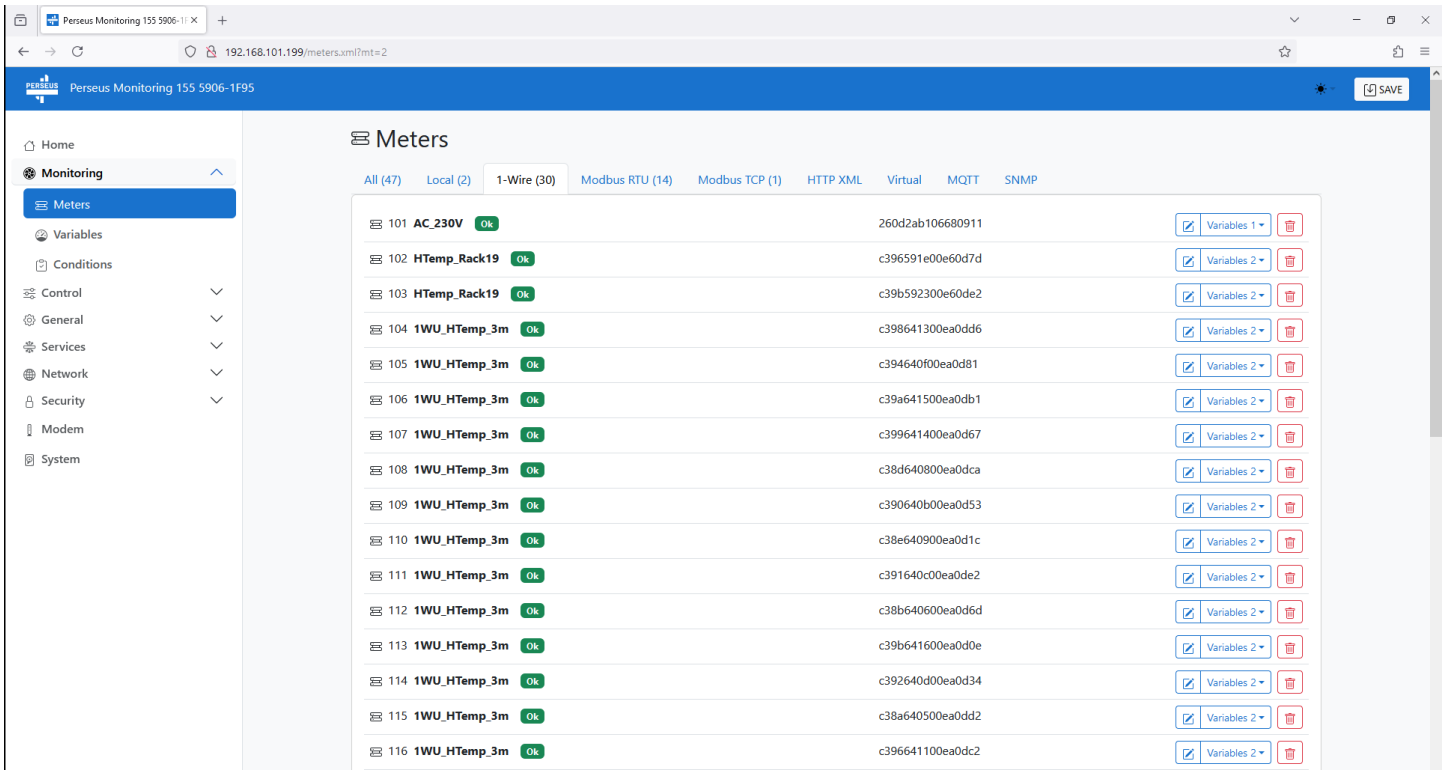
- **ID** - ID of the variable
- **Name** - The name of the variable
- **State** - The current state of the variable
- **Value** - The current value
- **Unit**
- **Options** - Options for working with the variable (Delete/Edit)

In the case of local meters, the meter can only be renamed and the reading period can be changed.

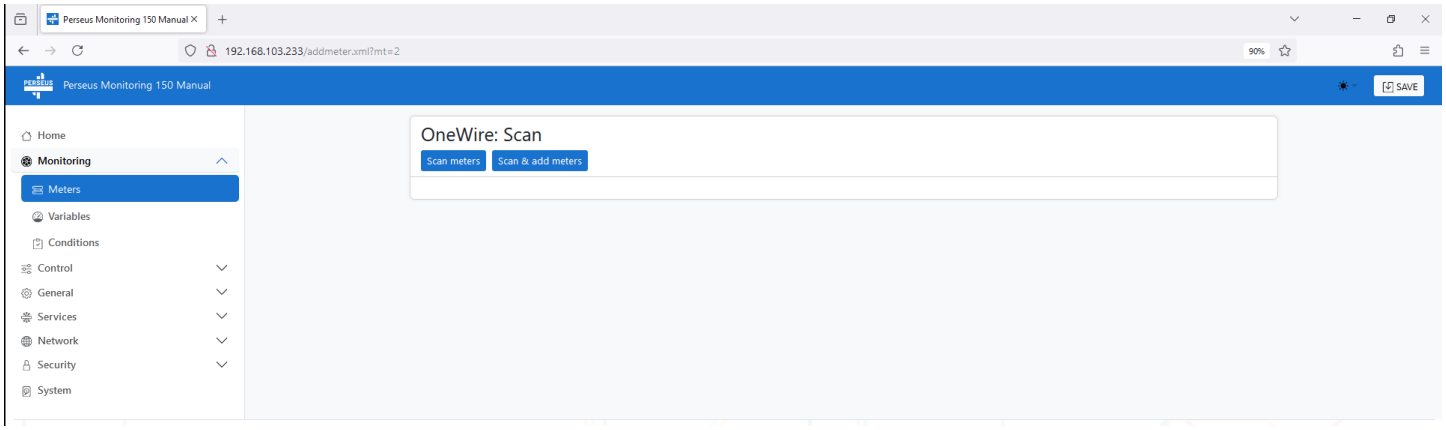
By clicking on the name or icon of the edit you can open the page for editing the variable – for more information, see chapter Variables.

1-Wire

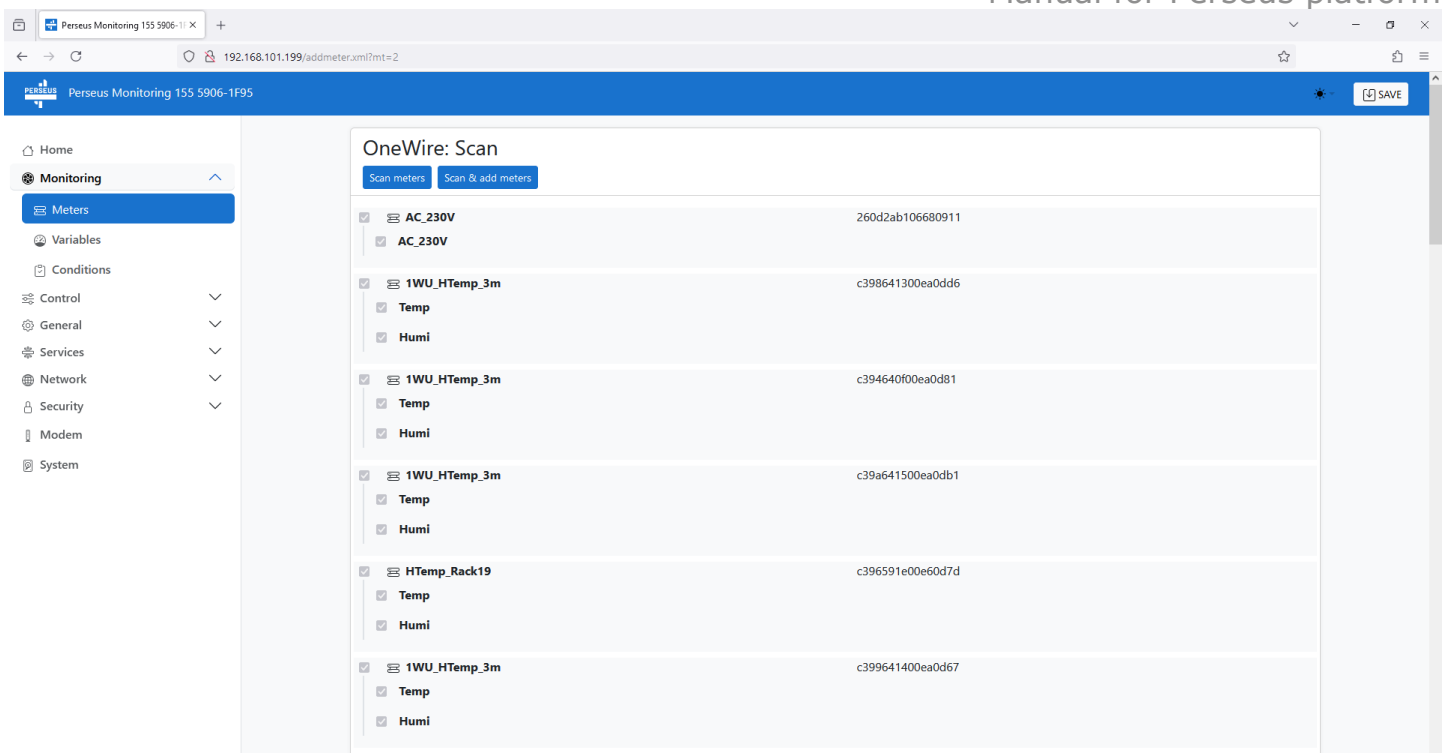
The 1-Wire tab contains meters connected via the 1-Wire interface.



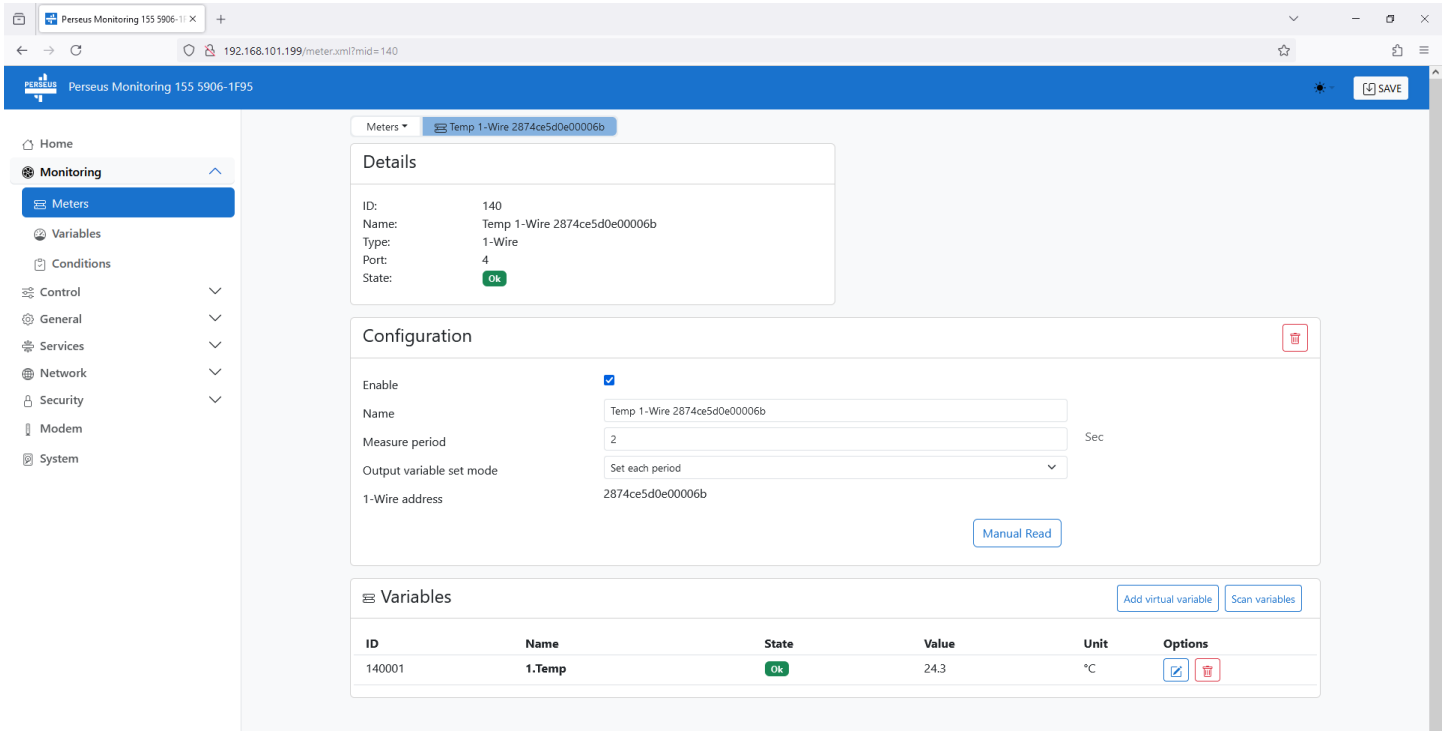
The 1-Wire: Find button opens the 1-Wire meter search page:



The *Scan & add meters* button scans the bus and adds all the sensors found, the *Scan meters* scan the 1-Wire buses and lists the connected meters and detected Variables:



Here you just need to select the individual meters and variables. By selecting the meter, all variables are selected, or you can select only specific ones (not all).



Details section

- **ID** – ID of the meter – see chapter Monitoring / Meters
- **Name** – The name of the meter. The name can be changed in the Configuration section
- **Type** – type of meter – see chapter Monitoring / Meters
- **Port** – The port number to which the meter is connected. For local or Ethernet sensors, this is an internal designation
- **State** – meter status (Unknown/OK/Device Invalid)

Configuration section

- **Enable** – Turns the meter on or off
- **Name** – The name of the meter for easier orientation
- **Measure Period** – Period of reading – the minimum period is 1s, but it may vary depending on the unit configuration, numbers and types of sensors in order to comply with the necessary timeouts
- **Output variable set mode –** Option for sending the value in set measure period or when the value has changed
- **1-Wire Address** – Unique, immutable 1-Wire ID read from the meter
- **Manual Read** – Button for manual reading of the meter

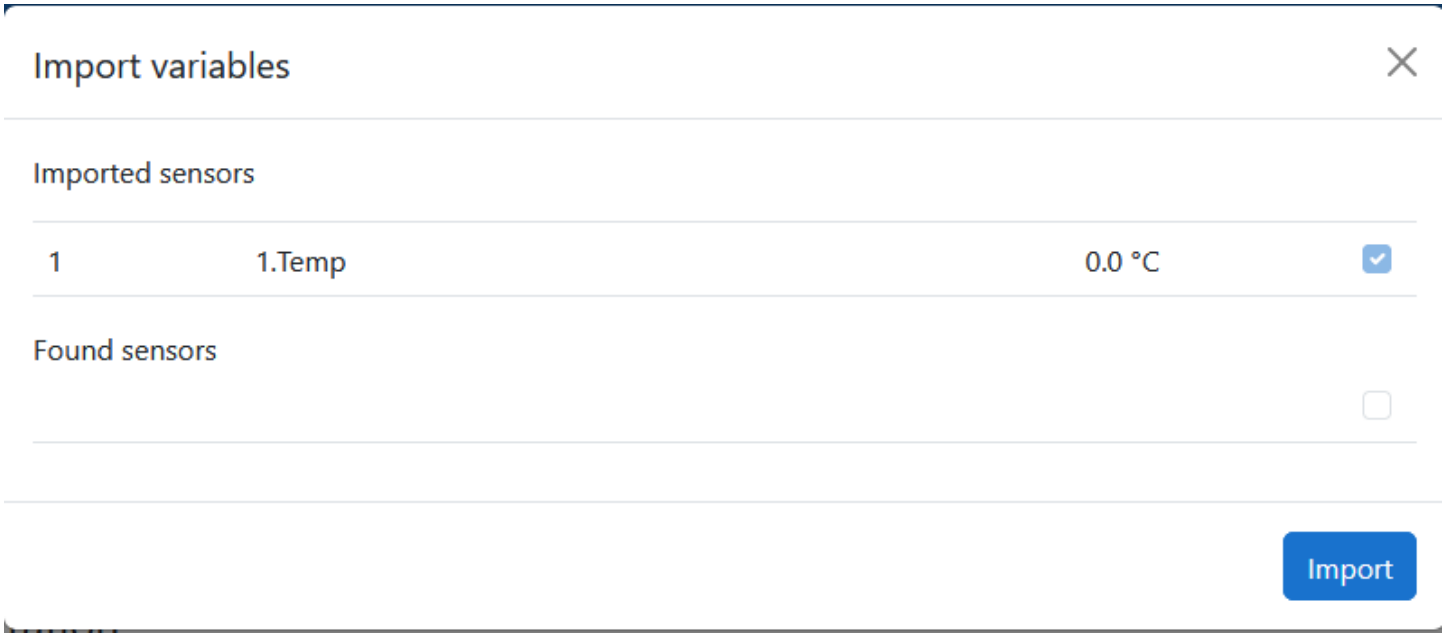
Variables section

A well-arranged table of connected variables, their states and values. Depending on the type of interface, it is possible to detect meter variables in this section (**Scan variables** button) or manually add them (**+ button**).

- **ID** – ID of the variable
- **Name** – The name of the variable
- **State** – The current state of the variable
- **Type** – The type of a variable
- **Value** – The current value
- **Unit** - The physical unit of the variable
- **Options** – Options for working with the variable (Delete/Edit)

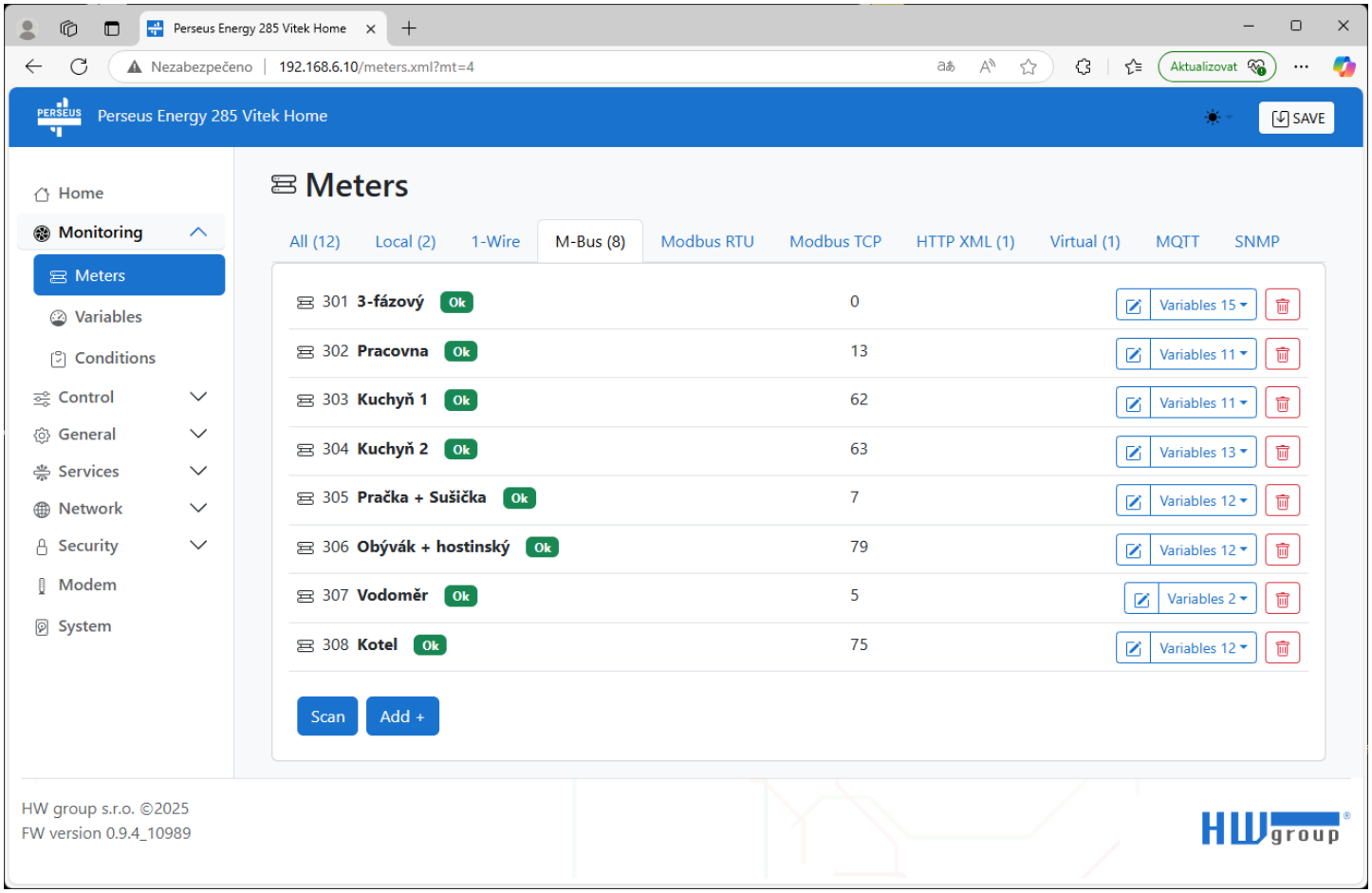
In the case of 1-Wire meters, you can just rename the meter and change the reading period.

Scan Variables



M-Bus

The M-Bus tab contains meters connected via the M-Bus interface. Meters equipped with M-Bus can be added manually or by searching for them on the bus.

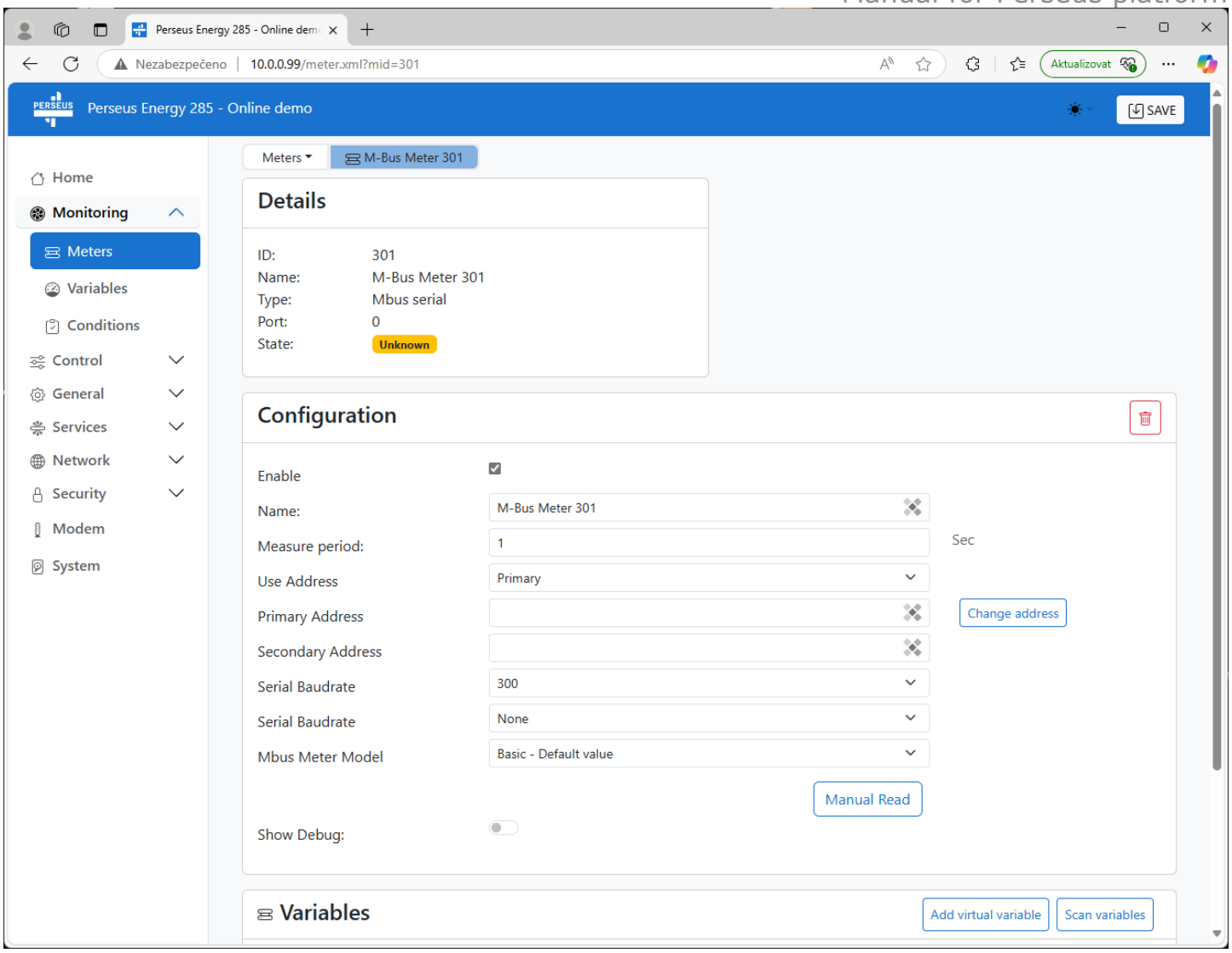


Meters can be added either manually by pressing the Add+ button or automatically by scanning the bus.

- Manual addition
- Scanning the bus

Manual addition

Adding is done by pressing the Add+ button



Details section

- **ID** – ID of the meter – see chapter Monitoring / Meters
- **Name** – The name of the meter. The name can be changed in the Configuration section
- **Type** – type of meter – see chapter Monitoring / Meters
- **Port** – The port number to which the meter is connected. For local or Ethernet sensors, this is an internal designation
- **State** – meter status (Unknown/ Disabled/OK/Device Invalid)

Configurations section

- **Enable** – Turns the meter on or off
- **Name** – The name of the meter for easier orientation
- **Measure Period** – Period of reading – the minimum period is 1s, but it may vary depending on the unit configuration, numbers and types of sensors in order to comply with the necessary timeouts
- **Use Address** - The Use Address option configures the addressing mode (Primary – shorter primary M-Bus address; Secondary – unique secondary address). For battery-powered meters, we recommend to use the primary address.
- **Primary Address** - Primary address on the M-Bus. The address is entered as one byte (0-255). Meters of the same type and by the same manufacturer are often

supplied with the same Primary address, which therefore cannot be used for addressing. The field allows changing the primary address as a convenience for battery-powered meters.

- **Secondary Address** - Secondary address on the M-Bus. This address is used to identify a meter if there are more meters with the same primary address (Address field) on the bus. Secondary address is not user-changeable.
- **Serial Baudrate** - M-Bus communication speed for the particular meter. By default, 2400 Baud is configured. Meters with different communication speeds can coexist on a single bus; this parameter sets the speed for the particular meter.
- **Serial Parity** – M-Bus communication parity. As with the baudrate, parity can be specified individually for each meter
- **Mbus Meter Model** - Meter type selection. This option is used to connect specific meters that require a different M-BUS query than the basic one, or, like INEPRO meters, require a combination of the two.
- **Manual Read** – Button for manual reading of the meter
- **Show Debug** - Displays a window with debugging information

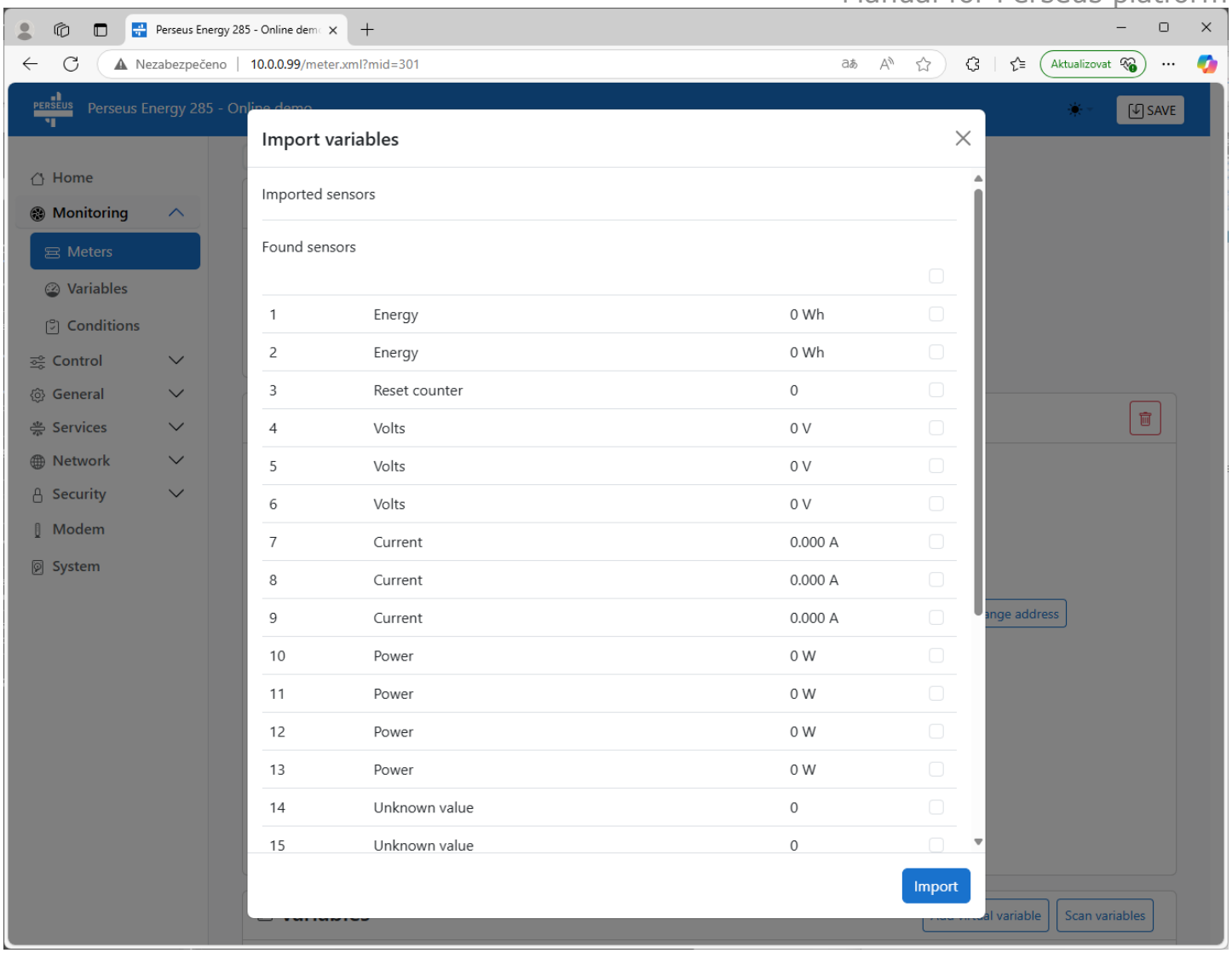
Variables section

A well-arranged table of connected variables, their states and values. Depending on the type of interface, it is possible to detect meter variables in this section (**Scan variables** button) or manually add them (**+ button**).

- **ID** – ID of the variable
- **Name** – The name of the variable
- **State** – The current state of the variable
- **Value** – The current value
- **Unit** – The physical unit of the variable
- **Options** – Options for working with the variable (Delete/Edit)

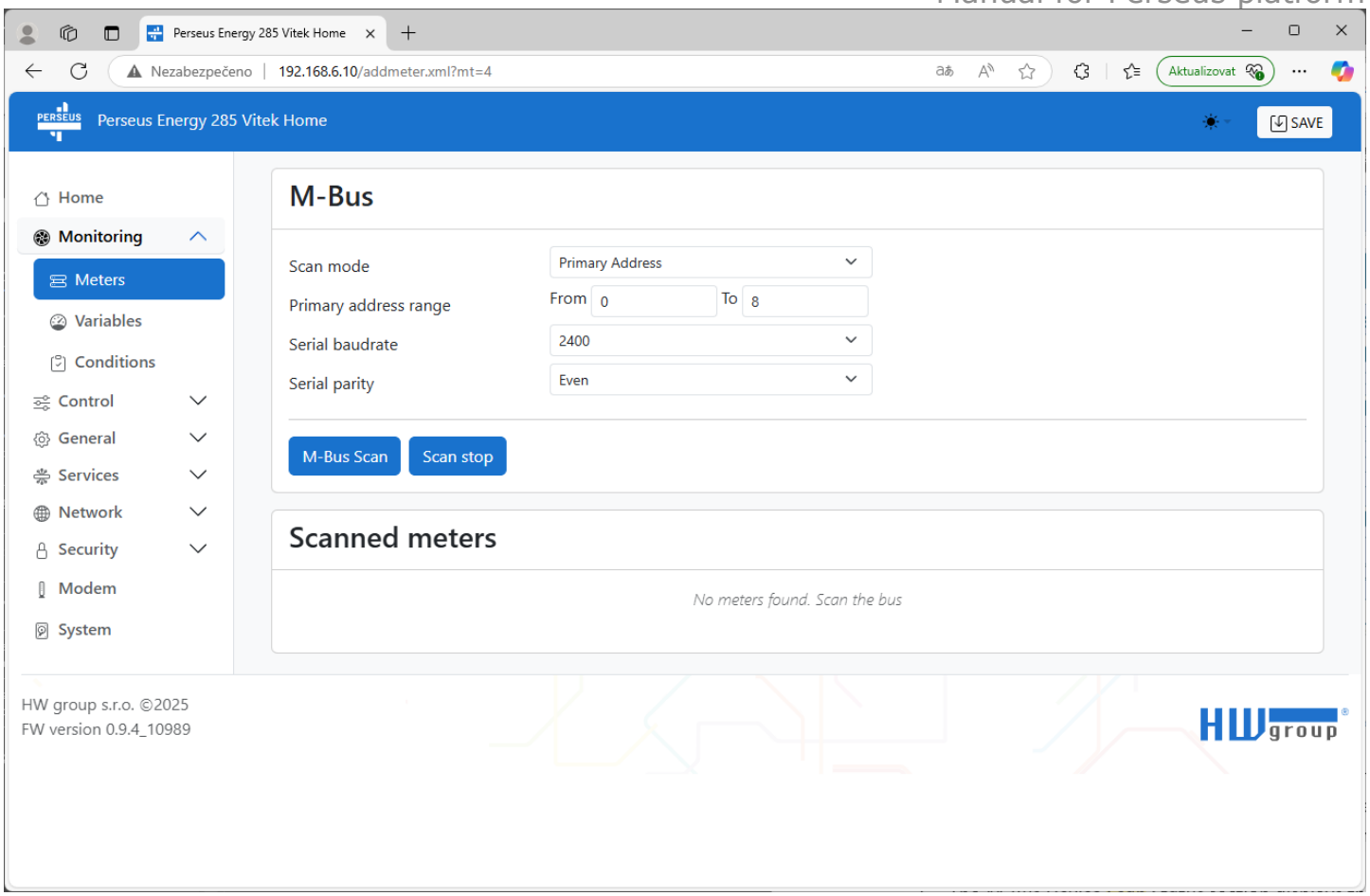
In the case of M-Bus meters, you can just rename the meter and change the reading period.

Scan variables searches for and allows adding variables in bulk.



Scanning the bus

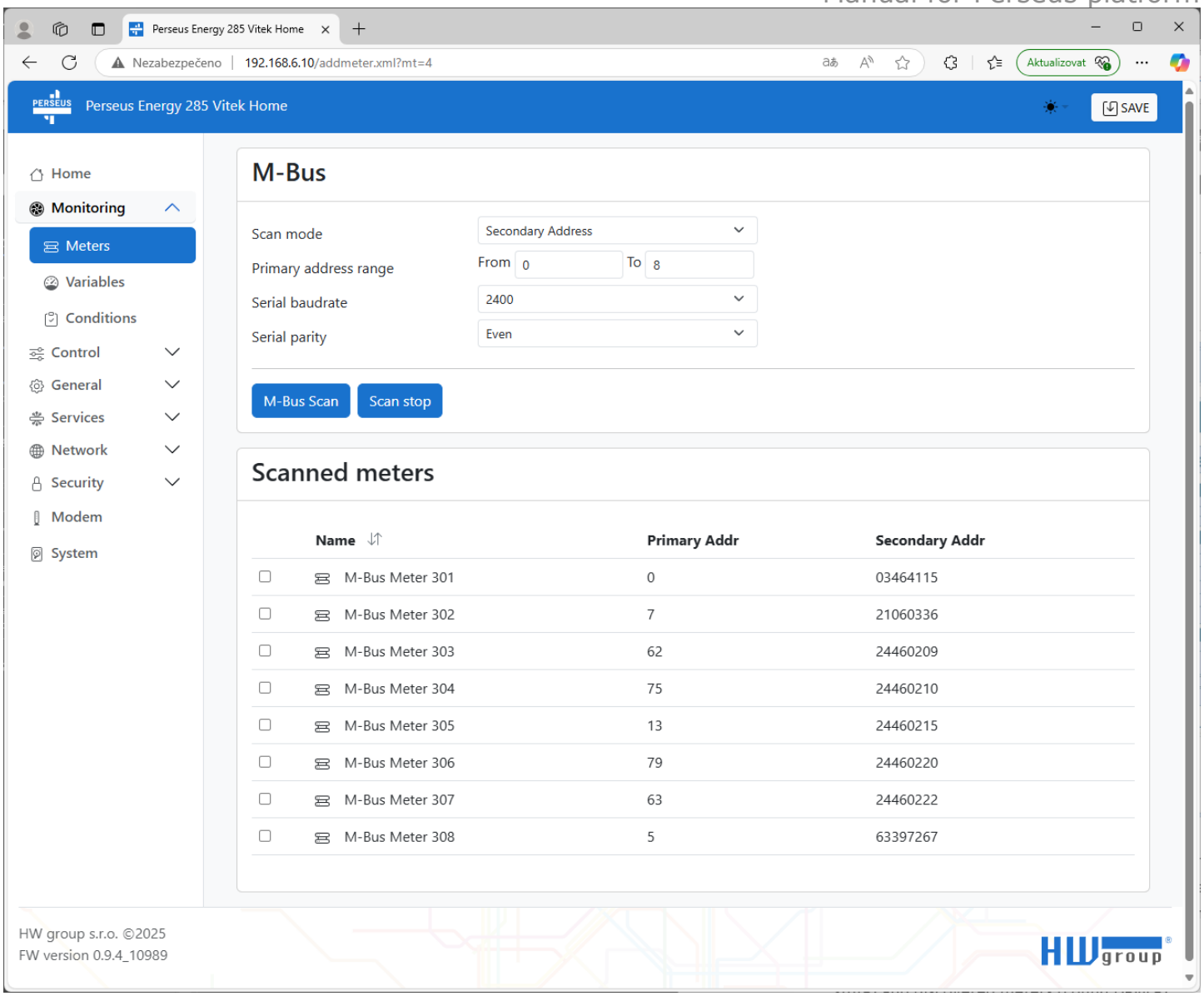
The M-Bus Scan button opens the M-Bus meter search page:



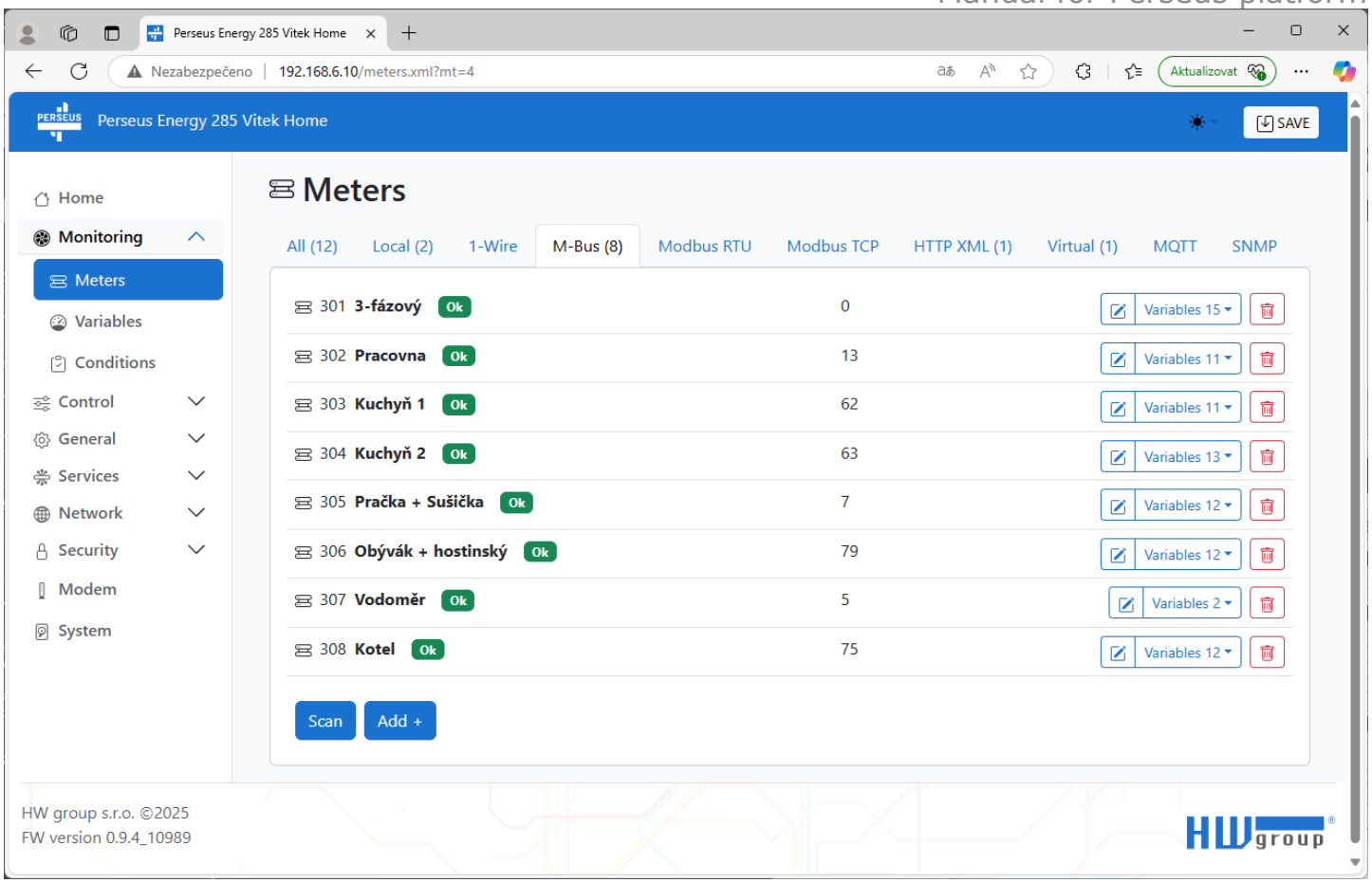
Here you need to select the search parameters:

- **Scan Mode** – Select whether the search will be performed using the primary (number 0-254) or secondary (typically serial number) address. Searching by primary address is supported by all meters, but takes longer and may need to be repeated for address collisions, searching by secondary addresses is more practical, but not all meters support it.
- **Primary address range** – When searching by primary addresses it is possible to limit the searched range and thus speed up the whole operation significantly.
- **Serial Baudrate** – Communication speed. The default is 2400 but this can differ for various meters – see the meter documentation.
- **Serial Parity** – Communication parity. The default is Even but this can differ for various meters – see the meter documentation.

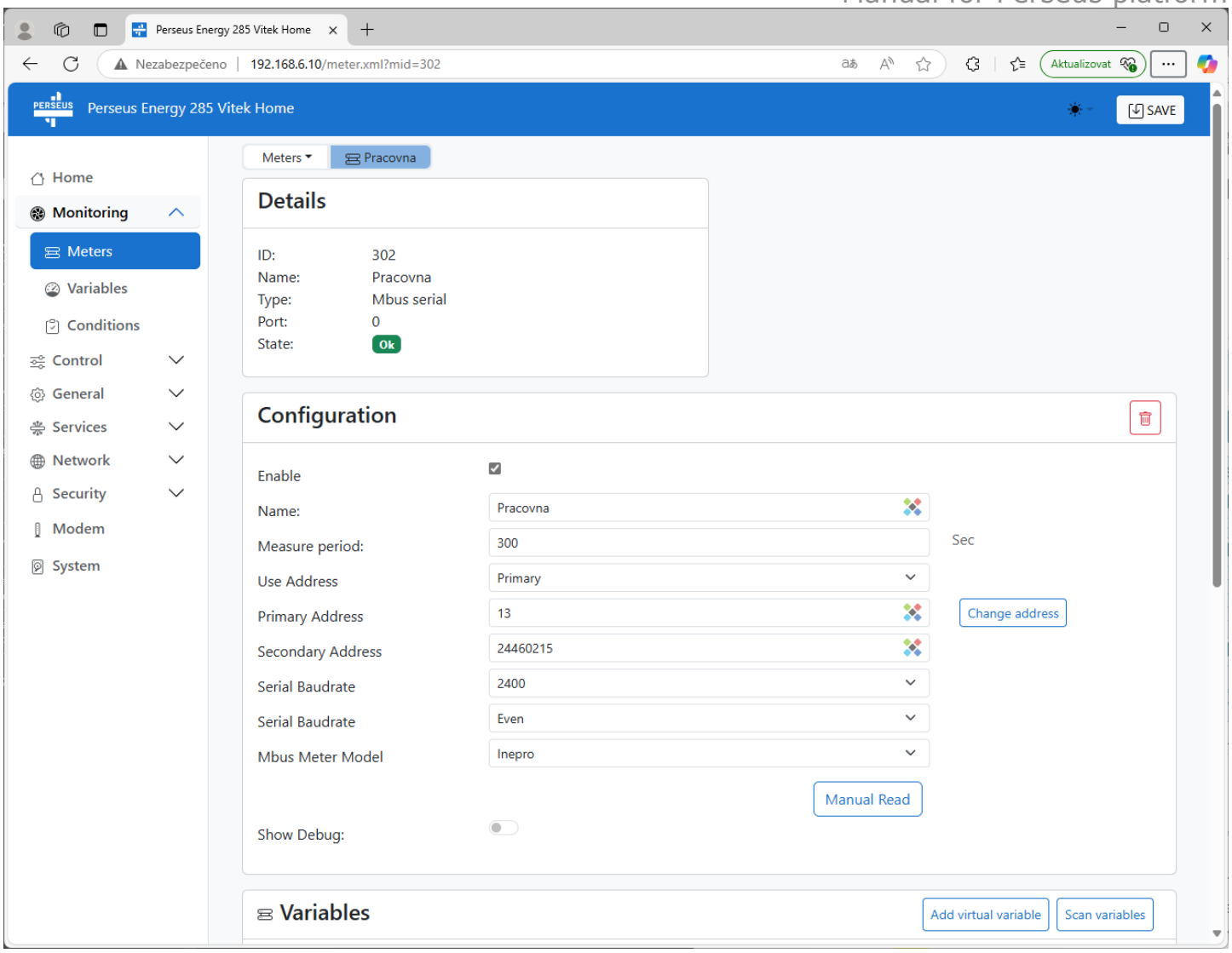
The result of the search is a listing of all found meters:



Now you can select and check all or only the ordered meters and save them with the Save button. This will create the individual meters and we can move on to editing them and adding variables.

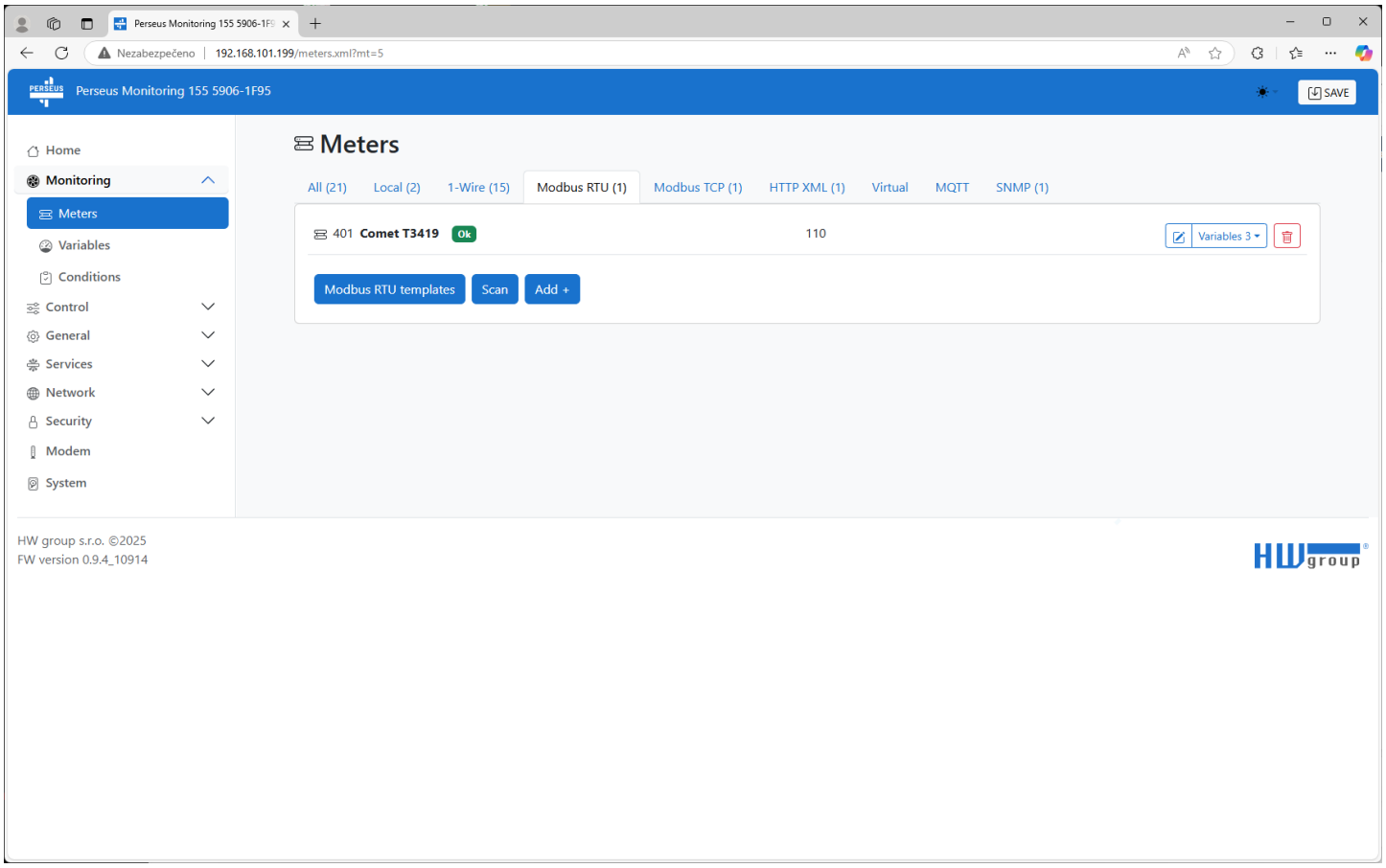


Here we open the meter editing:

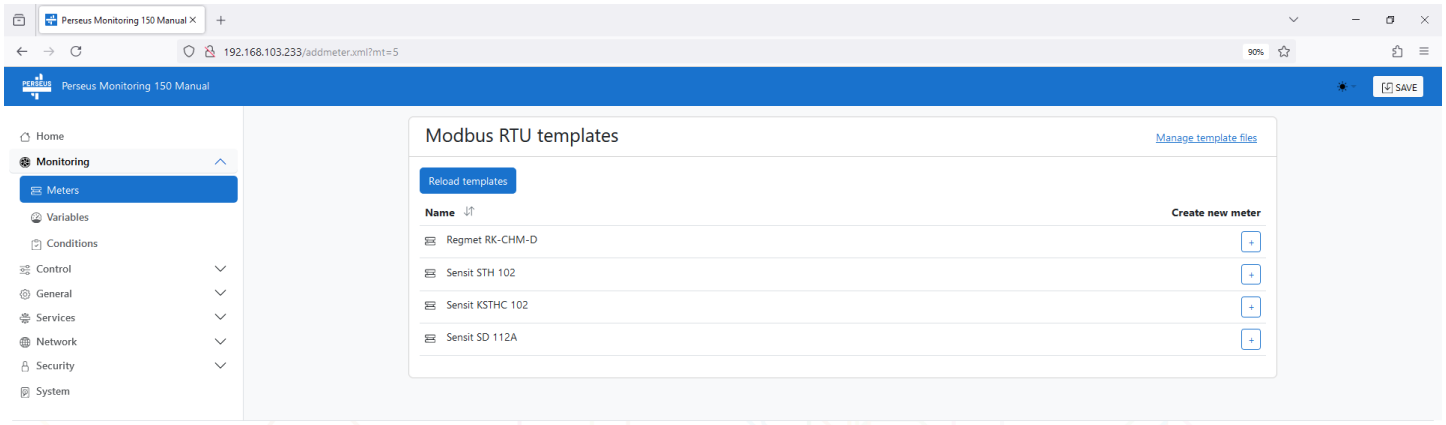

Perseus Energy 285 Vitek Home
Nezabezpečeno | 192.168.6.10/meter.xml?mid=302
Aktualizovat
Perseus Energy 285 Vitek Home
SAVE
Home
Monitoring
Meters
Variables
Conditions
Control
General
Services
Network
Security
Modem
System
Meters
Pracovna
Details
ID: 302
Name: Pracovna
Type: Mbus serial
Port: 0
State: Ok
Configuration
Enable
Name: Pracovna
Measure period: 300
Sec
Use Address: Primary
Primary Address: 13
Change address
Secondary Address: 24460215
Serial Baudrate: 2400
Serial Baudrate: Even
Mbus Meter Model: Inepro
Manual Read
Show Debug:
Variables
Add virtual variable
Scan variables

Modbus/RTU

The Modbus/RTU tab contains Modbus/RTU meters connected via the RS-485 interface.



The Modbus RTU Templates button opens a page of installed templates on the device.

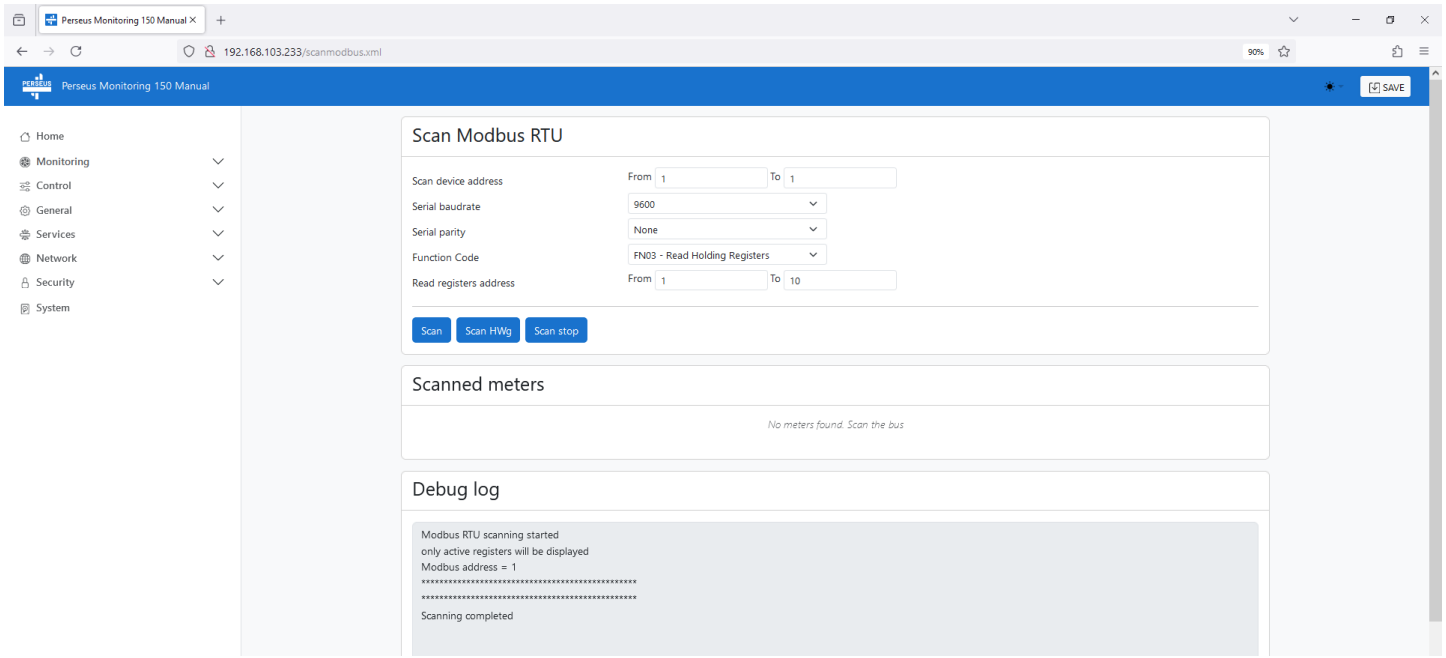


Here you can select meters and variables from the list of installed templates and use the Save button (bulk addition requires later editing of meters and entering their addresses), or add individual meters - you will be redirected to the meter edit page with a prompt to enter the meter address.

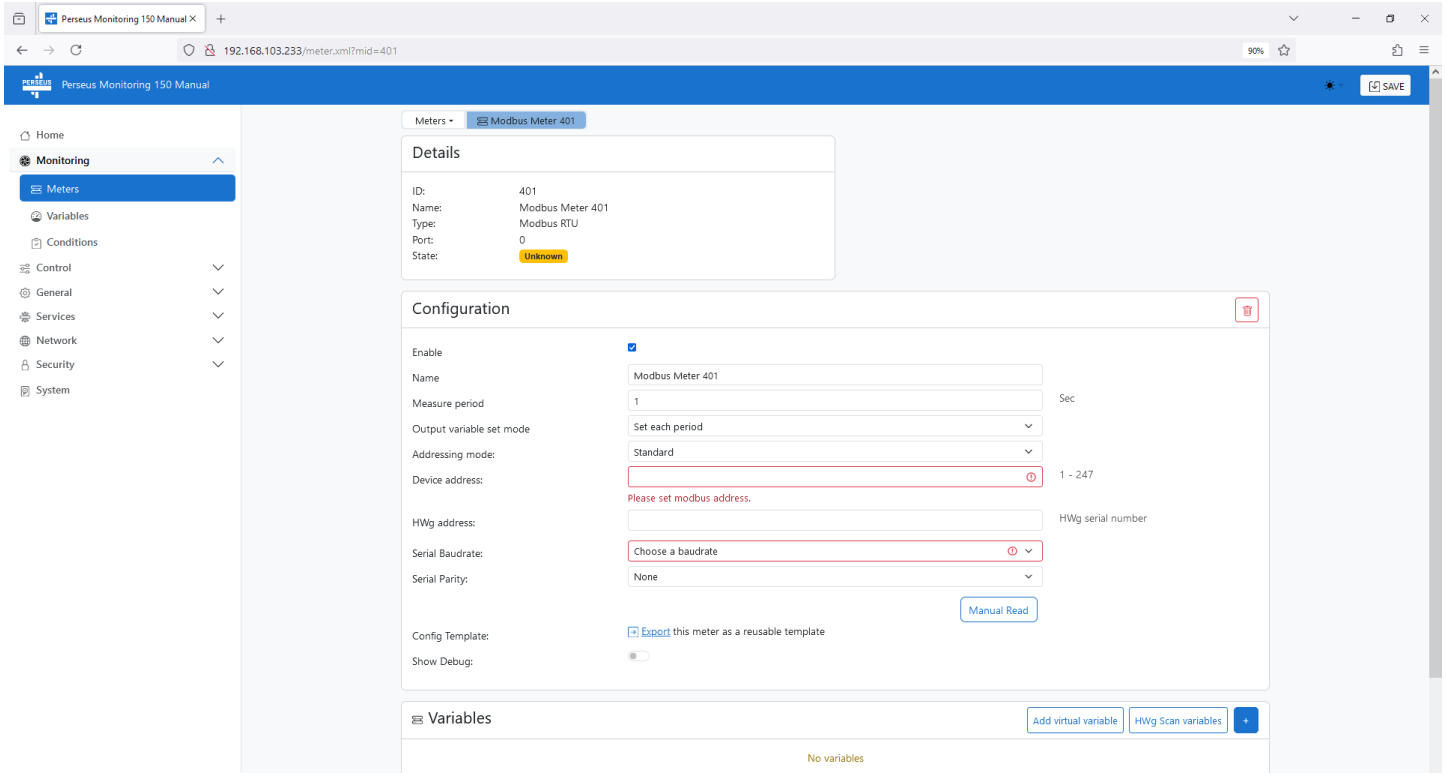
If a template for a specific meter is missing, it can be added on the General/Meter Templates page.

With the RS-485 bus, it is not currently possible to automatically search for and add a meter (there are no sensors yet that support this functionality). However, it is possible

to use the Modbus Scan button to scan the bus area. If the device encounters a valid search of a specific range, the registers will be shown:



It is also possible to create and add the meter and variables manually:



Details section

- **ID** – ID of the meter – see chapter Monitoring / Meters
- **Name** – The name of the meter. The name can be changed in the Configuration section
- **Type** – type of meter – see chapter Monitoring / Meters
- **Port** – The port number to which the meter is connected. For local or Ethernet sensors, this is an internal designation.
- **State** – meter status (Unknown/OK/Device Invalid)

Configuration section

- **Enable** – Turns the meter on or off
- **Name** – The name of the meter for easier orientation
- **Measure Period** – Reading period– the minimum period is 1s, but it may vary depending on the unit configuration, numbers and types of sensors in order to comply with the necessary timeouts
- **Addressing mode** - Switching the meter addressing type *Standard*/ *HWg Extended*. Standard is standard Modbus/RTU and uses the *Device Address*. *HWg Extended* is exclusively for HW group sensors and allows automatic loading of Variables and other options.
- **Device Address** – Modbus/RTU address in the range 1 - 247, which serves as a unique identifier of the meter on the bus. As a rule, the address is indicated on the meter label or in the manufacturer's documentation
- **HWg address** - Designed exclusively for HW group sensors and corresponds to the serial number. This address is unique and cannot be changed
- **Serial Baudrate –** Transmission rate of the meter on the bus
- **Serial Parity** – Data parity of the meter on the bus
- **Manual** Read – Button for manual reading of the meter
- **Config Template** – Export settings as a template (exports meter and variable settings).

Variables section

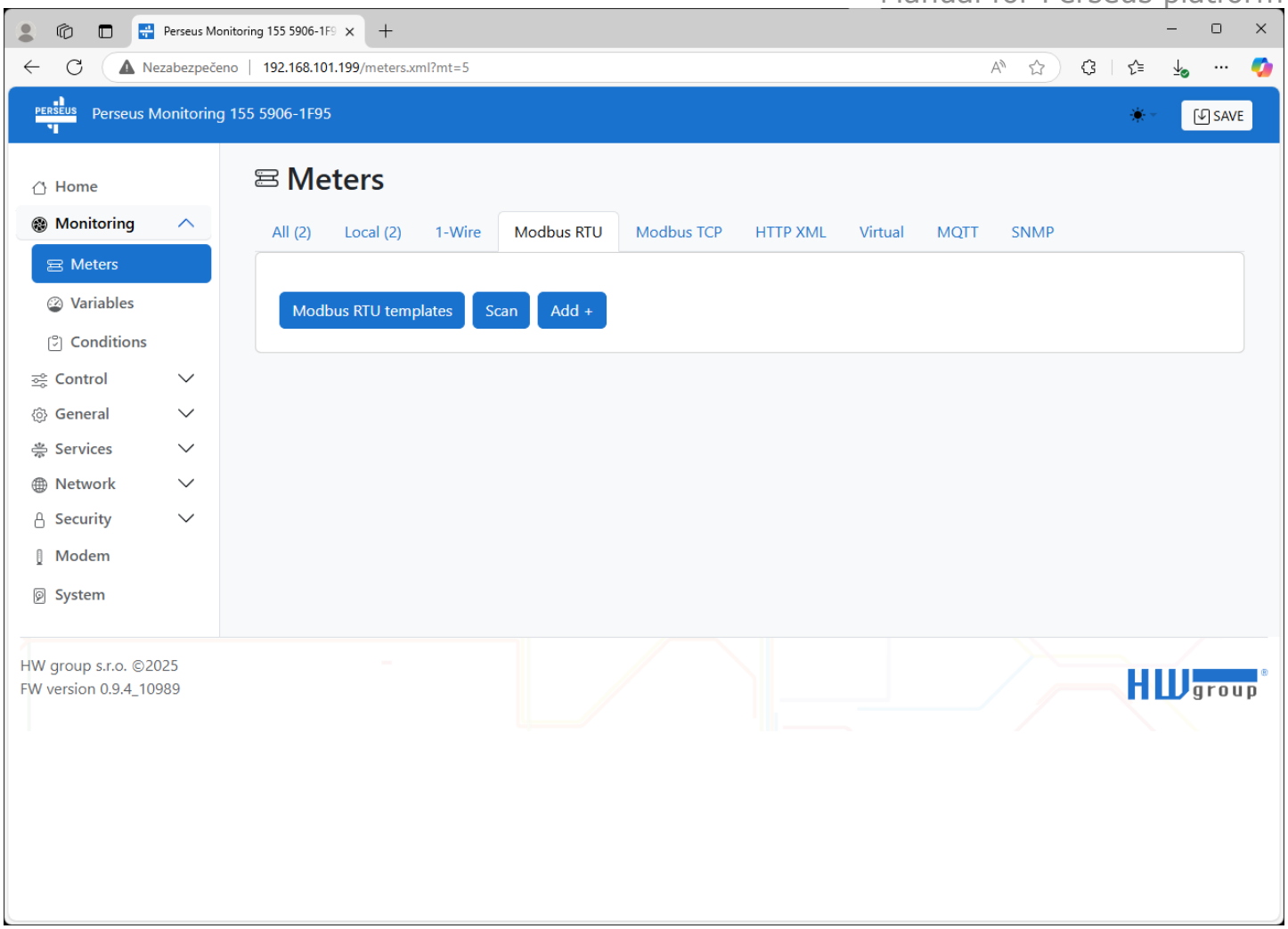
A well-arranged table of connected variables, their states and values. Depending on the type of interface, it is possible to detect meter variables in this section (**Scan variables** button) or manually add them (**+ button**).

- **ID** – ID of the variable
- **Name** – The name of the variable
- **State** – The current state of the variable
- **Value** – The current value
- **Unit**
- **Options** – Options for working with the variable (Delete/Edit)

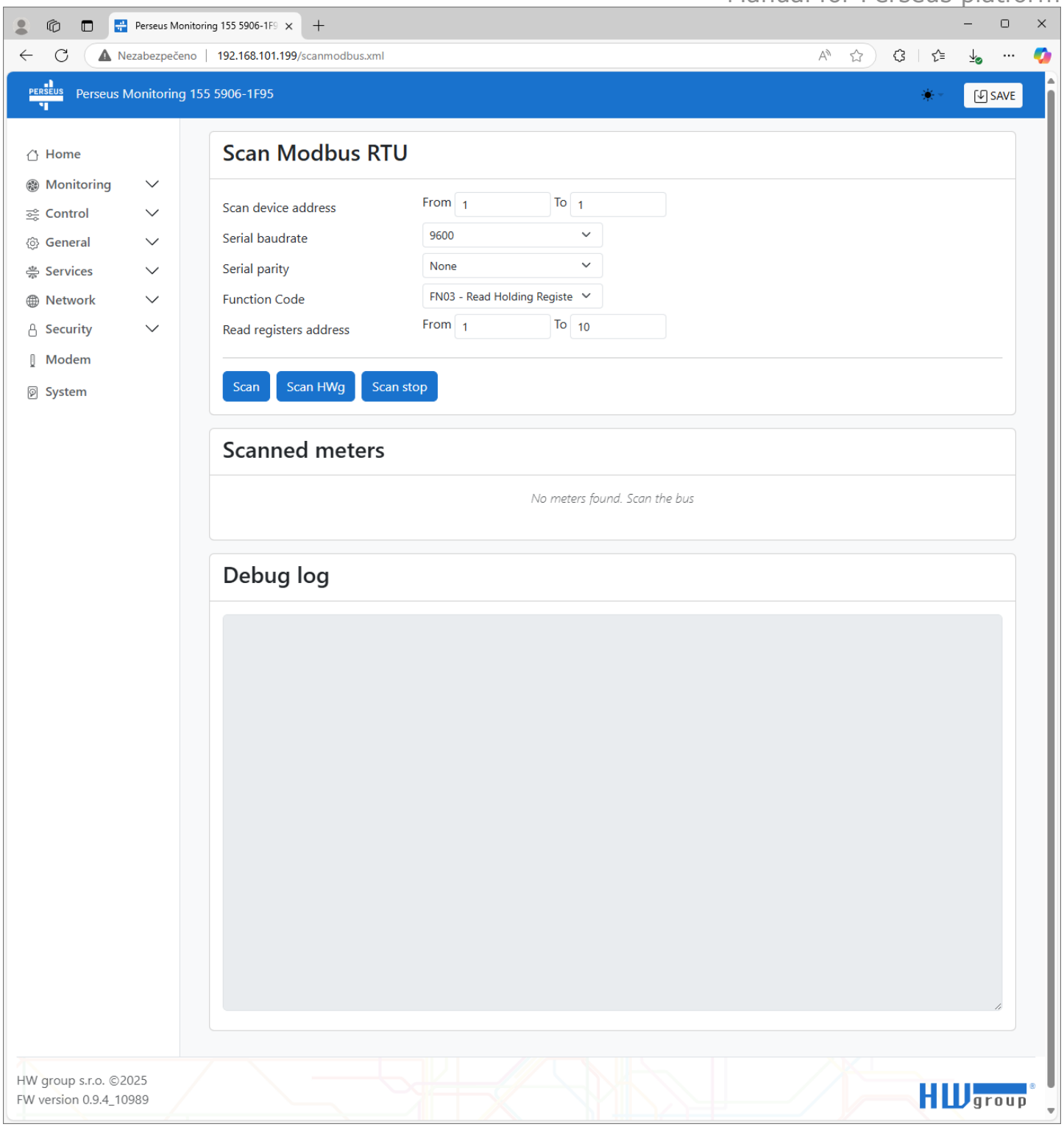
Modbus/RTU HWG

The sensors of HW group s.r.o. equipped with Modbus/RTU protocol have a special function for finding them on the RS-485 bus + adding variables without the need of template. This is a non-standard extension of the Modbus/RTU protocol designed exclusively for the needs of HW group s.r.o. devices and sensors. The only condition for using this function is that there **MUST NOT** be a sensor with ID 240 on the bus!

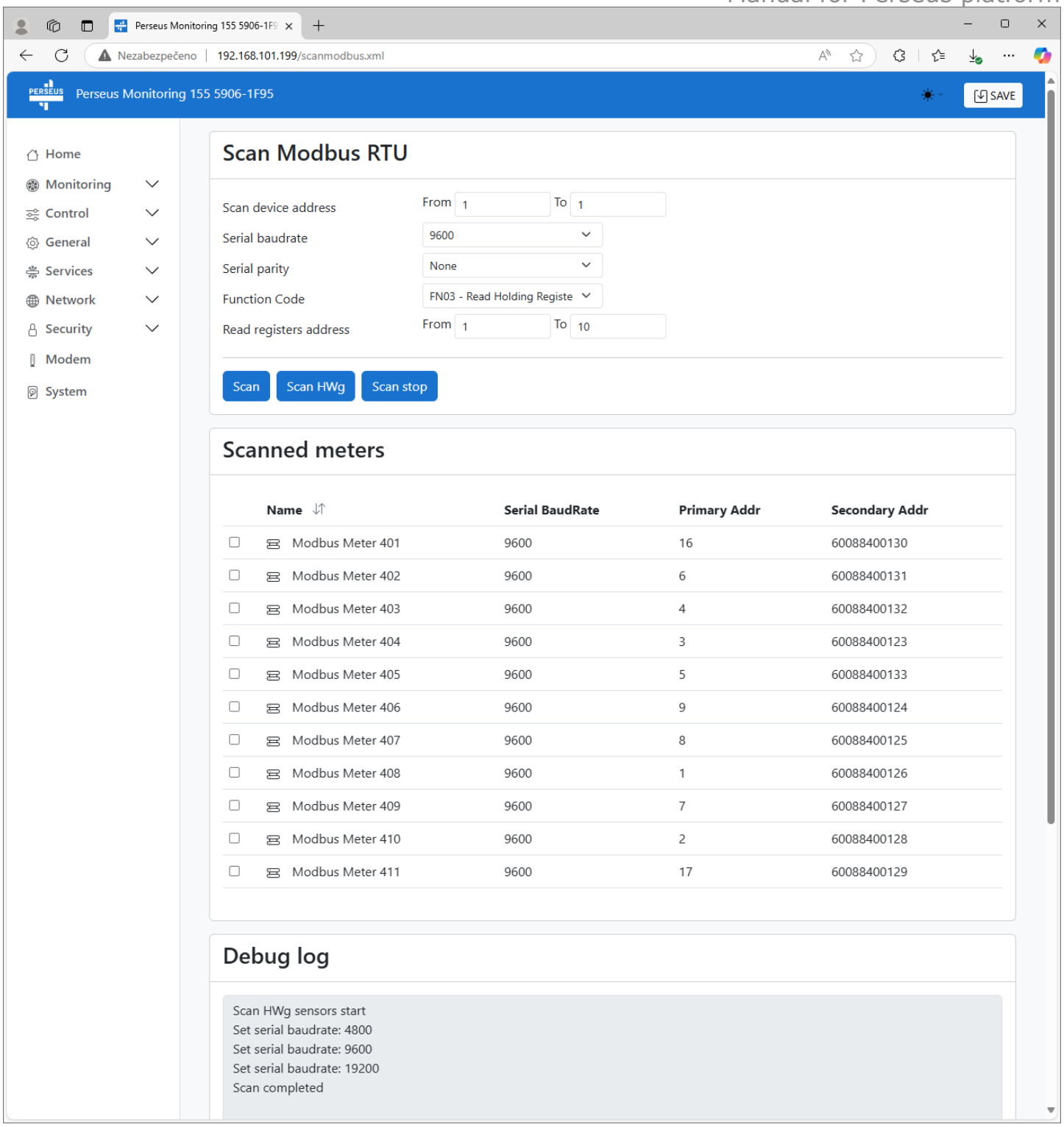
To add Modbus/RTU sensors from HW group, just select the Scan button on the Meters => Modbus RTU page



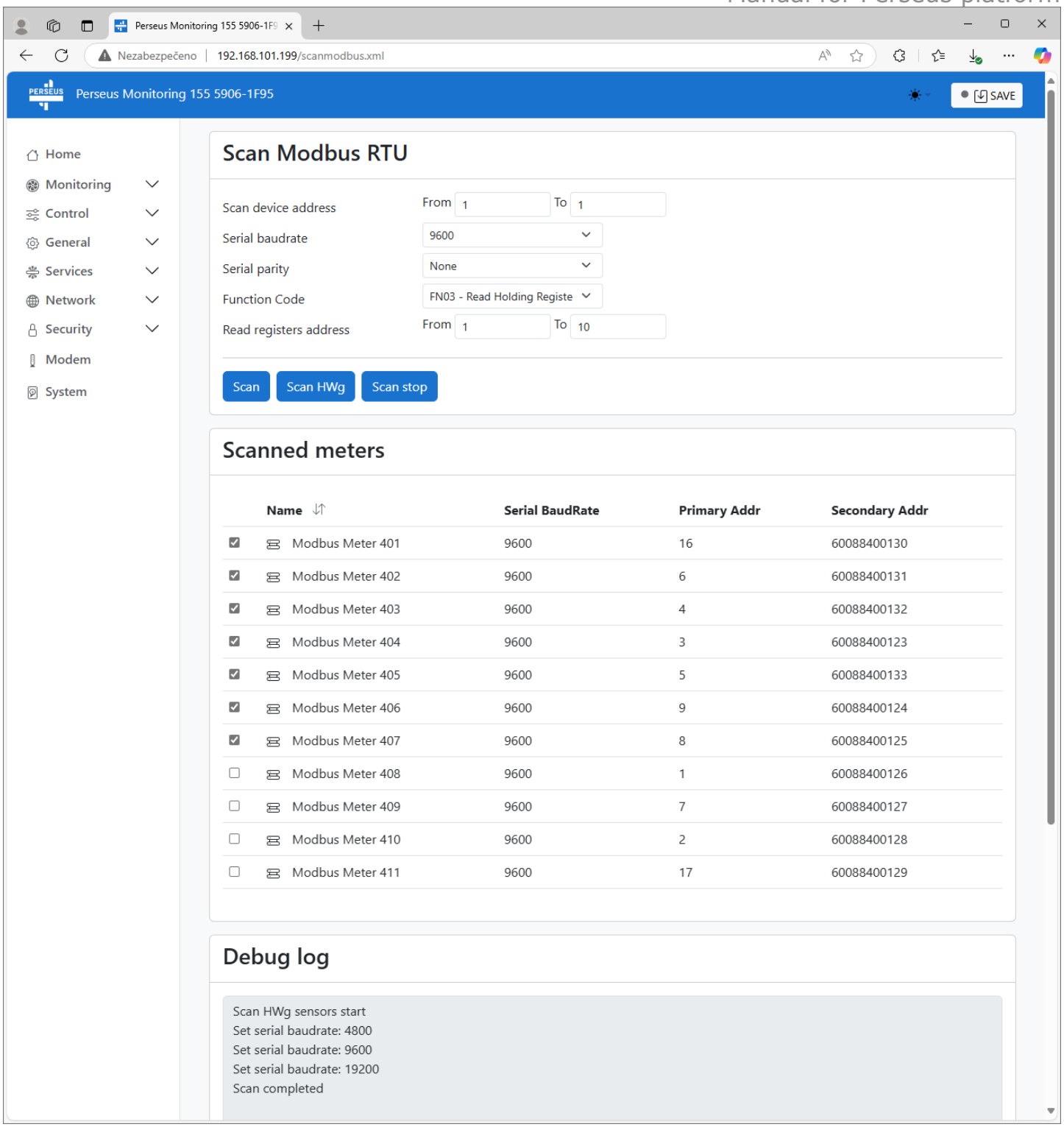
And here on the search dialog press the option Scan HWg



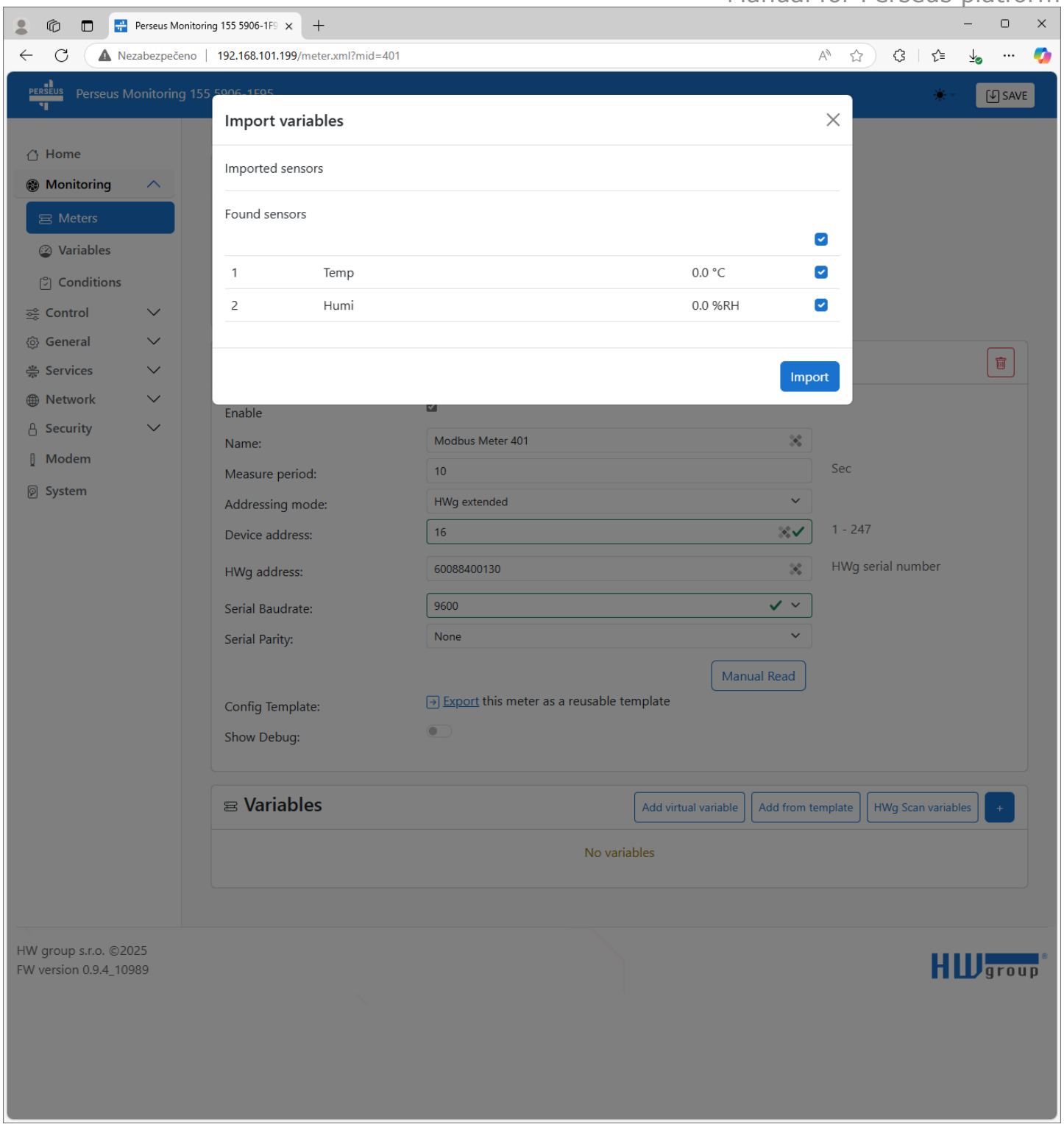
Perseus searches for HW group sensors connected to the bus at 4800, 9600 and 19200 baud rates regardless of the scan parameter settings in the form:



In the left column you can select which Meters you want to add to the system. Save is done by clicking Save.



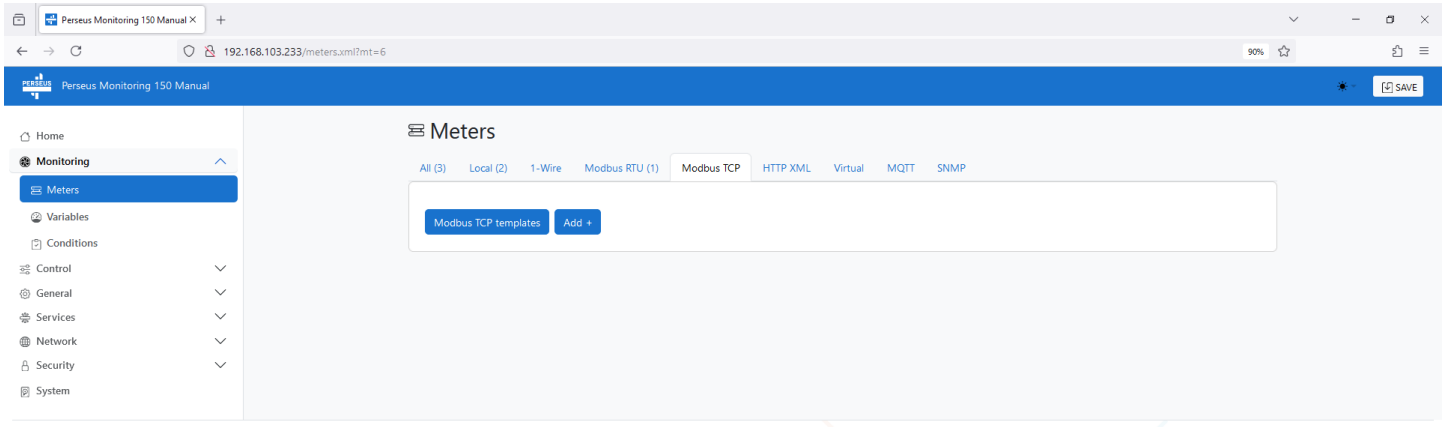
This saves the individual Meters and the Variables still need to be detected. This is done classically in Meter editing, using the Hwg Scan variables button.



The Import button will add the selected variables.

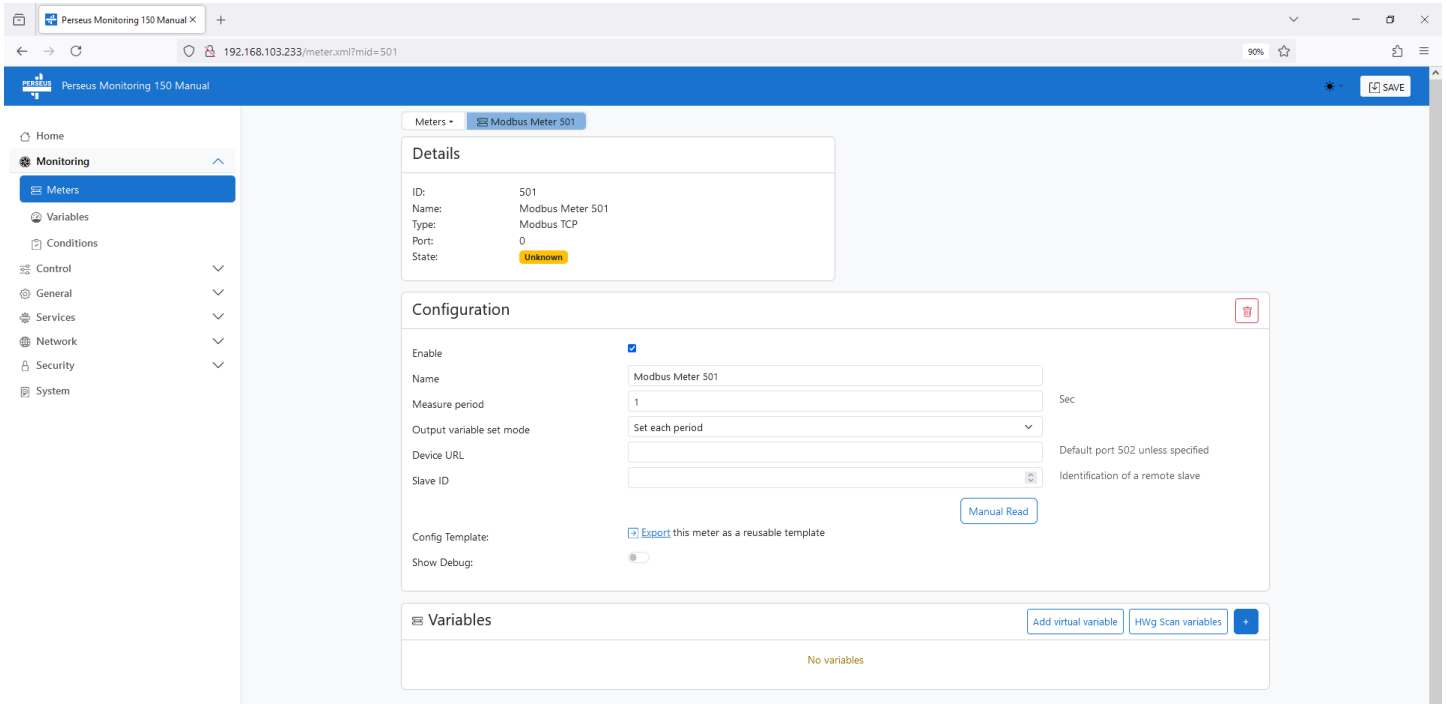
Modbus/TCP

The Modbus/TCP tab contains meters that are connected via the Ethernet interface and using the Modbus/TCP protocol (e.g. Poseidon2, Damocles2 and others).



Modbus TCP templates allow you to browse through the available templates.

The Add + button will allow you to add another meter:



Details section

- **ID** – ID of the meter – see chapter Monitoring / Meters
- **Name** – The name of the meter. The name can be changed in the Configuration section
- **Type** – type of meter – see chapter Monitoring / Meters
- **Port** – The port number to which the meter is connected. For local or Ethernet sensors, this is an internal designation.
- **State** – meter status (Unknown/OK/Device Invalid)

Configuration section

- **Enable** – Turns the meter on or off

- **Name** – The name of the meter for easier orientation
- **Measure period** – Reading period – the minimum period is 1s, but it may vary depending on the unit configuration, numbers and types of sensors in order to comply with the necessary timeouts
- **Output variable set mode –** Option for sending the value in set measure period or when the value has changed
- **Device URL** – IP address and port of the device to be read

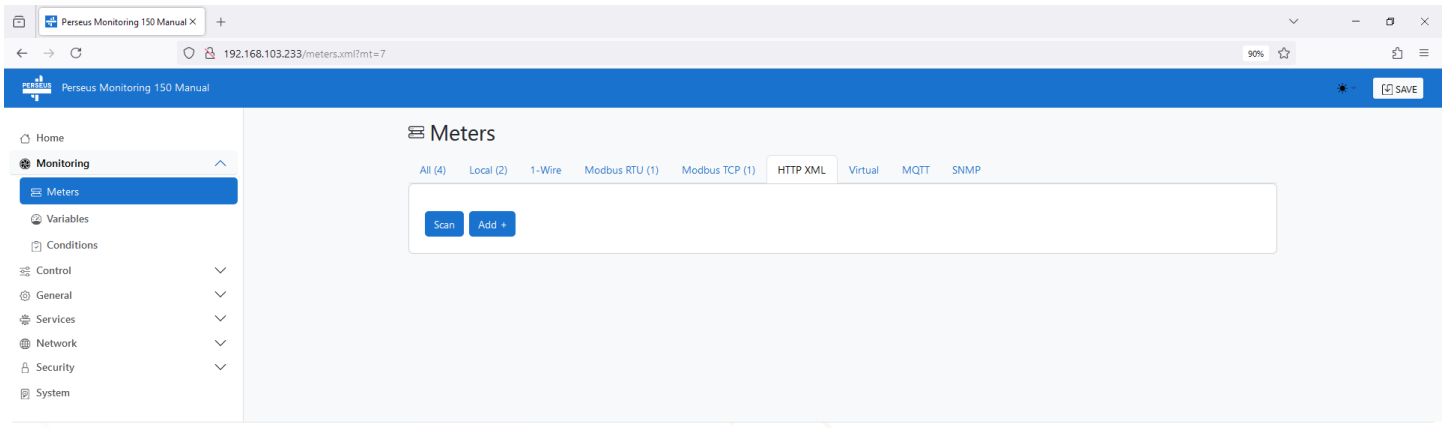
Variables section

A well-arranged table of connected variables, their states and values. Depending on the type of interface, it is possible to detect meter variables in this section (**Scan Variables** button) or manually add them (**+ button**).

- **ID** – ID of the variable
- **Name** – The name of the variable
- **State** – The current state of the variable
- **Value** – The current value
- **Unit**
- **Options** – Options for working with the variable (Delete/Edit)

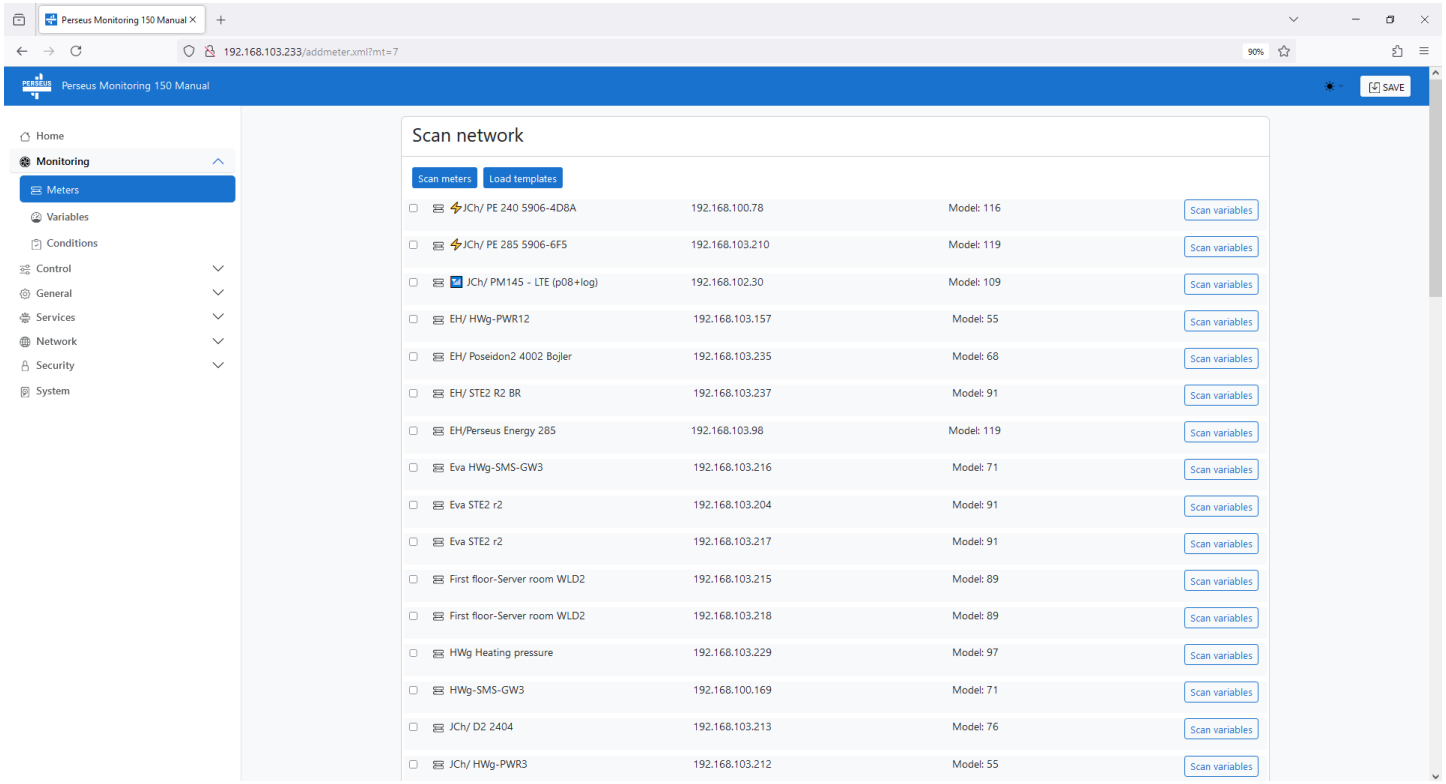
HTTP XML

Network meters allow you to connect remote HW group devices via XML.

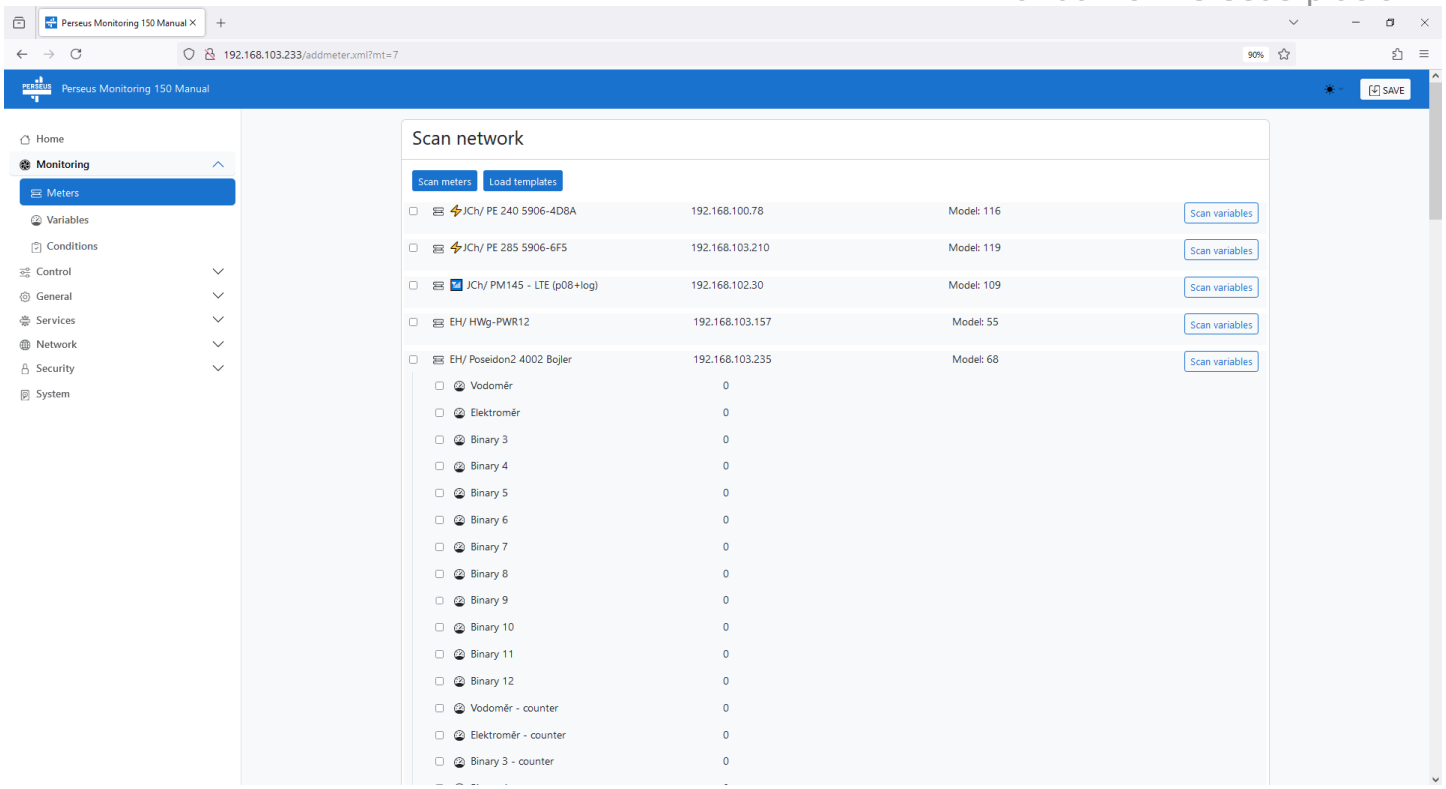


Automatic search for HW group devices

The *Network Scan* button scans the local network using the HWg search (UDP broadcast) function and lists all found devices.



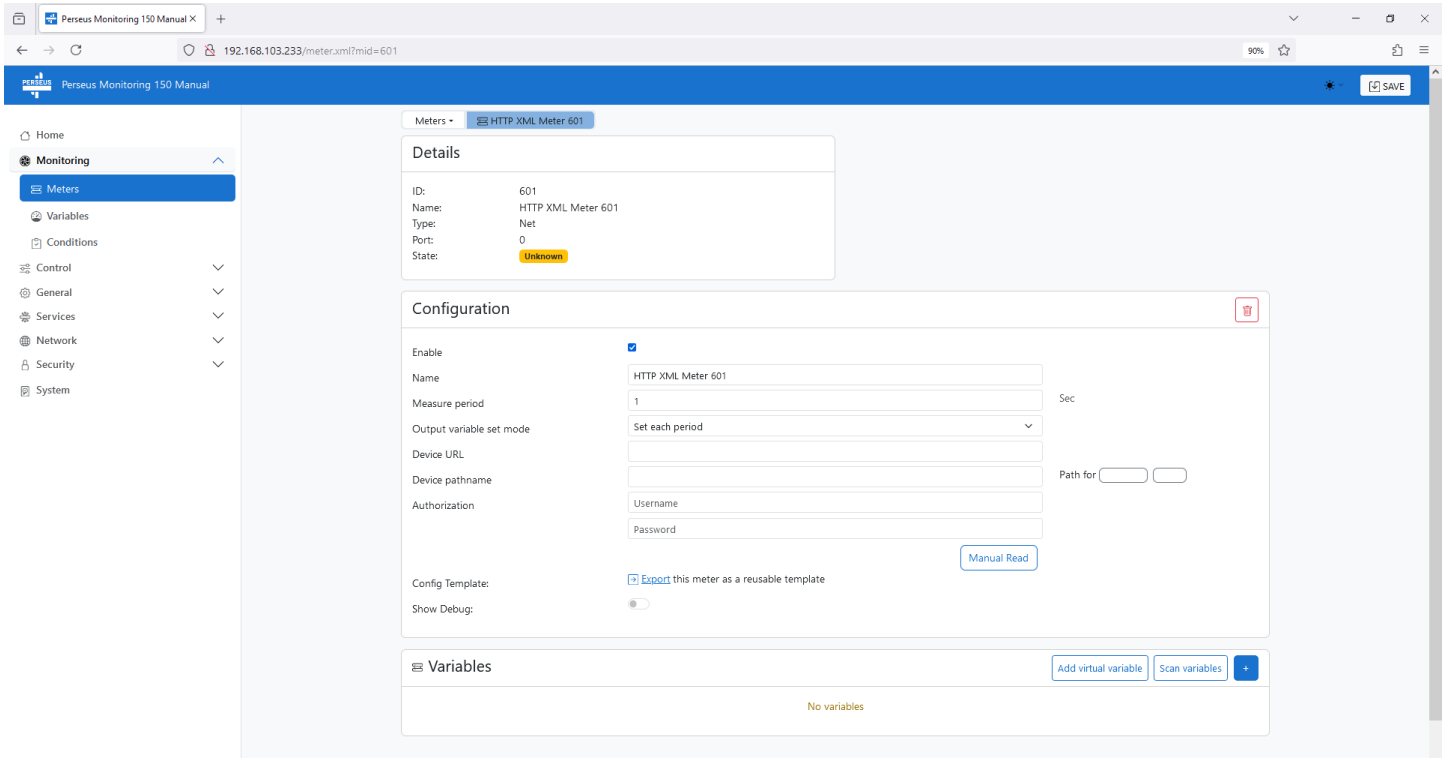
Here you can select Meters and confirm the selection with the *Save* button (bulk addition requiring later editing of meters and scanning of variables), or add individual meters with the *Scan variables button* and select specific variables:



Again, the selected items must be saved by clicking the *Save* button.

Manually adding HW group devices

Network: Add allows you to add another meter



Details section

- **ID** – ID of the meter – see chapter Monitoring / Meters
- **Name** – The name of the meter. The name can be changed in the Configuration section

- **Type** – type of meter – see chapter Monitoring / Meters
- **Port** – The port number to which the meter is connected. For local or Ethernet sensors, this is an internal designation.
- **State** – meter status (Unknown/OK/Device Invalid)

Configuration section

- **Enable** – Turns the meter on or off
- **Name** – The name of the meter for easier orientation
- **Measure period** – Reading period – the minimum period is 1s, but it may vary depending on the unit configuration, numbers and types of sensors in order to comply with the necessary timeouts
- **Device URL** – IP address and port of the device to be read
- **Device pathname** – the path to the downloaded file. It usually values.xml
- **Authorization** – name and password for authorization to the remote device – only if the device is protected by the authorization

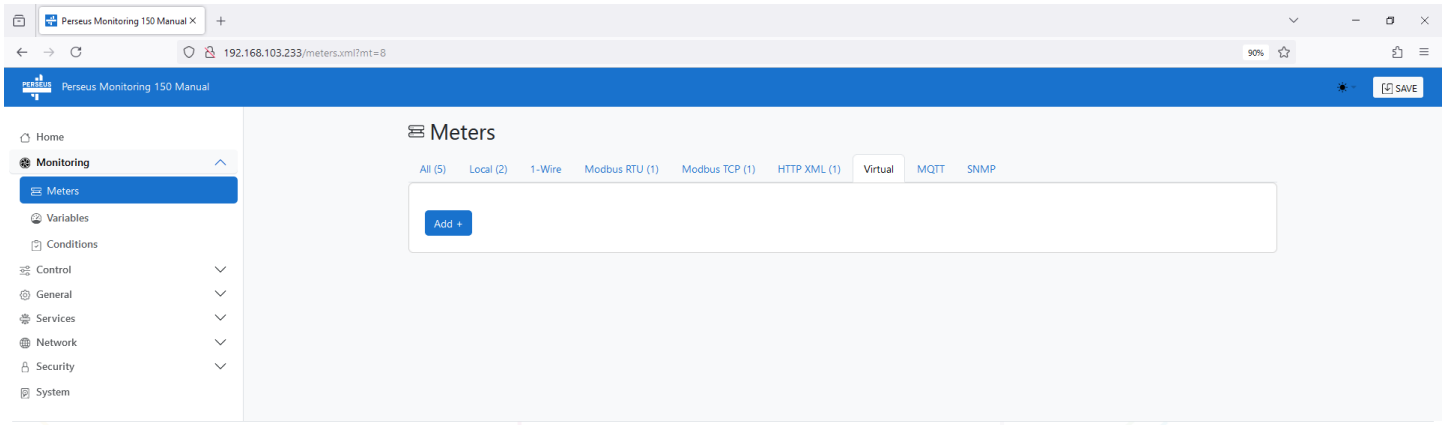
Section Variables

A well-arranged table of connected variables, their states and values. Depending on the type of interface, it is possible to detect meter variables in this section (**Scan Variables** button) or manually add them (**+ button**).

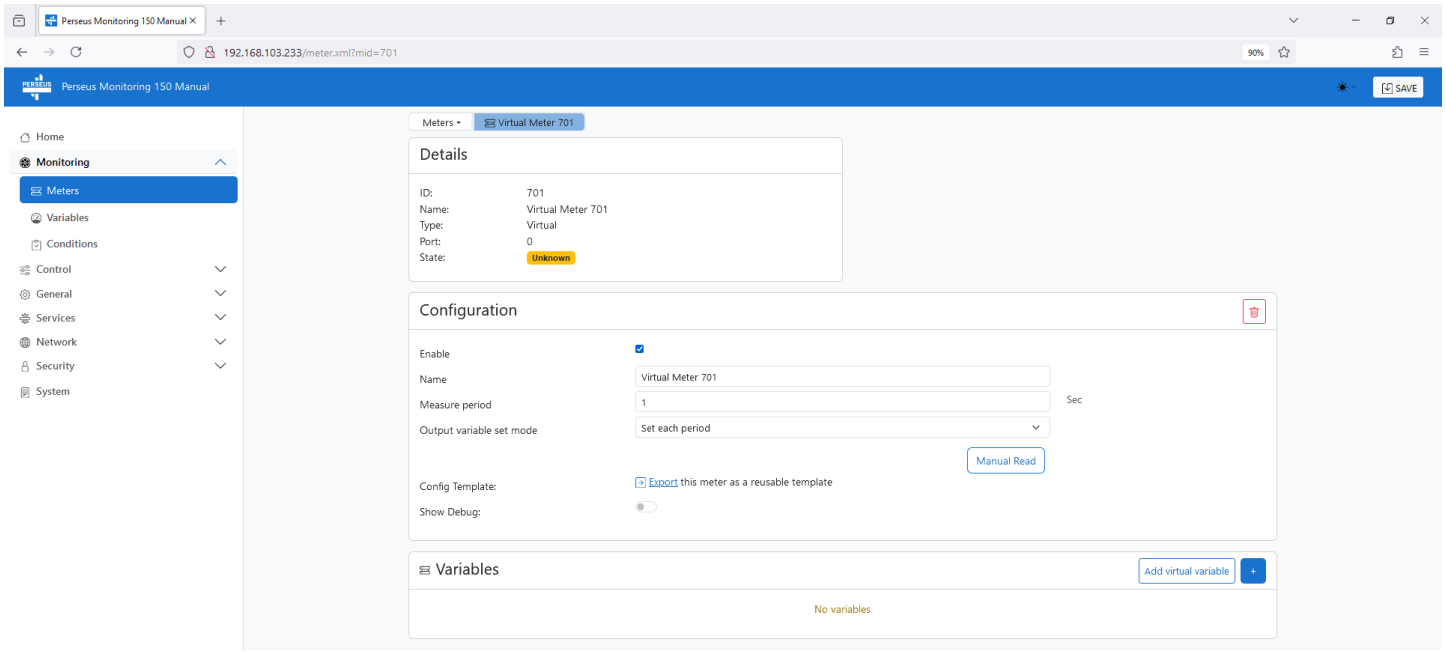
- **ID** – ID of the variable
- **Name** – The name of the variable
- **State** – The current state of the variable
- **Value** – The current value
- **Unit**
- **Options** – Options for working with the variable (Delete/Edit)

Virtual

The Virtual tab is used to create and manage virtual meters. Virtual meters are used to associate virtual variables, such as recalculated values and quantities, results of logical operations, and so on.



Use the **Add+** button to add a new virtual meter.



Details section

- **ID** – ID of the meter – see chapter Monitoring / Meters
- **Name** – The name of the meter. The name can be changed in the Configuration section
- **Type** – type of meter – see chapter Monitoring / Meters
- **Port** – The port number to which the meter is connected. For local or Ethernet sensors, this is an internal designation.
- **State** – meter status (Unknown/OK/Device Invalid)

Configuration section

- **Enable** – Turns the meter on or off
- **Name** – The name of the meter for easier orientation

- **Measure period** – Reading period – the minimum period is 1s, but it may vary depending on the unit configuration, numbers and types of sensors in order to comply with the necessary timeouts
- **Output variable set mode –** Option for sending the value in set measure period or when the value has changed

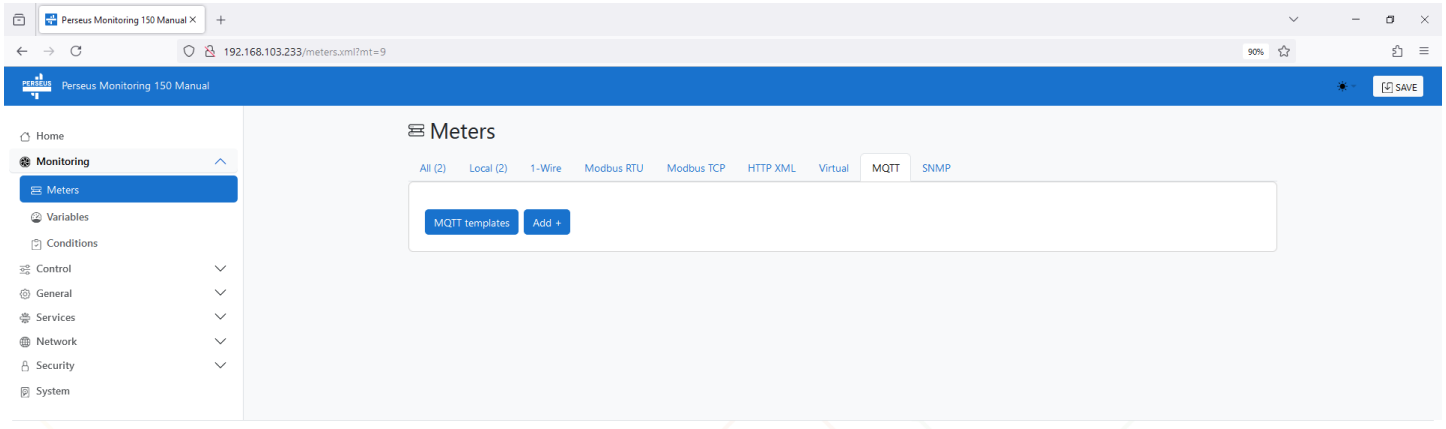
Variables section

A well-arranged table of connected variables, their states and values. Variables can only be added using the **+ button**.

- **ID** – ID of the variable
- **Name** – The name of the variable
- **State** – The current state of the variable
- **Value** – The current value
- **Unit**
- **Options** – Options for working with the variable (Delete/Edit)

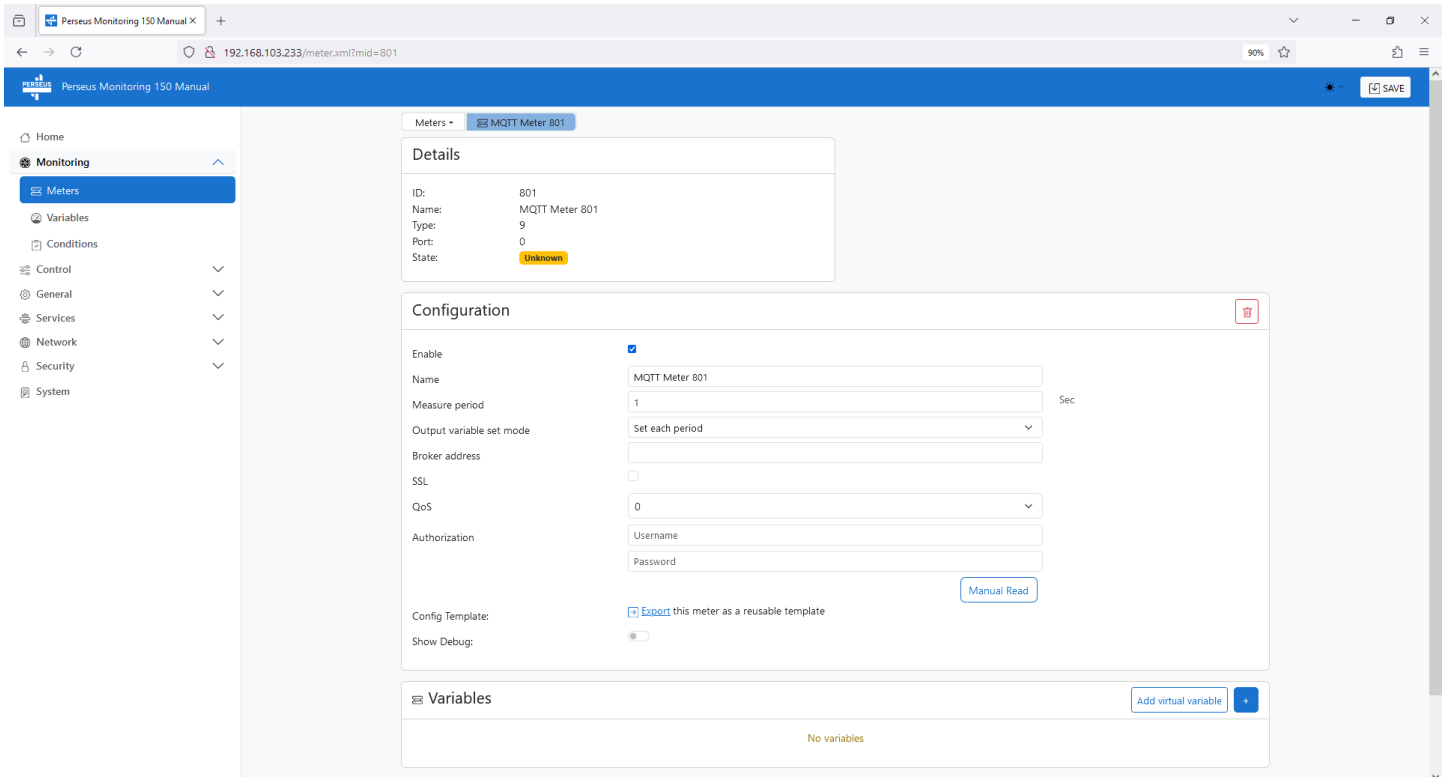
MQTT

The MQTT tab is used to create and manage meters connected via the MQTT protocol. Perseus then behaves like a standard MQTT subscriber.



MQTT templates are available with the MQTT templates button.

A new meter can be added with the **Add+** button.



Details section

- **ID** – ID of the meter – see chapter Monitoring / Meters
- **Name** – The name of the meter. The name can be changed in the Configuration section
- **Type** – type of meter – see chapter Monitoring / Meters
- **Port** – The port number to which the meter is connected. For local or Ethernet sensors, this is an internal designation.
- **State** – meter status (Unknown/OK/Device Invalid)

Configuration section

- **Enable** – Turns the meter on or off

- **Name** – The name of the meter for easier orientation
- **Measure period** – Reading period – the minimum period is 1s, but it may vary depending on the unit configuration, numbers and types of sensors in order to comply with the necessary timeouts
- **Broker address** – set the IP address and MQTT port of the broker to be read
- **Authorization** – name and password for authorization to the remote broker
- **Client ID** – ID of Perseus in MQTT network.
- **QoS** – Quality of Service (Perseus in publisher mode have always "0")
- **Retained messages** – MQTT Flag that tells MQTT broker to keep last value for any new subscriber, so it doesn't wait for new published value but instead the new subscriber receives last known value.
- **Publish Period** – the period of sending data to the MQTT broker.
- **Common Topic prefix** – prefix for MQTT Topic (address).

Debug log section

Used to display the debug dump

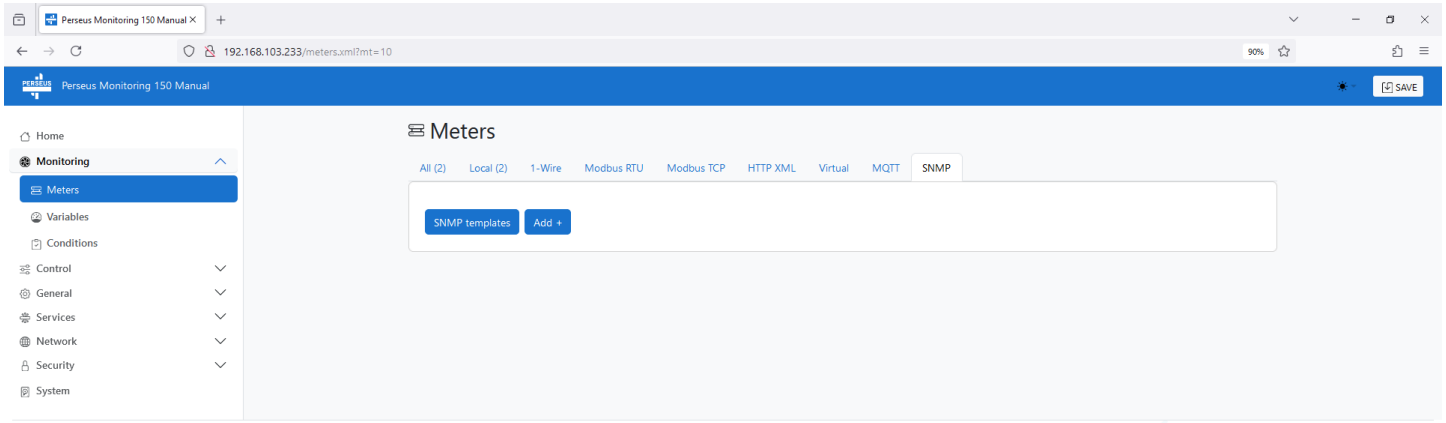
Variables section

A well-arranged table of connected variables, their states and values. Depending on the type of interface, it is possible to detect meter variables in this section (**Scan Variables** button) or manually add them (**+ button**).

- **ID** – ID of the variable
- **Name** – The name of the variable
- **State** – The current state of the variable
- **Value** – The current value
- **Unit**
- **Options** – Options for working with the variable (Delete/Edit)

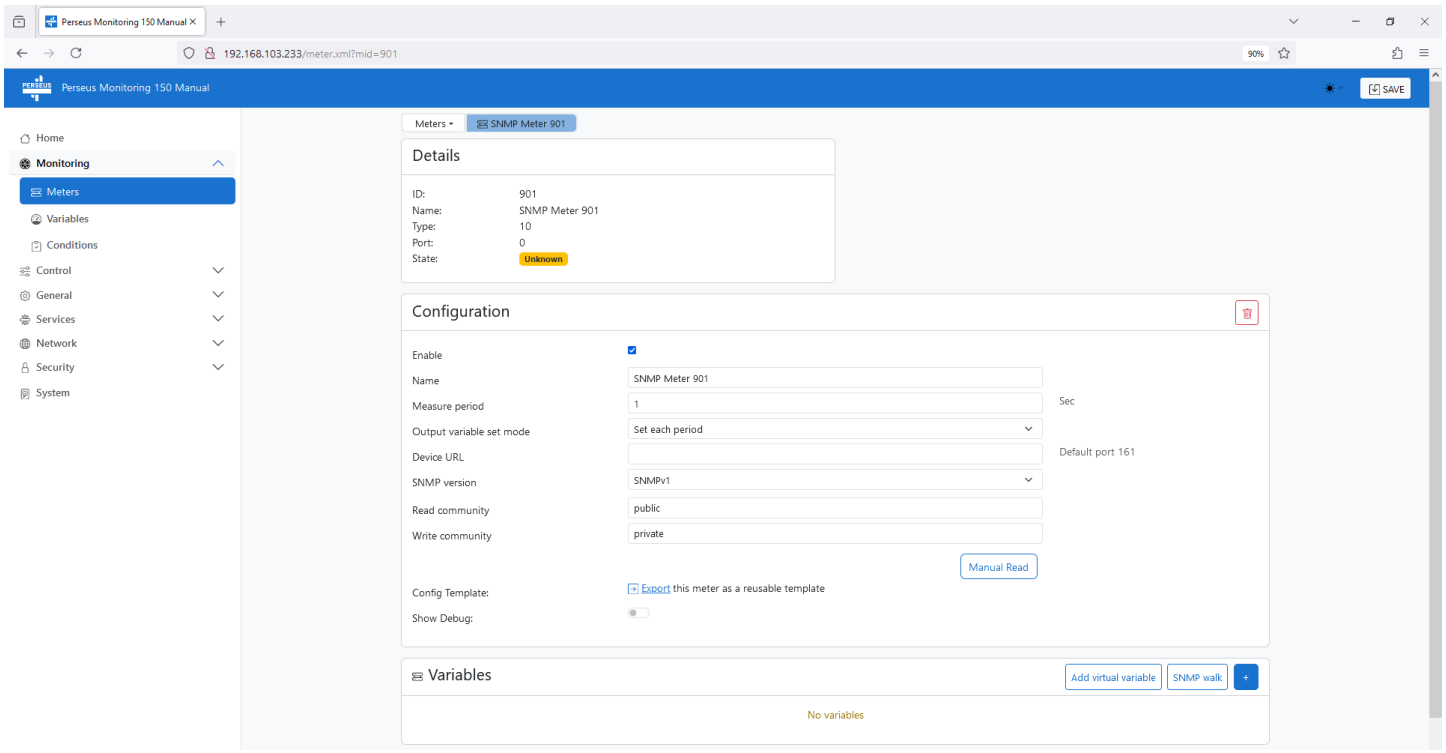
SNMP

SNMP meters allow you to connect any SNMP agents (SNMPv1 and SNMPv3), and read specific SNMP OIDs from them and work with the resulting value as with a standard sensor. The Perseus family is also equipped with the ability to scroll through the available values (SNMP walk).



SNMP templates are available with the SNMP templates button.

A new meter can be added with the **Add+** button.



Details section

- **ID** – ID of the meter – see chapter Monitoring / Meters
- **Name** – The name of the meter. The name can be changed in the Configuration section
- **Type** – type of meter – see chapter Monitoring / Meters
- **Port** – The port number to which the meter is connected. For local or Ethernet sensors, this is an internal designation.
- **State** – meter status (Unknown/OK/Device Invalid)

Configuration section

- **Enable** – Turns the meter on or off

- **Name** – The name of the meter for easier orientation
- **Measure period** – Perida reading – the minimum period is 1s, but it may vary depending on the unit configuration, numbers and types of sensors in order to comply with the necessary timeouts
- **Output variable set mode –** Option for sending the value in set measure period or when the value has changed
- **Device URL** – IP address and port of the device to be read
- **SNMP version** – use the SNMP protocol to be used for communication
- **Community –** Community for authorization after SNMP

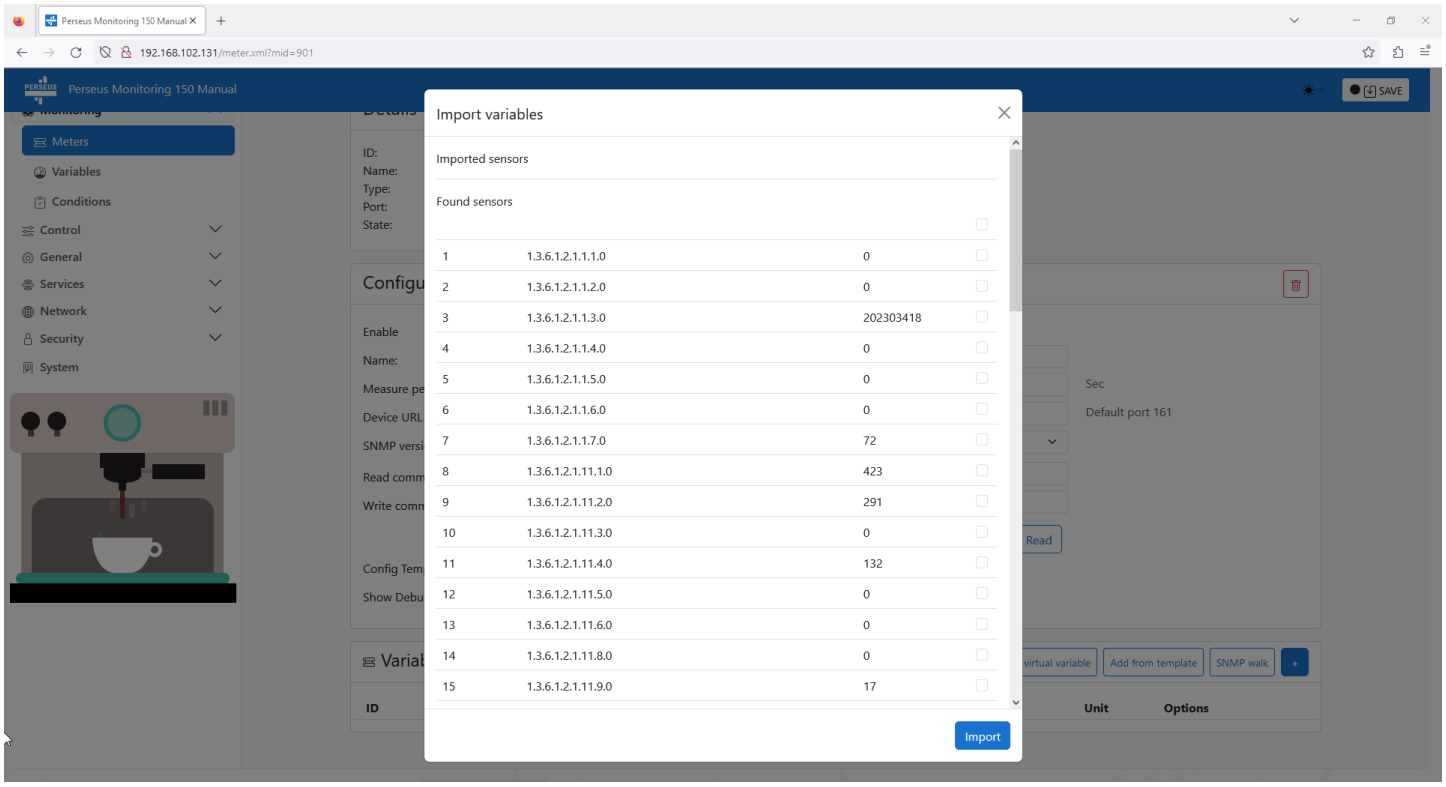
Debug log section

Used to display the debug dump

Variables section

A well-arranged table of connected variables, their states and values. Here it is possible to detect meter variables (**Scan Variables** button) or manually add them (**+ button**).

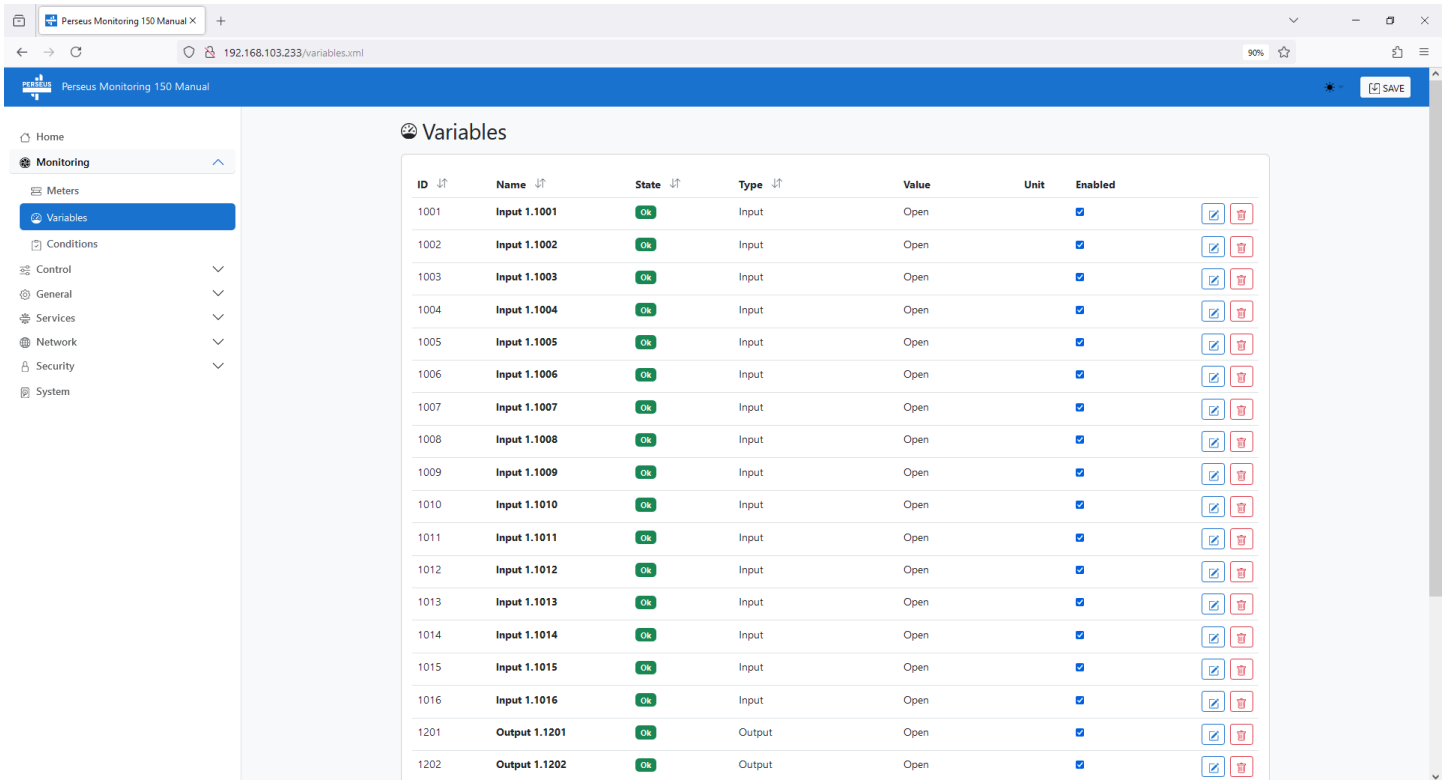
- **ID** – ID of the variable
- **Name** – The name of the variable
- **State** – The current state of the variable
- **Value** – The current value
- **Unit**
- **Options** – Options for working with the variable (Delete/Edit)



To scan the OID, it is possible to select individual values or select them in bulk and then confirm the selection with the Import button.

Menu Monitoring / Variables

This page is used to view, delete and sort monitored variables.



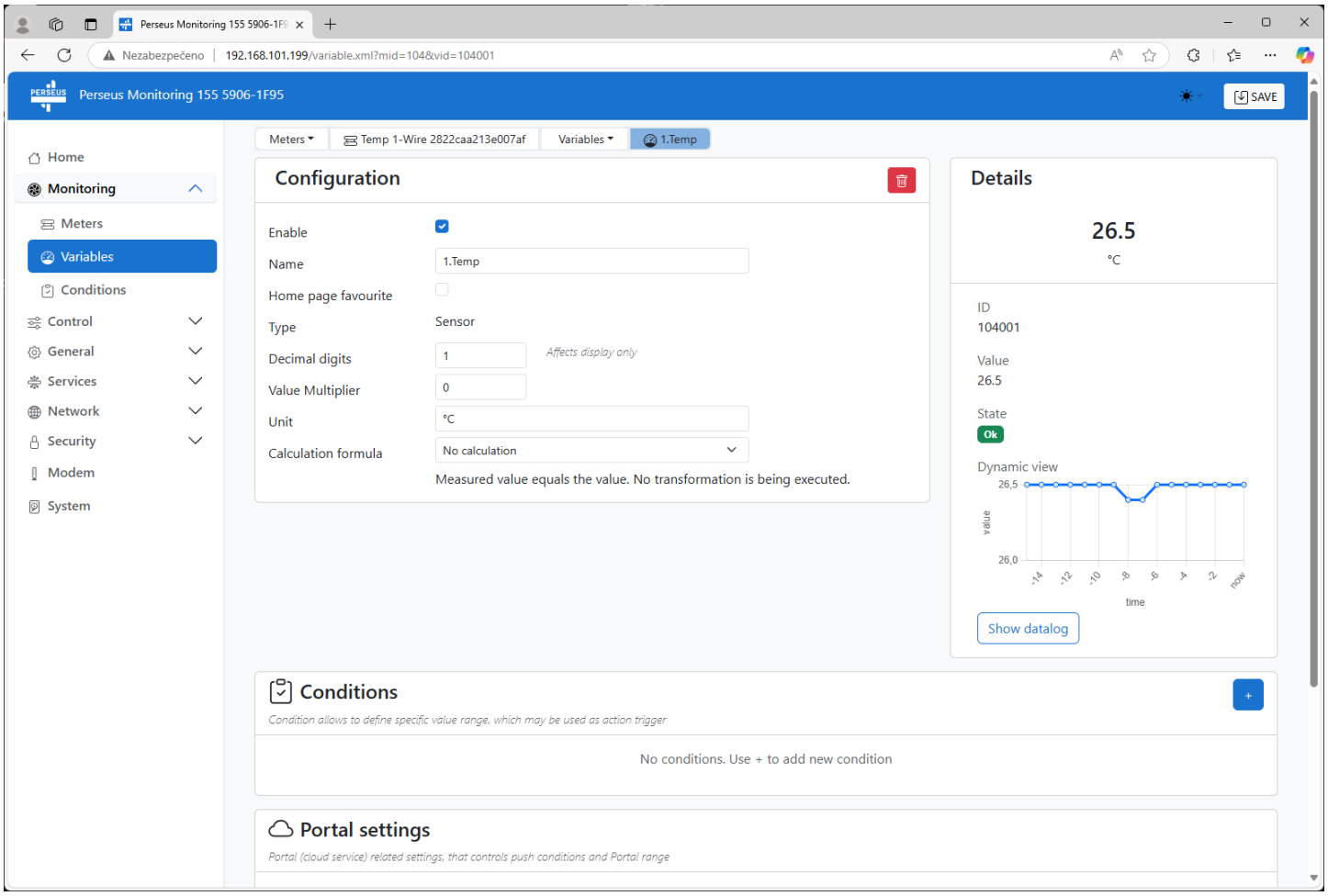
The information is presented in a clear table containing:

- **ID** – ID of the variable (variable ordering can be changed)
- **Name** – variable name (variable sorting can be changed)
- **State** – Current state of the variable (variable sorting can be changed)
- **Type** – variable type (Input/Output/Sensor, etc.) (variable ordering can be changed)
- **Value** – Current value (it is possible to change the sorting of values)
- **Unit** – Unit
- **Enabled** – options to enable or disable work with variables
- **Icons for deleting and editing variables**

For a description of variable states, see chapter Meter states, variables and conditions.

Variable

The page includes the ability to edit the variable.



Configuration section

- **Enable** – enable the variable to be used. If unchecked, Perseus retains the ID of the variable, but does not read, evaluate, or perform subsequent operations linked to it
- **Name** – The name of the variable for easier orientation
- **Home page favourite** – Enables priority display on the homepage (Favourites section)
- **Type** – The type of the variable. See chapter Types of Variables of the Perseus family
- **Decimal digits** – the number of decimal places. Affects the value for calculations and displays.
- **Unit** – The unit of the variable. For inputs and outputs, but also in other cases, it can be blank – the variable is dimensionless
- **Calculation formula** – Possibilities of variable conversion
  - **No calculation** – The displayed value is equal to the measured value. No recalculation is performed
  - **Function** – simple calculation by formula
  - **Calcultion by LUA script** – Recalculation using a function defined by a LUA script – For more information, see chapter Scripts

Depending on the interface, the Configuration section may have certain specifics:

Modbus specific

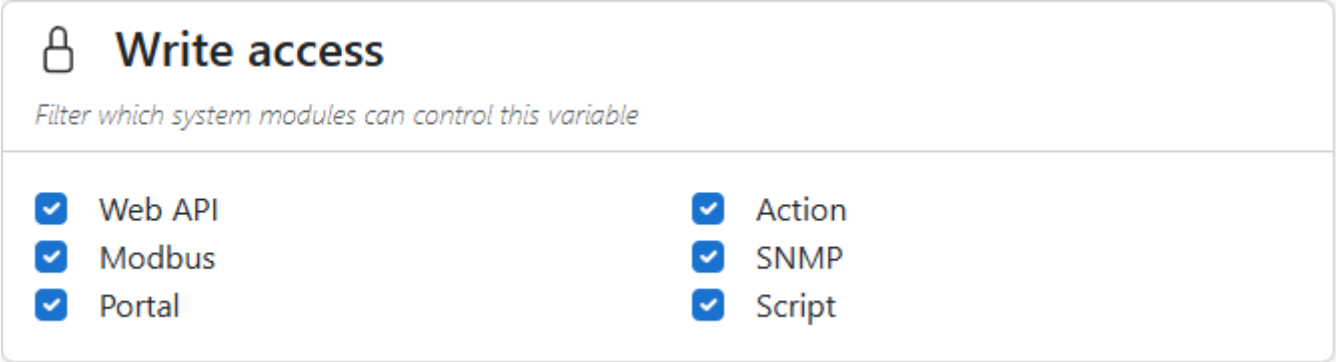
- **Function Code** – define the Modbus function used to work with the variable – See below
- **Modbus register** – set the Modbus register number. The absolute address and the Modicon address are automatically displayed next to the register number (e.g. Absolute Address: 5 [0x5], Modicon Register: 40006)
- **Data type** – define the data type of the variable (INT/UNIT/FLOAT/BIT) and bit length (8/16/32/64 bits)
- **Swap bytes** – Usually, when sending multibyte variables, the higher byte is sent first and then the lower one, but this is not always the rule. *Swap bytes* allow you to swap the order of byte values

SNMP specific

- **OID** – Entire SNMP OID (Object Identifier) of the required variable
- **Data type** – define the data type of the variable (Integer)

Write Access Section (Only variables of the output type)

The section serves as a filter for which system modules can control this variable



These settings allow you to choose which services can change the output value. By default, all are selected, so the value can be changed both by local condition using Action and by user using WWW interface or via portal, as well as 3rd party SW using API. This can result in the user changing the value but the Action changing it back. If you want to avoid this, select only those interfaces that are allowed to affect the value.

Value Names Section

The Value Names section is used to define text values for status sensors. For example, Open/Close, Flooded/Torn/OK and so on. The value is always determined by a number for the needs of machine readability, but it can also be named for visualization purposes. Value Names can also be defined for continuous sensors, but this option does not make much sense.

Conditions Section

The Conditions section serves as an overview and creation of conditions (safe ranges) for further processing. On this page, once added, you can only define:

- **ID** – ID of the condition within the given variable
- **Name** – Condition name – the entire condition ID is automatically added to the name. The name can be changed in the edit of the condition
- **State** – Displays the status of the condition after evaluation. For a description of variable states, see chapter Meter states, variables and conditions

- **Min** – Minimum allowed value
- **Max** – Maximum allowed value
- **Hysteresis** – Area of inertia
- **Delay** – Minimal delay for activation of the AlarmHi or AlarmLo state
- **Enabled** – Enable or disable the condition for the given variable
- **Icons for editing and deleting**

For more information about individual parameters, see the chapter Editing Conditions.

Each condition has its status given by the evaluation of the currently obtained value (after recalculation). It is not uncommon for one condition to be in the Alarm state while another is in the OK state. On the Homepage, the variable shows the number of Condition in the Alarm state.

A new action can be added by pressing the + button in the Conditions section, and existing ones can be edited by clicking on their name or on the appropriate edit icon.

If a condition to which actions are linked is deleted, the actions will remain defined in the system and are not just executed (unless another condition is linked to them).

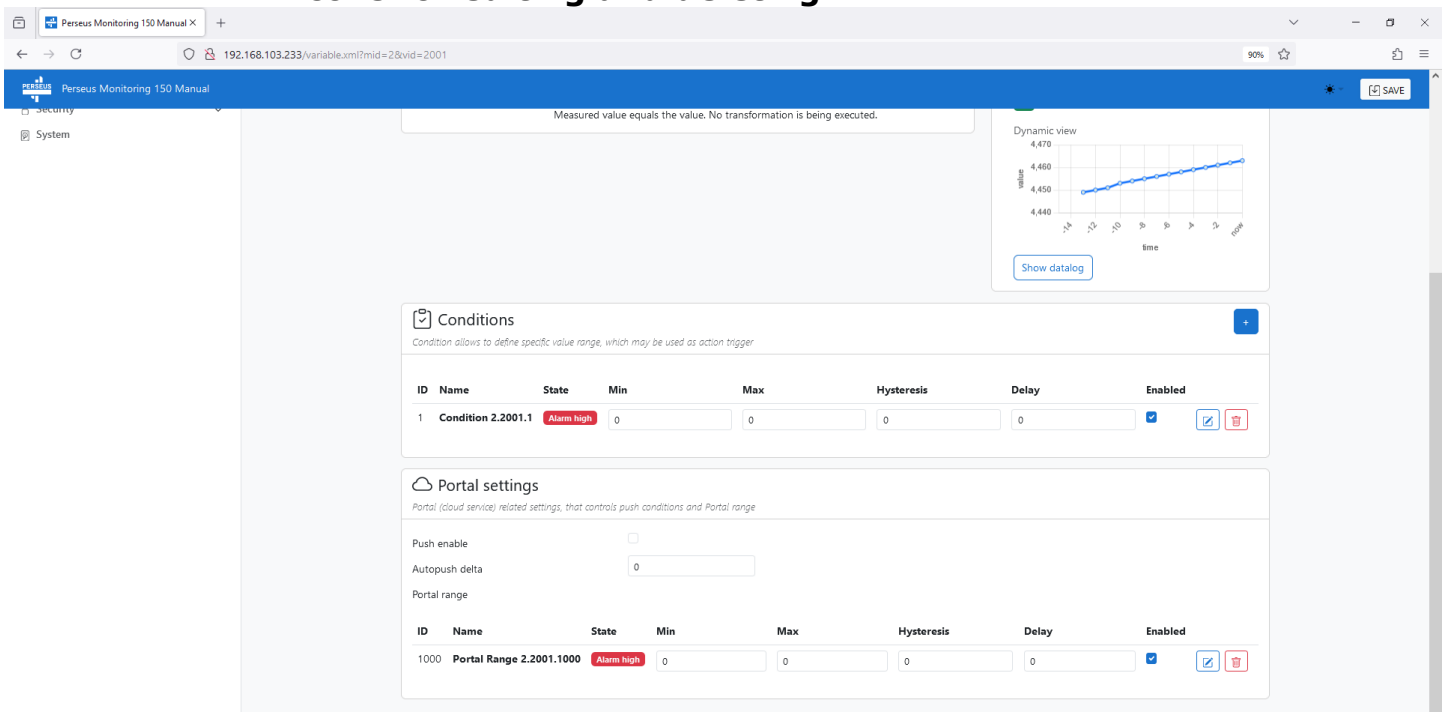
Portal settings Section

The section is used to enable sending a variable to the portal using the HWg Push protocol. The same parameters can be set on the General / Portal page

Due to the fact that Perseus units can work with a very high number of variables, but usually not all read variables are monitored, unlike other devices of the Perseus HW group units, they do not automatically send all measured variables to the portal, but only those for which this function is manually enabled are sent.

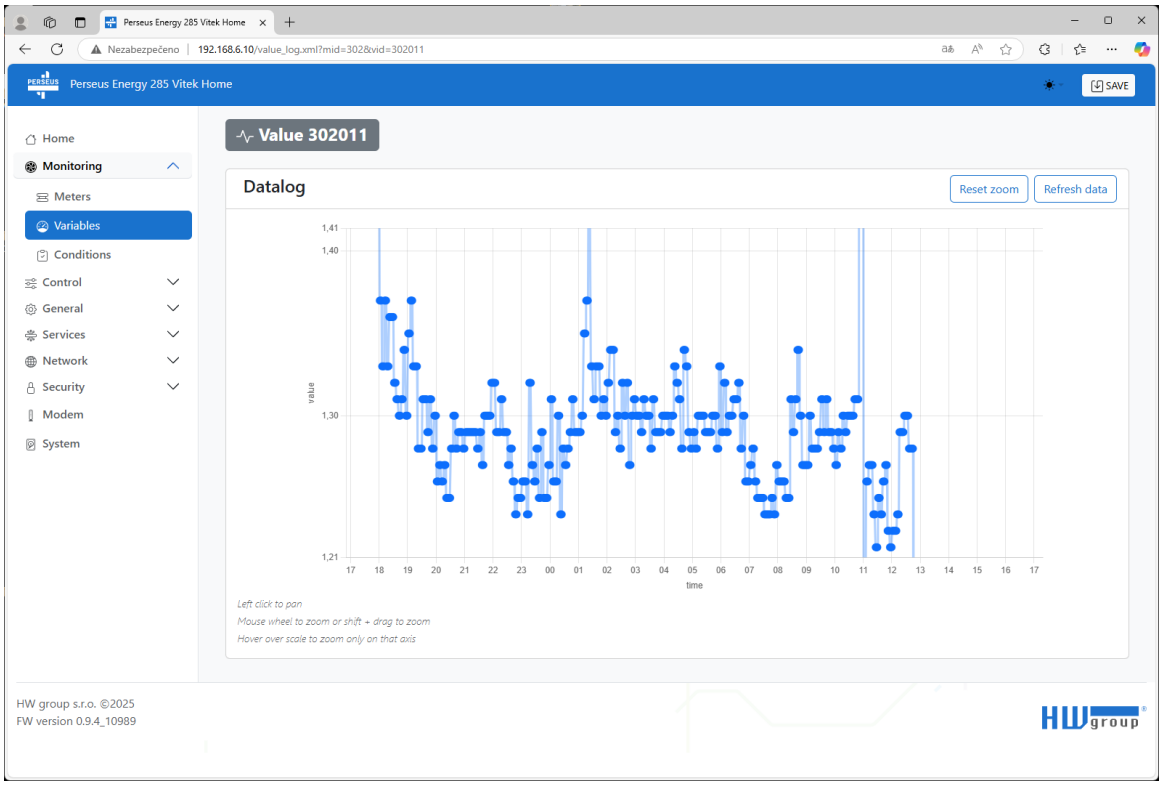
- **Push enable** – enable or disable the sending of the tag to the Sensdesk technology portal. This allows you to send only monitored parameters to the portal.
- **Autopush delta** – if the value increment is more than specified by the Autopush delta parameter since the last time data was sent to the portal, an extraordinary data push will occur. If the parameter is 0, the function is disabled for this variable.
- **Portal range** – set the allowed range of values for working with the portal. If not set, the state of the tag is not evaluated for the purposes of working with the *Sensdesk Technology portal*. If set, it is possible to monitor the status of the variable and, if the set range is exceeded, to call up an extrapush to the portal. Since evaluation is not necessary for most parameters, *the Portal range is not* set by default. The parameters are identical to the parameters in the Conditions section:
  - **ID** – ID of the condition within the given variable
  - **Name** – Condition name – the entire condition ID is automatically added to the name. The name can be changed in the edit of the condition

- **State** – Displays the status of the condition after evaluation. For a description of variable states, see chapter Meter states, variables and conditions
- **Min** – Minimum allowed value
- **Max** – Maximum allowed value
- **Hysteresis** – Area of inertia
- **Delay** – delay for activation of the AlaramHi or AlarmLo state
- **Enabled** – enable or disable the condition for the given variable
- **Icons for editing and deleting**



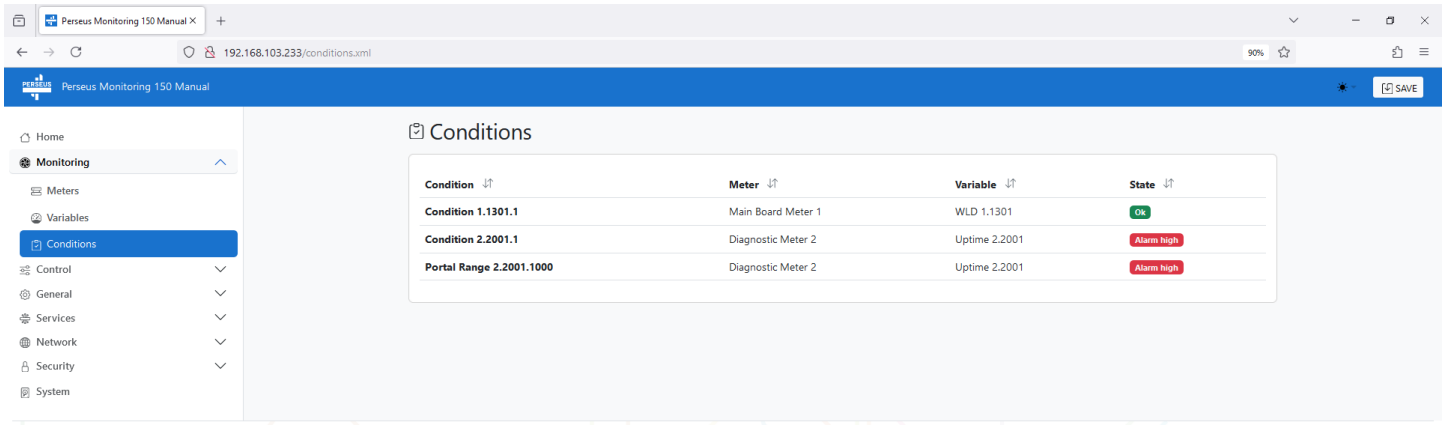
Details section

The Detail section allows you to view an overview of information about a variable, such as the value, its text representation (if any), status, and a chart.
For more information about variable editing parameters, see the Variables chapter. The Show Datalog button displays the chart page (only for Variables with logging enabled):


Perseus Energy 285 Vitek Home
Nezabezpečeno | 192.168.6.10/value_log.xml?mid=302&vid=302011
Perseus Energy 285 Vitek Home
SAVE
Home
Monitoring
Meters
Variables
Conditions
Control
General
Services
Network
Security
Modem
System
Value 302011
Datalog
Reset zoom
Refresh data
1,41
1,40
1,30
1,21
value
17 18 19 20 21 22 23 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17
time
Left click to pan
Mouse wheel to zoom or shift + drag to zoom
Hover over scale to zoom only on that axis
HW group s.r.o. ©2025
FW version 0.9.4_10989
HW group

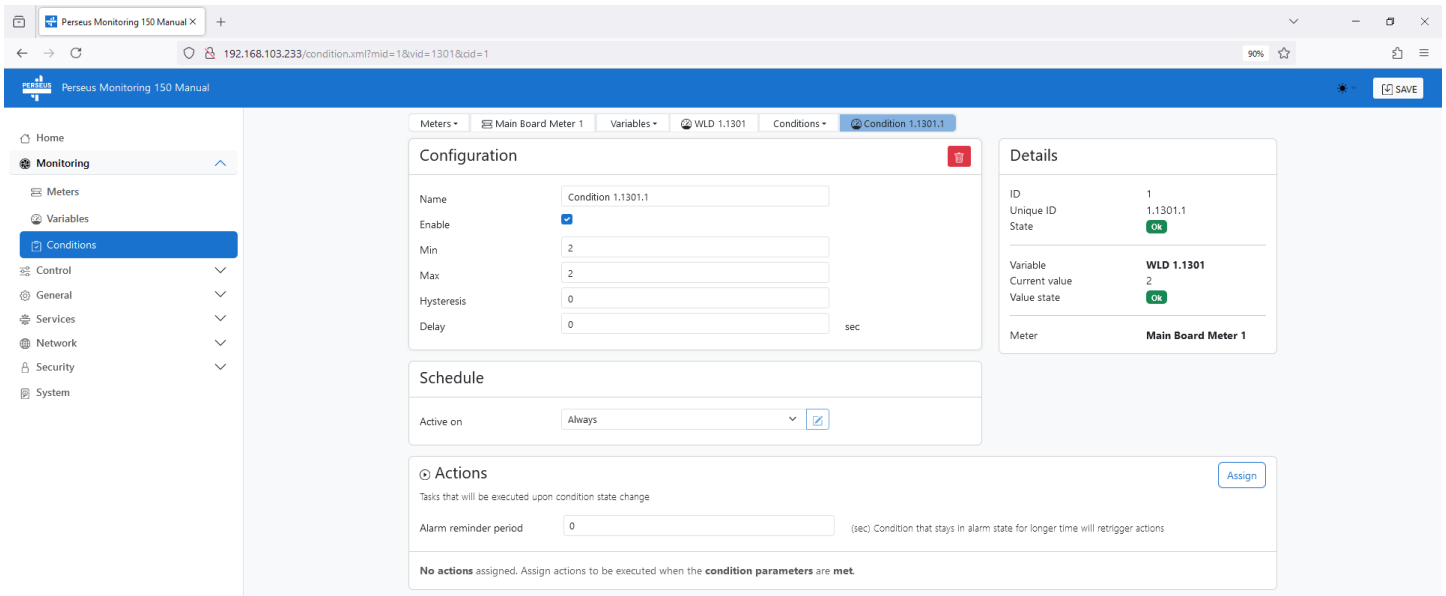
Menu Monitoring / Conditions

Since a *Condition* is always bound to a specific variable (*Variables), the* Conditions *page contains* only an overview of existing conditions, their status and links to *Meter* and *Variable*.



- **Meter** - Name *of the Meter* to which the condition is attached. Contains a link to edit *the Meter* (it is possible to change the order of meters)
- **Variable** - The name of the *Variable* to which the condition is bound. Contains a link to the *Variable edit* (variable ordering can be changed)
- **Condition** - The name of the *Condition* to which the condition is attached. Contains a link to edit the *Condition* (it is possible to change the order of the conditions)
- **State** - condition state (it is possible to change the sorting of states) A description of the condition states can be found in the chapter Meter states, variables and conditions

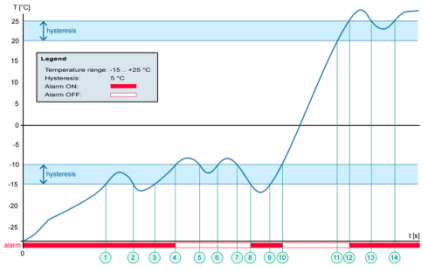
Condition Editing



Configuration section

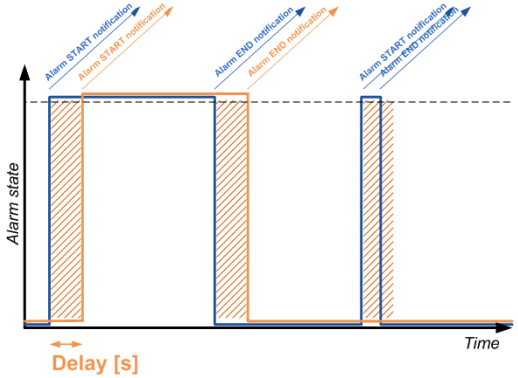
- **Name** – The name of the condition for easier reference. By default, it is pre-filled as a compound "Condition ID_meter.ID_values.ID_condition" (Condition 109.109001.1)

- **Enable** – enable (activates the condition). Disabling disables follow-up actions without having to turn them off or delete them individually, or even delete the conditions themselves. The function is suitable for temporary suspension of alarm actions, for example during a planned outage of the monitored technology.
- **Min** – The minimum allowed value. If the monitored value is lower, the ***AlarmLo status occurs***
- **Max** – The maximum value allowed. If the monitored value is higher, the ***AlarmHi status occurs***
- **Hysteresis** – The *Hysteresis value* defines the width of the tolerance band (inertia area) for returning the condition to the OK state. The function prevents multiple alarms from occurring in cases where the value oscillates around the set point. The function is evident from the graph.



Without the inner hysteresis band of 5°C, the alarm triggered at point 8 would end at point 9, thanks to the hysteresis function, the alarm is extended until the temperature rises to the end of the hysteresis band (point 10) 5°C + (-15°C) = -10°C.

- Hysteresis = 5°C - Unit sends 3 E-mails (SMS) - Alarm in points 0..4, 8..10, 12 and onwards
- No hysteresis (0°C) - Unit sends 8 E-mails (SMS) - Alarm at points 0..1, 2..3, 8..9, 12..13, 14 and more
- Delay - Delays sending information about the start and end of the alarm.



Details section

The details section contains an overview of the condition's properties:

- **ID** - ID of the condition within Variables.
- **Unique ID** – a unique ID within the perseus unit. It consists of ID_meter.ID_values.ID_condition
- **State** - current state of the condition
- **Meter** - The meter to which variables with this condition are connected
- **Variable** - Variable to which this condition is bound
- **Current value** - Current value variable
- **Value state** - This is the status of Variables, not specific conditions, unless it is a condition with ID=1
- **Condition State** - current condition status
- **State delayed** – if the Delay setting is set, displays how long the state change will be delayed

Schedule section

This section is used to set the validity of the condition depending on the real time. This makes it possible, for example, not to trigger an event outside of working days, or to perform different actions during and outside of working hours. The default value is Always, so the condition is always applied. A new schedule can be added on the Control=>Scheduler page. The Edit button is used for redirection and the changes made on the page are not applied and must be saved first using the Save button.

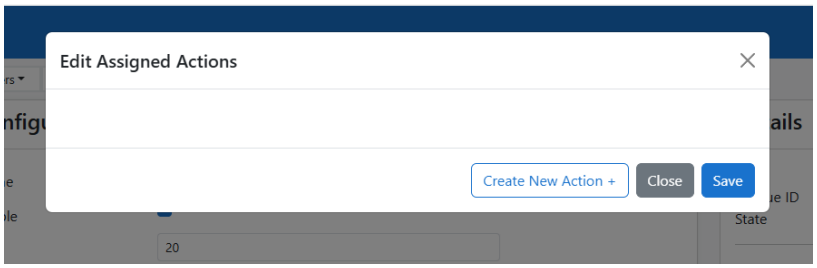
Actions section

Use the *Action* section to add one or more actions of the type:

- **E-mail** – sends a message by e-mail
- **SMS** – send a message via SMS.
- **SNMP Trap** – sends a message via SMS
- **Set Output** – sets the output to the desired value.
- **Script** – Runs the defined LUA Script (actually only Perseus Industrial models)

It is either possible to assign an existing action by selecting it at the *Assign action* position and pressing *the Assign button*, or you can open the dialog of creating a new action by clicking on the *Create Action button*.
For a description of condition states, see Meter states, variables and conditions.

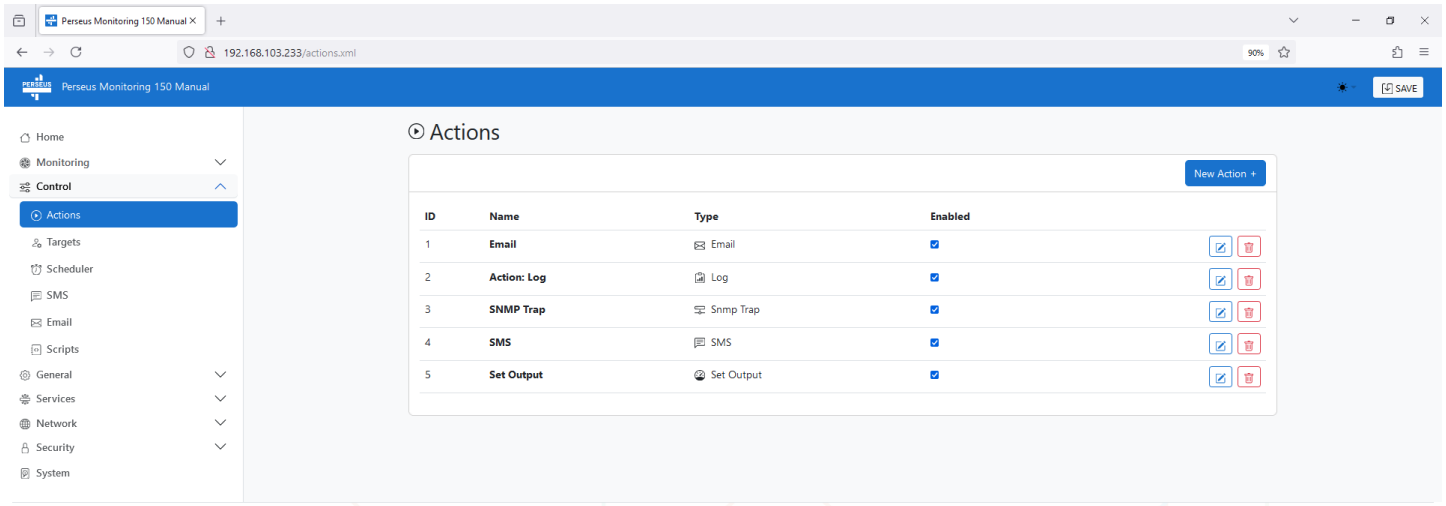
- **Alarm reminder period** - Condition that stays in alarm state for longer time will retrigger actions. Period is in seconds



For a description of condition states, see Meter states, variables and conditions.

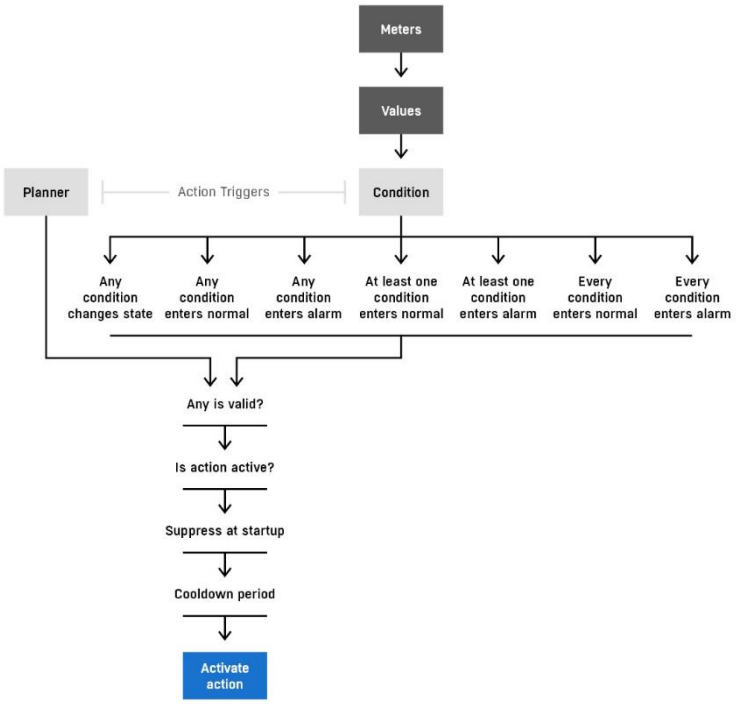
Menu Control / Action

This page provides an overview of existing events and their editing. The table contains information about the event's name and ID, type (Email/SMS/Set Output/SNMP Trap), and whether it is enabled.

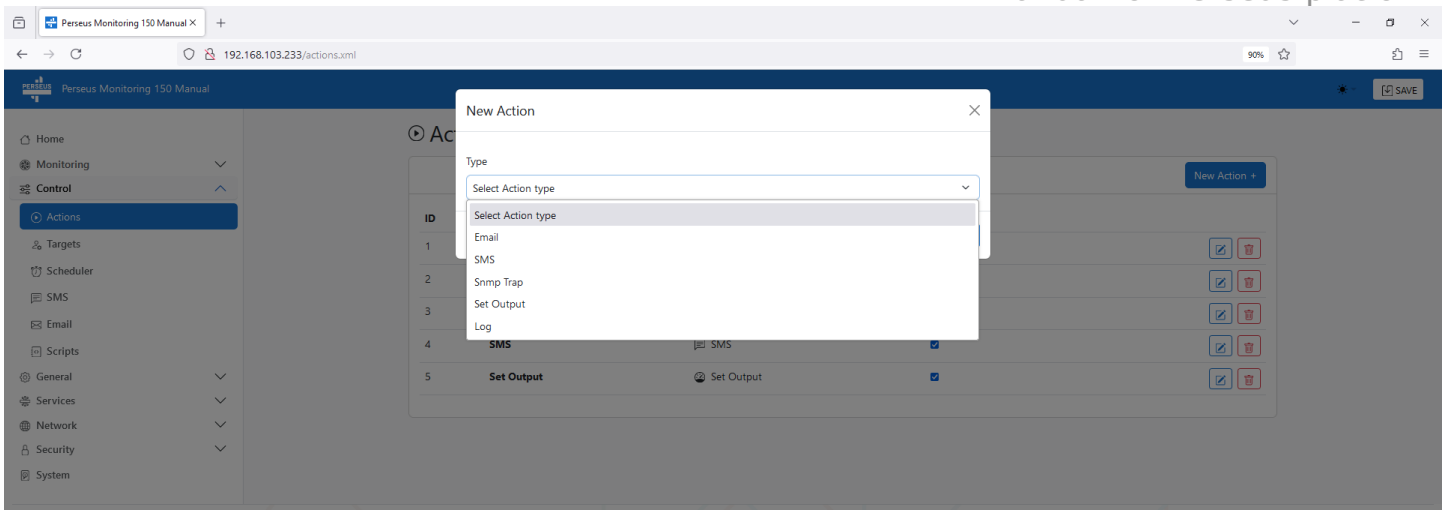


Action is an action element that is triggered either in real time by a scheduler, by changing the state of an assigned condition, or by a combination of both, and is complemented by a wide range of options and conditions:

Dependency scheme

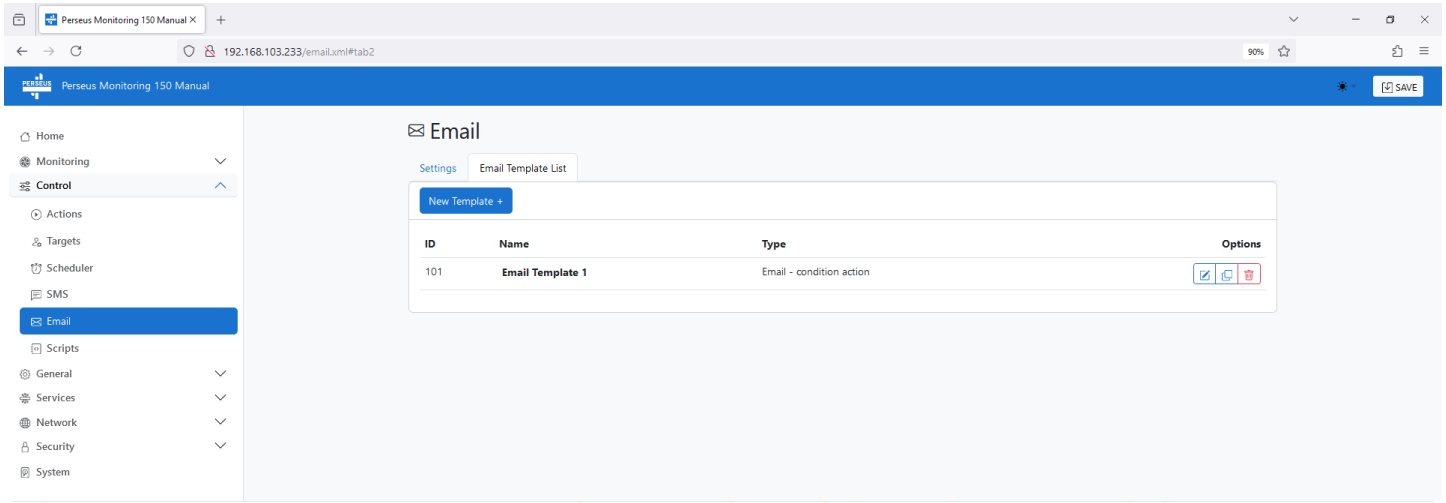


The action can be edited by clicking on the name of the existing action or on the corresponding edit button. To create a new action, use the "New Action +" button, which launches a simple wizard to add a new action.

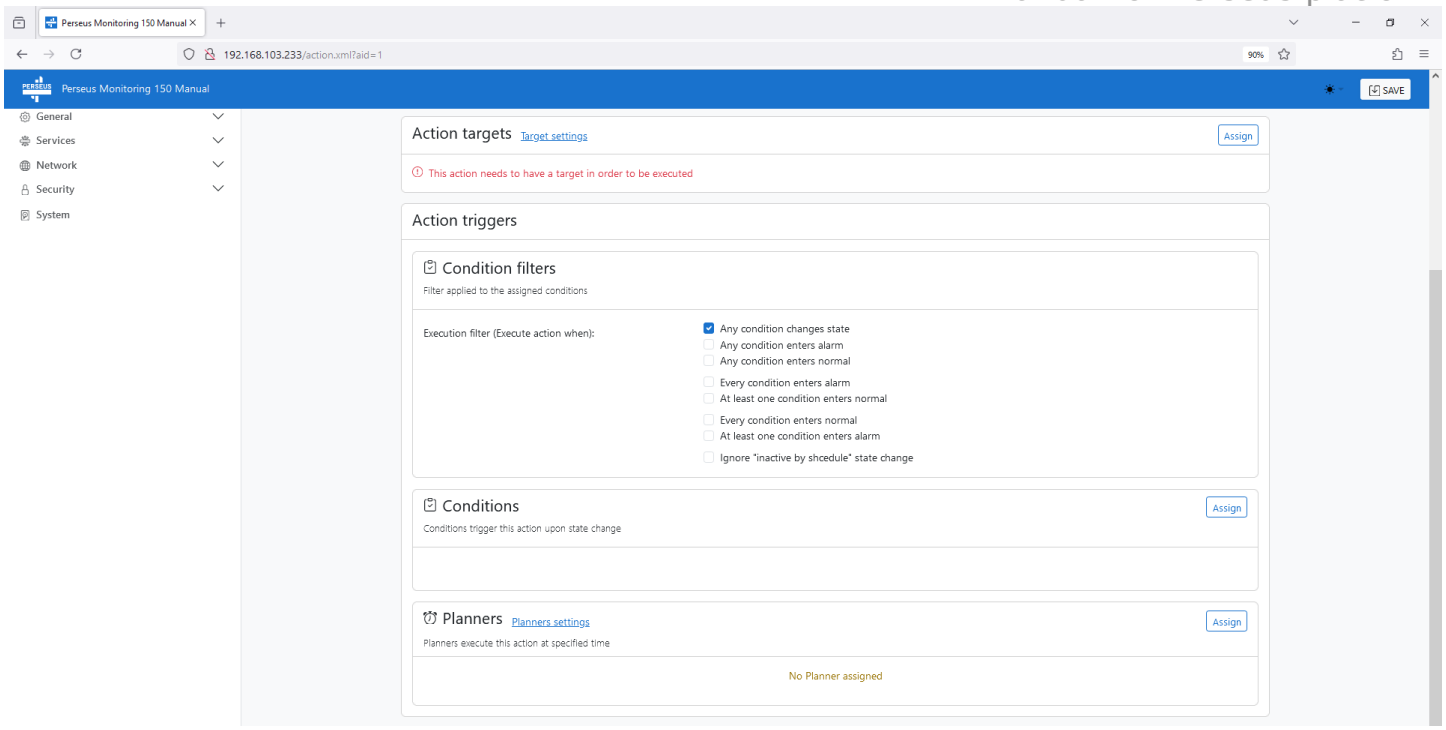


The modal window contains only limited options for setting action parameters, but it contains the basic ones, i.e.:

- **Type** – Action type – Can be switched to a different action type
- **Name** – The name of the action – The action has a naming option for easier orientation. If it is not filled in, it will be automatically filled in as "Action ID"



Clicking on the name of the action or the edit button will open the detailed settings of the action:



Action Features

- **Name** - The name of the action is used for clearer work with actions within the system
- **Enable** - Enables or disables the action. This functionality is suitable for suppression of the performance of the action, for example, during planned outages of monitored technologies, long-term failures, etc
- **Cooldown period** - the time in milliseconds for which the action execution is postponed - Equivalent to the Delay parameter for conditions.
- **Type** - type of action (E-mail, SMS, Set Output, SNMP Trap)
- **Template** (SMS and E-mail actions only)- Message template It is selected from a list of existing actions together with a link to edit or create a new template. By default, a default template for Alarm messages is always created on the device. In the future, a template for periodic reports will be added (a template type flag will be added).
- **Output Target** (Set Output only) - Select the output that will be controlled by the action. A single output can only be served by one action, so if you want to select an output already served by another action, this will not be possible with an indication of which action the output is controlling. For the needs of serving output based on multiple conditions, it is possible to use actions bound to a virtual value that executes the required logical function using LUA Script.
- **Ring out** (SMS type action only) - Ring - SMS message is always sent and when this flag is activated, the phone to which the message was sent is subsequently ringed to emphasize the existence of the message.
- **Suppress execution after device start** - suppress the execution of the action after the device starts. If the conditions for performing an action are met at system startup, this action will not be performed - useful for system restarts, upgrades, etc.

Action triggers

Action triggers represent action triggers. The trigger of the action can be the Scheduler, or in this case the planner, which will run the action periodically according to the relevant schedule, or the condition linked to values. Combinations of these triggers are also possible.

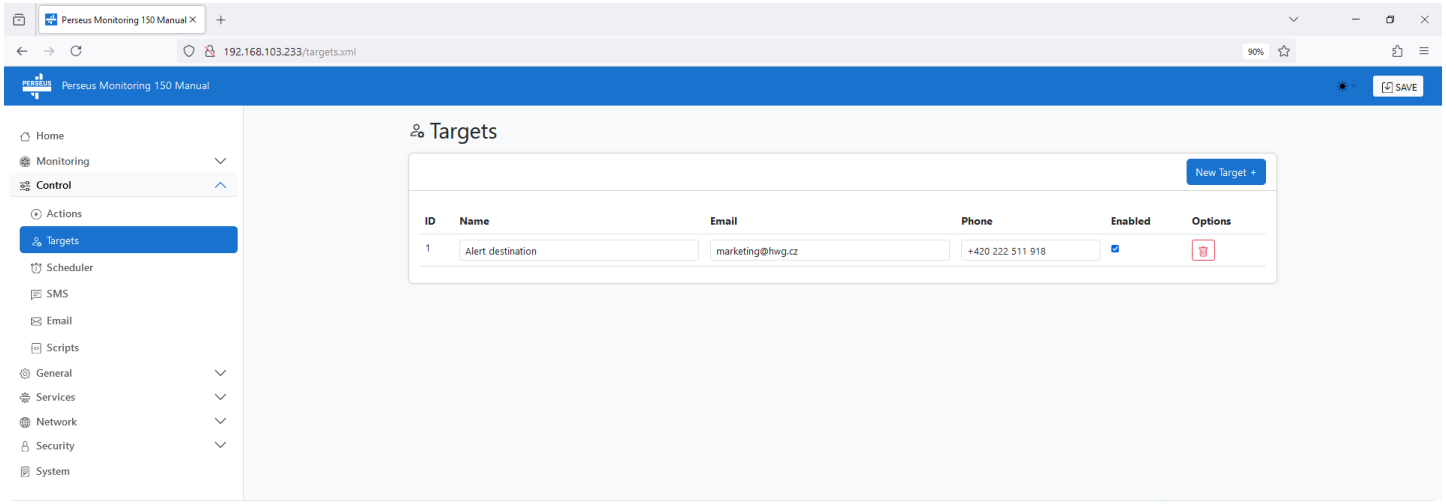
- **Conditions section** – Attaches or detaches from the condition action that triggers the action. If an action has at least one condition assigned, it does not have to have any planer assigned, and vice versa. However, both types of triggers can be freely combined.
  - **Execution filter (Execute action when)** - filter for starting the execution of the action
    - **Any condition changes state**
    - **Any condition enters alarm** - The action is executed if any condition enters the alarm (AlarmHi or AlarmLo) - the OR function is applied to all connected conditions (the Alarm state must be true on at least one condition)
    - **Any condition enters normal** - the action is executed if any condition enters normal (OK state) - the OR function is applied to all connected conditions (the OK state must be true on at least one condition)
    - **Every condition enters alarm** - The action is executed if all connected conditions enter the alarm (AlarmHi or AlarmLo) - the AND function is applied to all connected conditions (the Alarm state must apply to all connected conditions)
    - **At least one condition enters normal** - The action is executed if the first of the connected conditions returns to the normal state
    - **Every condition enters normal** - The action is executed if all connected conditions are normal (OK state) - the AND function is applied to all connected conditions (the OK state must be true on all connected conditions)
    - **At least one condition enters alarm** - The action is executed if the first of the connected conditions is set to the normal state
    - **Ignore "inactive by schedule" state change** - The action is not executed if the scheduler changes its state
- **Planner section** (old name Schedules) - Attaches or detaches from scheduler actions that trigger the action. If an event has a calendar assigned to it, it does not have to have a condition assigned to it and vice versa. However, both types of triggers can be freely combined.

Action targets section (only action types that require a cite definition)

Attaches or detaches from the target action for messages.

Menu Control / Targets

The page is used to create destinations for SMS and Email alarms. It is de facto a directory.

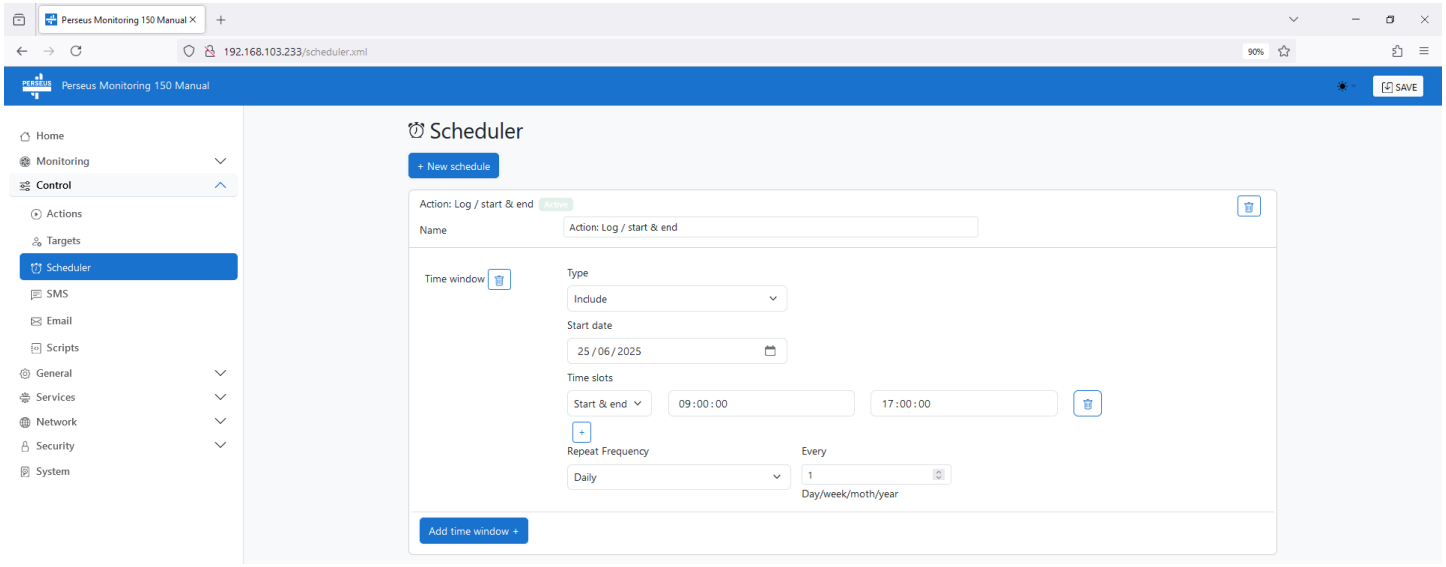


An email address and a phone number can be defined for each target (user). Individual recipients are then assigned to actions.

In the future, targets for SNMP traps will also be added here.

Menu Control / Scheduler

The page is used to define schedulers that can serve as a parameter of the Condition activity or as an action trigger.



Within the Perseus unit, one or more schedulers (calendars) can be defined and they can be used to condition the validity of Conditions or trigger actions. These schedulers can also be combined with each other to create more complex constructs such as activity conditions only during working hours, but not on public holidays.

For each *Scheduler*, the following is defined:

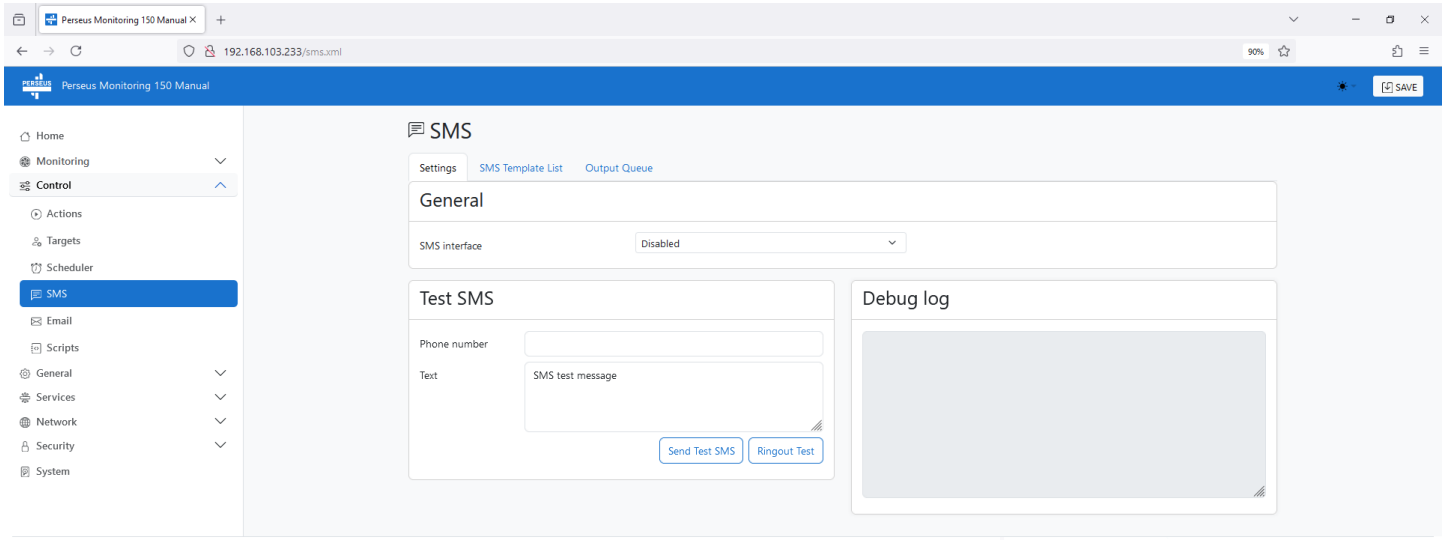
- **Name** - The name of the scheduler - used to make it easier to work within the Perseus unit
- Time window - define the time window that the scheduler interprets. Multiple time windows can be defined within one plan (for example, for the purposes of defining different working hours on different days – office hours)
  - **Type** - window type
  - **Include** - Inserted - Scheduler is active, inside the defined time window
  - **Exclude** - Scheduler is active, outside the defined time window
  - **Exclude Schedule** – Exclude another scheduler Exclude another schedule (e.g. the public holiday scheduler can be excluded from the scheduler of working hours Mon-Fri 7-17)
  - **Select schedule** - Selects the scheduler or time window that will be excluded from this
  - **Start date** – the start date of the planner's validity (it can be in the past) allows you to set the validity of the plan, for example, for company-wide holidays
  - **Time slots** – Range of time validity – multiple time ranges can be defined within the day (e.g. 0:00-07:00 and 16:00-23:59 for the definition of heating). However, it is possible to set, for example, the start or end time of the event, for example for scheduling the sending of periodic e-mails, or switching the output with the Pulse Timer.
  - **Repeat Frequency**
    - **Never** – this is a one-time event that will not recur
    - **Daily** – daily repetition

  - **Weekly** – Specific days of the week can be defined when the scheduler will be applied.
  - **Monthly** – Monthly recurrence – Specific numbers of days in the month when the scheduler will be applied can be defined.
  - **Yearly** – Yearly recurrence – Specific dates in the year when the scheduler will be applied can be defined.
- **Every** – recurrence period (Day/week/moth/year) – Every 2nd Tuesday, 1st Wednesday of the month, etc.

Most of the more complex schedules can be applied in multiple ways. For example, it is possible to create a working time planner, from 08:00 to 16:00 every working day, and use Exclude Schedule to insert a public holiday calendar into it as an exception (so it can be applied multiple times), as well as it is possible to define a scheduler containing 2 time windows, the first of the Include type with a time range from 08:00 to 16:00 every working day and the second of the Exclude type containing a list of public holidays, respectively, exceptions. Similarly, 2 Wednesdays of the month can be applied, where the definition of a monthly repetition from the 8th to the 15th day of the month together with a second window with weekly repetition every Wednesday can achieve the desired one.

Menu Control / SMS

Contains SMS gateway configuration.



General Section

SMS interface - allows you to define whether the SMS gateway function is active and whether the built-in modem or remote gateway is used.

- **Disabled** - The SMS function is disabled
- **Remote** SMS Gateway - HWg-SMS-GW3 (or newer models) will be used for sending messages
- **Internal** (only models with built-in LTE modem) - an internal modem will be used for sending messages. You can find its settings on the Modem page

SMS gateway settings section

This is where the parameters of the remote gateway are defined

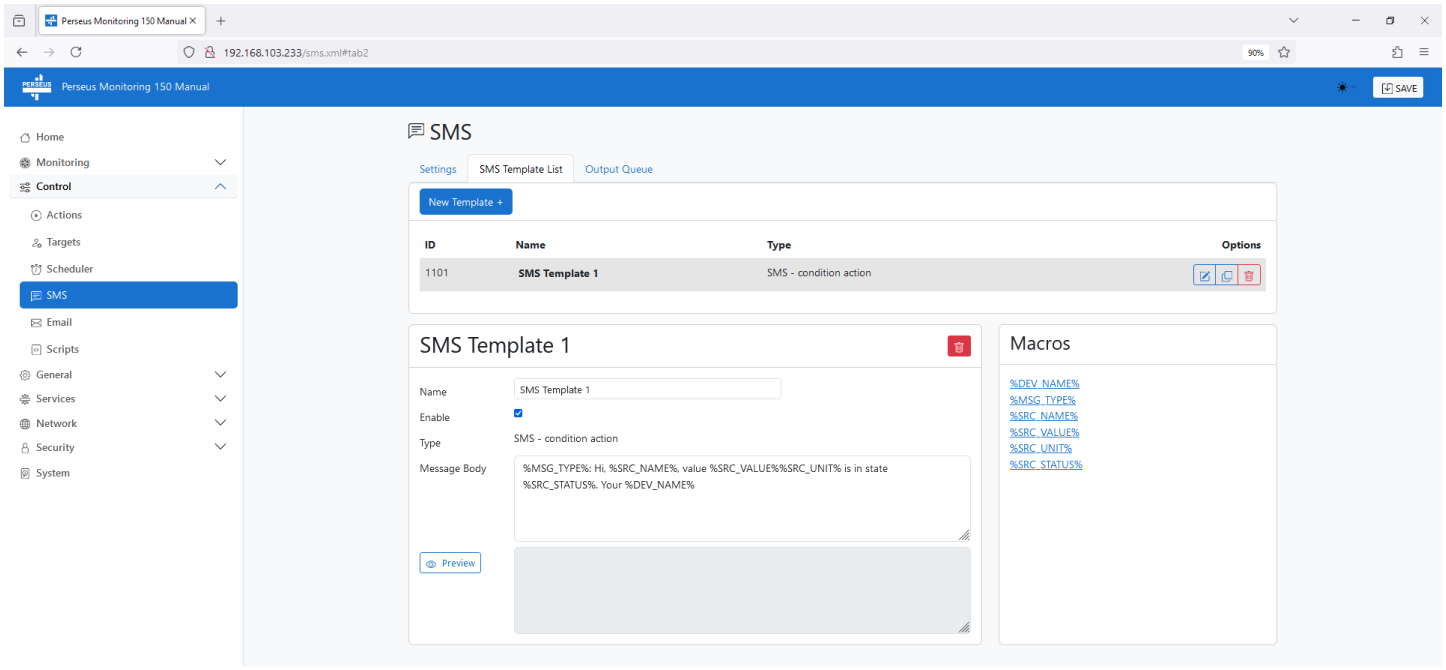
- **Host URL** - IP address or URL of the HWg-SMS-GW3 remote gateway. By default, everything is sent to the Service.xml service on port 80. If other parameters are to be set, the full path including the port should be included in the Host URL.
- **Username/Password** - Authorization parameters of the remote SMS gateway

SMS Test Section

It is used to send a test message (Text) to a specified phone number, or to test the ringing of a given number.

SMS Template List

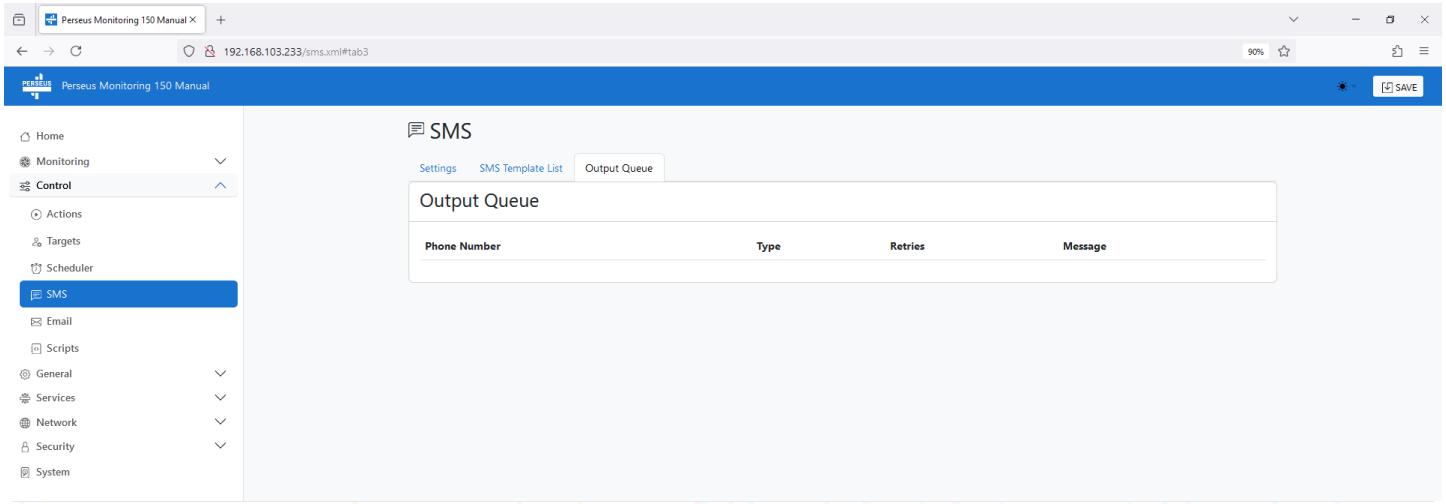
It contains a list of SMS message templates with the possibility of creating, cloning and editing them.



On the overview page, there is a list of current templates and the option to edit, clone or delete them. After pressing the "New Template +" button, a new template is created, which can be named and edited after clicking on the name or pressing the edit button.

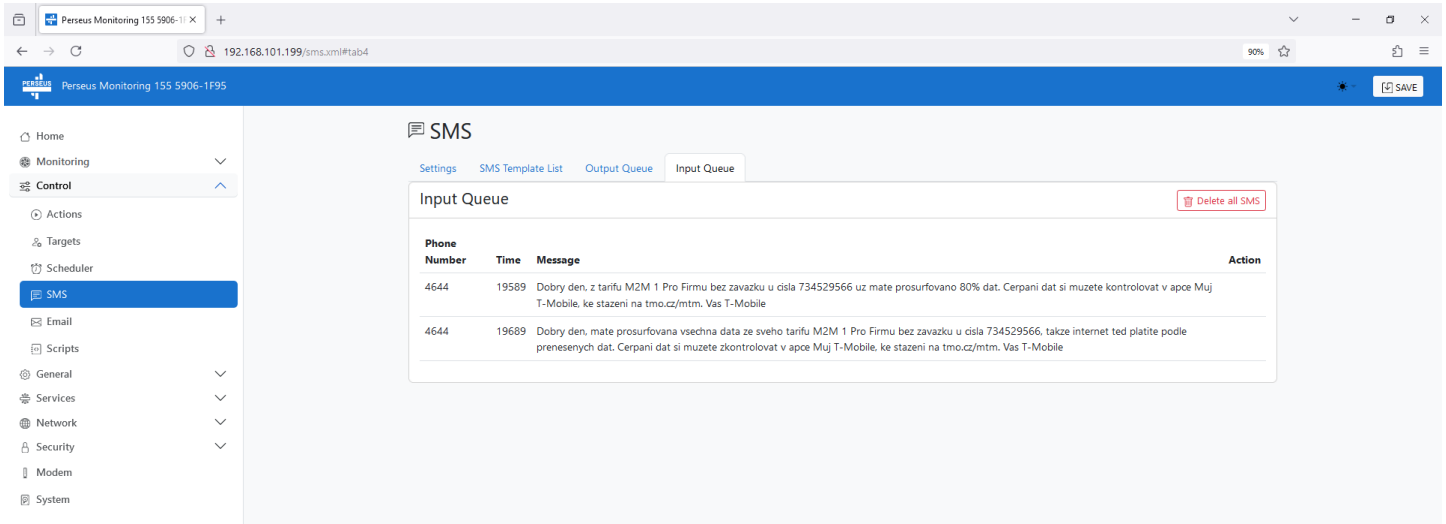
SMS Output Queue

The page contains a queue of outgoing messages.



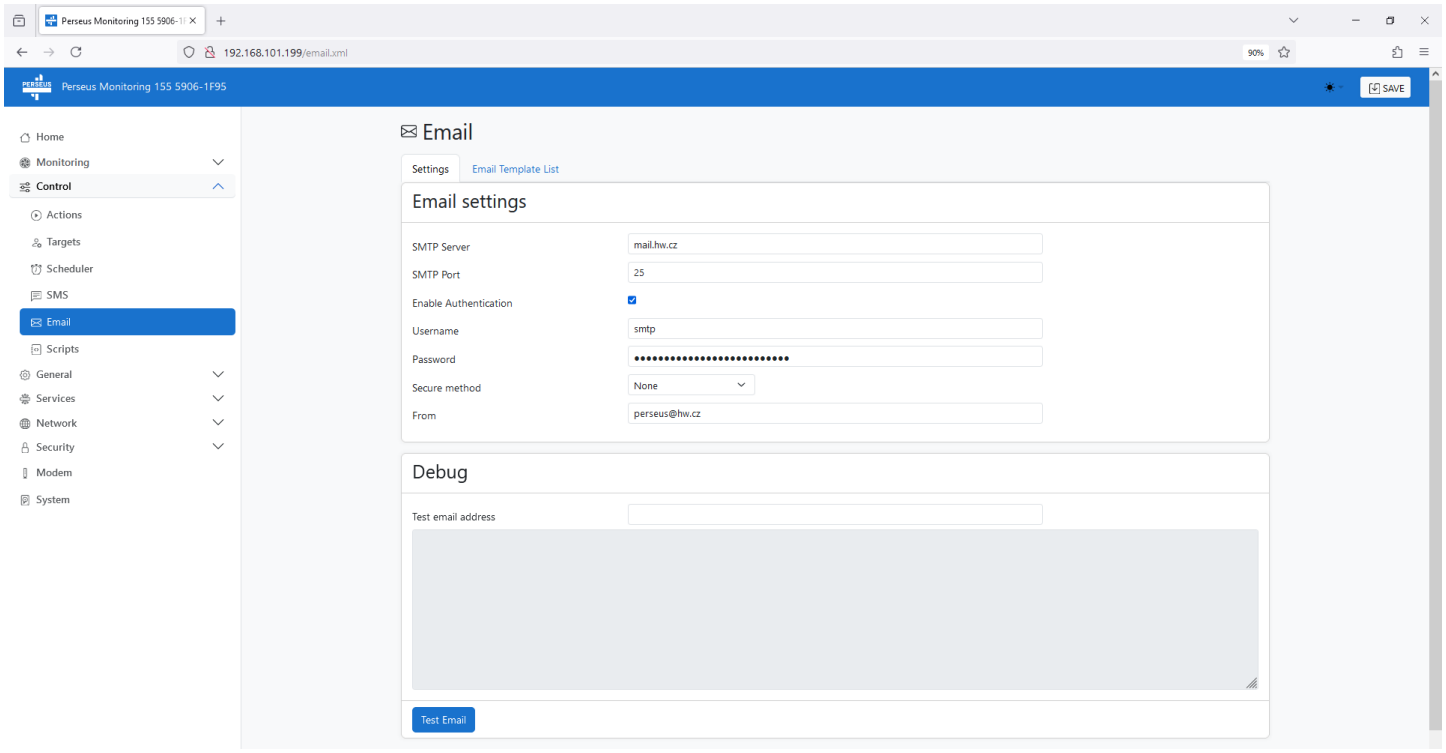
SMS Input Queue

The page contains a queue of ingoing messages.



Menu Control / Email

Contains SMTP server parameters for sending messages.



Email settings section

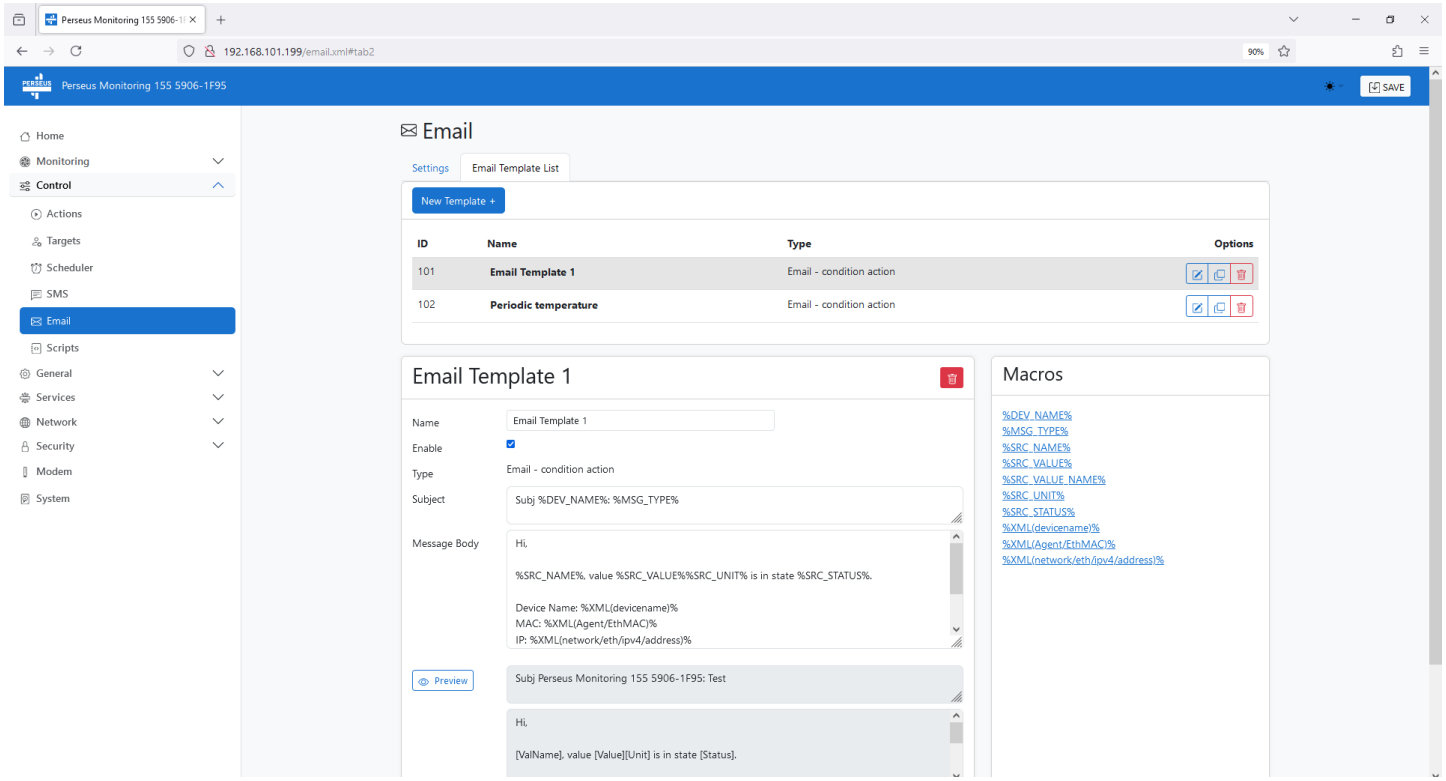
- SMTP Server – domain name or IP address of the SMTP server
- SMTP Port – set the port for communication with the SMTP server (25 by default)
- Enable Authentication – activate the use of a password if the SMTP server requires authorization.
- Username – name used for authorization on the SMTP server.
- Password – set the password for logging in to the SMTP server.
- Secure method – define the communication security method (“Automatic” may not be supported by target SMTP server, most common is “STARTTLS”)
- Email Sender Address – Email address to be listed as the sender in the email

Debug section

It contains debugging data and the ability to send a test email to a specified address.

Email Template List

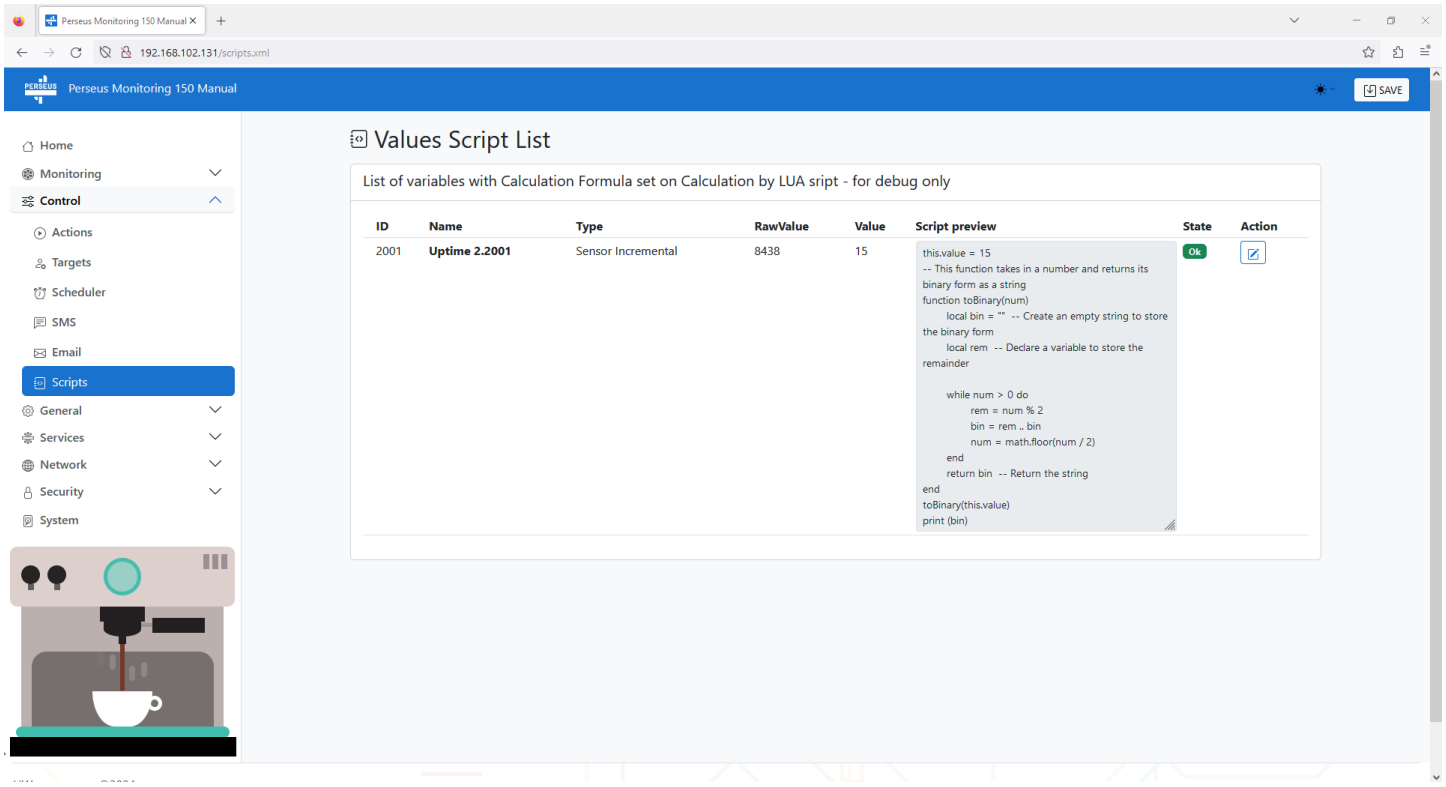
It contains a list of e-mail message templates with the ability to create, clone, and edit them.



On the overview page, there is a list of current templates and the option to edit, clone or delete them. After pressing the "New Template +" button, a new template is created, which can be named and edited after clicking on the name or pressing the edit button.

Menu Control / Scripts

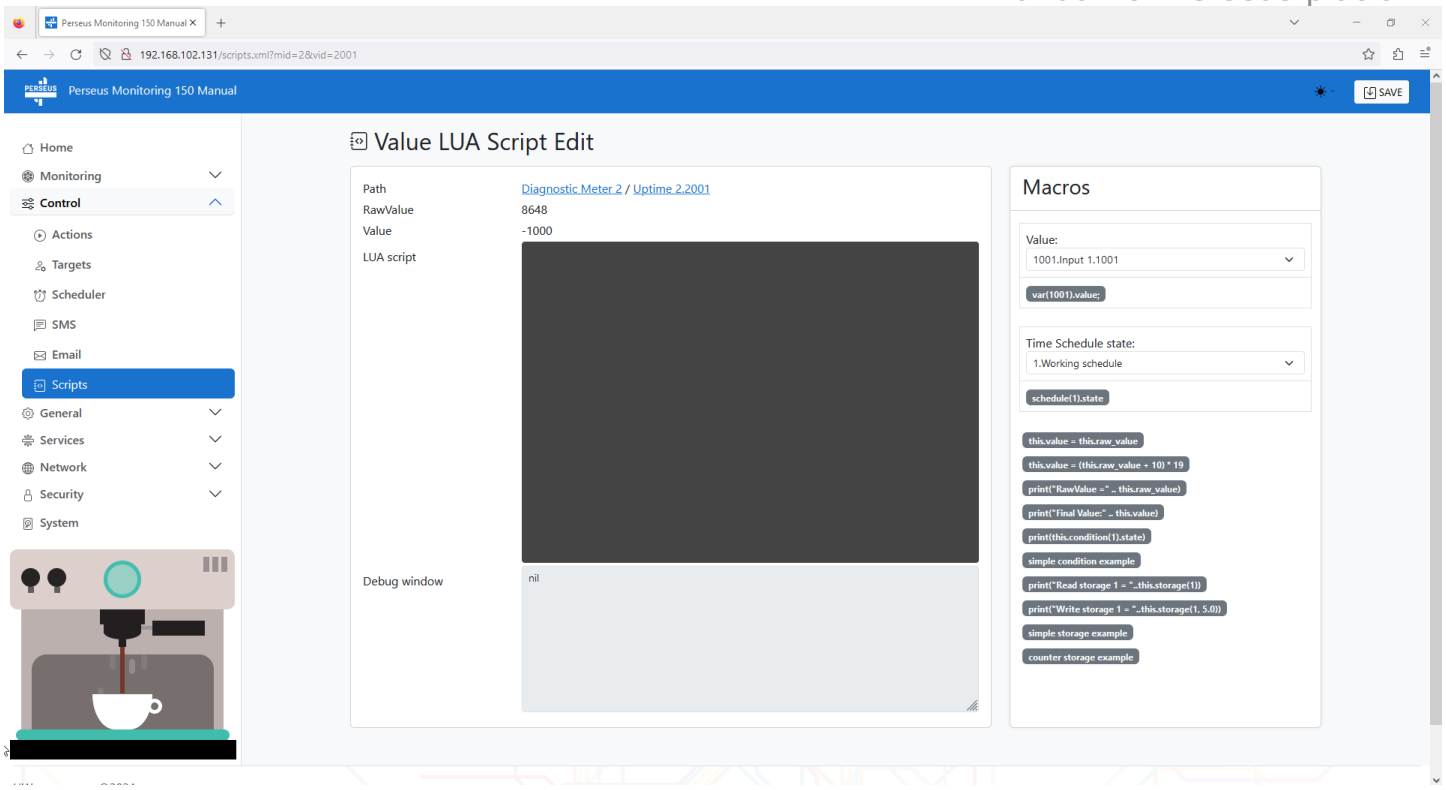
The page contains a list of values that have the Calculation formula parameter set to Calculation by LUA script with the possibility to edit their scripts.



The table clearly states:

- **ID** - ID of the value
- **Name** - Name of the value
- **Type** - type of value
- **RawValue** - Raw Value
- **Value** - the resulting value
- **Script preview** – preview of the script
- **State** - the state of the value
- **Action** - edit button

When you open the script editing, a window opens with the script code, a debugging window and macro help:



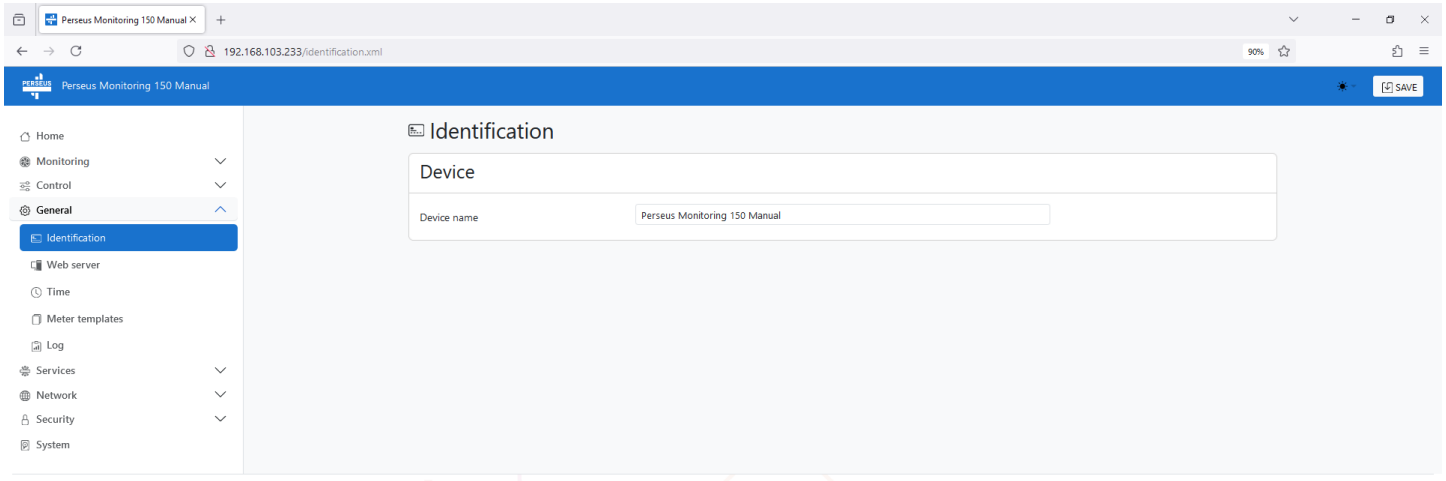
- **Path** – Link to the appropriate Meter / Variables
- **RawValue** – source value to which the script is applied
- **Value** – The resulting value including the unit
- **LUA Script** – custom script code
- **Debug window** – Debug window

Macros section

The section contains examples of macros as well as simple functions. By clicking on the "link", the corresponding macro or its code is inserted into the Lua Script window at the cursor position. To insert a specific Values or scheduler state, you can use the drop-down options, select a variable and then click on the row below the select.
For more information about working with scripts, see LUA Script.

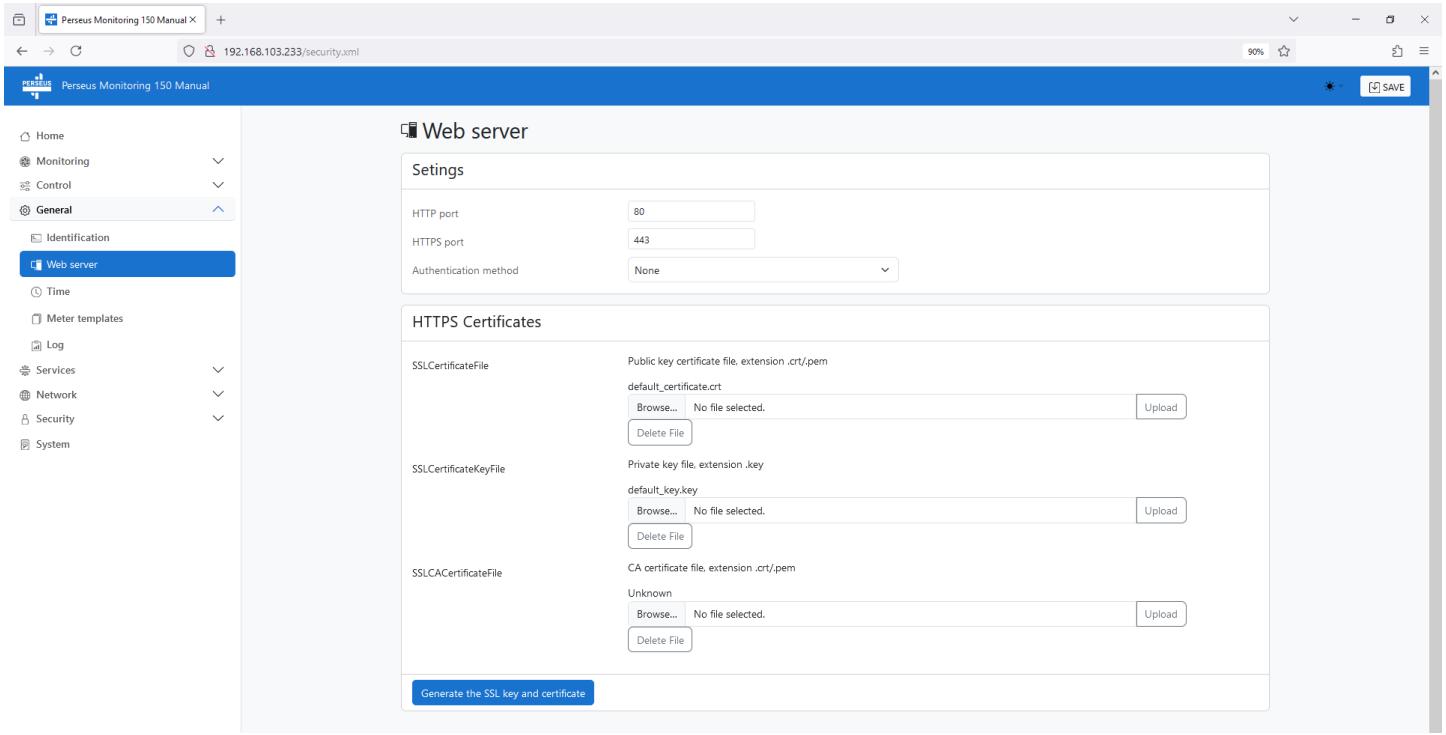
Menu General / Identification

On the page, it is possible to define the name of the device.



Menu General / Web Server

It defines the parameters of the web server of the device, i.e. the operation of the administration web.



Settings section

Sets the ports on which the website listens (both HTTP and HTTPS) and defines how users are authorized (auth selection is necessary, else the device User logins are not required even if they are defined in Security-Users)

- **None** - Perseus web interface does not require any authorization
- **Local authentication** - only local users defined on the Security / Users page can access
- **Radius authentication** - Only users authorized through the Radius server are allowed to access
- **Radius then Local authentication** - Can be accessed by users authorized through the Radius server or by local users

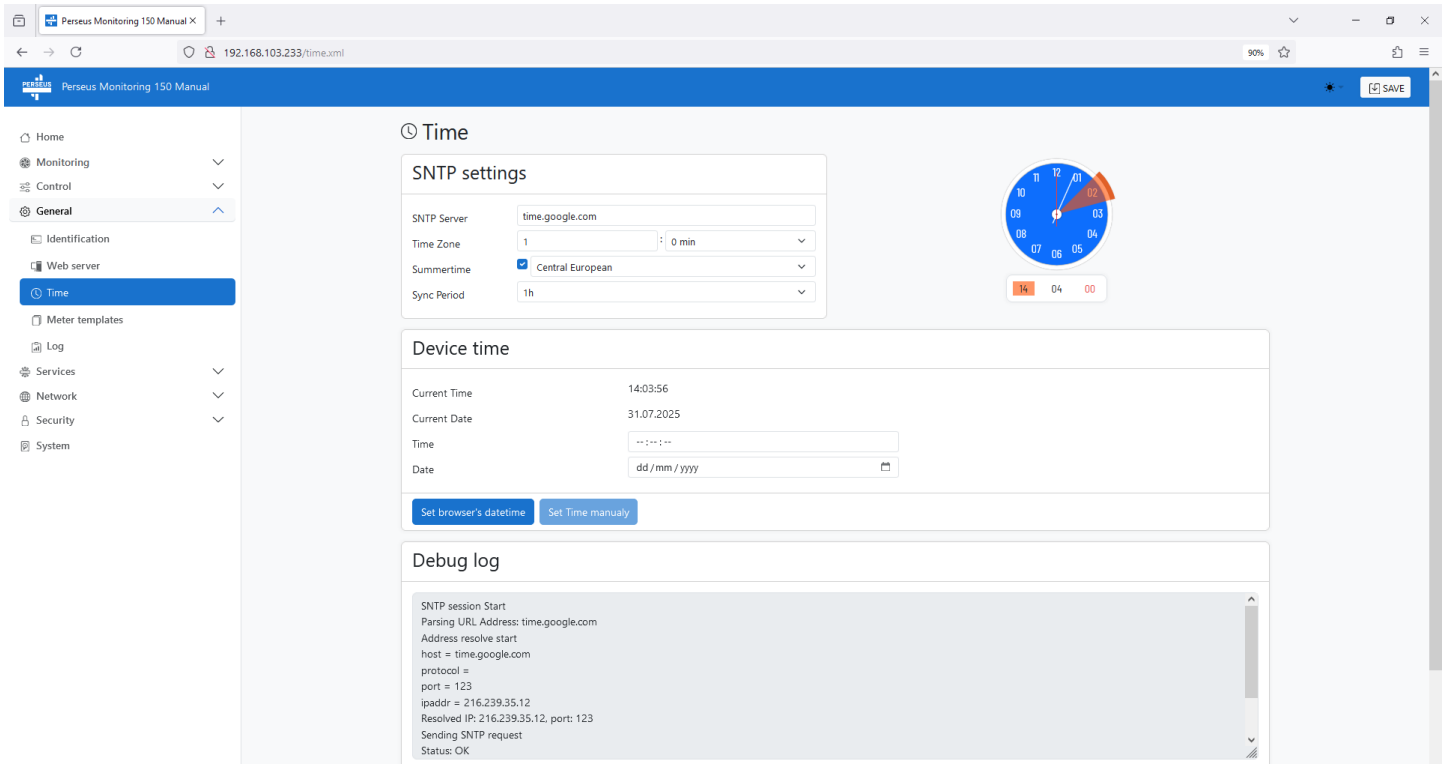
HTTPS Certificates Section

It is used to manage certificates needed for HTTPS server operation. It is possible to upload and delete the public key, the private key, and the certificate of the certification authority that issued the public and private keys (CAs).

Generate the SSL key and certificate - Generate the SSL private key and self-signed certificate for closed networks or testing purposes. The generated certificate is self-signed and appears as untrusted. Add the certificate to the exceptions list, or use a certificate signed by a trusted certificate authority. Please note that the generated data replaces SSLCertificateFile and SSLCertificateKeyFile.

Menu General / Time

Setting the actual date and time.



SNTP Settings section

- **SNTP Server** – IP address or domain address of the time synchronization server; time.google.com
- **Time Zone** – set the time zone of the device location. It is used to set the correct system time. Necessary for the correct recording of measured values
- **Summertime** – Daylight saving time permits. It is used to set the correct system time. Necessary for the correct recording of measured values
- **Sync Period** – set the time synchronization interval with the server

Device time section

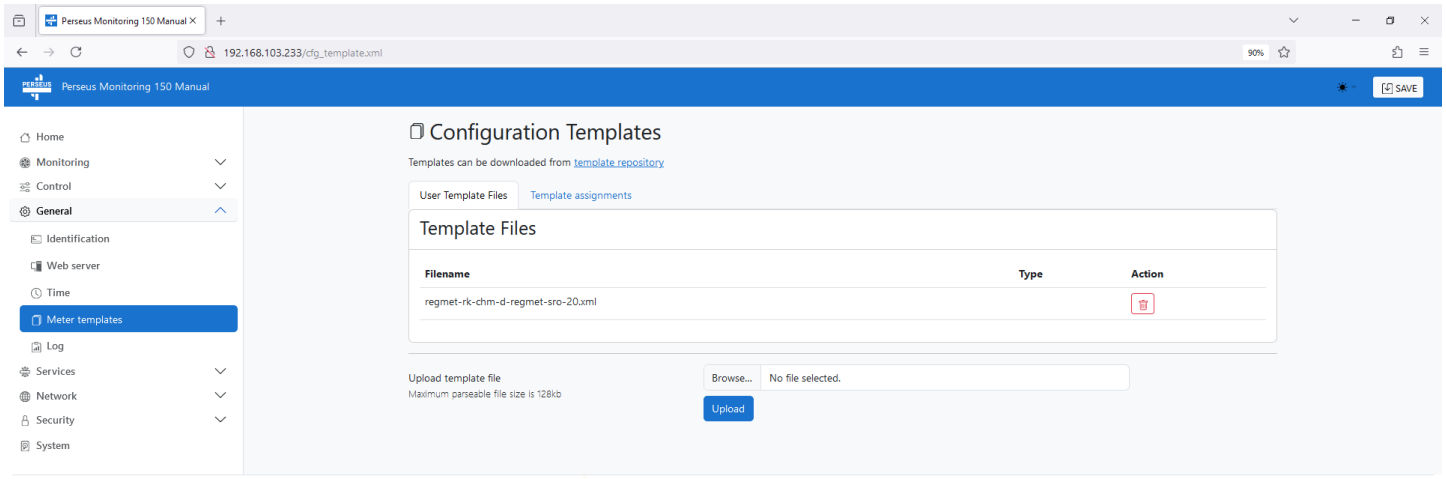
The Time Setup section allows you to fill in the current date and time manually (or take the browser's time) in case it is not possible to synchronize with a time server.

Menu General / Meter Templates

Use the page to view and manage uploaded Templates of meters and variables.

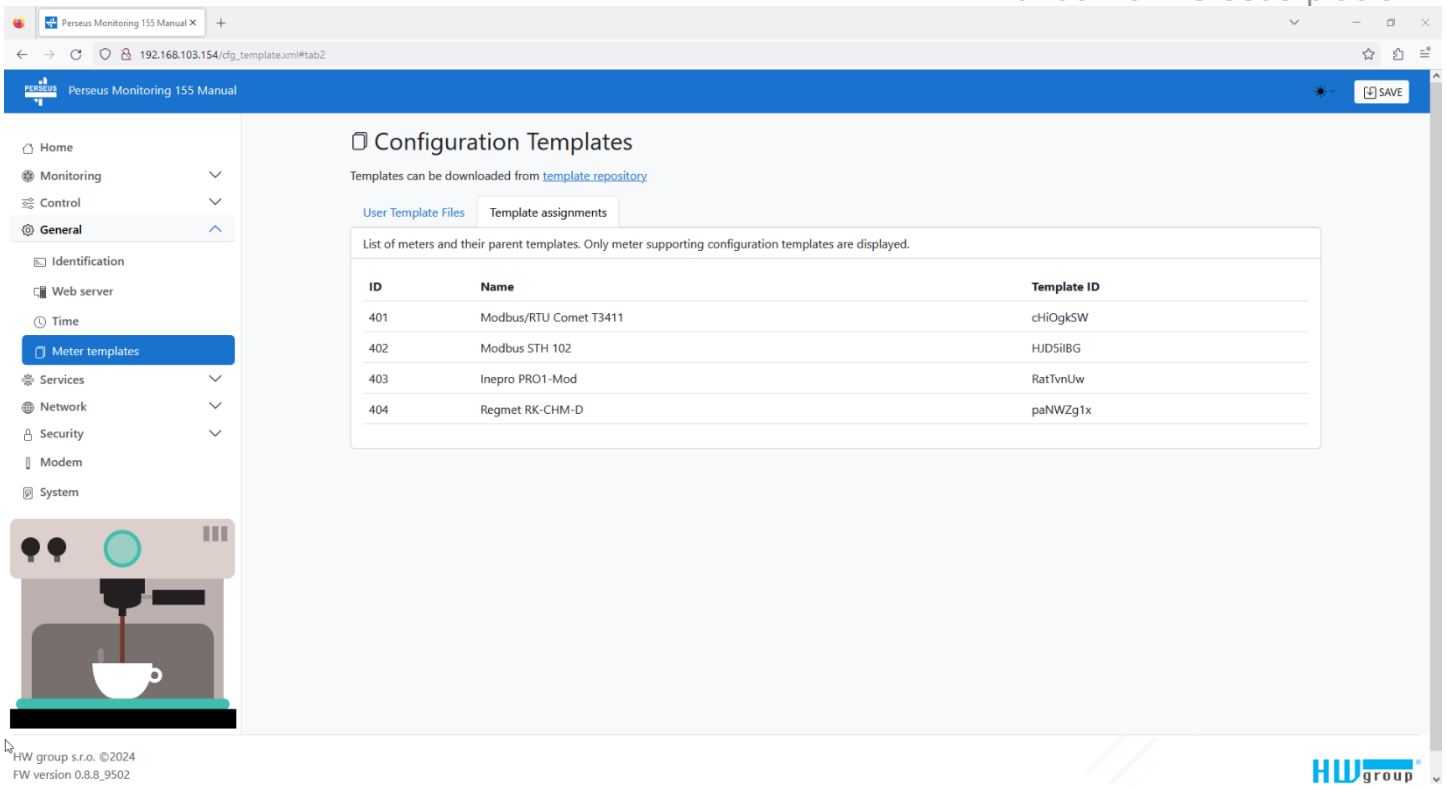
Because Perseus can support hundreds to thousands of 3rd party meters, only a very small sample of templates is loaded in Perseus units for easier handling, even for HW group devices. If the unit does not contain the necessary meter template, it can be downloaded from the https://perseus.hw-group.com portal and manually uploaded to the device via the configuration templates page, where you can also see a list of already uploaded templates and there is also the option to delete them.

Uploaded templates are stored in a part of the memory that is not subject to wiping the device to factory settings, and manually uploaded templates can only be deleted here.

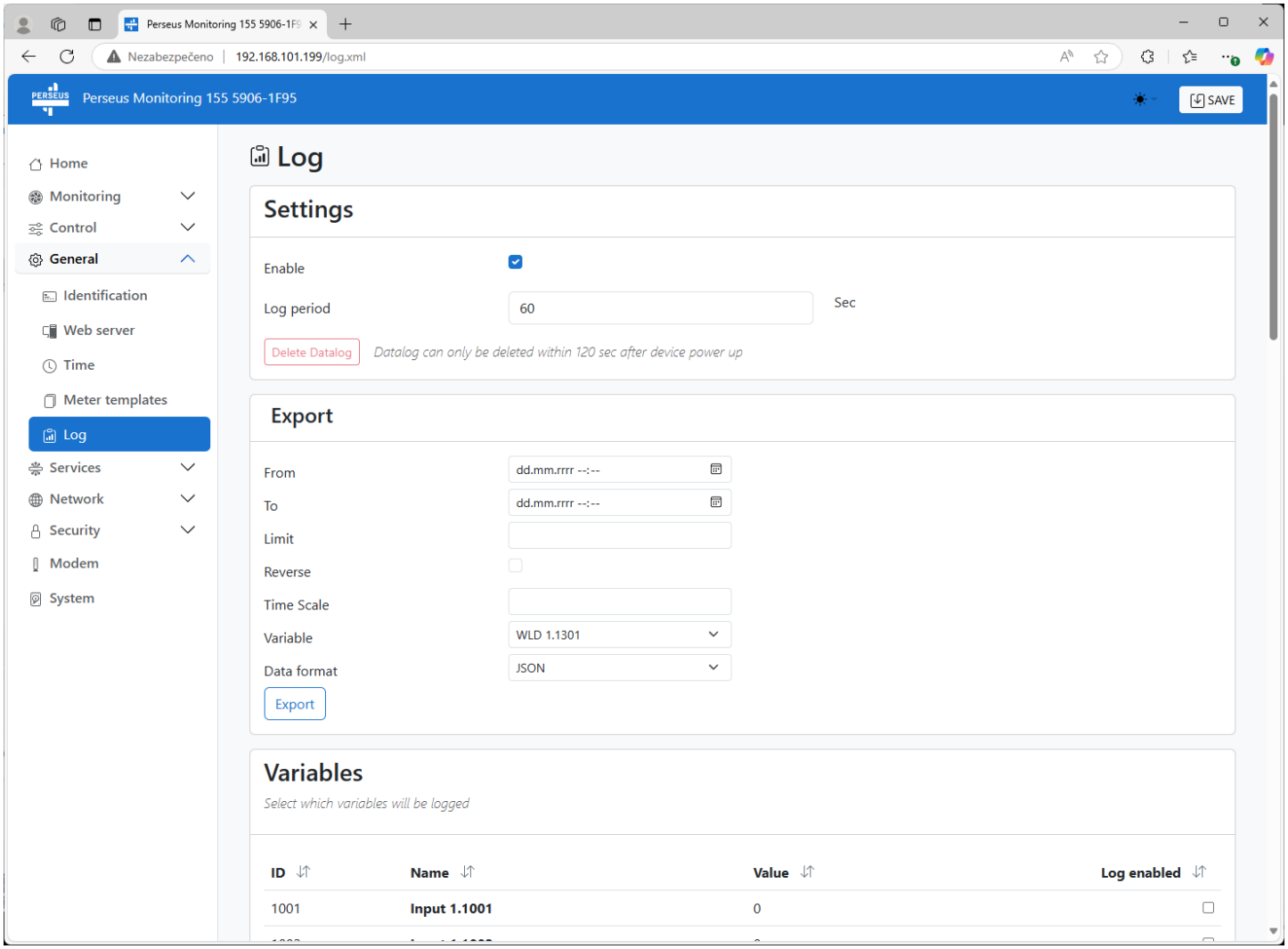


Template assignments

An overview of assigned Templates to the Meters for diagnostic purposes.


Perseus Monitoring 155 Manual
192.168.103.154/cfg_template.xml#tab2
Perseus Monitoring 155 Manual
SAVE
Home
Monitoring
Control
General
Identification
Web server
Time
Meter templates
Services
Network
Security
Modem
System
Configuration Templates
Templates can be downloaded from template repository
User Template Files
Template assignments
List of meters and their parent templates. Only meter supporting configuration templates are displayed.
ID
Name
Template ID
401
Modbus/RTU Comet T3411
cHiOgkSW
402
Modbus STH 102
HJD5iIBG
403
Inepro PRO1-Mod
RatTvnUw
404
Regmet RK-CHM-D
paNWZg1x
HW group s.r.o. ©2024
FW version 0.8.8_9502

Menu General / Log



This page is used for working with the datalog. It allows to enable which Variables will be logged, the logging period and also allows to select and download the ordered values for a specific variable.

Settings section

- **Enable** - Enables the logging function of the measured values. Which values will be logged must be selected in the Variables section
- **Log Period** - Value logging period. Global period for all meters and variables. The minimum period is 60s.
- **Delete Datalog** – Button to delete datalog. To prevent accidental data deletion, the button is only active for 120s after the device is powered on or restarted

Export section

- **From** - Required date and time for the start of the selection of the required values. *If not selected, all data from the beginning of logging is exported*
- **To** - Required date and time for the end of the selection of the required values. If not selected, all data to the ending of logging is exported
- **Limit** – Limit the number of required values. If not selected, all data is exported

- **Reverse** – Rotate the order of values in the export. The standard order is from oldest (top) to newest (bottom). The Reverse field allows you to reverse the order.
- **Time Scale** – Measurement scale - Allows you to issue, for example, every 5 logged values. If not selected, all data is exported
- **Variable**– Selection of a specific variable whose data is to be exported
- **Data format** – Selection of exported data format - CSV/JSON
- **Export** – Button to download exported data

Variables section

- **ID** - ID of the variable
- **Name** - Variable name - clickable to the variable detail where the datalog can be viewed in graphical form. see Variable
- **Value** – Current value of the variable.
- **Log Enabled** – Enables the function of recording the measured values of this variable.

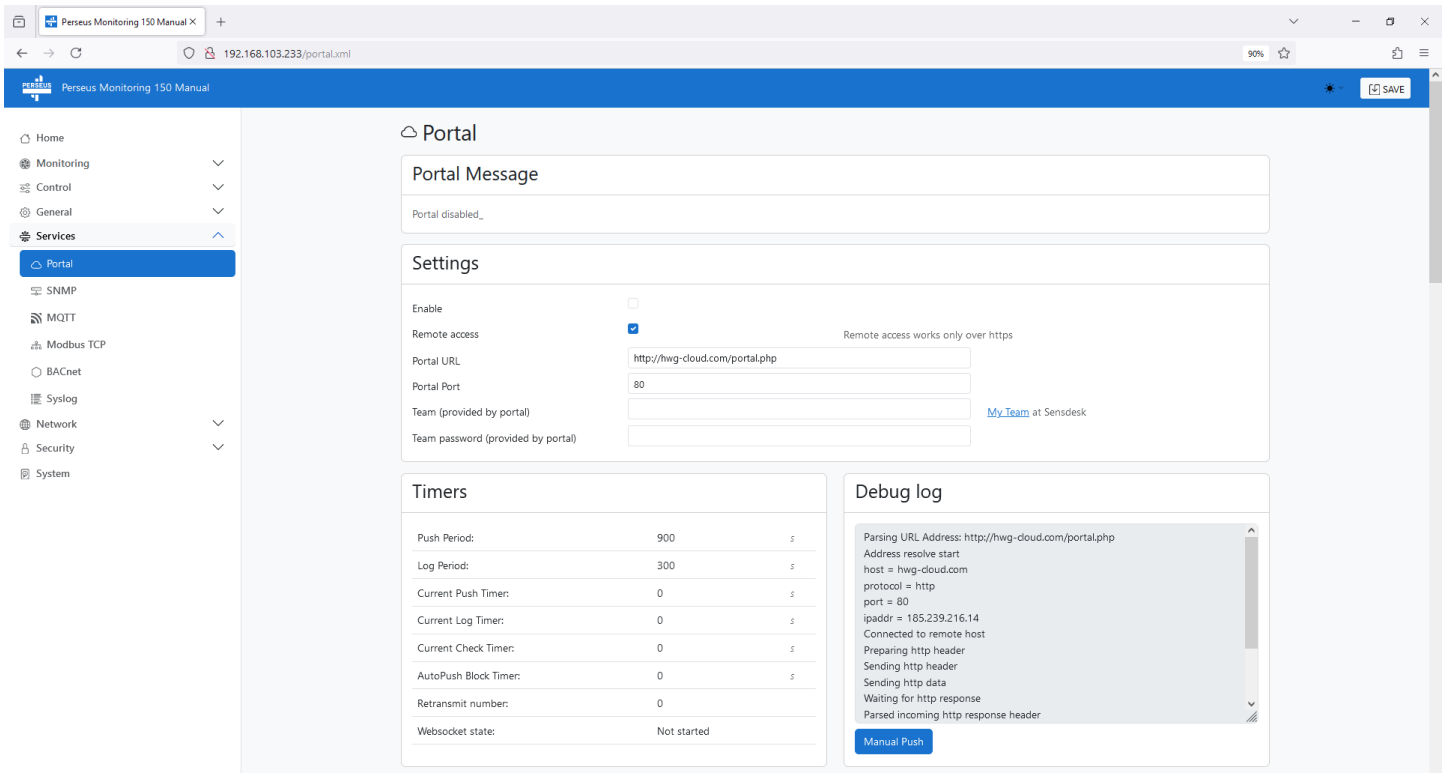
More about the Datalog on the page Datalog.

Menu Services / Portal

Setting up communication with the portal using the HWg-Push protocol. In communication with the portal, the device is an active element and automatically initiates communication and sends data to the portal. This happens at regular intervals set from the portal, when any alarm starts or ends, when the output status changes, or when the value changes by more than the set value.

Due to the fact that Perseus units can work with a very high number of variables, but usually not all loaded variables are monitored, unlike other devices of the Perseus HW group unit, they do not automatically send all measured variables to the portal, but only those for which this function is manually enabled in the Variables section are sent.

The connection to the hwg-cloud.com portal is pre-filled in the device.



Portal Message section

Feedback from the portal, including, for example, links to charts, etc. It depends on the type of portal.

Settings section

- **Enable** – enable or disable this feature.
- **Portal URL** - The full URL of the remote server. It is necessary to add the http://, optionally https:// and change the port number according to the type of communication.
- **Portal Port** – The port on which the portal is listening.
- **Team (provided by portal)** – The name of the team to which it should be assigned. You can find it by logging in to the portal on the My Team page.

- **Team password (provided by portal)** – the password for team assignment. You can find it after logging in to the portal on the My Team page.

Timers section

- **Push Period** - displays the period set by the portal to send data to the portal
- **Log Period** - displays the period of saving the measured data, which will be sent to the portal after the Push period
- **Current Push Timer** – shows how long it will take for the next standard data to be sent.
- **Current Log Timer** – shows how long it will take for the next data caching.
- **Current Check Timer** – show how long it will take for the next check of orders to change the status of outputs.
- **Autopush Block Timer** – shows how long it takes after sending an AutoPush to send additional information using AutoPush – the period is set retrospectively from the portal.

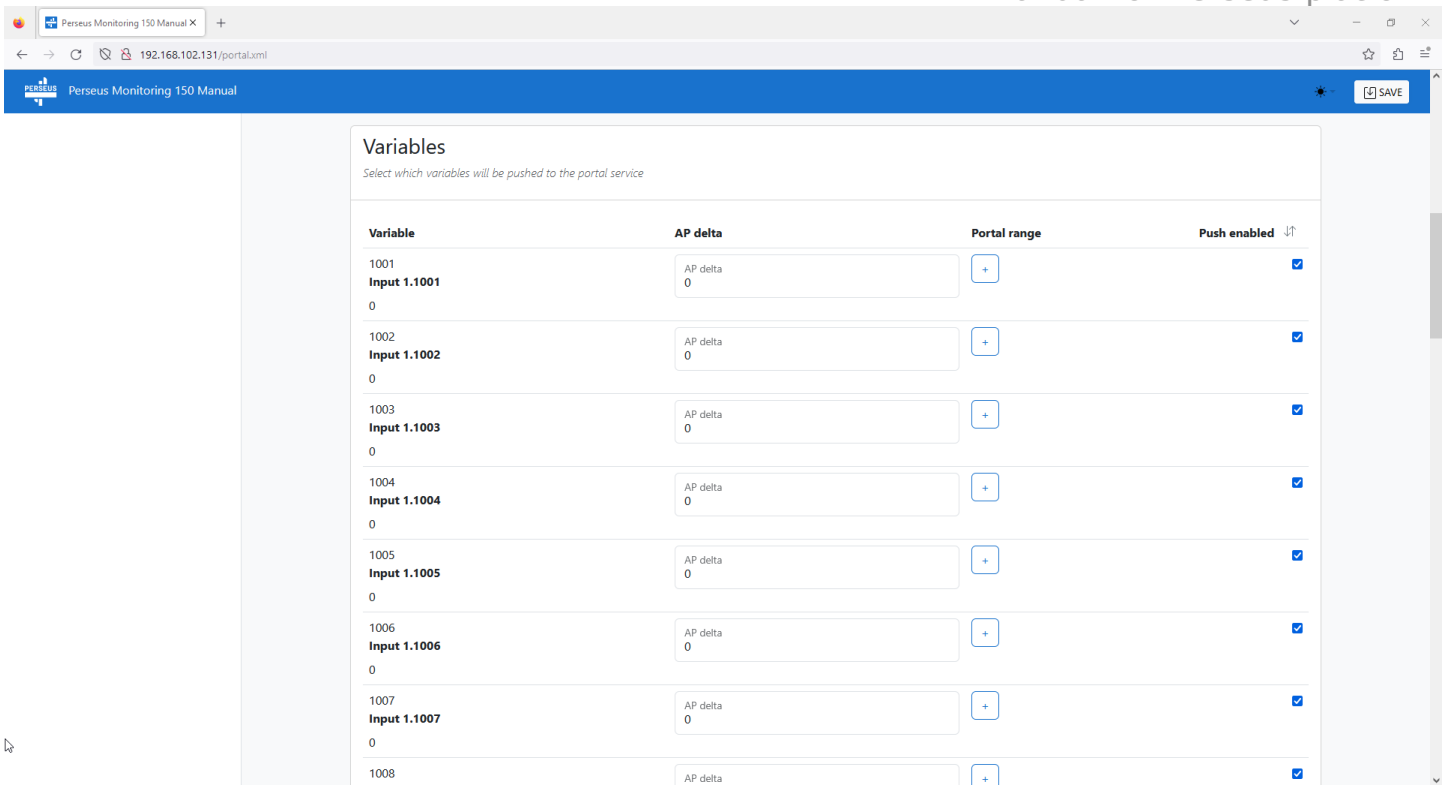
Debug Log section

Window for debugging device-portal communication

- **Manual Push** – button to manually send data to the portal immediately.

Variables section

- **Push enable** – enable or disable the sending of the tag to the Sensdesk technology portal. This allows you to send only monitored parameters to the portal.
- **AP delta** – if there is an increase in value more than specified by the Autopush delta parameter since the last time data was sent to the portal, an extraordinary data push will occur. If the parameter is 0, the function is disabled for this variable.
- **Portal range** – set an optional range of values to be used for working with the portal. If not set, the state of the tag is not evaluated for the purposes of working with the *Sensdesk Technology portal*. If set, it is possible to monitor the status of the variable and, if the set range is exceeded, to call up an extrapush to the portal. Since evaluation is not necessary for most parameters, *the Portal range is not* set by default. The parameters are identical to the parameters in the Conditions section:
  - **Min** – Minimum allowed value
  - **Max** – Maximum allowed value
  - **Hysteresis** – Area of inertia
  - **Delay** – delay for activation of the AlaramHi or AlarmLo state
  - **State** – Displays the status of the condition after evaluation. For a description of variable states, see chapter Meter states, variables and conditions.


Perseus Monitoring 150 Manual
192.168.102.131/portal.xml
Perseus Monitoring 150 Manual
SAVE
Variables
Select which variables will be pushed to the portal service
Variable
AP delta
Portal range
Push enabled
1001
Input 1.1001
0
AP delta
0
1002
Input 1.1002
0
AP delta
0
1003
Input 1.1003
0
AP delta
0
1004
Input 1.1004
0
AP delta
0
1005
Input 1.1005
0
AP delta
0
1006
Input 1.1006
0
AP delta
0
1007
Input 1.1007
0
AP delta
0
1008
AP delta

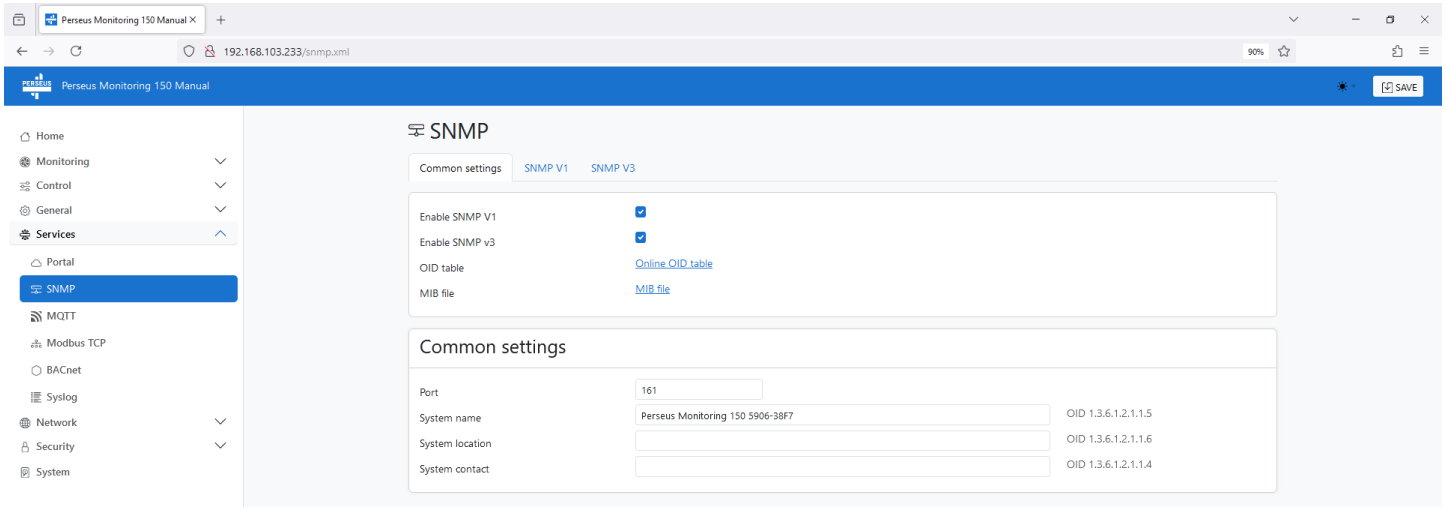
Menu Services / SNMP

The page contains the settings of the SNMP agent in the Perseus unit.

Common settings

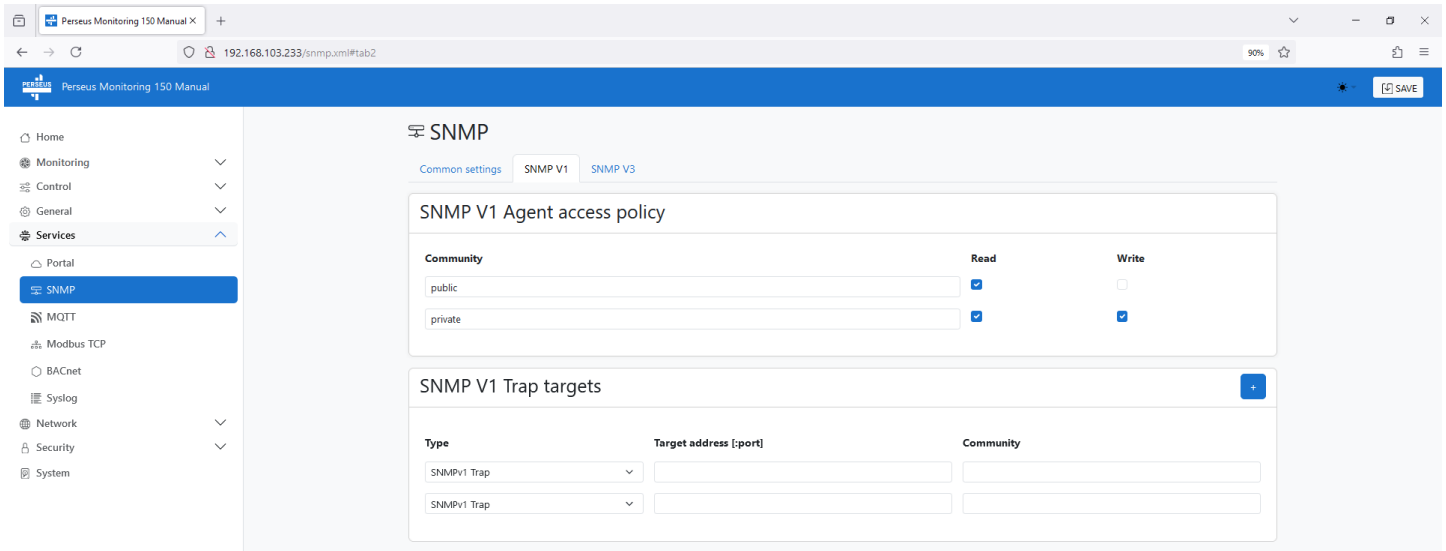
The page contains the option to enable the SNMPv1 and SNMPv3 agents (a separate tab is created for each version), the option to set the port on which SNMP listens and the definition of three system variables System name, System location and System contact.

- **OID table** – Opens a page with an overview of available OIDs and their meanings. – see SNMP OIDs page
- **MIB File** – download MIB file



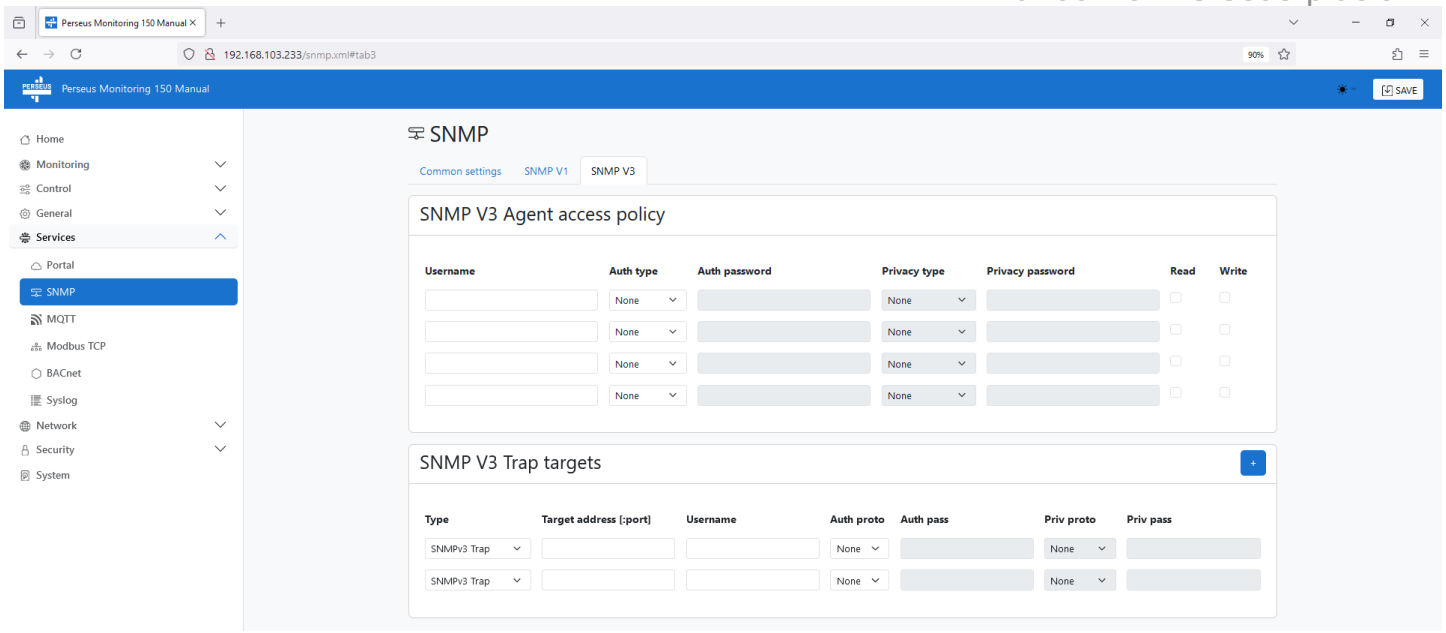
SNMP v1

It is used to set up SNMP v1 community, including their permissions.

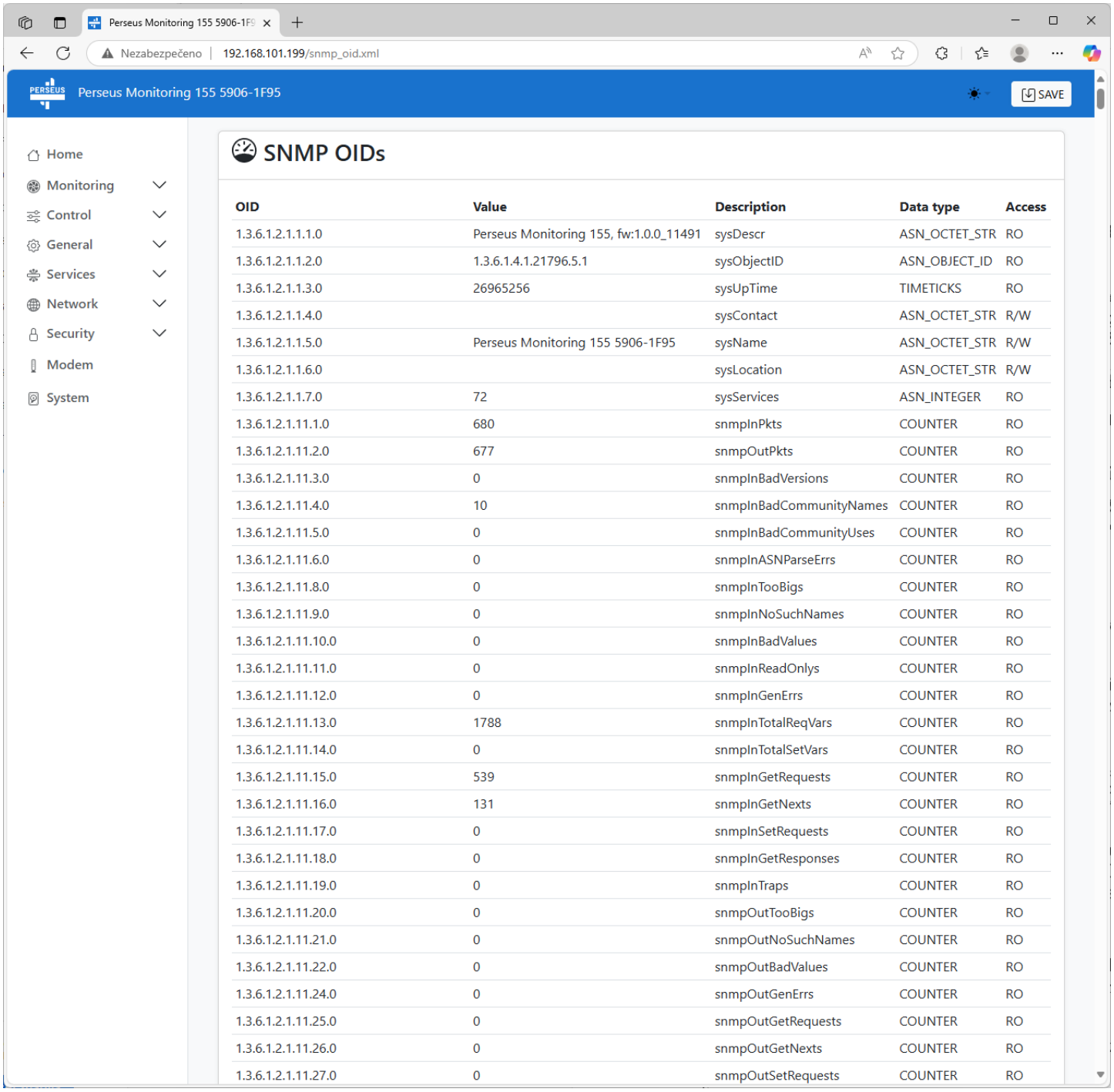


SNMP v3

It is used to set up SNMP v3 users and encryption, including their permissions.


Perseus Monitoring 150 Manual
192.168.103.233/snmp.xml#tab3
90%
Perseus Monitoring 150 Manual
SAVE
Home
Monitoring
Control
General
Services
Portal
SNMP
MQTT
Modbus TCP
BACnet
Syslog
Network
Security
System
SNMP
Common settings
SNMP V1
SNMP V3
SNMP V3 Agent access policy
Username
Auth type
Auth password
Privacy type
Privacy password
Read
Write
None
SNMP V3 Trap targets
Type
Target address [:port]
Username
Auth proto
Auth pass
Priv proto
Priv pass
SNMPv3 Trap

SNMP OIDs page

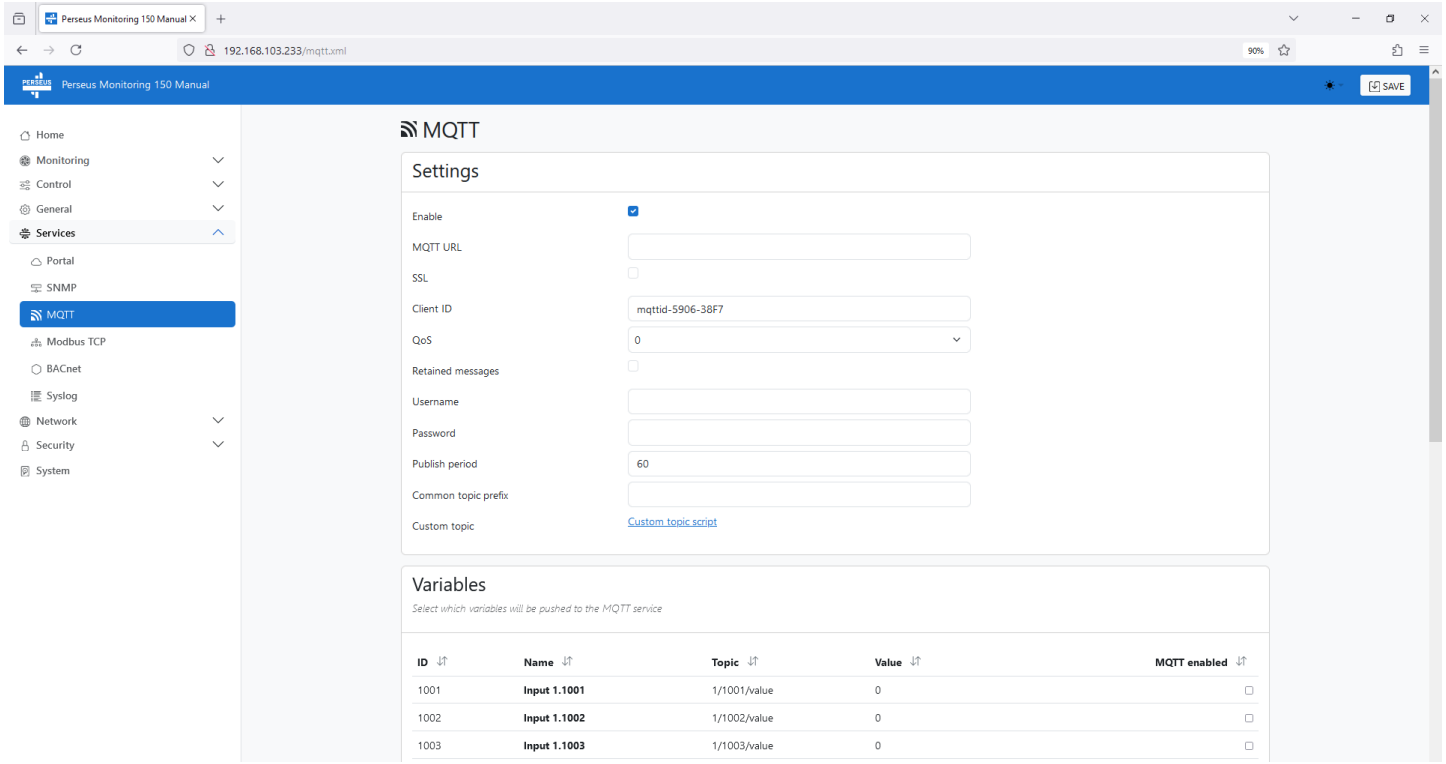


OID	Value	Description	Data type	Access
1.3.6.1.2.1.1.1.0	Perseus Monitoring 155, fw:1.0.0_11491	sysDescr	ASN_OCTET_STR	RO
1.3.6.1.2.1.1.2.0	1.3.6.1.4.1.21796.5.1	sysObjectID	ASN_OBJECT_ID	RO
1.3.6.1.2.1.1.3.0	26965256	sysUpTime	TIMETICKS	RO
1.3.6.1.2.1.1.4.0		sysContact	ASN_OCTET_STR	R/W
1.3.6.1.2.1.1.5.0	Perseus Monitoring 155 5906-1F95	sysName	ASN_OCTET_STR	R/W
1.3.6.1.2.1.1.6.0		sysLocation	ASN_OCTET_STR	R/W
1.3.6.1.2.1.1.7.0	72	sysServices	ASN_INTEGER	RO
1.3.6.1.2.1.11.1.0	680	snmpInPkts	COUNTER	RO
1.3.6.1.2.1.11.2.0	677	snmpOutPkts	COUNTER	RO
1.3.6.1.2.1.11.3.0	0	snmpInBadVersions	COUNTER	RO
1.3.6.1.2.1.11.4.0	10	snmpInBadCommunityNames	COUNTER	RO
1.3.6.1.2.1.11.5.0	0	snmpInBadCommunityUses	COUNTER	RO
1.3.6.1.2.1.11.6.0	0	snmpInASNParseErrs	COUNTER	RO
1.3.6.1.2.1.11.8.0	0	snmpInTooBigs	COUNTER	RO
1.3.6.1.2.1.11.9.0	0	snmpInNoSuchNames	COUNTER	RO
1.3.6.1.2.1.11.10.0	0	snmpInBadValues	COUNTER	RO
1.3.6.1.2.1.11.11.0	0	snmpInReadOnlys	COUNTER	RO
1.3.6.1.2.1.11.12.0	0	snmpInGenErrs	COUNTER	RO
1.3.6.1.2.1.11.13.0	1788	snmpInTotalReqVars	COUNTER	RO
1.3.6.1.2.1.11.14.0	0	snmpInTotalSetVars	COUNTER	RO
1.3.6.1.2.1.11.15.0	539	snmpInGetRequests	COUNTER	RO
1.3.6.1.2.1.11.16.0	131	snmpInGetNexts	COUNTER	RO
1.3.6.1.2.1.11.17.0	0	snmpInSetRequests	COUNTER	RO
1.3.6.1.2.1.11.18.0	0	snmpInGetResponses	COUNTER	RO
1.3.6.1.2.1.11.19.0	0	snmpInTraps	COUNTER	RO
1.3.6.1.2.1.11.20.0	0	snmpOutTooBigs	COUNTER	RO
1.3.6.1.2.1.11.21.0	0	snmpOutNoSuchNames	COUNTER	RO
1.3.6.1.2.1.11.22.0	0	snmpOutBadValues	COUNTER	RO
1.3.6.1.2.1.11.24.0	0	snmpOutGenErrs	COUNTER	RO
1.3.6.1.2.1.11.25.0	0	snmpOutGetRequests	COUNTER	RO
1.3.6.1.2.1.11.26.0	0	snmpOutGetNexts	COUNTER	RO
1.3.6.1.2.1.11.27.0	0	snmpOutSetRequests	COUNTER	RO

Menu Services / MQTT

The page contains the MQTT Publisher settings in the Perseus unit.

Due to the fact that Perseus units can work with a very high number of variables, but usually not all of the loaded variables are monitored, unlike other devices of the Perseus HW group units, they do not automatically send all measured variables to the MQTT broker, but only those for which this function is manually enabled in the Variables section are sent.



Settings section

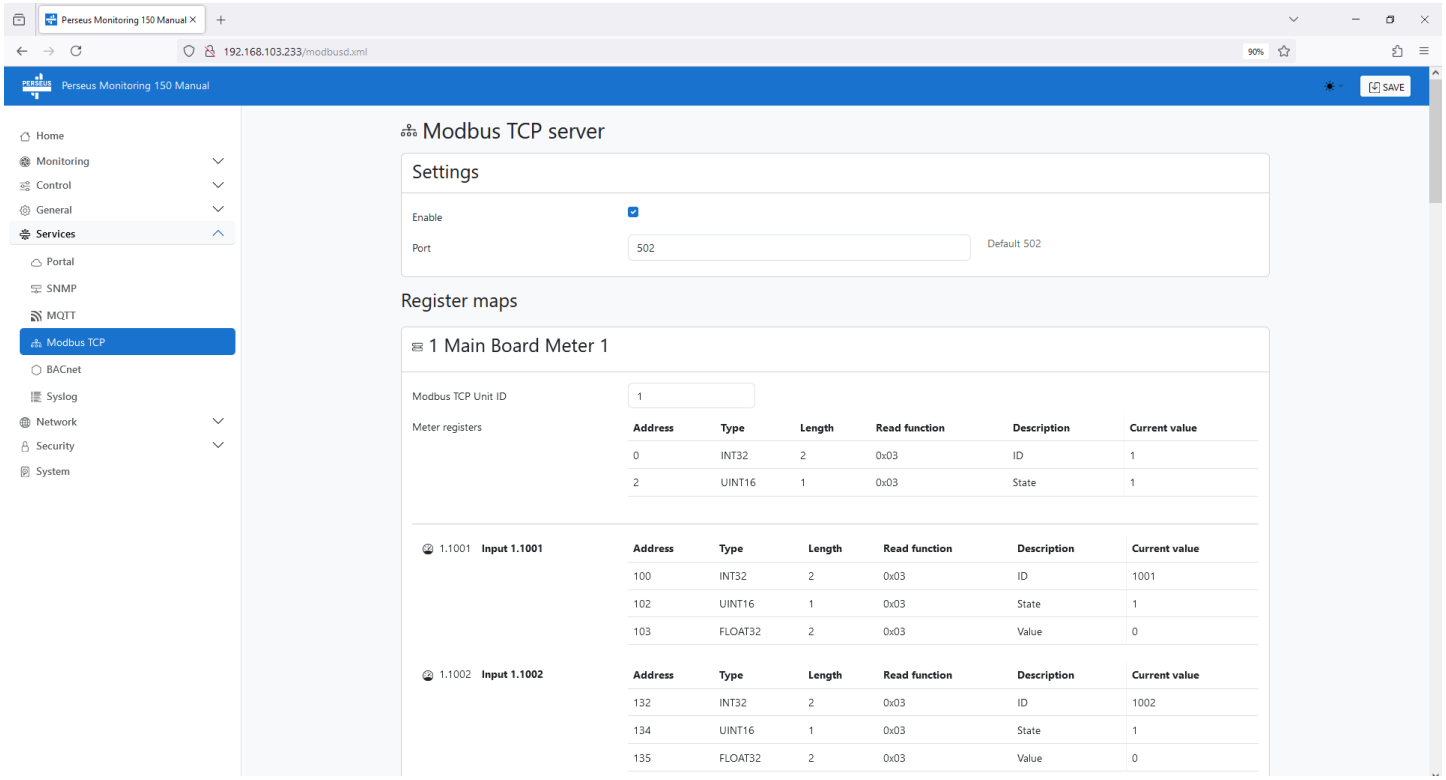
- **Enable** – enable or disable the MQTT data sending function.
- **MQTT URL –** IP address or domain address of the server with the MQTT broker. (for example test.mosquitto.org:1884)
- **SSL** – enable or disable SSL support when communicating with the MQTT broker.
- **Client ID** – ID of Perseus in MQTT network.
- **QoS** – Quality of Service (Perseus in publisher mode have always "0")
- **Retained messages** – MQTT Flag that tells MQTT broker to keep last value for any new subscriber, so it doesn't wait for new published value but instead the new subscriber receives last known value.
- **Username** – set the username to log in to the MQTT broker.
- **Password** – set the password to log in to the MQTT broker.
- **Publish Period** – the period of sending data to the MQTT broker.
- **Common Topic prefix** – prefix for MQTT Topic (address).

Variables section

- **ID** – ID of the relevant variable

- **Name** – The name of the variable
- **Topic** – the complete Topic after the prefix
- **Value** – The current value
- **MQTT enabled** – enable sending of a specific variable

Menu Services / Modbus TCP



This page shows available options for Modbus TCP server communication.

Since Perseus units can handle a very large number of Variables, the published Variables remain grouped within individual Meters, and in Modbus TCP, Meters can be distinguished by Meter ID or Modbus TCP unit ID. This allows configuration portability between Perseus units while maintaining the same register addresses across all units regardless of the number of Meters and Variables.

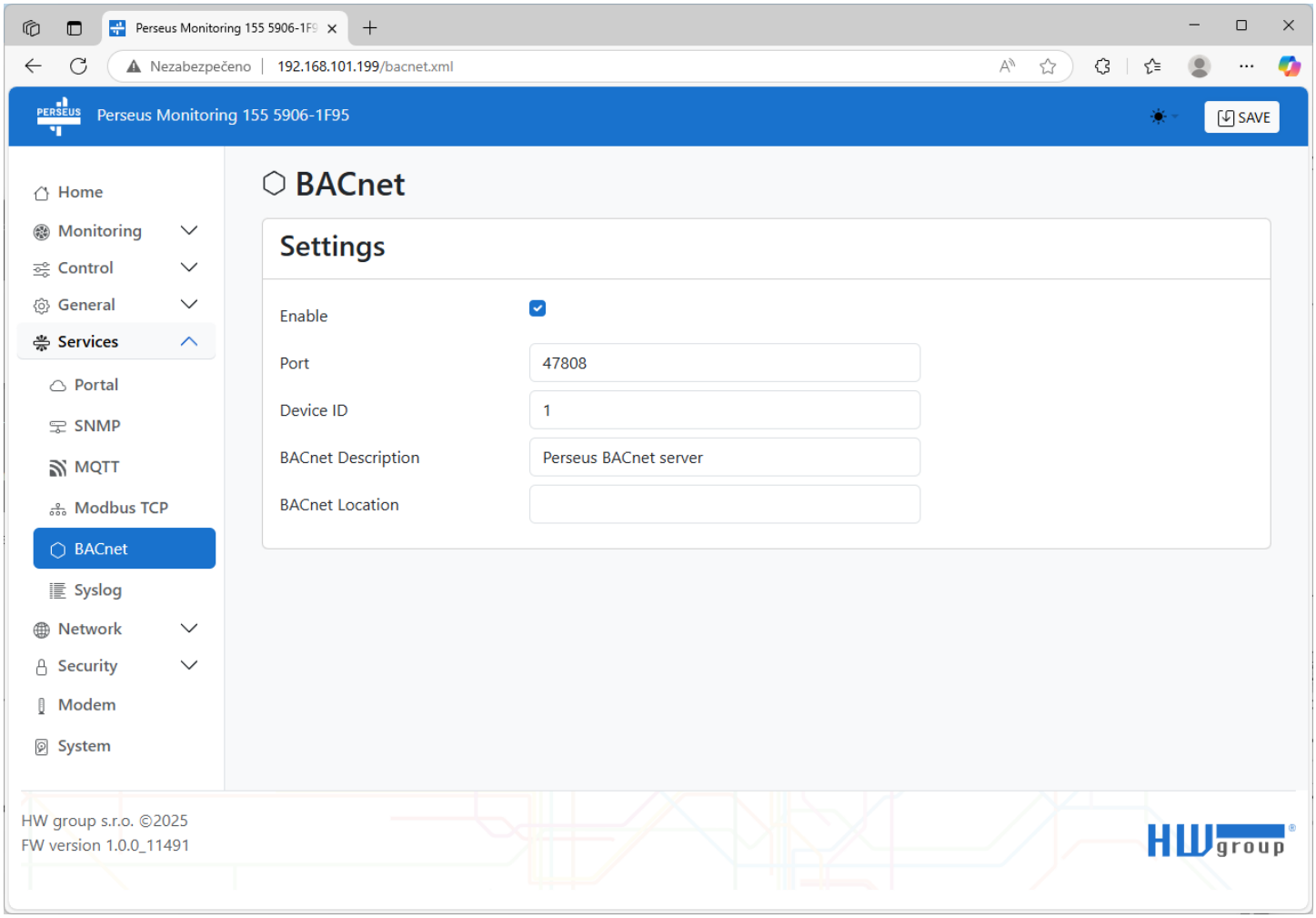
Settings section

- **Enable** – Enable/Disable the TCP server of Perseus
- **Port** – IP Port used for Modbus TCP server; default is 502

Register maps

- **Modbus TCP Unit** – Every Meter has its own ID; this allows Perseus to support almost any number of meters Perseus reads and provide them to the Modbus TCP clients
- **Meter registers** – predefined map of registers for existing meters Perseus have set, including their Address, type, Length, Read function, Description and Current state

Menu Services / BACnet



This page displays the available options for BACnet/TCP server communication.

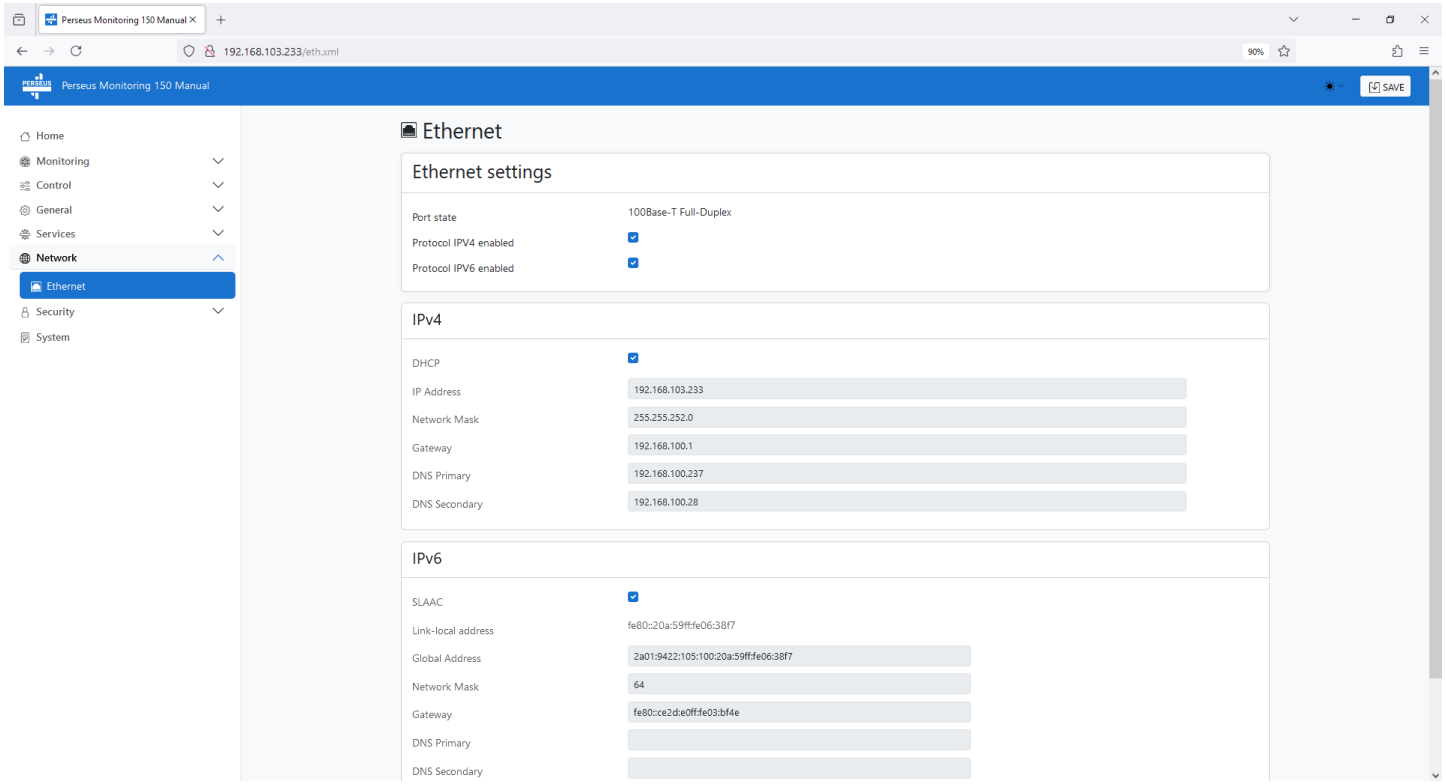
Settings section

- **Enable** – Enable/Disable the TCP server of Perseus
- **Port** – IP Port used for Modbus TCP server; default is 502
- **Device ID** – BACnet device ID
- **BACnet Decsription** – Description of device for identification in BACnet network

BACnet Location – Location of device for identification in BACnet network

Menu Network / Ethernet

On the Ethernet page, you can enable Ethernet communication over IPv4 and IPv6 and possibly set network parameters such as IP address, mask, gateway, etc.



Ethernet Settings Section

The block contains information about the status of the Ethernet connection and the option to enable IPv4 and IPv6 protocols

IPV4 Section

The block contains basic settings of network parameters for Ethernet communication:

- **DHCP** - Enables the IP address setting function of the DHCP server, if available. Enabling or disabling DHCP depends on the needs of the user and the network administrator.
- **IP address** – set the IP address of the unit, it is necessary to restart the device after changing the settings.
- **Network Mask** – set the local network mask, after changing the settings it is necessary to restart the device.
- **Gateway** – the default gateway for communication, it is necessary to restart the device after changing the settings.
- **DNS Primary/DNS Secondary** – set up the primary and secondary DNS servers, the Gateway must be set up correctly for proper functioning. A DNS server is required to convert a domain name to an IP address, without its settings the following services will not work:
  - Time synchronization (SNTP) is used in emails and SNMP traps for the time of the event.
  - E-mail Sending (SMTP).
  - Logging of values with time stamp.

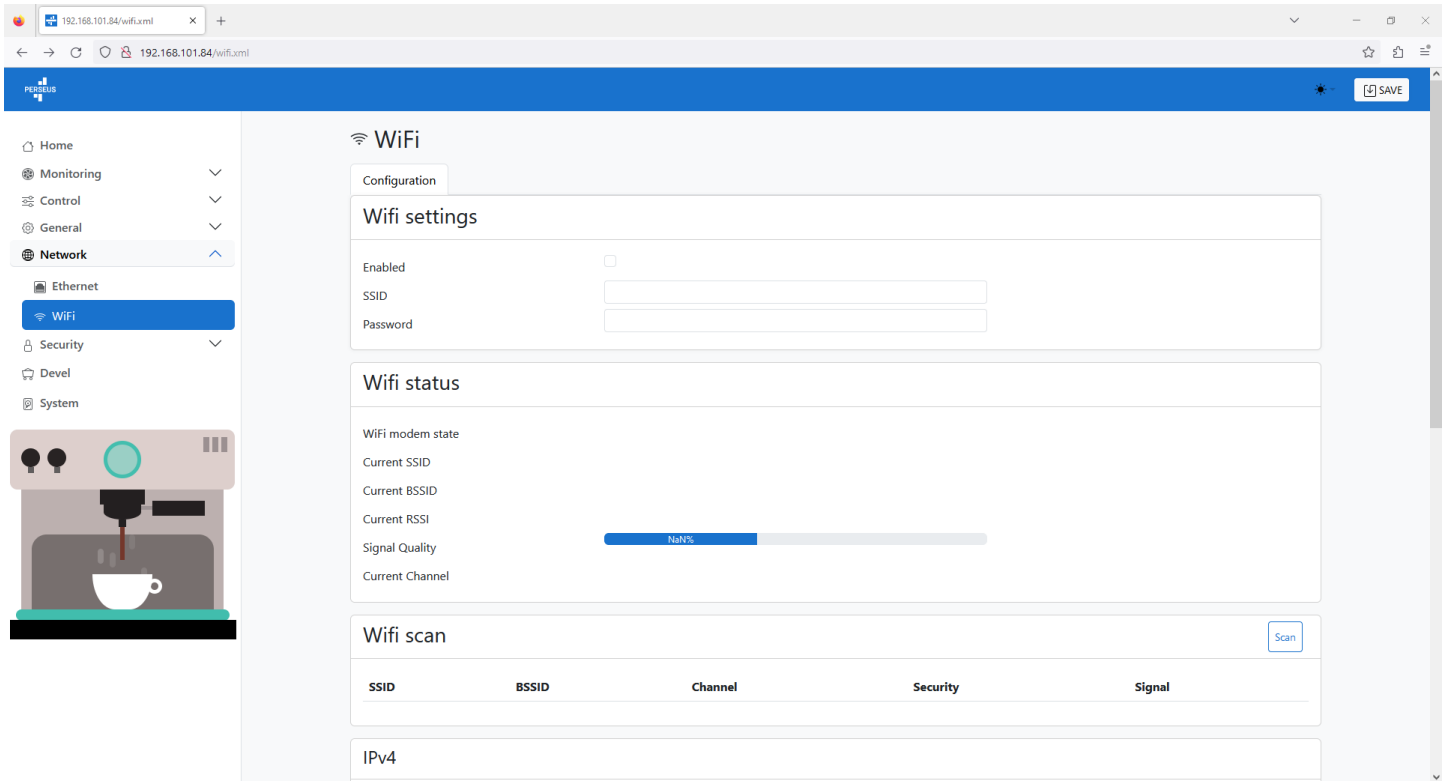
IPV6 Section

The block contains options for working on IPV6 networks.

- **SLAAC** - Enables support for automatic network configuration using SLAAC.
- **Link-local address** – link address of the device – visible only from the internal network.
- **Global Address** – global (public) IP address and network prefix. It consists of a line address and a network prefix.
- **Gateway** – the default gateway for communication, it is necessary to restart the device after changing the settings.
- **DNS Primary/DNS Secondary** – setting up the primary and secondary DNS servers, the Gateway must be set up correctly for proper operation.

Menu Network / WiFi

Setting up a WiFi connection – WiFi-equipped models only.



WiFi Setup Section

- **Enable** – Turns WiFi on or off. The wireless interface is turned off by default.
- **SSID** – The name of the WiFi network to which the device is to be connected. If you don't know your network name, use the Scan AP function at the bottom of the page.
- **Password** – Network security password. If you don't know it, contact your network administrator.

WiFi status section

WiFi modem state

- *Disable* – WiFi off.
- *Wait for power on* – wait for the WiFi module to turn on.
- *Init* – initialize the WiFi module.
- *Connecting*.
- *SSID check* – controlling the SSID.
- *Connected* – Connected to the selected WiFi network.
- *Network wifi scan* – scan available WiFi networks.
- *Wait for scan* – wait for WiFi scanning to start.

- **Current SSID** – The current name of the network to which the device is connected. If the parameter is missing, the device is not connected to any WiFi network.
- **Current BSSID** – The current identifier of the WiFi connection point of the device is connected to. If the parameter is missing, the device is not connected to any WiFi network.

- **Current RSSI** – The relative strength of the received signal. The higher the RSSI, the stronger the signal.
- **Signal Quality** – WiFi signal strength in % with graphical indicator.
- **Current Channel** – the WiFi channel on which the device communicates. If the parameter is missing, the device is not connected to any WiFi network.

WiFi Scan section

- **SSID** – The name of the WiFi network found.
- **BSSID** - Mount Point Identifier (MAC Address).
- **Channel** – set the WiFi channel on which the access point communicates.
- **Security** – type of WiFi communication security.
- **Signal** – set the signal level in DB. The higher the value, the better. ATTENTION, -60 is more than -90! A highlighted line is provided.

IPV4 Section

The section contains basic settings of network parameters for communication:

- **IP address** – set the IP address of the unit, it is necessary to restart the device after changing the settings.
- **Network Mask** – set the local network mask, after changing the settings it is necessary to restart the device.
- **Gateway** – the default gateway for communication, it is necessary to restart the device after changing the settings.
- **DNS Primary/DNS Secondary** – set up the primary and secondary DNS servers, the Gateway must be set up correctly for proper functioning. A DNS server is required to convert a domain name to an IP address, without its settings the following services will not work:
  - Time synchronization (SNTP) is used in emails and SNMP traps for the time of the event.
  - E-mail Sending (SMTP).
  - Logging of values with time stamp

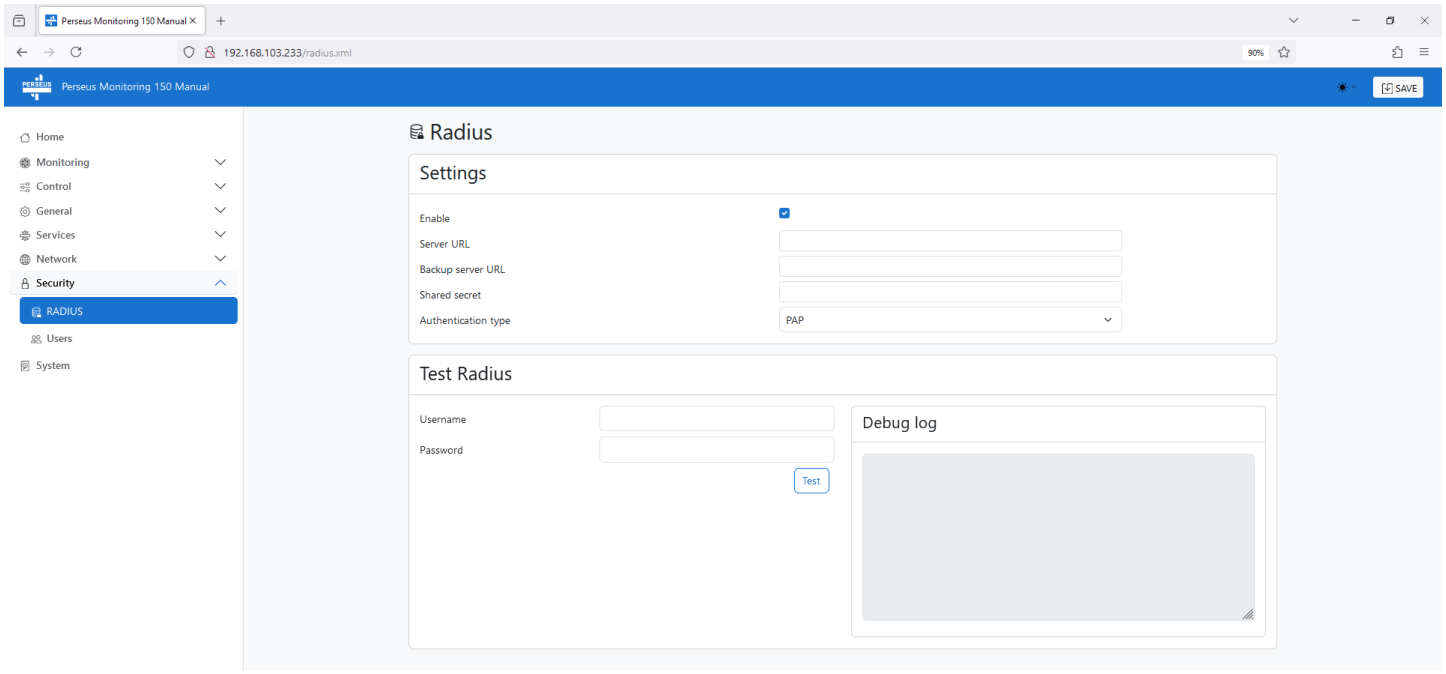
IPV6 Section

This section contains options for working on IPV6 networks.

- **SLAAC** - Enables support for automatic network configuration using SLAAC.
- **Link-local address** – link address of the device – visible only from the internal network.
- **Global Address** – global (public) IP address and network prefix. It consists of a line address and a network prefix.
- **Gateway** – the default gateway for communication, it is necessary to restart the device after changing the settings.
- **DNS Primary/DNS Secondary** – setting up the primary and secondary DNS servers, the Gateway must be set up correctly for proper operation

Menu Security / Radius

Contains parameters for setting up a RADIUS authorization server.



Settings section

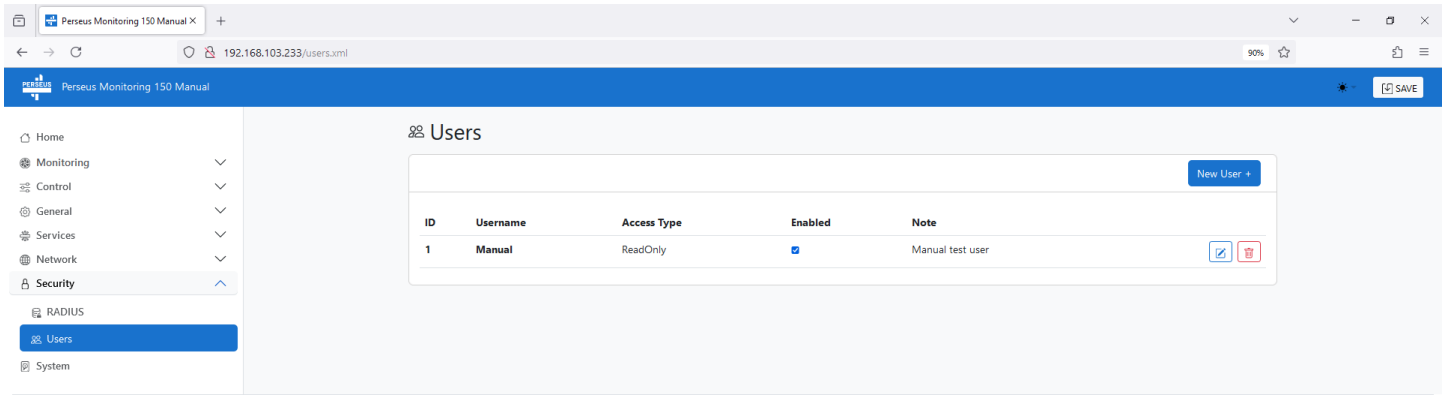
- **Enable** – Enables/disables the client's Radius function
- **URL Server** - The address of the primary authorization server
- **Backup server URL** - Secondary authorization server address (optional)
- **Shared secret** – shared password to the RADIUS server.
- **Authentication type** – type of authorization protocol (PAP/EAP MD5)

Test Radius Section

- **Username** – username to connect to the RADIUS server.
- **Password** – password for connection to the RADIUS server

Menu Security / Users

The page defines users and their rights for working with the Perseus unit.

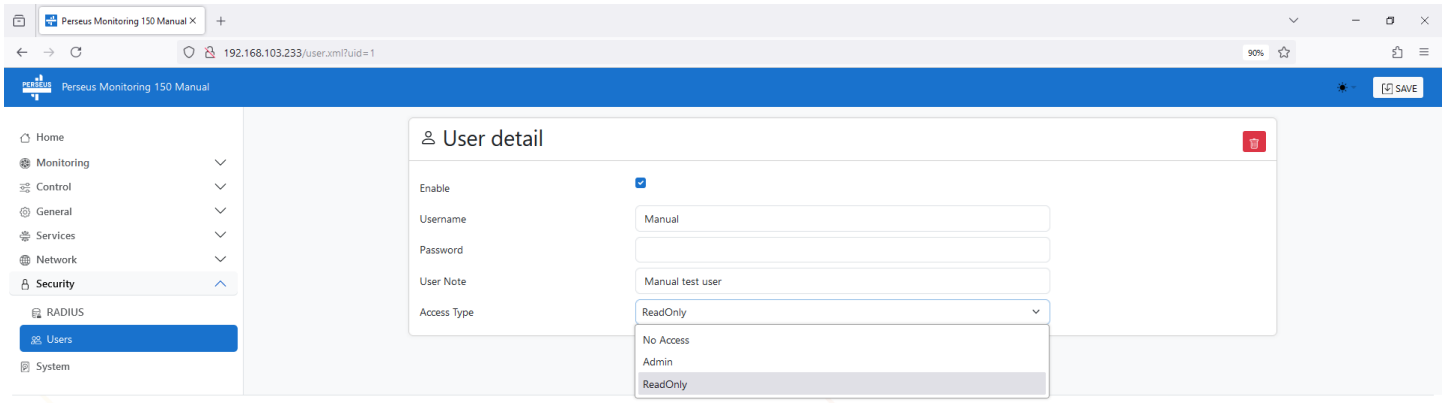


The page contains a clear list of users, their permissions, and the ability to quickly turn them on and off (for example, to remove permissions). It is advisable to delete users in order to preserve the history of their activities within the unit (the audit trail will record logins, parameter changes, etc. over time).

This page coexists with the other one, General - Web server, where you need to select Authentification method.

User Editing

New users can be created by pressing the New Users + button.

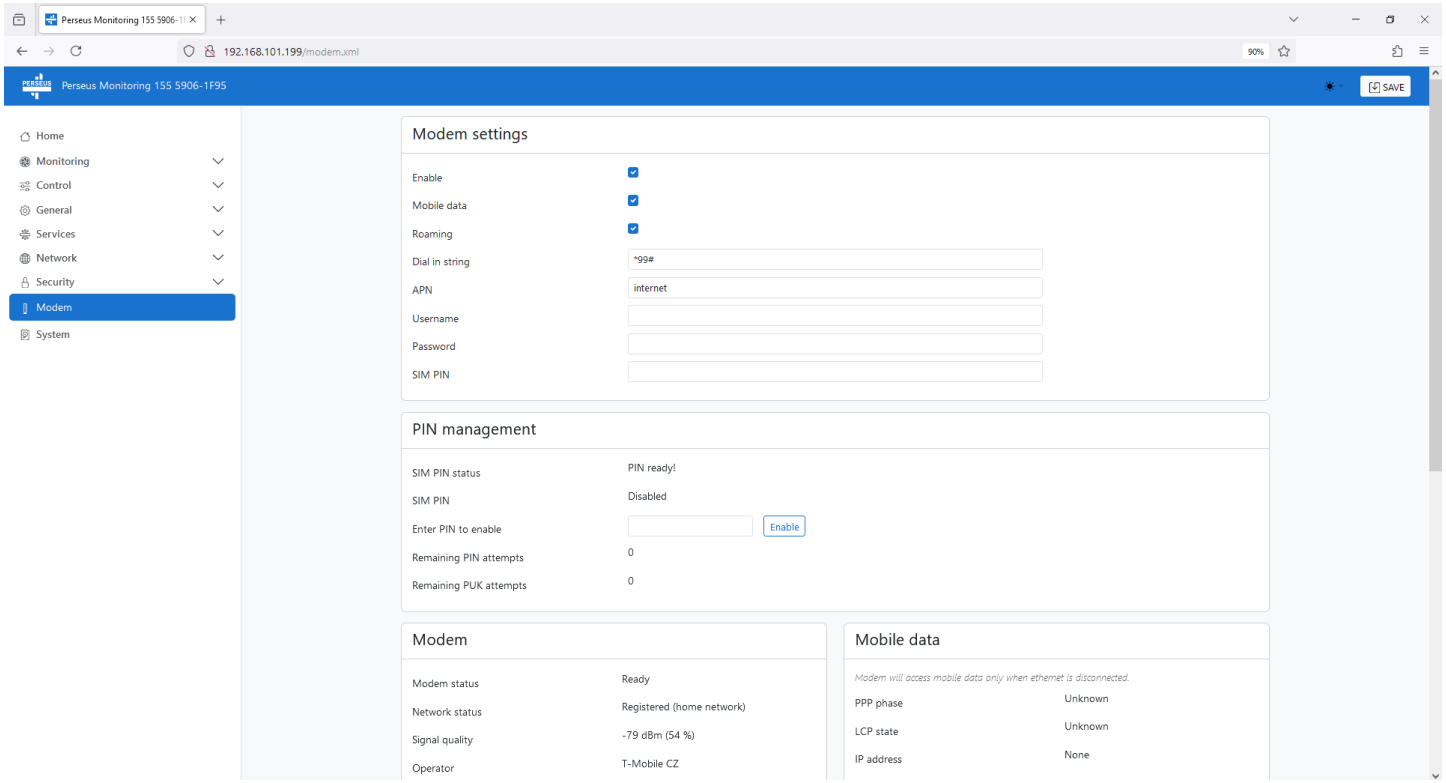


- **Enable** – enable/disable the user
- **Username** – username
- **Password** – password
- **User Note** – A note, e.g. job title
- **Access Type** – Permission type

No Access – the user has no access – equivalent to shutting down
Admin – The user has the option to make any changes to the unit, including downloading and uploading the configuration
Read Only – the user does not have the right to change any settings, nor does he have access to download the configuration and can only browse the website

Menu Modem

Only models with an LTE modem. The page is used to set the parameters of the built-in LTE modem.



Settings section

- **Enable** – Enables/Disables built-in modem
- **Mobile data** - allows data transfers (if disabled, it is only possible to send SMS and ring)
- **Roaming** - Allow device traffic in the roaming network
- **Dial in string** - phone number for dial-up connection (read from SIM or received from operator)
- **APN** - APN of the data network (read from the SIM or received from the operator)
- **Username** - Username to connect to the operator's network
- **Password** - password to connect to the operator's network
- **SIM PIN** – PIN for SIM card (Enter PIN here if its Enabled on SIM)

PIN management section

In this section you can Enable/Disable PIN on SIM card, read status and get remaining PIN/PUK attempts.

Modem section

Contains information about the status of the modem

- **Modem status** – State of LTE modem
- **Network status** – information about device registration in the operator's GSM network.
- **Signal quality** – GSM signal level. The information can help you debug network issues.
- **Operator** – set the name of the operator to which the GSM modem is connected.

- **Access technology** – actualy used connectivity (EDGE/…/LTE)
- **IMEI** – device IMEI
- **Modem fw version** – fw version of LTE modem

Mobile data

- **PPP phase** – status of connection to the operator's data network
- **LCP state** – link connection status
- **IP address** – IP address obtained from the operator

GNSS section

Contains information about the global positioning (only models with special modems)

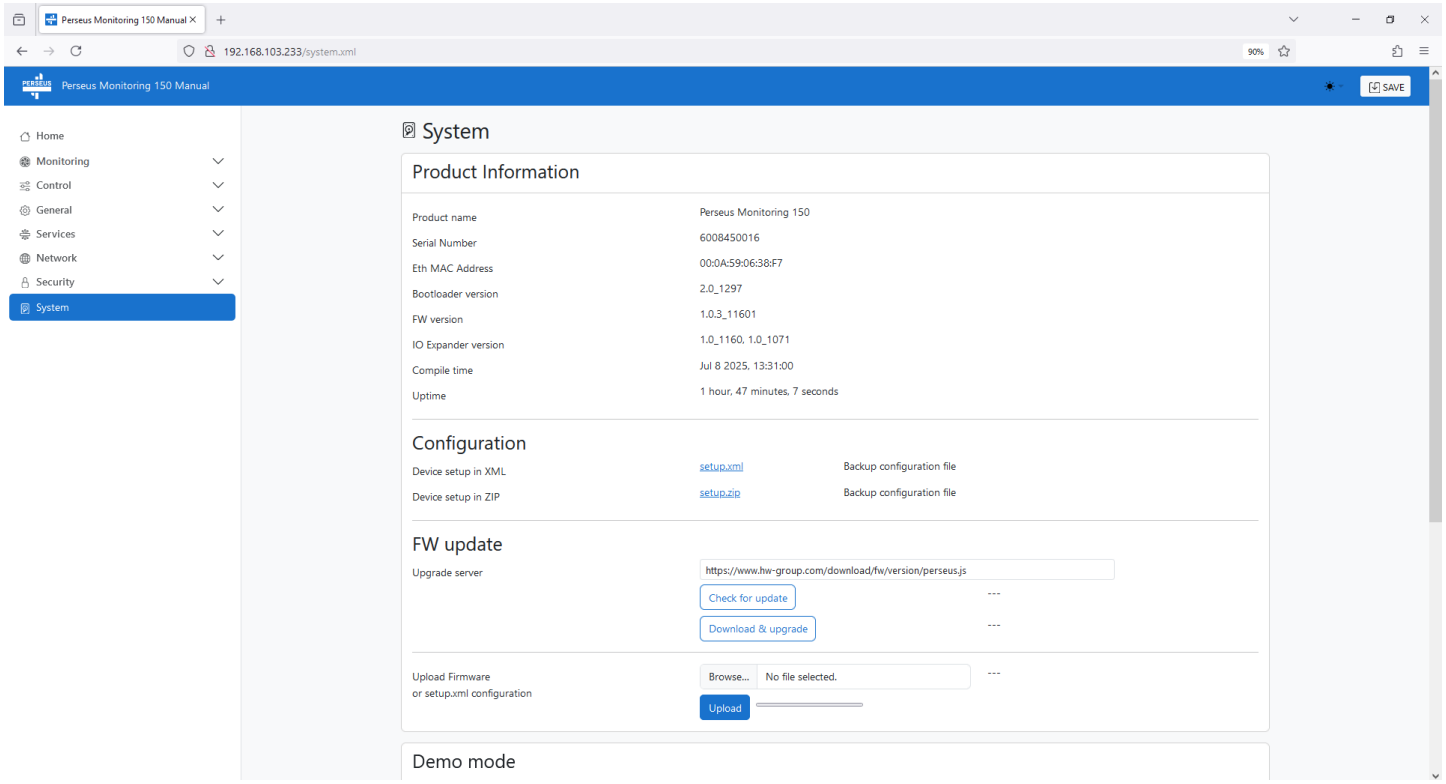
- **Enable** - enable modem position support. The obtained data will be available to the unit as part of the new local meter
- **State** - Receiver status
- **UTC** - read time from the positioning system
- **Latitude** - latitude
- **Longitude** - longitude
- **Precision** - accuracy
- **Altitude** - altitude
- **Speed** - speed of movement
- **Show location in Google Maps** - Link to view Google maps

Debug section

Contains tool for testing modem connectivity when the Ethernet is standard connection. Modem is used for the ping test and result is shown.

- **Ping address** – Enter destination server address in the GSM/LTE network for ping test (for example www.google.com or www.hwg.cz)

Menu System



System Section

- **Product Name** – The name (type) of the device
- **Serial Number** – Serial number of the device
- **Eth MAC Address** – The MAC address of the wired device
- **WiFi STA MAC Address** –MAC address of the device for WiFi connection (only for models equipped with wifi)
- Bootloader version – Bootloader version. It is used for diagnostic purposes in case of troubleshooting
- **FW version** – Firmware and Build version. It is used for diagnostic purposes in case of troubleshooting
- **IO Expander version** – Firmware of Perseus built-in IO Expanders. It is used for diagnostic purposes in case of troubleshooting
- **Compile time** – set the firmware compilation time. It is used for diagnostic purposes in case of troubleshooting
- **UpTime** – The running time of the device since it was last powered on or rebooted. It is used for diagnostic purposes in case of troubleshooting
- **Configuration** – for downloading of device setup (backup, support, copy…)
  - **Device setup in XML** – limited setup excluding Templates (this option may be removed in future)
  - **Device setup in ZIP** – full device setup including Templates packed in the zip archive
- **FW update** – firmware update via the Internet
  - **Upgrade** server – full path to HW group server with an upgrade script
  - **Check for update** – print the latest firmware version on the HW group update server
  - **Download & Upgrade** – start the firmware upgrade from the HW group update server

- **Upload Firmware or Configuration** – Upload newer firmware or configuration file to the device. Restoring the configuration may fail if there is too much difference in firmware versions

Demo mode section

Allows to activate Demo mode secured by password.
In Demo mode the device only shows the saved Meters / Variables / Conditions and Actions and does not allow any changes.

System section

- **Restart** – Restarts the device.
- **Factory Reset** – Restores factory settings. By default, the setting is enabled by DHCP. If the device does not receive an address within 60 seconds of powering on, it defaults to 192.168.10.20. Neither the username nor the password is defined by default.

Description of formats

Datalog

The datalog for measured values in Perseus units currently has a capacity of 32MB which represents approximately 1,000,000 values. Not all values are automatically logged, but only those Variables for which logging is enabled by checking the General => Log page

Data is stored in a uniform period regardless of the reading period. If the period of the Variable or Meter is longer than the logging period, the last known value is logged. The logging period is in relation to system time, so for example log period 28800s (=8hrs) means that the Perseus will save selected values to log in these real times: 00:00, 08:00 and 16:00.

The Datalog contains the following information:

Source ID – 1Byte

Status – 1Byte

Variable ID - 4Bytes

Value – 8Bytes

+ each logging contains 14Byte header containing timestamp, record type and so on + 8byte footer:

14Byte header

14 Byte value of the variable

14 Byte value of the variable

.

.

.

.

14 Byte variable value

14 Byte variable value

8Byte footer

If you are logging only one Variable, 36Bytes are needed for one record. Therefore, for 32MB of available memory, there are 1 000 000 records.

Datalog API

If you don't want to use the WWW download, you can use the API in the form of HTTP GET commands.

The datalog can be retrieved using the selector on the Log page, or by using HTTP Get commands:

URL of the entire datalog

1. /**datalog.json** – for datalog in JSON format
2. /**datalog.csv** – for datalog in CSV format

Parameters to select specific data:

- **vid** - ID of the variable we want the data to be from - e.g.
- **from** - linux timestamp of the time from which the data should be exported
- **to** - linux timestamp of the time to which the data should be exported
- **limit** - limit of the number of exported values. It is used in conjunction with the from or to parameter
- **reverse** - reverse sort order (reverse=0 - sort from oldest to newest, reverse=1 - sort from newest to oldest)
- **time_scale** – interval between exported values. Not all logged values are exported, but e.g. every fifth **- Can only be used in conjunction with a specific variable**

Examples:

http://192.168.101.199/datalog.csv?vid=113001&from=1741122000&to=1741125600&limit=300&reverse=1&time_scale=60

http://192.168.101.199/datalog.json?vid=113001&from=1741122000&to=1741125600&reverse=1&time_scale=60

Datalogs formats

CSV format

```
Timestamp;Source ID;Variable ID;State;Value
1737712860;0;1301;1;0
1737712860;0;2001;1;95
1737712860;0;101001;1;0
1737712860;0;102001;1;22,12
1737712860;0;113001;1;24,06
1737712860;0;114001;1;23,75
1737712860;0;115001;1;32,2
```

JSON format


```
{ "status": "success",
    "data": [
        {
            "time": 1739808660,
            "sensors": [
                {
                    "id": 119001,
                    "data": [1, 24.1]
                }
            ]
        },
        {
            "time": 1739808720,
            "sensors": [
                {
                    "id": 119001,
                    "data": [1, 24.1]
                }
            ]
        },
        {
            "time": 1739808780,
            "sensors": [
                {
                    "id": 119001,
                    "data": [1, 24.1]
                }
            ]
              {
            "time": 1740067860,
            "sensors": [
                {
                    "id": 119001,
                    "data": [1, 25.3]
                }
            ]
        }
    ]
}
```


Read the application note: Logger in the Perseus Platform.

How to add Modbus/RTU 3rd party meters to Perseus units

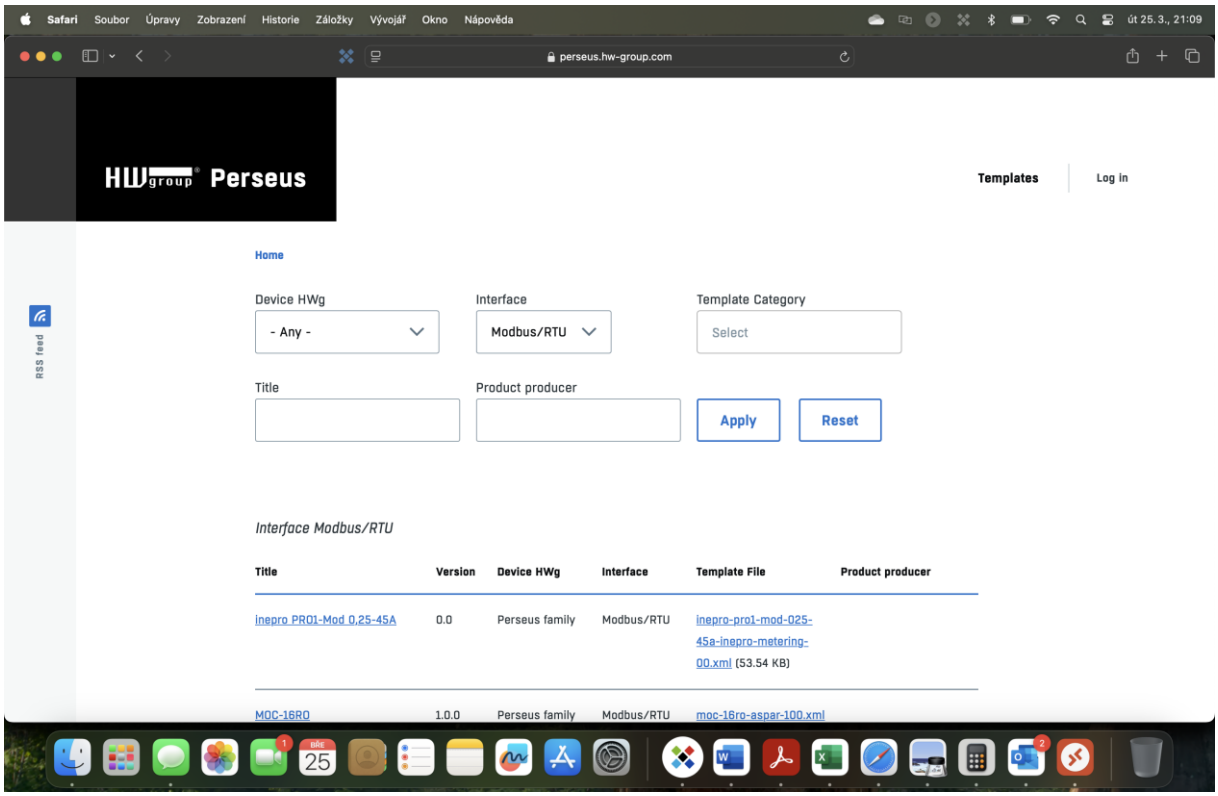
Any Modbus/RTU or Modbus/TCP-enabled device can be added as a meter to Perseus units. As this is an industrial protocol that requires a knowledgeable user, adding them is not always straightforward. This document aims to help users of Perseus units to connect and install Modbus/RTU and Modbus/TCP units from manufacturers other than HW group s.r.o.

There are 2 ways to add a 3rd party Modbus device:

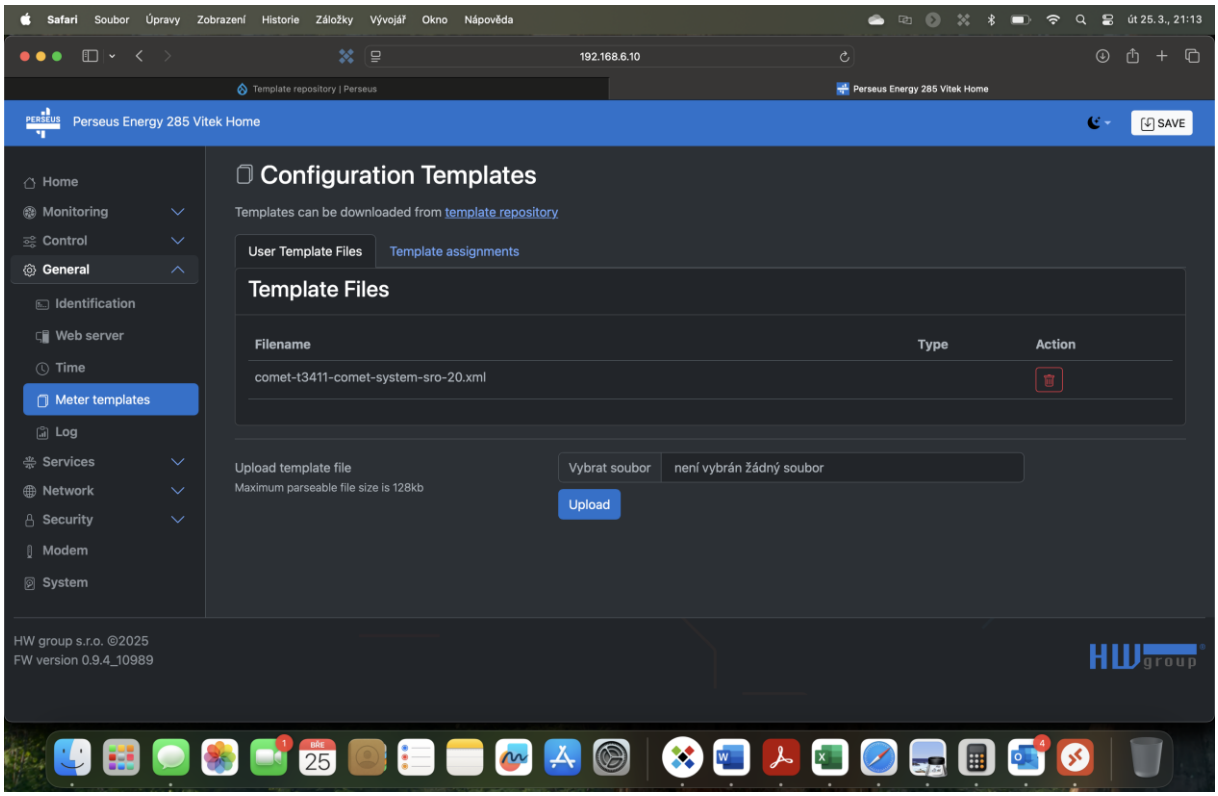
- Using an existing template (Template)
- Manual meter addition (with Template export option)

Using an existing template (Template)

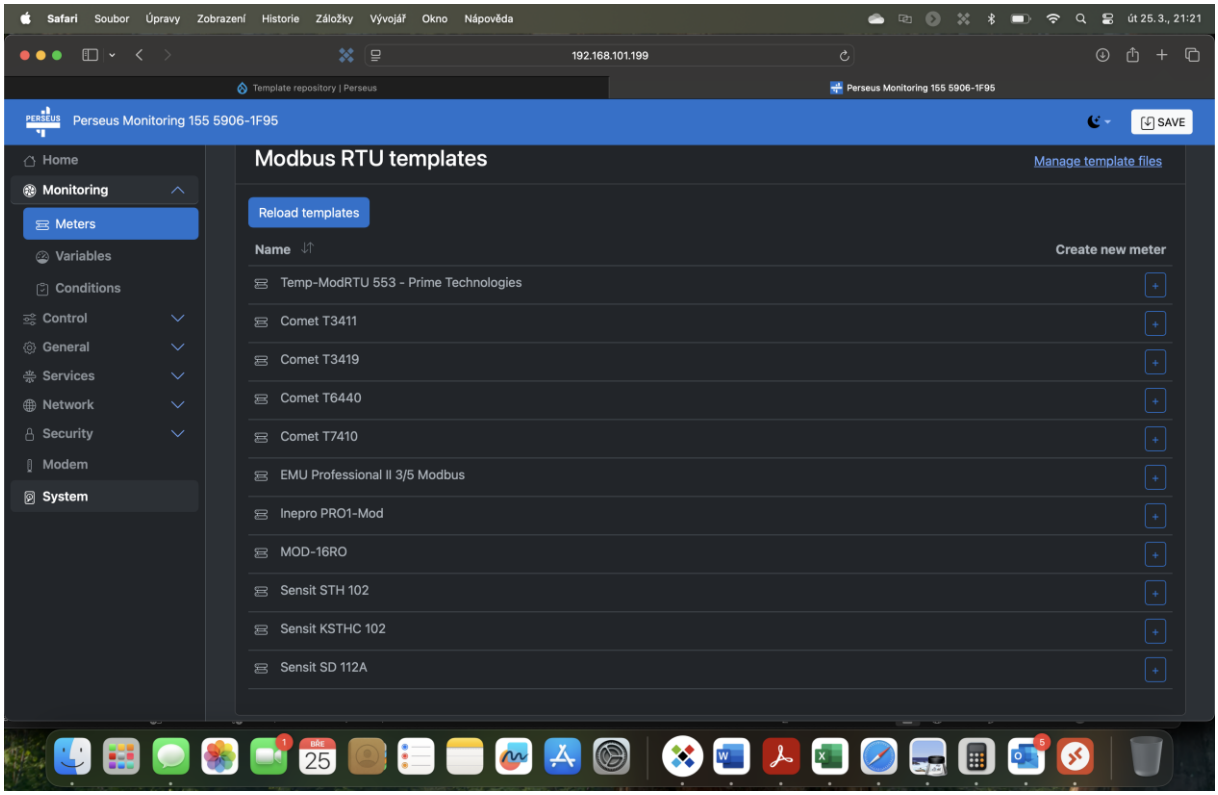
For many Modbus/RTU or Modbus/TCP-enabled devices, a template containing all the necessary information already exists. A repository of existing templates is available at https://perseus.hw-group.com/template-repository.



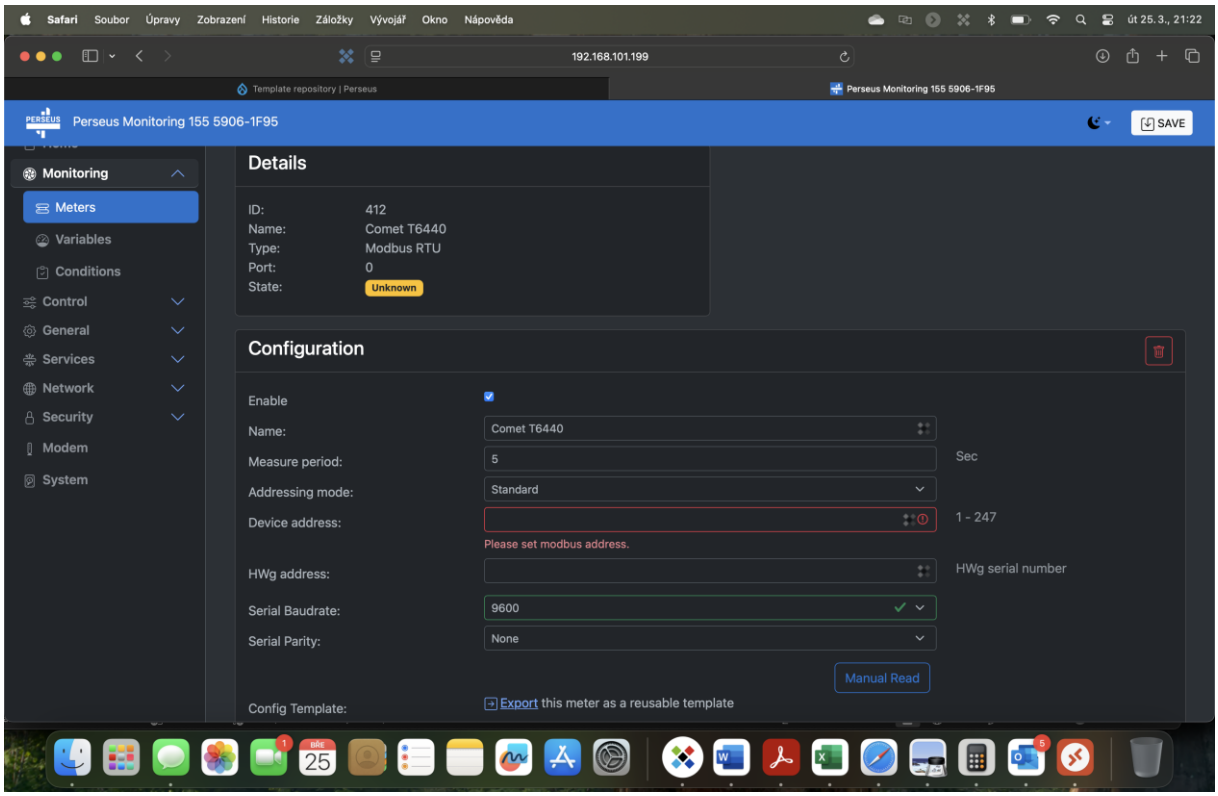
Here you can search for a template of a specific meter by name, manufacturer or protocol. Just download the relevant XML (ZIP) file to your computer and then upload it to the Perseus unit on the Meter Templates page (General menu), http://your_perseus_ip/cfg_template.xml



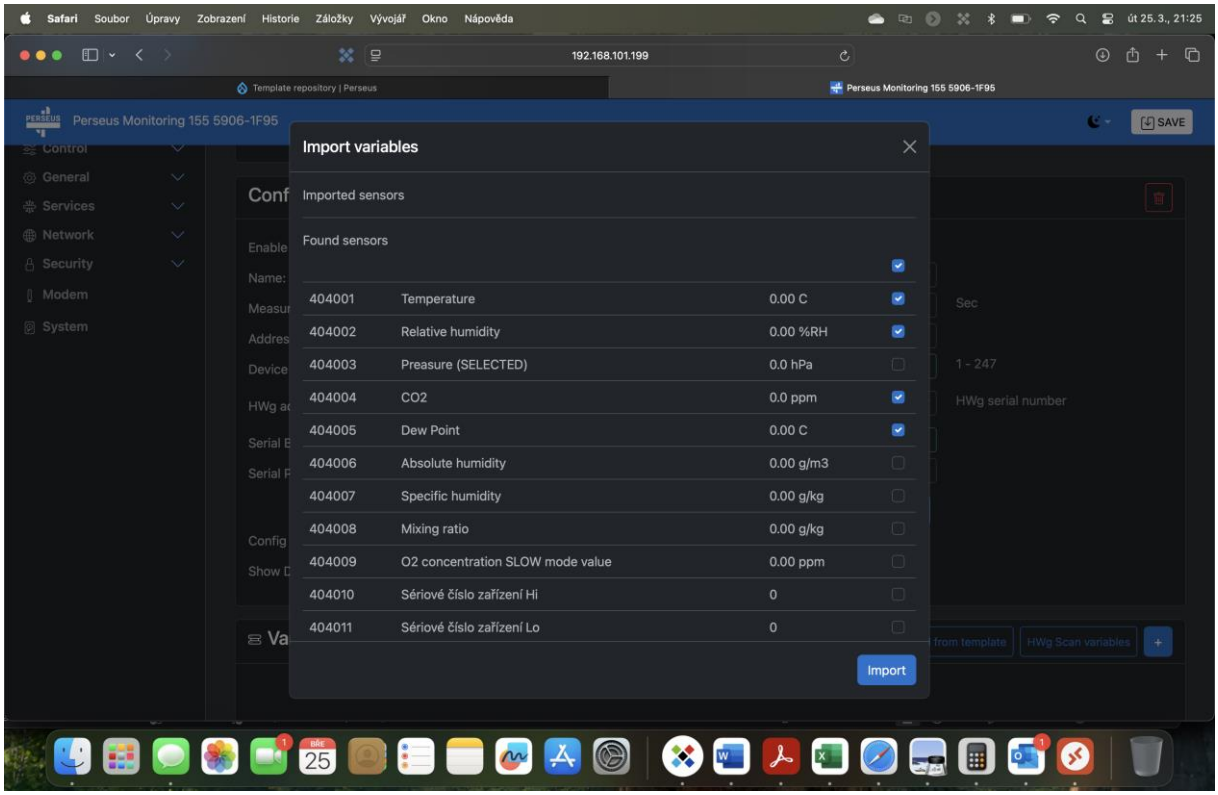
You can then add the meter on the Monitoring>Meters> Modbus/RTU or Monitoring>Meters> Modbus/TCP page by clicking the Modbus RTU templates button.



Here, press the + button next to the template to create a meter.



Now just fill in the Device Address (indicated on the meter or in its manual) and save the meter with the Save button. Then press the Add from templates button in the Variables section to bring up the known variables dialog



The pre-selected ones are those that the template creator considered important, but you can add or remove any known ones and confirm the selection by pressing the Import button.

This completes the addition and you can add another meter or set Conditions.

Manual meter addition (with Template export option)

For manual addition it is necessary to know the basic communication parameters of the meter (Modbus Address, baudrate and Parity for Modbus/RTU or Device URL (IP address) and Slave ID for Modbus/TCP) and the map of Modbus registers and their data types.

All required information can be found in the datasheet of the respective meter. As an example, we will use the Comet T6440 sensor.

First, download the manufacturer's Instruction manual at https://www.cometsystem.com/products/temperature-humidity-co2-transmitter-with-rs485-interface/reg-t6440 -

On page 3 you can find the communication parameters of the unit:

Device setting from the manufacturer

If special setting was not required in the order device is set from the manufacturer to the following parameters:

communication protocol:	Modbus RTU
device address:	01
communication speed:	9600Bd, without parity, 2 stop bits
display:	switched ON
value displayed at higher line:	CO_2, temperature/CO_2 – by device type
value displayed at lower line:	relative humidity
temperature unit:	°C
preset computed value:	dew point temperature
measurement mode:	SLOW
display:	switched ON
LED indication:	up to 1000 ppm lights green LED, between 1000 and 1200 ppm lights yellow LED and over 1200 ppm lights red LED

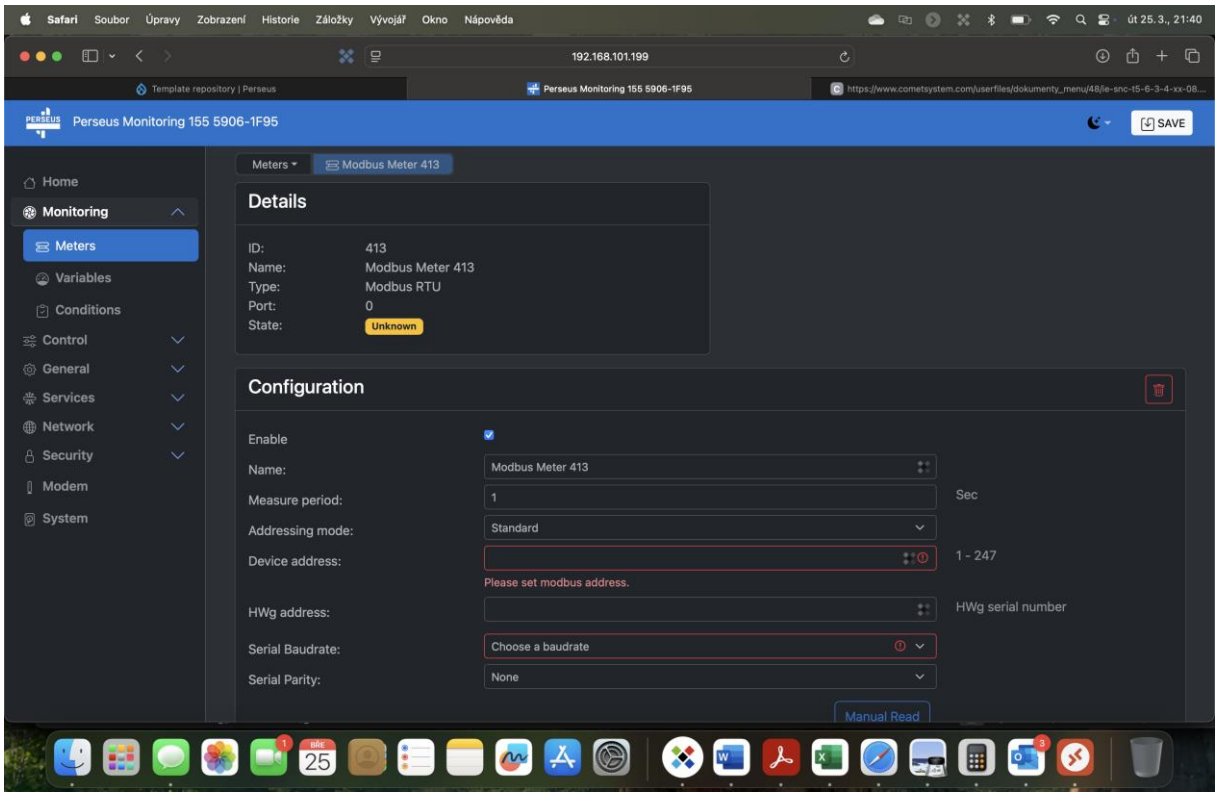
Modification of the setting is possible to do by means of the PC and TSensor program.

Here we are interested in:

Device address: 1

Communication speed: 9600Bd, without parity.

Based on this, create a new meter on the Monitoring>Meters page> Modbus/RTU using the Add+ button:



The dialog prompts us to fill in the Device Address (fill in the default value of 1 or the value specified on the sensor) and Serial Baudrate (fill in 9600) fields. Serial Parity remains none (without parity). Save the settings using the Save button.

Now we can proceed to add variables, again starting in the manual from the sensor. Here we will be interested in page 11:

Modbus registers of the device

Variable	Unit	Address[hex]^X	Address[dec]^X	Format	Size	Status
Measured temperature	[°C] [°F]*	0x0031	49	Int*10	BIN16	R
Measured relative humidity	[%]	0x0032	50	Int*10	BIN16	R
Computed value *	[*]	0x0033	51	Int*10	BIN16	R
Dew point temperature	[°C] [°F]*	0x0035	53	Int*10	BIN16	R
Absolute humidity	[g/m^3]	0x0036	54	Int*10	BIN16	R
Specific humidity	[g/kg]	0x0037	55	Int*10	BIN16	R
Mixing ratio	[g/kg]	0x0038	56	Int*10	BIN16	R
Specific enthalpy	[kJ/kg]	0x0039	57	Int*10	BIN16	R
CO_2 concentration displayed on LCD	ppm	0x0034	52	Int	BIN16	R
CO_2 concentration „FAST“ mode value	ppm	0x0054	84	Int	BIN16	R
CO_2 concentration „SLOW“ mode value	ppm	0x0055	85	Int	BIN16	R
Address of device	[-]	0x2001	8193	Int	BIN16	R/W*
Code of communication speed	[-]	0x2002	8194	Int	BIN16	R/W*
Serial number of device Hi	[-]	0x1035	4149	BCD	BIN16	R
Serial number of device Lo	[-]	0x1036	4150	BCD	BIN16	R
Version of Firmware Hi	[-]	0x3001	12289	BCD	BIN16	R
Version of Firmware Lo	[-]	0x3002	12290	BCD	BIN16	R

Explanation:

- * depends on device setting (by User’s software)
- Int*10 register is in format integer*10
- R register is designed only for reading
- W* register is designed for writing, for details see file “*Description of communication protocols of Txxxx series*”
- X register addresses are indexed from zero – register 0x31 is physically sent as value

Meaning of columns:

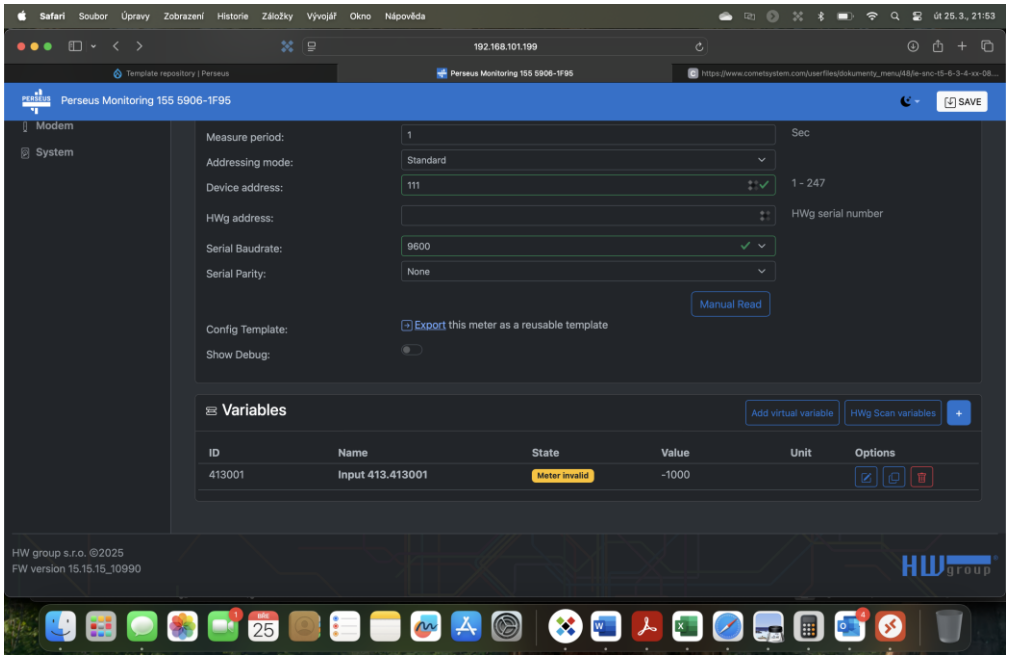
Variable - Variable name
Unit - unit of the variable
Address[hex] - Address in HEX
Address[dec] - Address in DEC
Format - value format
Size - Data size
Status - Status (Read/Write)

For Modbus RTU, I still lack information about the read function. It is listed on page 10:

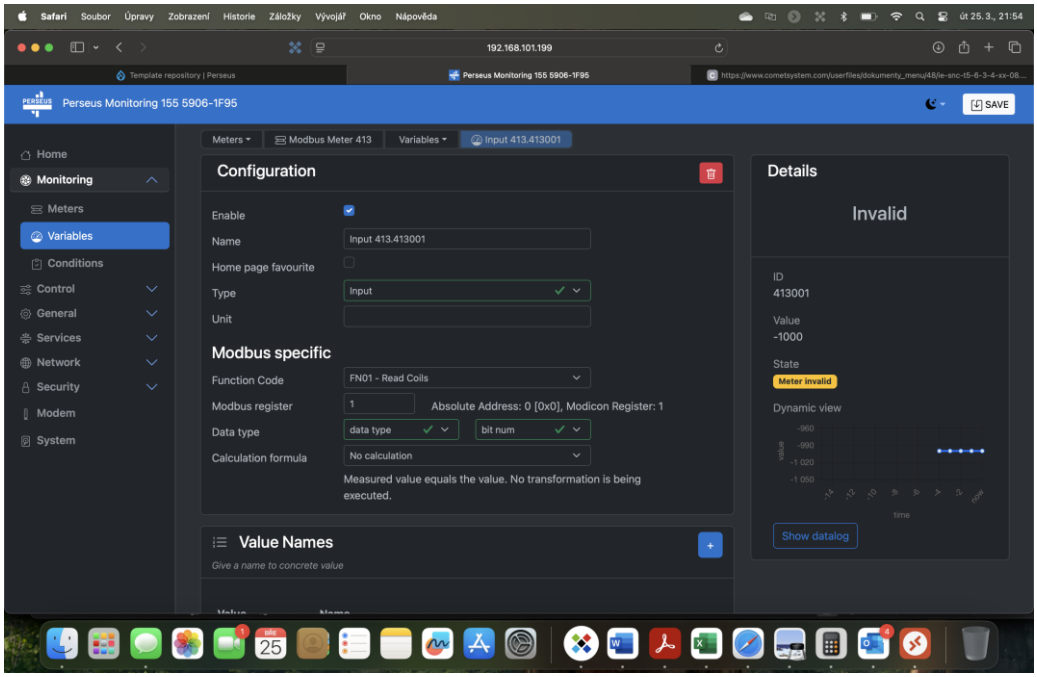
Supported
03 (0x03): Reading of 16-bit registers (Read Holding Registers)
04 (0x04): Reading of 16-bit input gates (Read Input Registers)
16 (0x10): Setting of more 16-bit registers (Write Multiple Registers)

It follows that function 03 will be used to read the values from the above table.

In Perseus, on the relevant Meter page, press the + button in the Variables section to create a Variable:



And we'll open his edit:



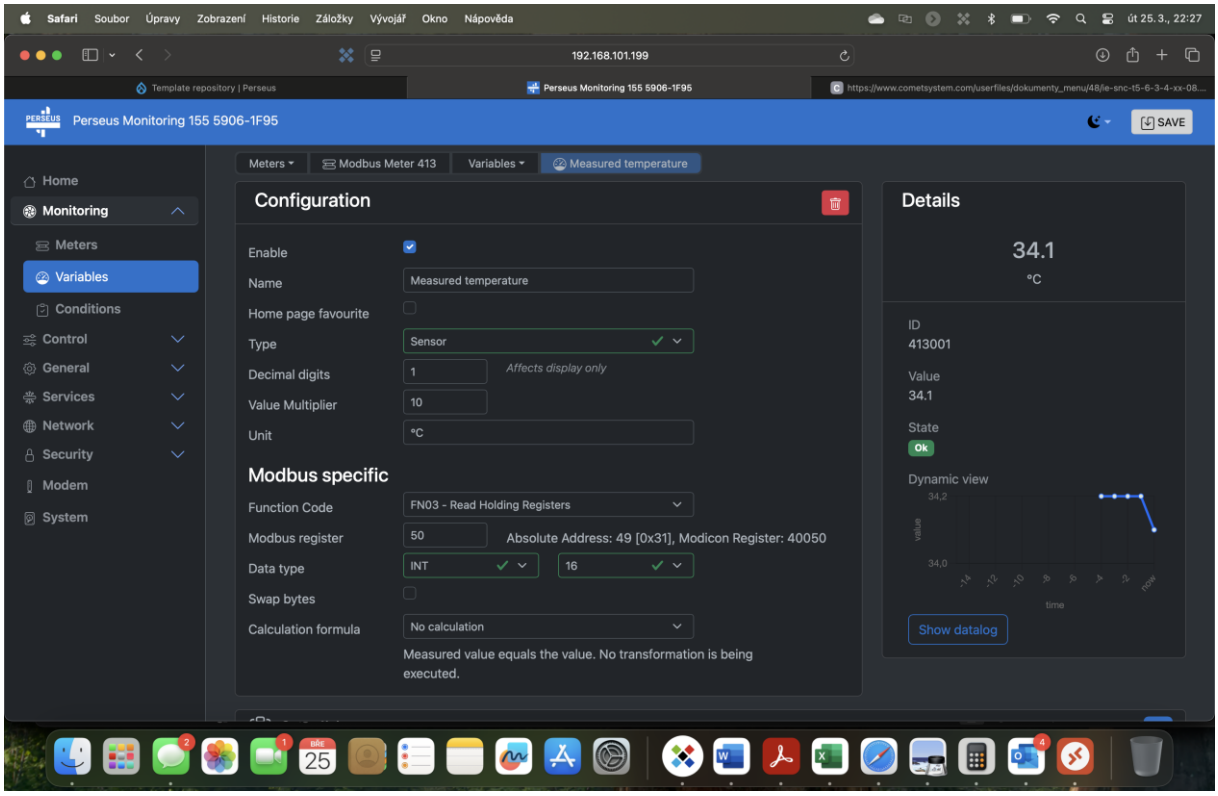
We will add here:

- **Name**: Measured
- **Type:** Sensor - It is a continuous value i.e. Sensor
- **Unit**: °C
- **Decimal Digits**: 1 - We know that the value has one decimal place because of the note in the register description that says that if there is more than one decimal place, the Float data type is used
- **Value Multiplier**: 10 - The register description note states that the value is *10, so a multiplier of 10 is required
- **Function Code**: 03

- **Modbus register**: 50 - From the register map we know that the value is stored in register 49DEC or 0x0031 hex. The value of the register is given 1 higher, but you can check the correctness of the entry on the line next to the entered value: "Absolute Address: 49 [0x31], Modicon Register: 40050".
- **Data type**: INT 16 - From the register table we know that this is INT and BIN16, i.e. 16-bit value

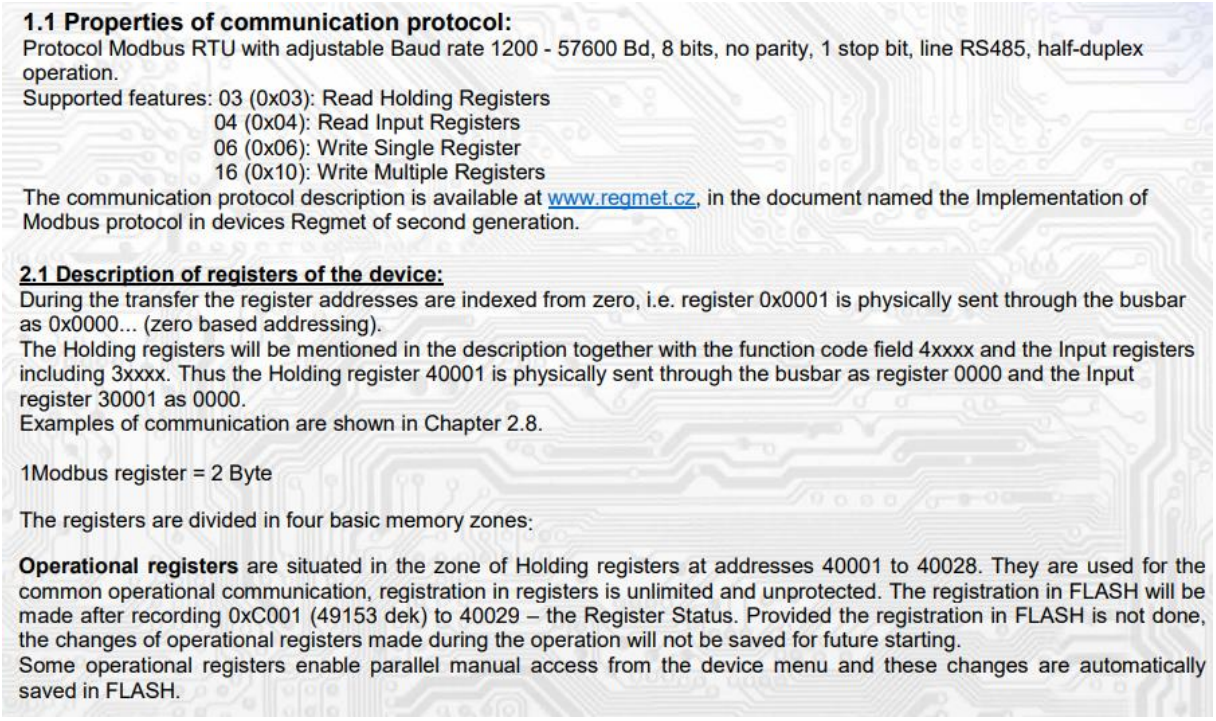
After entering all the data, press the Save button and in a moment we can see the result of the reading.

The result looks like this:



This completes the entry and the next variable can be continued.

For example, the information for the meter Controller with touch display - type RK-CHM-D | Regmet can be obtained similar way. From the data sheet RK-CHV_en.pdf on page 4 you can find following:



1.1 Properties of communication protocol:
Protocol Modbus RTU with adjustable Baud rate 1200 - 57600 Bd, 8 bits, no parity, 1 stop bit, line RS485, half-duplex operation.
Supported features: 03 (0x03): Read Holding Registers
04 (0x04): Read Input Registers
06 (0x06): Write Single Register
16 (0x10): Write Multiple Registers
The communication protocol description is available at www.regmet.cz, in the document named the Implementation of Modbus protocol in devices Regmet of second generation.

2.1 Description of registers of the device:
During the transfer the register addresses are indexed from zero, i.e. register 0x0001 is physically sent through the busbar as 0x0000... (zero based addressing).
The Holding registers will be mentioned in the description together with the function code field 4xxxx and the Input registers including 3xxxx. Thus the Holding register 40001 is physically sent through the busbar as register 0000 and the Input register 30001 as 0000.
Examples of communication are shown in Chapter 2.8.

1Modbus register = 2 Byte

The registers are divided in four basic memory zones:

Operational registers are situated in the zone of Holding registers at addresses 40001 to 40028. They are used for the common operational communication, registration in registers is unlimited and unprotected. The registration in FLASH will be made after recording 0xC001 (49153 dek) to 40029 – the Register Status. Provided the registration in FLASH is not done, the changes of operational registers made during the operation will not be saved for future starting.
Some operational registers enable parallel manual access from the device menu and these changes are automatically saved in FLASH.

The values of temperature, humidity, etc. are read by function 03, 1 register has 2Byte, i.e. 16 bits and is located at addresses 40001-40028 (modicon address, i.e. including function - decade 1-28). A detailed description and register map is given on page 5:

2.2 Description of operational registers:
Saving to FLASH is done just after writing 0xC001 (49153 dek) to 40029 – the Register Status.

				Modbus register [dek]
Measured temperature	Measured humidity	-	-	1 - 4
-	Measured CO2	-	-	5 - 8
Required temperature	Required humidity	-	-	9 - 12
-	Required CO2	-	-	13 - 16
Required mode 1	Required mode 2	Required mode 3	-	17 - 20
Digital input (DI)	-	-	Digital output (DO)	21 - 24
Symbol at position 1	Symbol at position 2	Symbol at position 3	-	25 - 28

4001 (R) – Measured temperature:
is detected with an inbuilt digital sensor that is built into the front panel of the cover and connected to the circuit board through the connector via the flexible cord. The value from the sensor is displayed on OLED and at the same time it is accessible on line RS485.
It is sent in °C in form of 16-bit number with sign (signed integer) multiplied by the constant 10:
0x00FB = 251dek = 25.1°C.

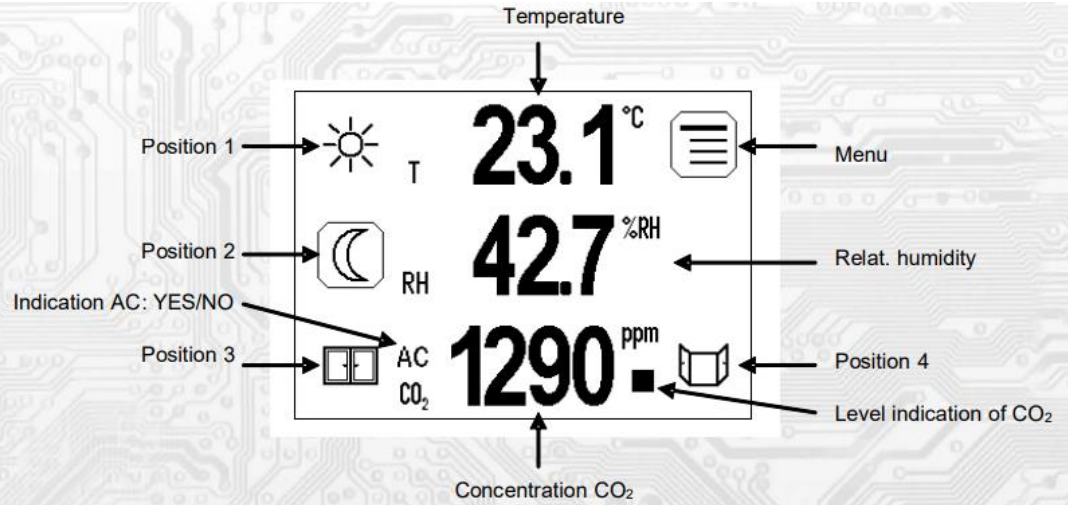
40002 (R) – Measured relative humidity of air:
is sensed with built-in digital sensor that is built into the front panel of cover and connected to the circuit board through the connector via the flexible cord. The value from the sensor is displayed on OLED and at the same time it is accessible on line RS485.
It is sent in % in form of 16-bits number with sign (signed integer) multiplied by the constant 10: 0x0164 = 356dek = 35.6%.

4006 (R) - Measured CO2 concentration:
is detected with an a module that is embedded under the cover. The value is sent in ppm in form of 16-bits number with sign (signed integer): 0x0237 = 567dek = 567ppm.

40009 (R,W,M) – Required temperature:
or possible temperature correction.The limits of the range for setting by buttons are defined by configuration registers.
40053 – Required temperature, lower limit and 40054 – Required temperature, upper limit.
The value is in°C in form of 16-bits number with sign (signed integer) multiplied by the constant 10.
For example,when setting the configuration registers 40053 – Required temperature, lower limit at -50 and the register 40054 - Required temperature, upper limit at 50 will be the range of settingby keys +/-5.0°C and the value will be shown as the required temperature correction. When setting the configuration registers 40053 –Required temperature, lower limit for example at 100 and register 40054 – Required temperature, upper limit at 300 the range of setting by keys will be10 -3 0 °C and the value will be shown as the required temperature.

40010 (R,W,M) – Required value of relative humidity of air:
The limits of the range of setting by buttons are defined by configuration registers 40055 – Required humidity, lower limit and 40056 – Required humidity, upper limit.The value is in% in form of 16-bit unsigned integer.

Device address and communication speed information is missing. A search for the word "Baud" in the datasheet shows that this information can be found in the sensor menu:



Position 1-4: space for displaying up to 4 possible symbols indicating modes, functional states or other information from the control system.

Temperature:	current measured temperature
Relat. humidity:	current measured relative air humidity
Concentration of CO2:	current measured concentration of CO2 in the air
Menu:	entering the device menu
Indication AC: YES/NO:	indication activ/inactive automatic calibration of CO2
Level indication of CO2:	indicating the level of CO2 concentration using a three-color display on the LCD

1.2 Touch panel control:

If some positions 1 ÷ 4 are preselected as buttons for quick mode change (the position or the displayed symbol is highlighted with a white frame), it is possible to directly change modes with these buttons. By touching the temperature, relat. humidity or CO2 concentration, the LCD switches to the menu for setting the correction / desired value of the measured values.
By pressing the button , the drop-down menu of the device is displayed, which can be scrolled through with the and butto
Writing to FLASH is done by pressing . After inactivity of the buttons for longer than 10s or by pressing them gradually will return the display to the previous level until finally exiting the menu.

Menu - Network information ..... Information on current setting of the network address and Baud rate.
▼ ▲
device info ↵ Information on HW and SW of the device
LCD Settings ↵ Color... color settings for each LCD field
*Brightness during the day.... LCD backlight setting when the display is inactive (power save mode) during the day ***
*Brightness during the night.... LCD backlight setting when the display is inactive (power save mode) during the night ***
*100% brightness time ....setting of the active display time, after which the LCD backlight without activity on the touch panel switches to power saving mode ****