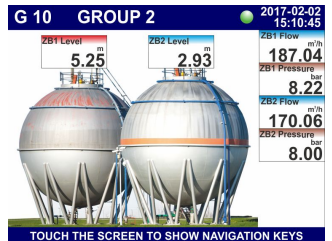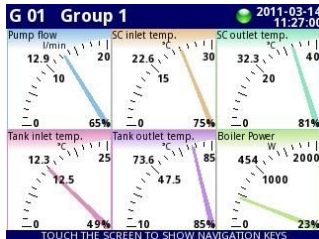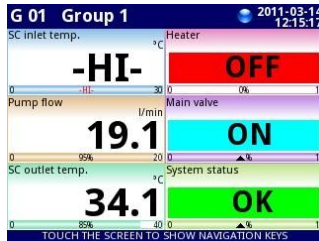# User manual
CONTROLLER/DATA RECORDER
MultiCon CMC-N16

- Firmware: **v.5.19** or higher

CE

Read the user's manual carefully before starting to use the unit or software.
The producer reserves the right to change any part of this document at any time without prior notice.

CONTENTS

Explanation of the symbols used in the manual:

- This symbol denotes especially important guidelines concerning the installation and operation of the device. Not complying with the guidelines denoted by this symbol may cause an accident, damage or equipment destruction.

IF THE DEVICE IS NOT USED ACCORDING TO THE MANUAL, THE USER IS HELD RESPONSIBLE IN ACCORDANCE WITH THIS MANUAL FOR POSSIBLE DAMAGE.

- This symbol denotes especially important characteristics of the unit.
Read any information regarding this symbol carefully

1. BASIC REQUIREMENTS AND USER SAFETY

- The manufacturer is not responsible for any damage caused by inappropriate installation, not maintaining the proper environmental conditions and using the unit contrary to its assignment.

- Installation must be performed by qualified personnel . During installation all available safety requirements must be considered. The fitter is responsible for executing the installation in accordance with this manual, local safety and EMC regulations.

- Protective conductor terminal of the device must be connected to an external

protective earthing system .

- The unit must be properly set-up, according to the application. Incorrect configuration may cause defective operation, which can lead to unit damage or an accident.

- **In case of a unit malfunction there is a risk of a serious threat to the safety of people or property additional. Independent systems and solutions to prevent such a threat must be used.**

- **The unit uses dangerous voltage that can cause a lethal accident. The unit must be switched off and disconnected from the power supply prior to starting installation of troubleshooting (in case of malfunction).**

- Neighbouring and connected equipment must meet appropriate standards and regulations concerning safety and be equipped with adequate overvoltage and interference filters.

- Do not attempt to disassemble, repair or modify the unit yourself. The unit has no user serviceable parts. Defective units must be disconnected and submitted for repairs at an authorized service centre.

- In order to minimize a fire or electric shock hazard, the unit must be protected against atmospheric precipitation and excessive humidity.

- Do not use the unit in areas threatened with excessive shocks, vibrations, dust, humidity, corrosive gasses and oils.

- Do not use the unit in areas where there is a risk of explosions.

- Do not use the unit in areas with significant temperature variations, exposure to condensation or ice.

- Do not use the unit in areas exposed to direct sunlight.

- Make sure that the ambient temperature (e.g. inside the control box) does not exceed the recommended values. In such cases forced cooling of the unit must be considered (e.g. by using a ventilator).

The unit is designed for operation in an industrial environment and must not be used in a household environment or similar.

1.1. TOUCH-SCREEN USE

Do not use pointers with sharp edges (like tips of pencils and pens, knives, scissors, needles, wires, nails, screws, bolts etc.) while working with the touch-screen. It is strongly recommended to use a special stylus made of plastic or another soft material with rounded ends (for example the stylus delivered with the device). The display of the **MultiCon CMC-N16** should also be protected against aggressive substances and extremely high and low temperatures (see **Chapter 3TECHNICAL DATA**).

2. GENERAL CHARACTERISTICS

The **MultiCon CMC-N16** is a sophisticated multichannel unit which allows a simultaneous measurement, visualisation and control of numerous channels. This device can operate autonomously or in cooperation with external measurement devices and actuators. Essential features of the **MultiCon CMC-N16** are listed and briefly described below.

- *Advanced processing unit and system based on LINUX*
  The powerful **MultiCon CMC-N16** processor allows the device to run under the control of a LINUX operating system. Such a solution makes the firmware flexible and gives the possibility of a simultaneous operation of many processes (such as: measurement, communication, visualisation). The use of LINUX also makes software independent of hardware configurations.

- *Color TFT display with Touch-panel*
  The **MultiCon CMC-N16** displays all data and dialogue on a legible, 320x240 pixels, color TFT screen. Full control of the device is realised using the built-in touch-panel which makes operating the **MultiCon CMC-N16** easy and intuitive.

- *Hardware flexibility and a large variety of possible configurations*
  MultiCon CMC-N16 is designed as a modular device consisting of a *base* and optional *input and output configurations*. The *base* contains:
  - main processor,
  - display with touch-screen,
  - Switch Mode Power Supply
    - 19V...**24**...50V DC, 16V...**24**...35V AC
    - 85V...**230**...260V,
  - basic communication interfaces (USB and RS485).
  - three blocks (marked as A, B, C) designed for installation of input and output hardware configurations.
  - one block (marked as D) designed for installation of communication hardware configurations.

All hardware configurations are optional and can be installed inside the device according to the customer's needs.

Optional hardware configurations for block A are:
  - 2/4x universal inputs with analogue flowmeter mode (flowmeter/tachometer 0/4 ÷ 20 mA),

Optional hardware configurations for block B are:
  - 2x counter inputs with flowmeter mode,
  - 4x binary inputs,

Optional hardware configurations for block C are:
  - 2/4x Relay module 1A/250V,
  - 2/4x Passive current output,
  - 2/4x OC-type outputs with PWM mode,
  - 2/4x Passive current output + 2/4x Relay module 1A/250V,
  - 2/4x Passive current output + OC-type outputs with PWM mode,
  - 2/4x Relay module 1A/250V + 2/4x OC-type outputs with PWM mode,

<u>Optional hardware configurations for block D are:</u>
- – Ethernet port with or without IP67 connector on the housing,
- – RS-485 port + Ethernet port with or without IP67 connector on the housing,

- **Full freedom of data sources, presentation modes and controlling methods**
  The multi level structure of the **MultiCon CMC-N16** firmware allows to select presented data sources, presentation modes and controlling methods. The **MultiCon CMC-N16** displays the values of virtual *logical channels* which can be fed with:
  - – measurement data from built-in physical channels,
  - – measurement data from remote channels (other devices connected to the **MultiCon CMC-N16** by RS-485 interface),
  - – generate profiles/timers,
  - – output states and quantities (looped back results of controlling processes),
  - – mathematical combination of one or more *logical channels*.

  Every *logical channel* (displayed or not) can be used as input data for one or more controlling processes. The **MultiCon CMC-N16** implements many different controlling methods:
  - – above a defined level,
  - – below a defined level,
  - – within a defined range,
  - – out of a defined range,
  - – PID control.

  Process control with built-in outputs can be done with programmable hysteresis and delays of the outputs control. It is possible to control (linearly or bistably) remote modules. Controlling processes can drive built-in physical outputs or virtual outputs which can be used as inputs to *logical channels*.

Built-in analog input
Built-in binary input
External input (RS-485)
Profiles/timers
Controller
Mathematical & logical combination data
States of hardware & virtual outputs
Set point values

Interface

Display	Numeric, logical or text value
	Charts & bars
	Needle indicator
	Grouping data

Interface

Reading data via Ethernet
Reading data stored on the Flash drive

Interface

Built-in analog output
Built-in binary output
External output (RS-485)

Fig. 2.1. Basic structure of the multichannel device

All of these can be freely named and described by the user, and presented in many forms:
– as numerical values,
– as needle dials,
– as horizontal and vertical linear charts,
– as horizontal and vertical bars,
– as phasor charts,
– as **SCADALite** view,
– as controllers view.

3. TECHNICAL DATA

Power supply voltage (depending on version)	85...**230**...260V AC/DC; 50 ÷ 60 Hz or 19...**24**...50V DC; 16V...**24**...35V AC
External Fuse (required)	T - type, max. 2 A
Power consumption	typically 15 VA; max. 20 VA
Display (depending on version)	3.5", TFT color graphic display, 320 x 240 pixels, with LED backlight
Sensor power supply output	24V DC ± 5% / max. 200 mA,
Basic communication interfaces	RS 485, 8N1/2, Modbus RTU, 1200 bit/s ÷ 115200 bit/s USB Host port,
Digital input (DI)	1 input 0/15..24V DC, galvanic isolation (low state: 0÷1V, high state:8÷24V) power consumption: 7,5 mA / 24V, isolation: 1min @ 500V DC.
Optional communication hardware configurations*	Additional Ethernet port with or without IP 67 connectors** Additional RS-485 port + Ethernet port with or without IP 67 connectors**
Optional input hardware configurations*	2/4x universal inputs with flowmeter mode ** 2x counter inputs with tachometer mode** 4x binary inputs**
Optional output hardware configurations*	2/4x Relay 1A/250V** 2/4x Passive current outputs** 2/4x OC-type outputs with PWM mode** 2/4x Passive current output + 2/4x Relay module 1A/250V** 2/4x Passive current output + OC-type outputs with PWM mode** 2/4x Relay module 1A/250V + 2/4x OC-type outputs with PWM mode**
Protection level	IP 65
Internal memory capacity	2 GB, optional 4 GB
Housing type	Wall mounted
Housing material	ABS + polycarbonate
Housing dimensions (L x h x D)	166 x 161 x 103 mm 166 x 195 x 103 mm

Glands

Maximal amount of cable glands (cg):	10
Glands for "cg" from 0 to 1:	1 x M25 + 1 x M16
Glands for "cg" from 2 to 5:	2 x M25 + 1 x M16
Glands for "cg" from 6 to 10:	3 x M25 + 1 x M16

Inserts for one and many-

M25	1 x ⌀ 13-18 mm or 3 x ⌀ 7 mm or 4 x ⌀ 6 mm
M16 replaceable for	1 x ⌀ 4-8 mm
M12	Ethernet connector

Operating temperature (depending on version)	0°C to +50°C or -20°C to +50°C
Storage temperature (depending on version)	-10°C to +70°C or -20°C to +70°C
Humidity Altitude	5 to 90% no condensation up to 2000 meters above sea level
Screws tightening max. torque	0.5 Nm
Max. connection leads diameter	2.5 mm^2
Safety requirements	In accordance with to: PN-EN 61010-1 installation category: II pollution degree: 2 voltage in relation to ground: 300V AC isolation resistance: >20MΩ isolation strength between power supply and input/output terminal: 1min. @ 2300V (see **Fig. 4.1**)
EMC	PN-EN 61326-1

* check the current list of measurement modules at the producer's website
** see the full specification in the Appendices

4. DEVICE INSTALLATION

The unit has been designed and manufactured in a way assuring a high level of user safety and resistance to interference occurring in a typical industrial environment. In order to take full advantage of these characteristics, installation of the unit must be conducted correctly and in accordance with the local regulations.

- Read the basic safety requirements on page **4** prior to starting the installation.

- Ensure that the power supply network voltage corresponds to the nominal voltage stated on the unit's identification label.

- The load must correspond to the requirements listed in the technical data.

- All installation works must be conducted with a disconnected power supply.

- Protecting the power supply connections against unauthorized persons must be taken into consideration.

This is a class A unit. Class A equipment is suitable for use in all establishments other than domestic and those directly connected to a low voltage power supply network which supplies buildings used for domestic purposes. This equipment is not intended for use in residential environments and may not provide adequate protection to radio reception in such environments.

This is group 1 unit. Group 1 contains all equipment in the scope of this standard which is not classified as group 2 equipment. Group 2 contains all ISM RF equipment in which radio-frequency energy in the frequency range 9 kHz to 400 GHz is intentionally generated and used or only used, in the form of electromagnetic radiation, inductive and/or capacitive coupling, for the treatment of material or inspection/analysis purposes.

Carefully check that the isolation used with the unit (**4.1**) meets the expectations and if necessary use appropriate measures for over-voltage protection. Additionally, insure the appropriate air and surface insulation gaps while installing.

▨▨▨▨▨▨	Isolation strength 1min @ 2300V AC
—————————	Isolation strength 1min @ 500V AC
-------------------	No isolation
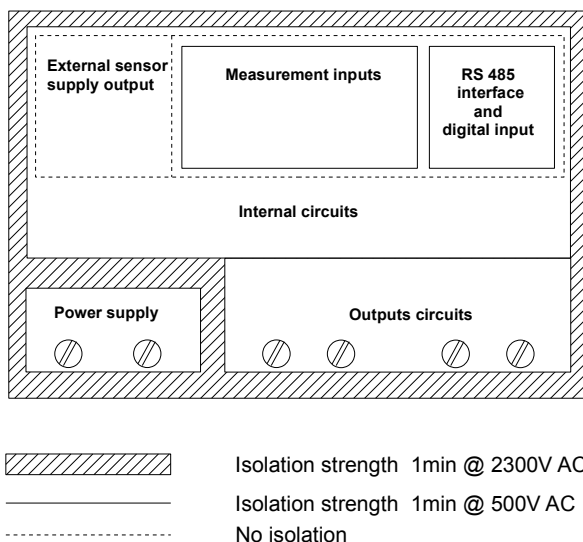
Fig. 4.1. Schematic diagram showing the isolation between individual circuits of the unit.

4.1. UNPACKING

After removing the unit from the protective packaging, check for transportation damage. Any transportation damage must be immediately reported to the carrier. Also, write down the unit serial number located on the housing and report the damage to the manufacturer.

Attached with the unit please find:
   – pointer for touch screen,
   – user's manual for **MultiCon CMC-N16** unit (device) in **pdf.**,
   – set of connectors for the device (delivered loosely)
   – set of inserts for M25 glands with plugs
   – 24 V output installation connectors

4.2. ASSEMBLY

⚠ - Disconnect the power supply prior to starting assembly.

- Check the connections are wired correctly prior to switching the unit on.

⚠ In order to install the unit a mounting hole must be prepared as it is shown in **Fig. 4.2**. First drill central hole and mount a hook, bolt or screw and hang the device. Then mark drill places for rest of the mounting holes, drill them and mount the device. It is recommended to use ϕ8 dowels and ϕ4 screws. After assembling the device, remove protective films from door and display.
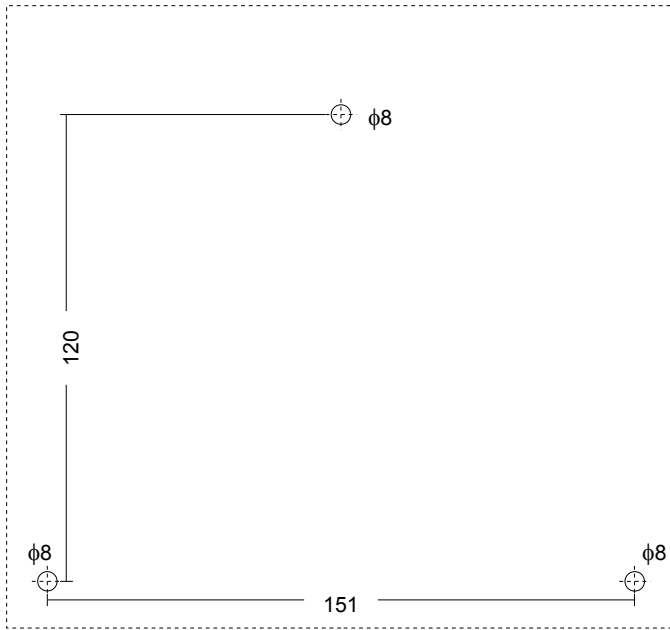
Fig. 4.2. Placement of the mounting holes
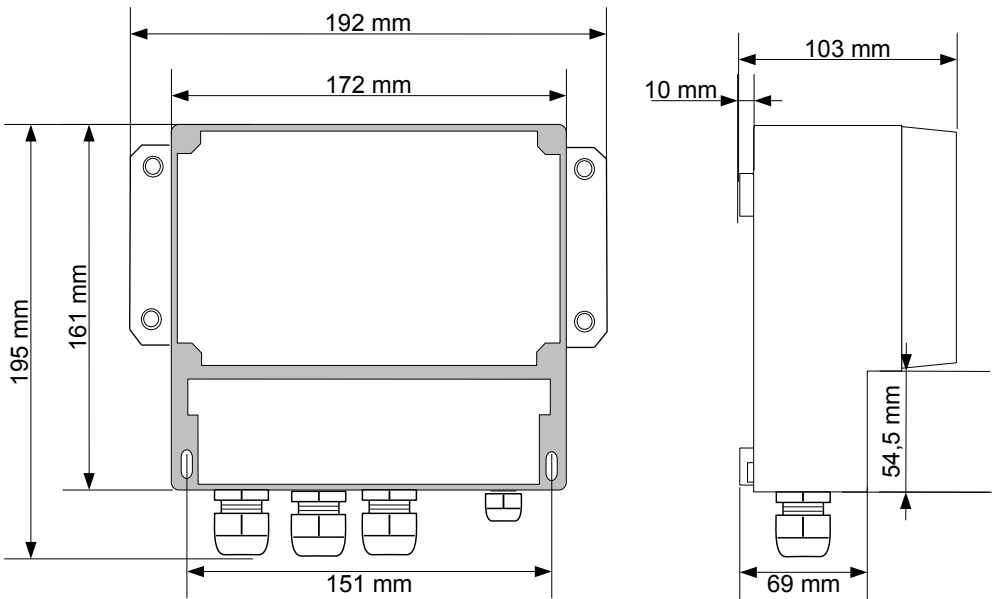
Fig. 4.3. Enclosure dimensions

To avoid connectors slots destruction use the method shown bellow.
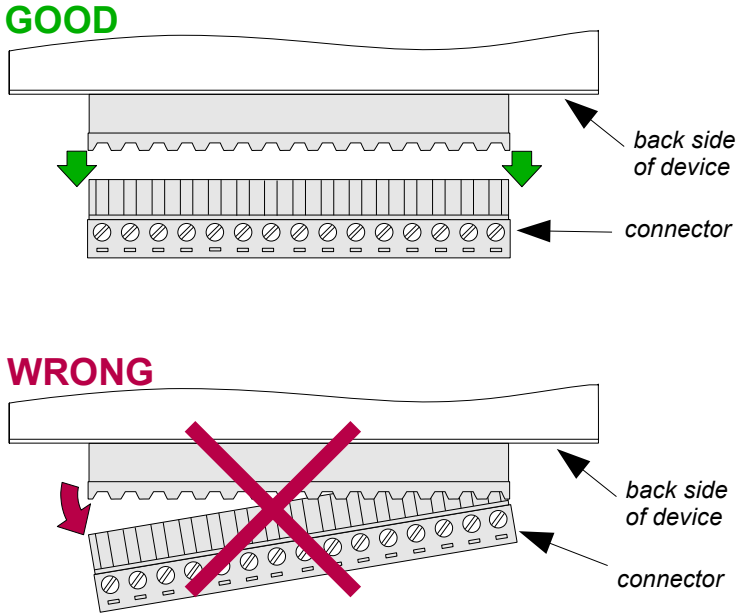
Fig. 4.4. Connectors removing method

Optionally, the insert with the key can be included to the device to reduce the risk of tampering. To assembly the insert first remove the end cap then insert the insert (see **Fig. 4.5**). Inserting the insert may require using force.

a)

b)

Press lightly

Slide out the end cap

c)

Insert the insert with the key

d)

Turn and remove the key
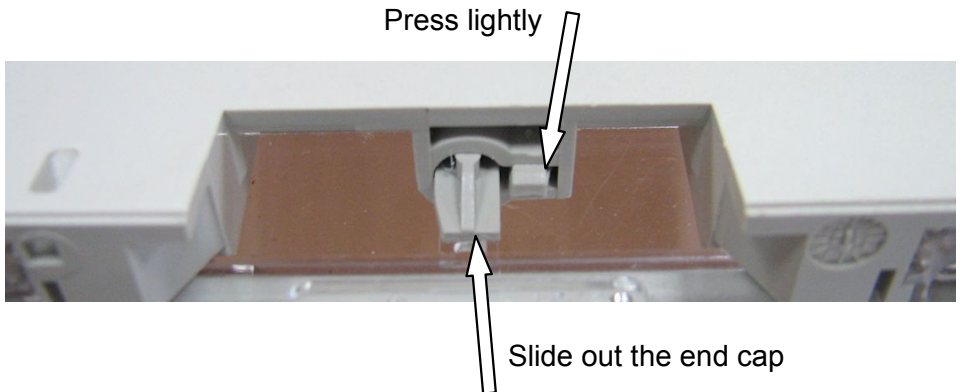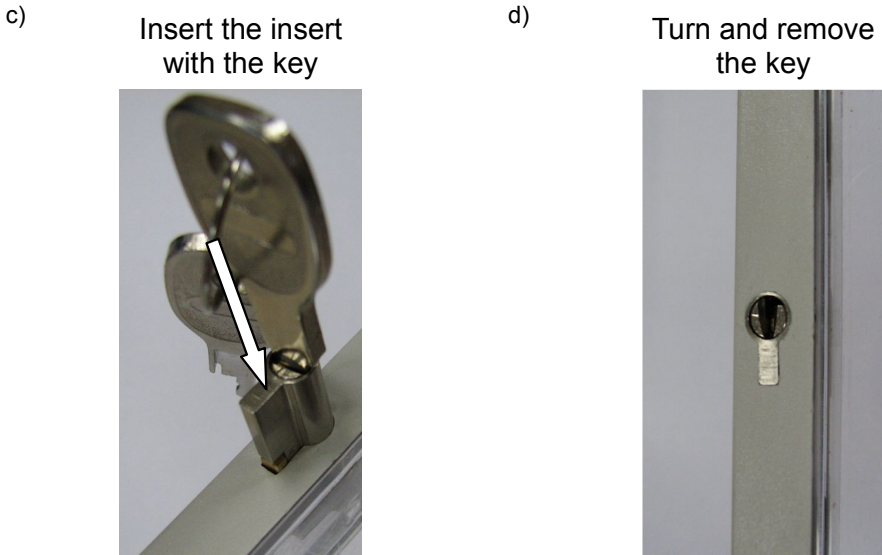
Fig. 4.5. Method of the insert assembling

4.3. CONNECTION METHOD

Caution

⚠

- Installation should be conducted by qualified personnel. During installation all available safety requirements should be considered. The fitter is responsible for executing the installation in accordance with this manual, local safety and EMC regulations.

- The unit is not equipped with an internal fuse or power supply circuit breaker. Because of this, an external time-delay cut-out fuse with a small nominal current value must be used (recommended bipolar, max. 2A) and a power supply circuit-breaker located near the unit (**4.6**). In case of using a monopolar fuse it must be mounted on the active wire (L).

- The power supply network cable diameter must be selected in such a way so that in case of a short circuit of the cable from the side of the unit, the cable shall be protected against destruction by an electrical installation fuse.

- Wiring must meet appropriate standards and local legal regulations and laws.

- In order to secure against accidental short circuit, the connection cables must be terminated with appropriate insulated cable tips.

- Tighten the clamping screws. The recommended tightening torque is 0.5 Nm. Loose screws can cause fire or defective operation. Over-tightening can lead to damaging the connections inside the units and breaking the thread.

- In case of the unit fitted with separable clamps they should be inserted into appropriate connectors in the unit, even if they are not used for any connections.

- Unused terminals must not be used for connecting any connecting cables (e.g. as bridges), because this can cause damage to the equipment or electric shock.

- If the unit is equipped with a housing, covers and sealing to protect against water intrusion, pay special attention to their correct tightening or clamping. In case of any doubt consider using additional preventive measures (covers, roofing, seals, etc.). Carelessly executed assembly can increase the risk of electric shock.

- After the installation is completed do not touch the unit's connections when it is switched on, because it brings the risk of electric shock.

Due to possible significant interference in industrial installations appropriate measures assuring correct operation of the unit must be applied. To avoid, the unit of improper indications, keep recommendations listed below.
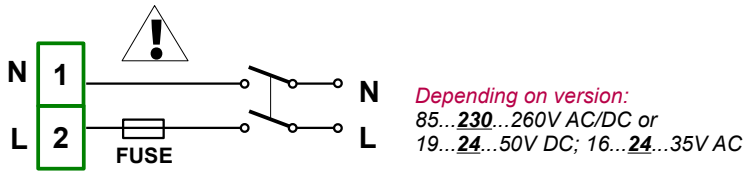
Fig. 4.6. Connection of power supply

- Avoid running signal cables and transmission cables together with power supply cables and cables controlling inductive loads (e.g. contactors). Such cables should cross each other at a right angle.
- Contactor coils and inductive loads should be equipped with interference protection systems, e.g. RC-type.
- Use of screened signal cables is recommended. Signal cable screens should be connected to the earthing only at one of the ends of the screened cable.
- In case of magnetically induced interference the use of twisted pair signal cables is recommended. Twisted pair (best if shielded) must be used with RS-485 serial transmission connections.
- If measurement or control signals are longer than 30m or go outside of the building, then additional safety circuits are required.
- In case of interference from the power supply side, the use of appropriate interference filters is recommended. Bear in mind that the connection between the filter and the unit should be as short as possible and the metal housing of the filter must be connected to the earth with the largest possible surface. The cables connected to the filter output must not be run together with cables with interference (e.g. circuits controlling relays or contactors).

Connections of power supply voltage and measurement signals are performed using the screw connections at the back of the unit's housing (see **4.7**).
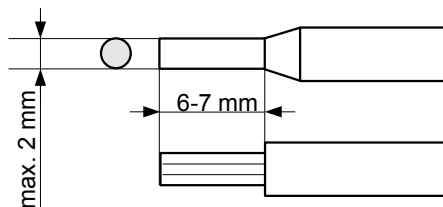
Fig. 4.7. Method of cable isolation replacing and cable terminals dimensions

⚠ **All connections must be made while power supply is disconnected !**
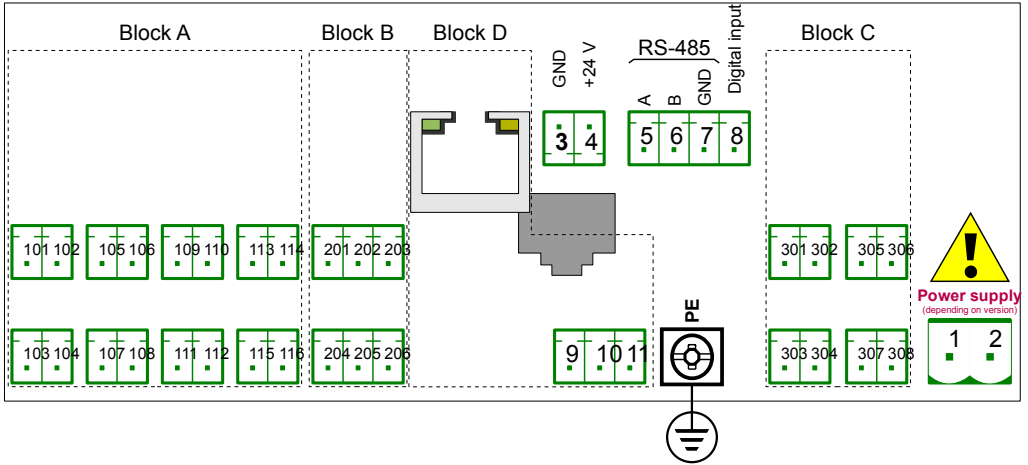
Fig. 4.8. Terminals description

The basic performance of the device (see **Fig. 4.9.**) contains:
  – Power supply,
  – Sensor supply output +24V DC Imax=200mA,
  – Digital input 0V...15...24V DC (low state: 0÷1V, high state:8÷24V),
  – Interface RS-485.
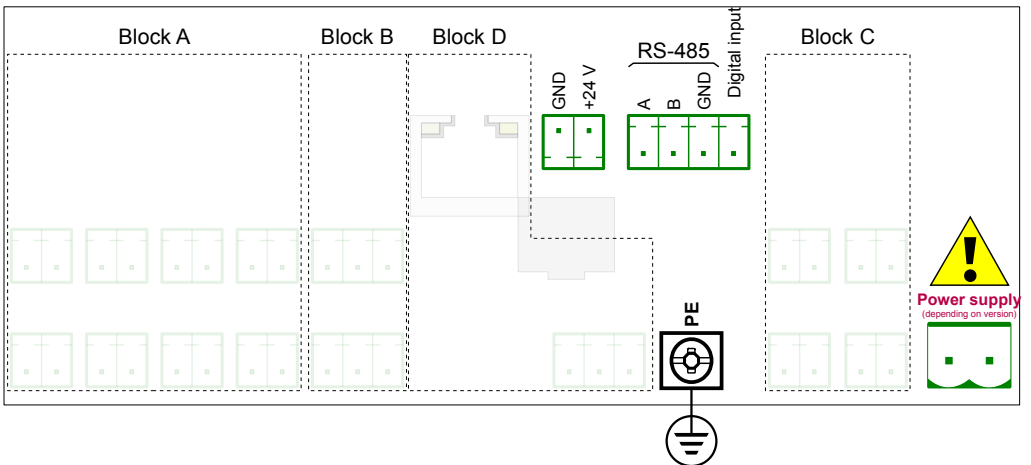
Fig. 4.9. Basic performance of the device

Depending on customer's needs, the basic version of the device can be upgraded with:
  – up to three I / O modules (installed in a place designated as Block A, Block B, Block C),
  – an additional hardware communication configuration (installed in a place designated

as **Block A**).

According to the order these terminals can look different than those shown in **Fig. 4.8.** or be not present. Terminals and connections of available hardware configurations are shown in **Fig. 4.10. ÷ Fig. 4.14.** in **Section 4.3.1Configurations of MultiCon CMC-N16**.

Shown below is an example of hardware configuration installed in the device:
– base,
– Block A - FUN4 (4 universal inputs with flowmeter mode),
– Block B - D4 (4 digital input),
– Block C - R21IO2B (2 relay outputs 1A/250V + 2 passive current outputs),
– Block D - ETEC (additional serial port + additional Ethernet interfaces with IP67 something on housing).

4.3.1. Configurations of MultiCon CMC-N16
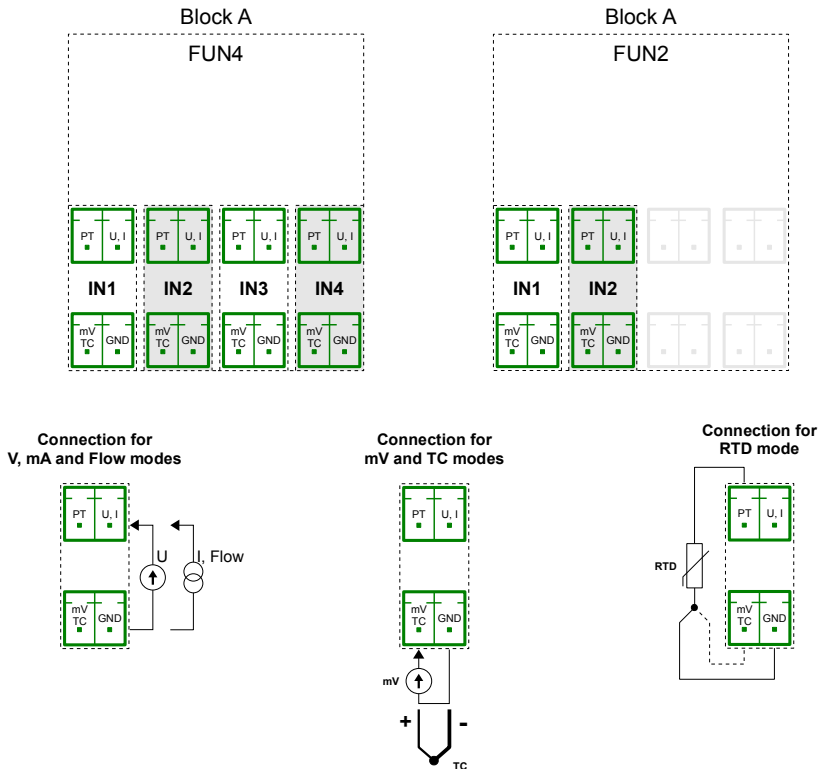
Fig. 4.10. Input configurations for Block A

Block B — DU2

IN1
| A | B | GND |

IN2
| A | B | GND |

Block B — D4

| IN1 | IN2 | GND |

| IN3 | IN4 | GND |

Fig. 4.11. Input configurations for Block B

Block C — R21

R1

R2

Block C — R41

R1   R3

R2   R4

Block C — COP2

IO1

IO2

Block C — COP4

IO1   IO3

IO2   IO4

Block C — OC2

OC1

OC2

Block C — OC4

OC1   OC3

OC2   OC4

Block C — R21COP2

IO1   R3

IO2   R4

Block C — COP2OC2

IO1   OC3

IO2   OC4

Block C — R21OC2

R1   OC3

R2   OC4
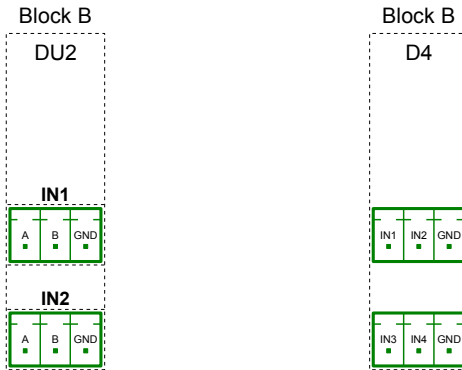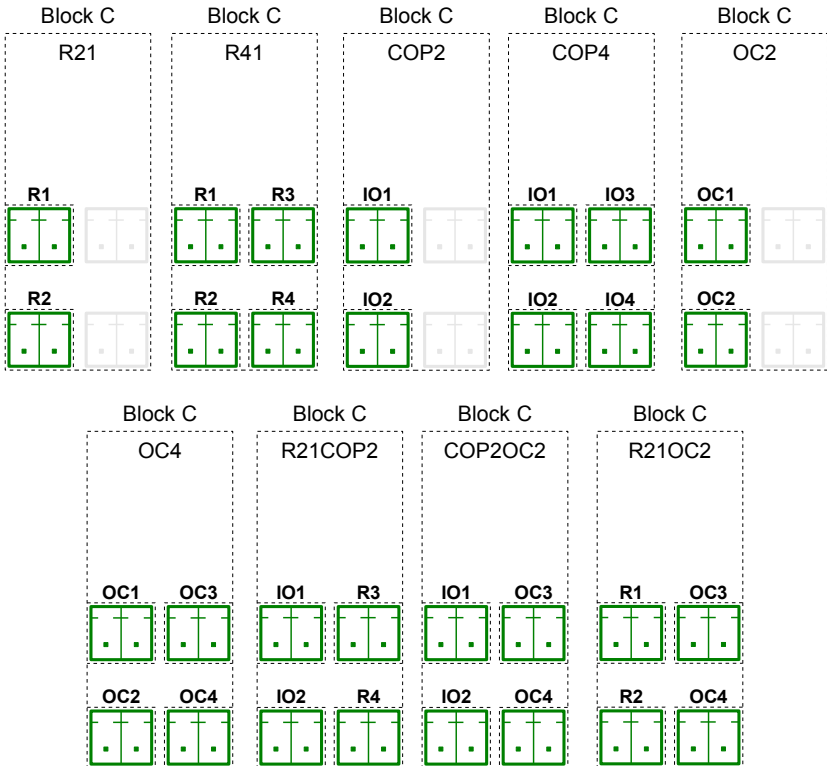
Fig. 4.12. Input configurations for Block C

a)

b)
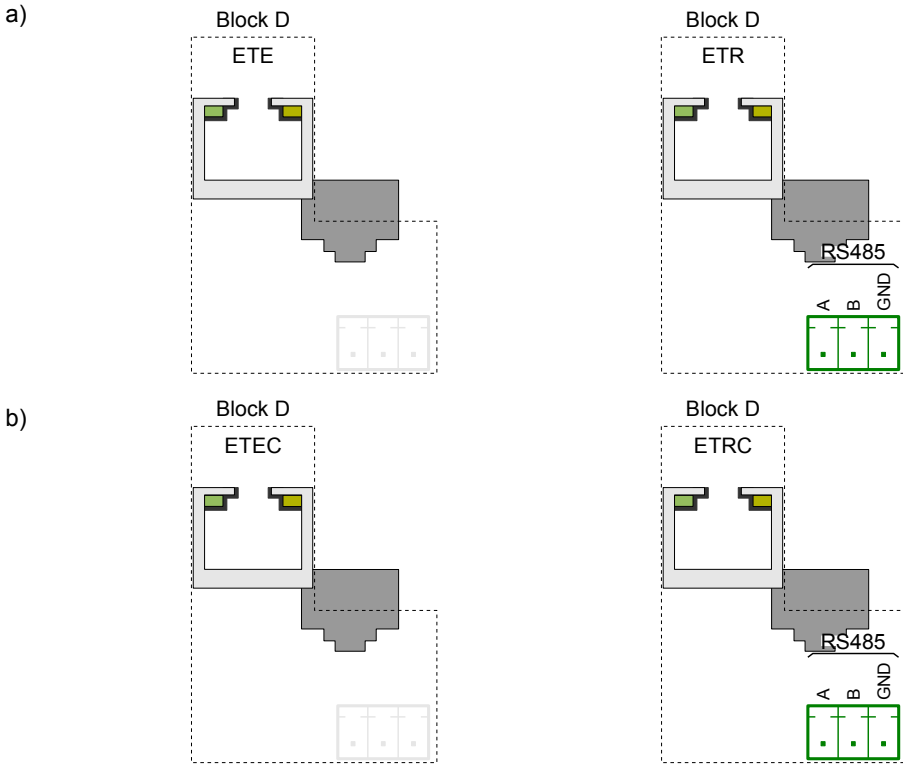
Fig. 4.13. Communication configurations for Block D

a) with IP67 connector on the housing, b) without IP67 connector on the housing,
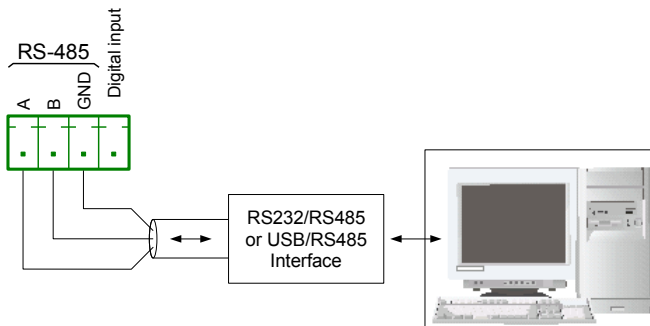
Fig. 4.14. Connection of RS-485 transmission signals

The **MultiCon CMC-N16** device supports the following converters:
  – USB / RS-485 converter **(SRS-U4)**
  – RS-232 / RS-485 converter **(SRS-2/4-Z45)**

4.4. MAINTENANCE

The unit does not have any internal replaceable or adjustable components available to the user. Pay attention to the ambient temperature in the room where the unit is operating. Excessively high temperatures cause faster ageing of the internal components and shorten the fault-free time of the unit's operation.

In cases where the unit gets dirty, do not clean with solvents. For cleaning use warm water with a small amount of detergent, or in case of more significant contamination ethyl or isopropyl alcohol.

Using any other agents can cause permanent damage to the housing.

Products marked with this symbol should not be placed in municipal waste. Please check local regulations for disposal of electronic products.

5. INTRODUCTION TO MultiCon CMC-N16

5.1. UNDERSTANDING MultiCon CMC-N16

The **MultiCon CMC-N16** device has been developed as a universal multichannel controller. To maintain this concept its firmware has been written with a multi-level structure. The device runs under the control of a LINUX operating system keeping all subsystems ready to use and allowing independent and simultaneous operation of many processes (communication, data acquisition, post-processing, visualisation etc.). Such an approach offers great advantages to high level applications, making it flexible and dynamically configurable. Similarly, data structures and streams have been implemented in quite a different way than in most similar devices. The main difference is the concept of using *Logical Channels* as a bridge: physical inputs and outputs - visualisation and controlling processes. Designers of **MultiCon CMC-N16** decided to use such a solution to increase the functionality of the device and make the software almost fully independent from the hardware.

5.1.1. Logical channels

A **Logical Channel** is a data stream existing in the memory of the device, having its own name and can be displayed in almost any way. **Logical Channels** can be used as:
– measurement inputs,
– data source of control loop,
– control source of the physical outputs,
– input data to other **Logical Channels**,
– data source for visualisation and logging.

Various device functionalities are related to the number of **Logical Channels**. Their amount is shown in table below:

Type	Description	N16
LC	logical channels	60
Di	digital inputs	4
Ai	analog inputs	4
Rel/SSR	relays / SSR	4
Ao	analog outputs	4
MR/MT	Modbus ports RTU/TCP	2
F	math functions	34
G	groups	10
P/t	profiles / timers	8
C	controllers	8

Tab.5.1 Maximum amount of functions included in the device
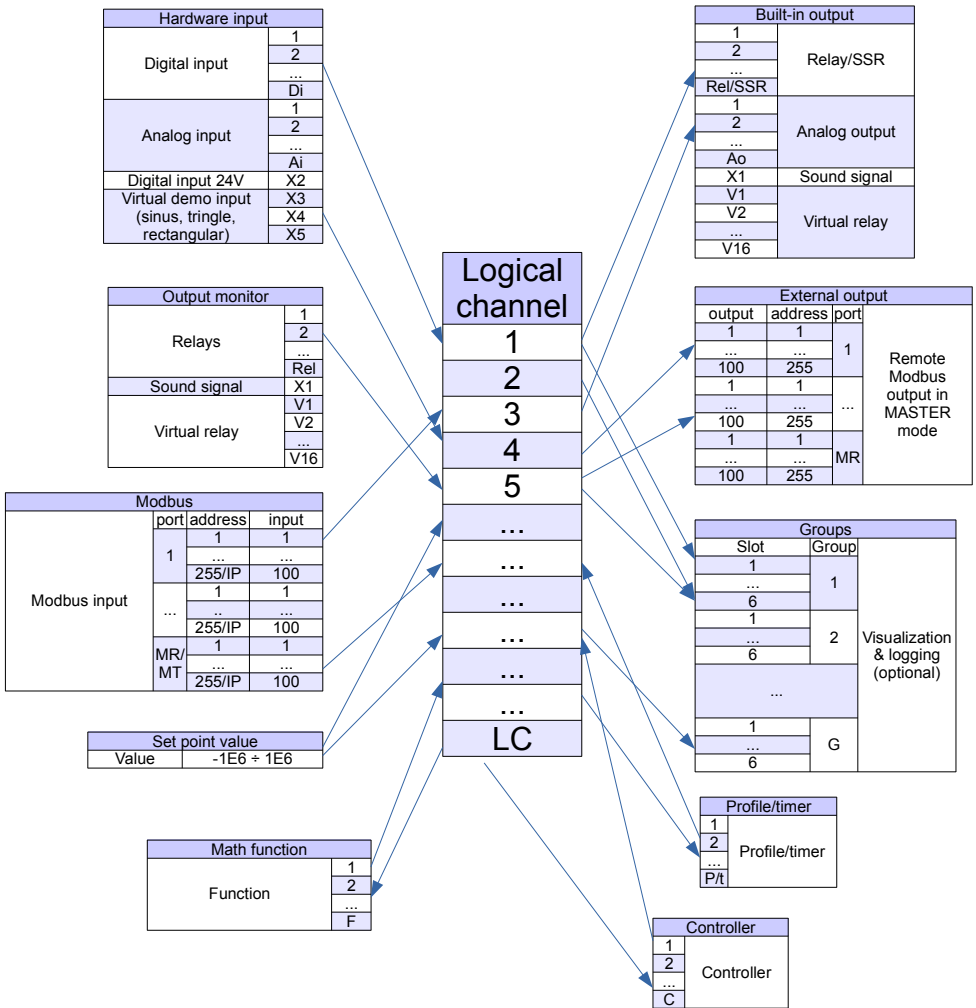
Fig. 5.1. The overall connections structure of the Logical channel with the device I /O

5.1 shows the general structure of connections between logical channels and device inputs/outputs. Each of the *Logical Channels* can be configured to represent:
- measurement data from built-in physical input channels,
- output data and states of physical output channels,
- output data and states of external modules connected to **MultiCon CMC-N16** via RS-485 interface,
- generated profile/timer
- states and data coming from outputs of controlling processes,
- states of virtual input channels and timers,
- mathematical combination of other *Logical Channels*.

More information about Logical channels and samples Logical channels configuration are presented in **Chapter 7.9LOGICAL CHANNELS**.

To make the visualization clearer, *Logical Channels* can be gathered into *Groups*.

5.1.2. Groups

      A *Group* is a set of 1-6 *Logical Channels*. The **MultiCon CMC-N16** can show on the same screen only channels belonging to the same *Group,* additionally each *Group* has its own individual name making operation with the device very clear. Every *Logical Channel* can belong to one or more groups simultaneously, and also not to belong to any group (then it will not be shown, but it can still be used for other processes). It is common that channels belonging to the same *Group* are related to one another in some way (for example representing parameters of a single object or representing similar parameters of a few separate objects) but it is also possible to create a *Group* consisting of completely unrelated channels. Overview of the concept is presented in **5.2**.
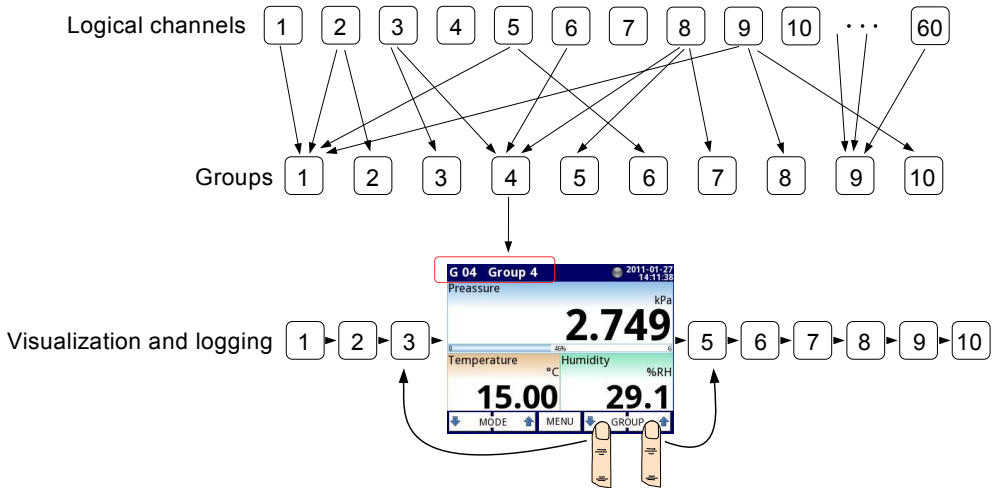
Fig. 5.2. An overview of the concept of Group in the device

Using *Groups*, *Logical Channels* and mathematical combinations of them offers an incredible flexibility to the software, allowing an easy designing of advanced control methods and visualisation at a low cost **MultiCon CMC-N16**.

More information about **Groups** and samples of Group configurations are presented in **Chapter 7.15. GROUPS**.

5.2. HARDWARE CONFIGURATIONS

The functionality of **MultiCon CMC-N16** can get adjusted to the user's needs. The *base* of the **MultiCon CMC-N16** contains: the main processor, display with a touch-screen, Switch Mode Power Supply (in one of two versions: 19V...**24**...50V DC 16V...**24**...35V AC and 85V..230..260V AC) and basic communication interfaces: USB and RS485, see **4.12**. All the other blocks are optional and can be installed inside the device according to customer's needs.

In Block D there is a possibility to install an additional ports: 10Mb Ethernet and RS-485.
An additional IP67 connectors exit on the housing (see **4.13.**) is also optional.

Two Blocks designed for built-in hardware input configurations are placed on the left side of the housing (see **4.9** - connector groups marked as: **Block A** and **Block B**). Block designed for built-in hardware output configurations is placed on the right side of the housing (see **4.9** - connector group marked as **Block C**). The number and size of these terminals varies depending on hardware configuration. Brief descriptions of available hardware configurations are shown in **Fig. 4.10. ÷ Fig. 4.14.**. Measurement and actuator hardware configurations are constantly being developed, so the current list of available configurations varies (visit the manufacturer's website to check the current list of **MultiCon CMC-N16** configurations).

Optional hardware configurations for Block A are:
– 2/4x universal inputs with analogue flowmeter function (flowmeter/tachometer $0/4 \div 20$ mA),

Optional hardware configurations for Block B are:
– 2x counter inputs with tachometer function,
– 4x binary inputs,

Optional hardware configurations for Block C are:
– 2/4x 1A/250V relay outputs,
– 2/4x passive current outputs,
– 2/4x OC-type outputs with PWM mode,
– 2x passive current outputs + 2x 1A/250V relay outputs,
– 2x passive current outputs + 2x OC-type outputs with PWM mode,
– 2x 1A/250V relay outputs + 2x OC-type outputs with PWM mode,

An optional hardware configurations for Block D are:
– Ethernet port with optional IP67 connector exit on the housing,
– RS-485 port + Ethernet port with optional IP67 connector exit on the housing.

6. WORKING WITH THE MultiCon CMC-N16

6.1. MultiCon CMC-N16 POWER ON AND OFF

After powering up a starting Logo is showed on the **MultiCon CMC-N16** display. While the operating system is being loaded a progress bar is visible in the middle of the screen. During this process the view of the screen may stay dimmed for 3-5 seconds. Please wait until the end of this operation before starting to operate the device. Additionally, in the last phase of loading the software version is displayed in the bottom left corner. After that the main application is started. The view of the main program depends on **General settings** (see **Chapter 7.8. GENERAL SETTINGS**) and **Group** settings (see **Chapter 7.15. GROUPS**). An example view of the main program is shown in **6.4**.

The **Safe shutdown** button allows to safe power down of the device. After pressing the button and accepting the **warning message** after a short time the screen will look like in **6.1**. Now, the user can power off the device. The manufacturer recommends turning off the device this way. This method is especially recommended when data logging is enabled. Failure to adapt to these instructions could cause the loss of saved data samples.
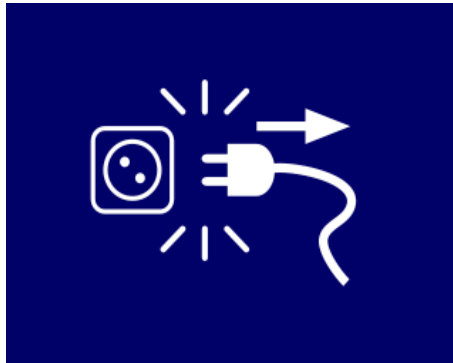
*Fig. 6.1. The view of the screen after pressing the **Safe shutdown** button*

Incorrect device shutdown occurs when the power supply to the device is stopped before pressing the **Safe shutdown** button and the message from **6.1** appears. It may occur as a result of e.g.: disconnecting the power supply with an external switch, automatic disconnection of the fuse from the supply circuit of the device or failure of the power supply network. When the device is restarted after incorrect turning off, a message contained in **6.2** will be displayed.
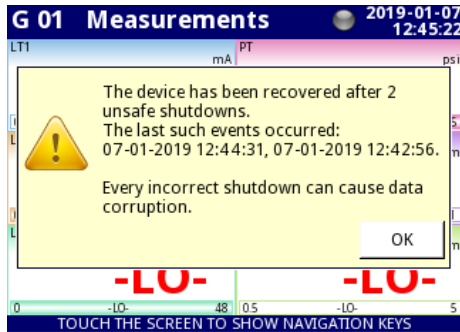
Fig. 6.2. The view of the screen with a message about unsafe shutdowns of the device

This message shows the number of incorrect shutdowns and the dates and times of the restarts. If the number of such events increases above 5, the message will take the form showing in **6.3,** giving the last five recovery dates. The counter of such events increases until the message acknowledgment.
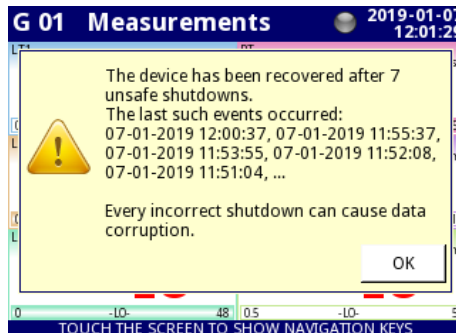
Fig. 6.3. The view of the screen with a message about 7 unsafe shutdowns

The correct connection of the device to the mains is described in **Chapter 4.3. CONNECTION METHOD**.

6.2. THE USE OF THE TOUCH-SCREEN

Do not use pointers with sharp edges (like tips of pencils and pens, knives, scissors, needles, wires, nails, screws, bolts etc.) while working with the touch-screen. It is strongly recommended to use a special stylus made of plastic or another soft material with rounded ends (for example the pointer delivered with the device) or a finger. The display of the **MultiCon CMC-N16** should also be protected against aggressive substances and extremely high and low temperature (see technical data in **Chapter 3. TECHNICAL DATA**).

To clean the LCD screen you should use a special detergent designed for LCD screens and a soft cloth.

6.3. DISPLAY

The **MultiCon CMC-N16** displays all data and dialogue on a 3,5" 320x240 pixel, color TFT screen with an embedded touch-screen panel. New devices have the display protected with a thin transparent plastic cover. This protective layer should be removed before the first use to ensure perfect visibility of pictures and sensitivity of the touch-screen.

During normal operation the **MultiCon CMC-N16** displays data in a form selected by user, at any time it is possible to switch a presentation mode and group or show the configuration menu. All the details of the user interface are designed to make the use of the device easy and intuitive. To change a display mode, group or to enter the menu, touch the screen of the **MultiCon CMC-N16** and press an appropriate button in the **Navigation bar** (see **6.11**).

By touching the screen twice will appear the **Home-Back** menu available only in **SCADALite** display mode (see **Chapter 7.15.5. Groups - SCADALite**).

Further information about the menu and presentation modes is described in further chapters.
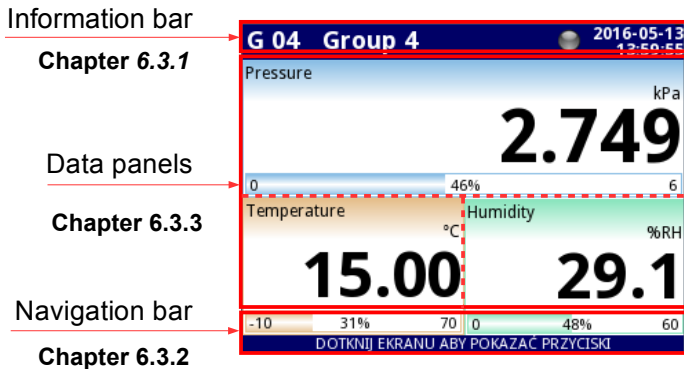
*Fig. 6.4. Typical view of the **MultiCon CMC-N16** main page, after touching the display*

6.3.1. Information bar

The **Information bar** informs the user about currently displayed group, logging, curren date and time.

name of the group which is visible on the display

date

group number   logging data indicator   time
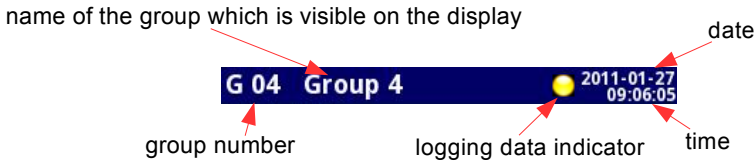
Fig. 6.5. View of the information bar

The information bar displays (**6.5**):
- – **name of the Group** visible on the screen, in place of a standard name (e.g. **Group 4**). It is possible to enter a more descriptive name for clarity (for more information see **Chapter 7.15. GROUPS**),
- – **group number** - number of the currently displayed Group, to change the displayed **Group** press button **[↓GROUP]** or **[GROUP↑]** in the **Navigation bar** (see **Chapter 6.3.2. Navigation bar**),
- – **time** and **date** - current time and date displayed in the top right corner on the screen can be set in **General settings** (see **Chapter 7.8. GENERAL SETTINGS**),
- – **logging data indicator** - located in the **Information bar** the **logging data indicator** changes color depending on the state of logging:
  - • **gray color** - data logging option is not activated (to activate the data logging option you need to enter the licence key provided by the manufacturer - see **Chapter 7.5. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION**), or when the logging option is activated but disabled.
  - • **green color** - after activation the data logging option indicator changes to green when the data logging is enabled (for more information about setting data logging see **Chapter 7.15.2. Groups - Logging**),
  - • **yellow color** - It is possible to log data in the device but there is less than **20%** of total free memory remaining (to clear the memory you need to move any important data logging files and possibly the Modbus templates onto a removable flash drive, after which remove them from the device – for more information see **Chapter 7.4. FILES MANAGEMENT**). Also at the moment when free memory drops below 20% a warning will be displayed on the screen.
  - • **red color** - warning about the lack of space on a memory card, meaning data logging will not be possible until space is freed in the memory (how to remove data and exchange data with a flash drive is shown in **Chapter 7.4. FILES MANAGEMENT**)
  - • **alternately blinking green with a blue color** - when the indicator flashes blue, the logged data is moved to memory (Note! at this time you must not turn off the device because it may cause a loss of recently logged data).

⚠ In order to turn off the device, especially when the data logging is **ON**, it is recommended to use the safe-off device by pressing the button **Safe-shutdown** in the main menu (see **6.1**).

6.3.2. Navigation bar

Touching the screen at any place causes the **Navigation bar** to display (see **6.6**) which allows the user to switch between visualisation modes, groups and to enter the menu.
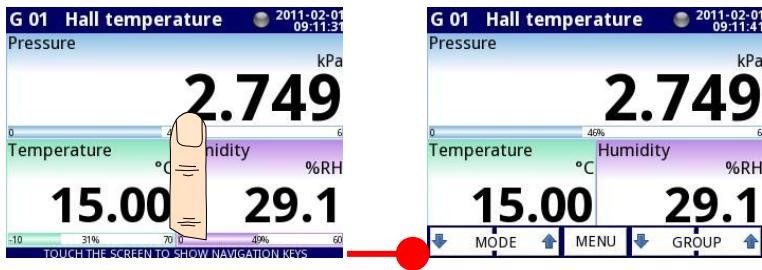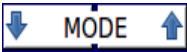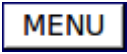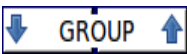
*Fig. 6.6. Main window of the device – displaying the **Navigation bar***

This bar contains three kinds of buttons shown in **Tab. 6.1**.

⬇ MODE ⬆	switching between visualisation modes of a current group (for possible modes see **Chapter 6.3.3. Data panels** and **Chapter 7.15. GROUPS**)
MENU	entering the main menu (see details in **Chapter 7. CONFIGURATION OF THE MultiCon CMC-N16**)
⬇ GROUP ⬆	switching between presented groups of logical channels (activation and settings for **Group** view see **Chapter 7.15. GROUPS**)

Tab. 6.1 Buttons of the navigation bar

ⓘ To enter the configuration menu of a particular **Logical channel** directly, press and hold the screen over the channel data panel for 3-4 seconds (see option **(1)** in **6.7** entering configuration of the **logical channel** named **'Temperature'**). Similarly, to go directly to the configuration of a displayed **Group**, touch and hold **the group number** or **group name** in the upper **Information bar** for a few seconds (see option **(2)** in **6.7** entering configuration of the **Group** named **'Group 4'**). In both cases if a password is set (see **Chapter 7.18. ACCESS OPTIONS – MultiLevel Access FUNCTIONALITY**) then the user has to enter the password before entering the configuration.
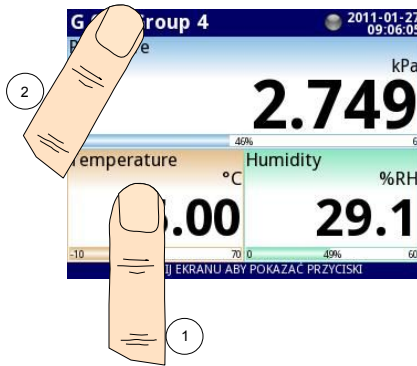
*Fig. 6.7. Methods for direct entry to **Logical channel** configuration **(1)***
*and **Group** configuration **(2)***

6.3.3. Data panels

A significant part of the screen is used for channel visualisation. Data can be presented in one of the following modes:
– as numerical values,
– as needle dials,
– as linear charts,
– as bars,
– as phasor charts,
– as **SCADALite** view,
– as controllers view.

All the channels of the current group are simultaneously presented in the same mode. In the current version of software there is no possibility to mix different modes in one view. Figures **6.9 ÷ 6.16** show examples of different views. Switching between visualisation modes can be performed by pressing the buttons **[MODE↑]** or **[MODE↓]** in the **Navigation bar** (see **Chapter 6.3.2. Navigation bar**). Switching between groups can be performed by pressing the buttons **[GROUP↑]** or **[GROUP↓]**.

*Fig. 6.8. View of the **Data panel***

In all data panels (a sample of a data panel is shown inside the frame marked **(1)** in **6.8**) the following information is available:
  – value of the logical channel (denoted by **(2)** in **6.8**),
  – data unit (denoted by **(3)**),
  – channel's name (denoted by **(4)**),
  – on some modes there is also a visible percentage indicator of the value in relation to its full scale (denoted by **(5)**).

Every **Group** of **Logical Channels** can be presented in one of the following modes:
  – as numerical values **6.9**
  – as needle dials **6.14**
  – as horizontal charts **6.11**
  – as vertical charts **6.11**
  – as horizontal bars **6.10**
  – as vertical bars **6.10**
  – as phasor charts **6.12**
  – as **SCADALite** view **6.15**
  – controllers - 1 group **6.13**
  – controllers - 4 groups **6.13**
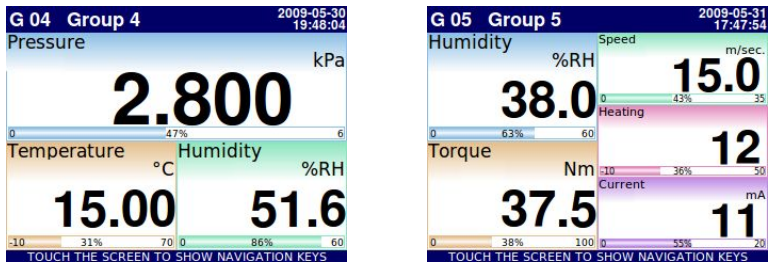
*Fig. 6.9. Examples of **Numerical Values** presentation mode*
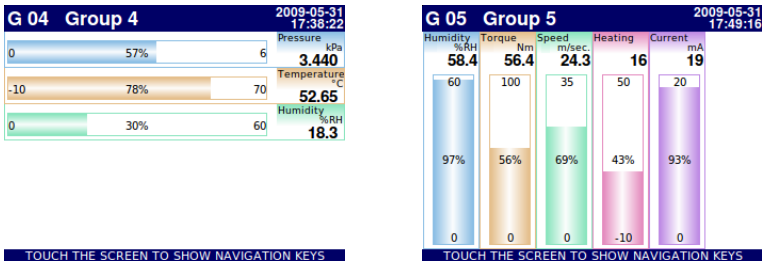
*Fig. 6.10. Examples of **Horizontal** (for 3 channels) and **Vertical Bars** (for 5 channels)*
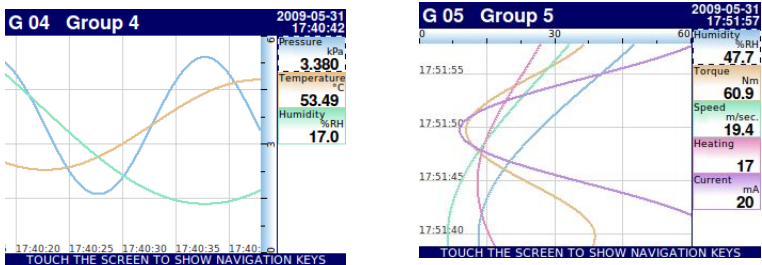
*Fig. 6.11. Examples of **Horizontal and Vertical Charts***
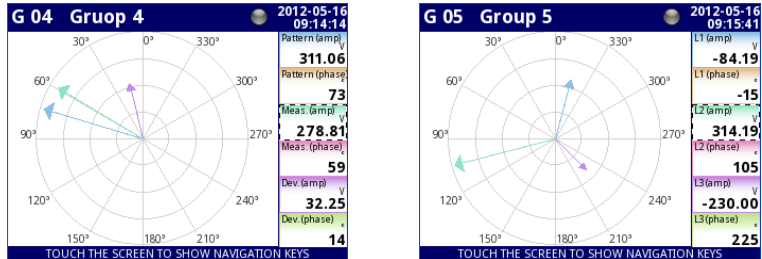
*Fig. 6.12 Examples of **Phasor Charts** presentation mode*
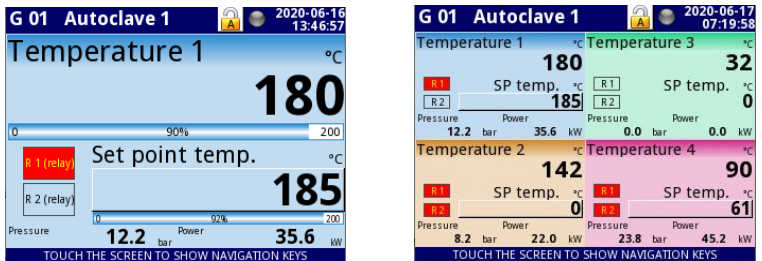
*Fig. 6.13 Example of 1 and 4 groups as the **Controllers** presentation mode*
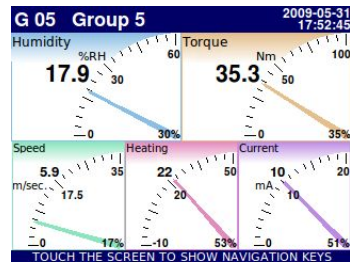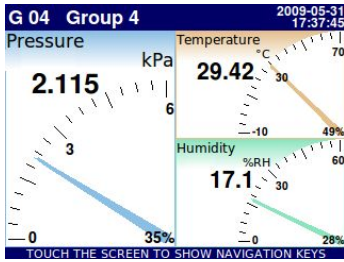
*Fig. 6.14. Examples of **Needle Dials** for 3 channels and for 5 channels*
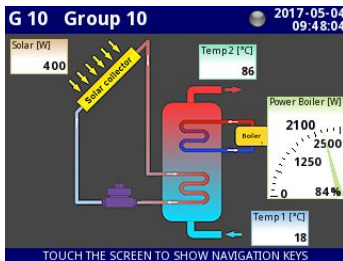
*Fig. 6.15. Examples of **SCADALite** view*
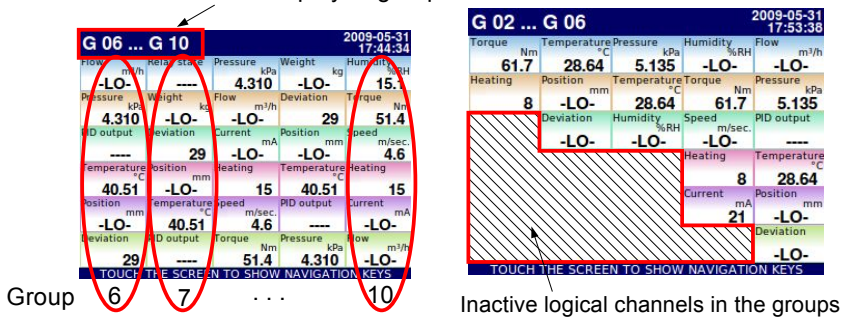
*Fig. 6.16. Examples of simultaneous presentation of **Many Groups***

It is also the possible to show many groups on a single screen (**6.16**). In this mode channels belonging to the same group are displayed under one another, and groups are placed side by side. As much as 5 groups can be displayed simultaneously on a single screen (for example, groups starting from group 8 will be displayed on the screen, starting from the left side of the page, groups: 8, 9, 10, 1, 2).

See **Chapter 7.15. GROUPS** for more information about **Groups**.

6.3.4. Important messages

The user will sometimes be asked, informed and alerted about a variety of events by messages displayed on the screen. The figures below (**6.17÷6.20**) present examples of message types.
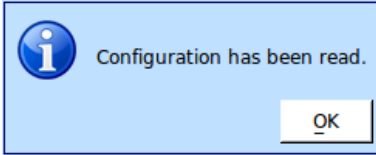
Fig. 6.17. Information message
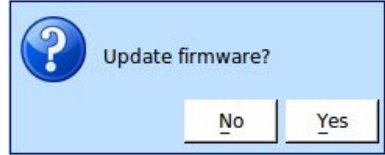
Fig. 6.18. Question message
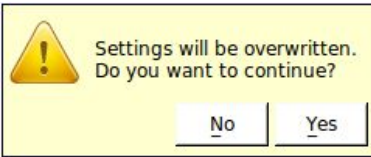
Fig. 6.19. Warning message
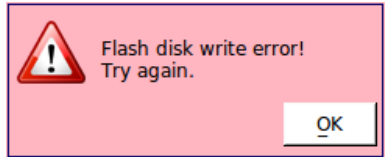
Fig. 6.20. Alert message

7. CONFIGURATION OF THE MultiCon CMC-N16

7.1. EDIT DIALOGUES

The configuration process is based on edit dialogues. Some of the dialogues are common to different menus, such dialogues are:
– text editor, which is divided into tabs:
  • letters, see **7.1**,
  • numbers and arithmetic signs, see **7.2**,
  • special symbols, see **7.3**,
  • diacritical signs (if occur in the selected language), see **7.4**,
– style of slot editor window, see **7.5**,
– values editor, which is divided into tabs:
  • decimal form, see **7.6**,
  • hexadecimal form, see **7.8**,
  • binary form, see **7.9**,
– switch editor, which is divided into:
  • single choice type options, see **7.10**,
  • multiple choice type options, see **7.11**,
– file editor, which is divided into:
  • single file selection, see **7.12**,
  • multiple files selection, see **7.13**,
– log file description window see **7.4**.
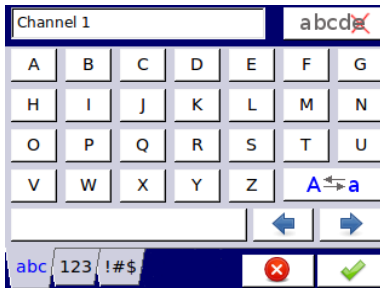
Fig. 7.1. Text editor – letters
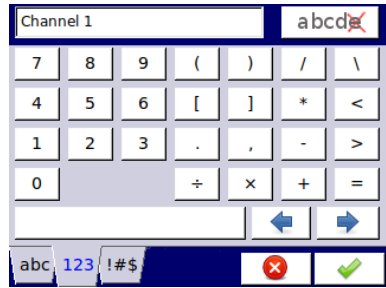
Fig. 7.2. Text editor – numbers
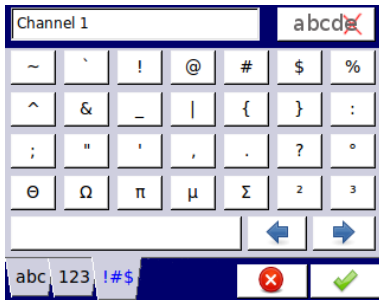
Fig. 7.3. Text editor – special symbols
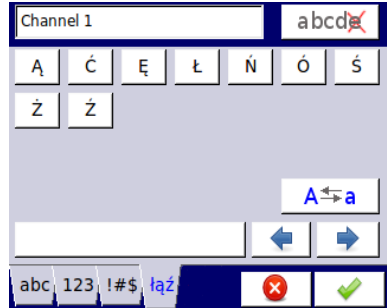
Fig. 7.4. Text editor – diacritical signs
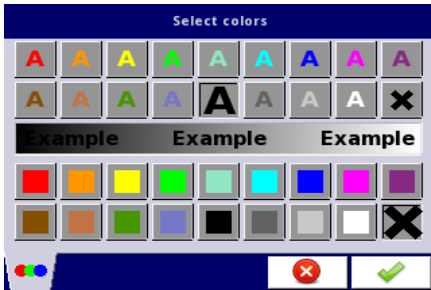
Fig. 7.5. Style of slot editor
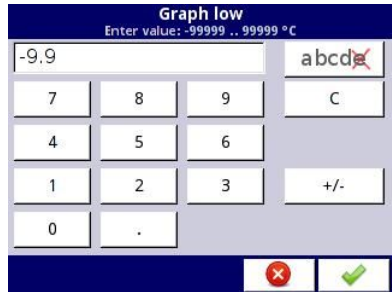
Fig. 7.6. Value editor – decimal form
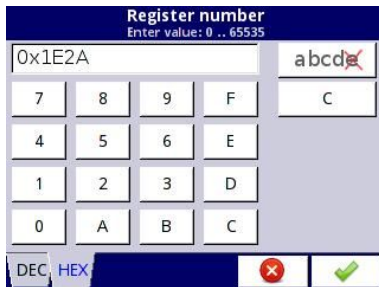
Fig. 7.7. Value editor – hexadecimal form

Fig. 7.8. Value editor – binary form

Fig. 7.9. Single choice type editor
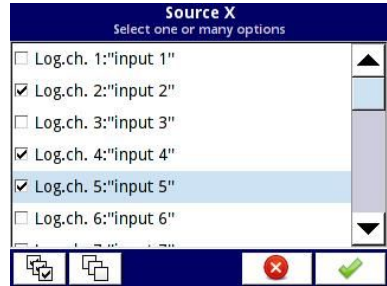
Fig. 7.10. Multiple choice type editor

Fig. 7.11. File editor - single file selection

Fig. 7.12. File editor - multiple file selection
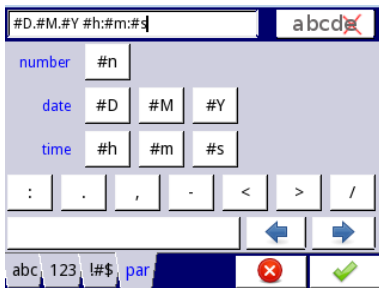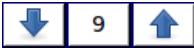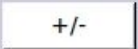
Fig. 7.13. Log file description

Functions of common buttons

	"**Exit**" - exits from current menu or sub-menu
	"**OK**" - accept the choice or changes of edit dialogue (and exit from this dialogue)
	"**Cancel**" - reject the entered choice or changes of edit dialogue (and exit from this dialogue)
	Selection of an element for editing. Arrow buttons allow the user to select successive elements (groups, logical channels, controllers or outputs). The middle button allows a direct selection of a particular element from the list.
	Navigation keys in choice type dialogues.
	Move arrows. Allow to move the cursor along the edited text.
	"**Caps lock**" - switches between lower and upper case letters.
	"**Backspace**". When editing values, pressing this button deletes the last visible number. When editing text, the last edited symbol shown directly before the cursor is deleted.
	"**Clear**" - clears the whole number when editing values.
	"**Sign**" - changes the sign of the edited value.
	This button deletes the selected file.
	"**All**" - selects all the available options.
	"**None**" - deselects all the available options.
	Press this button to enter the Text editor window.
	Add a new object
	Delete a selected object

	„Home" – moves to the home group (button available only in **SCADALite** display mode)
	„Back" – moves to the last displayed group (button available only in **SCADALite** display mode)

Tab.7.1 Button functions common for different views

7.2. ENTERING CHARACTERS NOT AVAILABLE FROM THE DEVICE MENU

When it is necessary to enter characters encoded in UTF-8 which are not available from the device GUI level, there is an option to enter them from the user's operating system using any text editor and **Xming** program (details of installation and configuration the **Xming** program can be found in **Chapter 7.5**).

The following example shows changing the name of logical channel (**Fig. 7.14**).
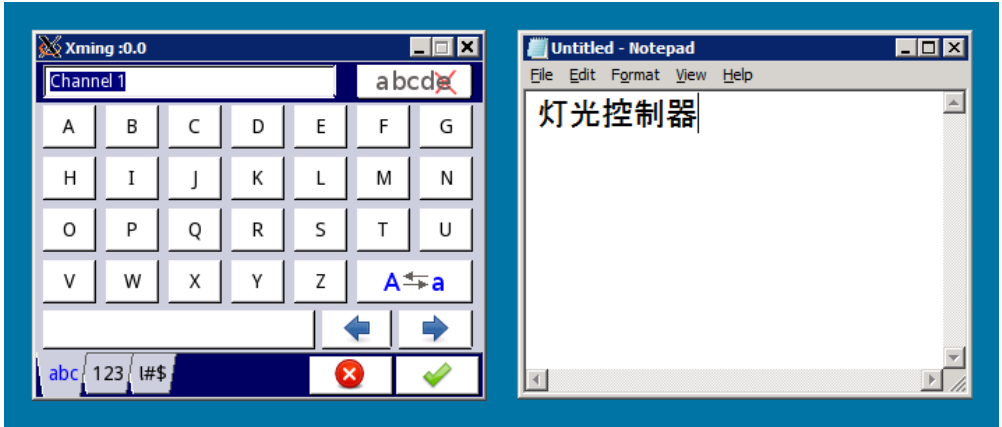
Fig. 7.14. The Xming window with logical channel name opened for editing and the Notepad window with the text prepared to replace that name

The typical procedure for entering extended characters consists of the following steps:

1. Start to edit via **Xming** program any device's text setting (**Fig. 7.15 – 2**).

2. Enter an appropriate text into any text editor in your OS. The **Notepad** can be used (available by default on every version of Microsoft Windows).

3. Select the text in **Notepad**, copy it (**Ctrl+C** or **Ctrl+Insert**) (**Fig. 7.15 - 1**) and paste it (**Ctrl+V** or **Shift+Insert**) into an appropriate input field visible in **Xming** program (**Fig. 7.15 - 2**,**3**).

4. The changed text of chosen setting should appear after applying changes in edit window (**Fig. 7.15 – 4**).

Fig. 7.15. The flowchart for entering characters that are not available from the device menu

7.3. MAIN MENU SELECTION PANEL

Pressing the **[MENU]** button on the *Navigation bar* (see **Chapter 6.3.2. Navigation bar**) enters the main selection panel (see **7.16**). This panel allows users to select between entering the **Device Configuration** menu, **Files management** menu and **Device Information** window.

Further information about the different menus is described in further chapters.
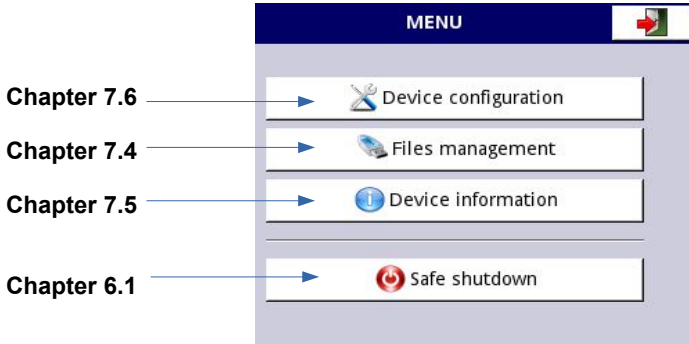
Fig. 7.16. Main menu window

7.4. FILES MANAGEMENT
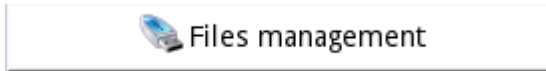
Fig. 7.17. This button allows to enter the files management menu

Pressing **MENU->Files Management** button (see **7.17**) opens the **File Management** menu, which is used to exchange data with the portable flash drive.

Flash drive requirements:
– Maximum current consumption 100mA. Some large capacity flash drives are not supported by the device (in this case, an external USB hub with power supply can be used). The manufacturer recommends using flash drives up to 32 GB.
– The flash drive must be formatted under Windows as FAT or FAT32.
– Update, configuration and template files must be in the root folder.

The view of the main **File Management** menu is shown in **7.18**. The following buttons may appear in this window:
– The **Logging files** button is visible if there is an active license for recording or if the license has expired (for more information about the license activation see **Chapter 7.5. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION**).
– The **Bundled Configuration** and **Configuration files** buttons are always visible and active.
– The **SCADALite File** button is visible when the **SCADALite** license is active on the device.
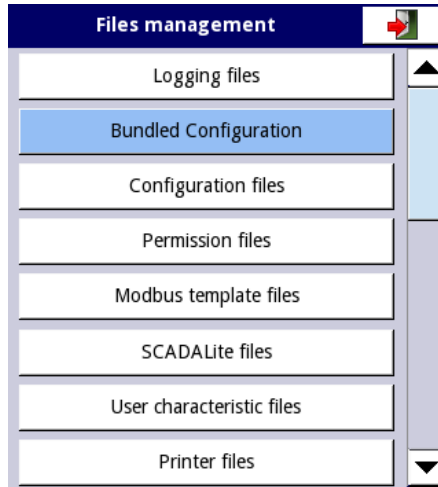
*Fig. 7.18. **Files management** menu*

Logging files:
        The **Logging files** button (see **7.18**) allows access to the recording files management menu (see **7.19**). The management of the recording files is performed by:
–   selecting a file or group of data recording files in a given group,
–   selecting successive files in successive groups,
–   exporting selected data recording files to a portable flash drive and/or deleting selected data recording files.

*Fig. 7.19. **Logging files** menu*
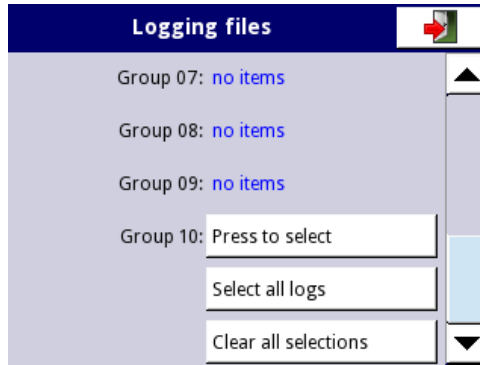
*Fig. 7.20. **Logging files** menu (cont.)*

The **Logging files** menu is shown in **7.19**. The menu consists of buttons:
–   **Export selected files** - after pressing this button the selected logging files by the user will be exported to a flash drive,
–   **Delete selected files** - after pressing this button the selected logging files by the user will be removed from the device. This button may be inactive if the user doesn't have permission to delete log files,
–   **Press to select** at the **Damaged files** label - (the **Damaged files** label appears if there is at least one damaged logging file) after pressing this button, the user goes to the window with the list of damaged files (e.g. logging files that contain errors caused by inappropriate switching off of the device during data recording) that do not guarantee the correct reading of the recorded data,
–   **Press to select** next to the labels that indicate the **Group number** - in case of switching on the recording (description of the configuration of switching on the recording for a specific group of channels can be found in **Chapter 7.15.2. Groups - Logging** of a specific logical channel group, in the Logging files menu of this group, the button Press to Select will appear instead of the label "no items". Depending on how many groups loggings have been enabled (now or in the past), there will be so many active Press to select buttons. After pressing this button, you will see a list of recordings made in this group,
–   **Select all logs** – after pressing this button, all logging files from all groups will be selected,
–   **Clear all selections** – after pressing this button, all logging files from all groups will be unselected.
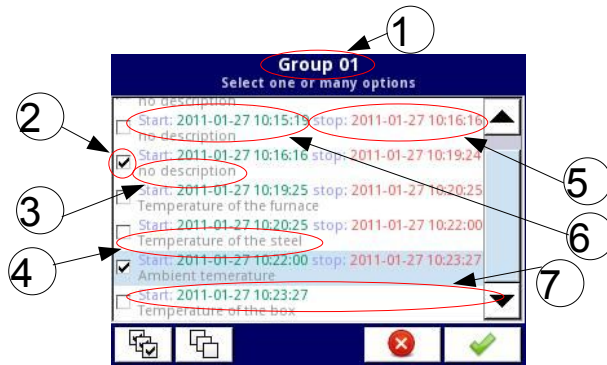
Fig. 7.21. Example view of the data logging files selection window for Group 1

7.21 shows an example view of the data logging files selection window for Group 1.
The next numbers mean:
   – **(1)** - group number,
   – **(2)** - logging file selected,
   – **(3)** - no description for the logging file,
   – **(4)** - user-defined description (the description of the logging file is the same as defined in the Groups menu - see **Chapter 7.15.2. Groups - Logging**,
   – **(5)** - date and time of stopping data recording of the selected file,
   – **(6)** - date and time of starting data recording of the selected file,
   – **(7)** - date and time of starting data recording of the selected file, the registration of which has not yet been completed.

Example of data export to flash drive:
An example of exporting 2 logging files from group 1 is shown in **7.22**. First insert the flash drive into the USB connector on the front or rear (if the device is equipped with one of the hardware communication configurations) device panel.
   – In the first step **(1)**, press the button **Press to select** on the **Group 01** label,
   – In the second step **(2),** select 2 files by pressing them and then accepting the selection with the button in the lower right corner of the screen:
      • File 1. Name: "no description", Start: 2010-12-01 3:24:58 p.m., stop: 2010-12-01 4:34:11 p.m. "
      • File 2. Name: "Ambient temperature" Start: "2010-12-03 9:53:15 a.m.", stop: "2010-12-03 9:55:00 a.m."
   – In the third step **(3)**, press the **Export selected files** button and wait for the message that the export operation to the flash drive is finished.
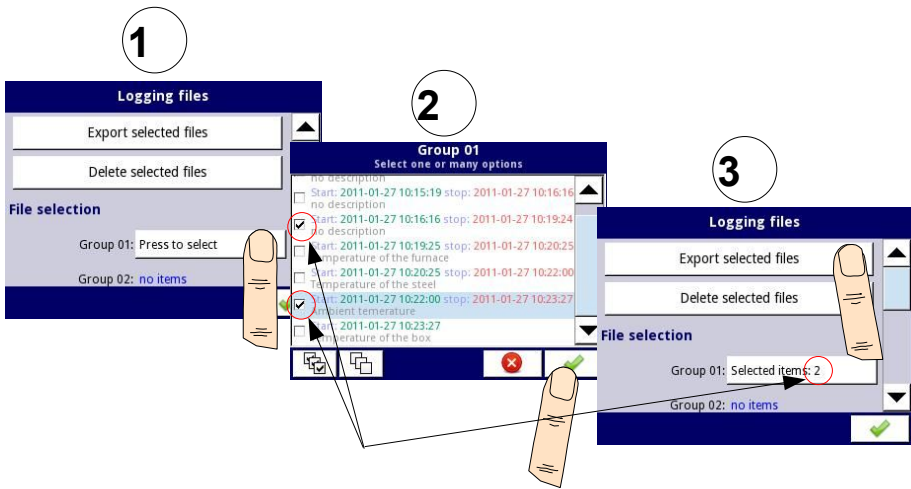
Fig. 7.22. Steps of exporting logged files to flash drive

After exporting the logging files to the flash drive, a folder with the same name as the device identification number is created, in which folders with the selected registrations are located.

Deleting files from the device is done in a similar way as exporting logging files with the difference that instead of pressing the **Export selected files** button in the third step **(3)** (see **7.22**), you should press the **Delete selected files** button.

Bundled Configuration:
The **Bundled Configuration** button allows you to go to the file management menu related to the device configuration.
The**Bundled Configuration** menu allows you to transfer a full, complete configuration between **MultiCon CMC-N16** devices and allows you to upload multiple files to the device in two steps. Use of this menu is recommended to transfer the complete configuration between devices.

The **Bundled Configuration** menu is shown in **7.23** and consists of buttons:

Load bundled configuration - allows you to load selected configuration files to the device from the bundled configuration archive located on the flash drive; the process of loading bundled configuration consists of two steps:
- selecting the archive to load from the list of available files on the flash drive,
- overview and selection of additional files to be loaded from the list available in the archive.

Save bundled configuration - allows you to save selected configuration files as a bundled archive to a flash drive; the process of saving bundled configuration consists of two steps:
- overview and selection of additional files to be saved in the archive from the list of available configuration files in the device memory,

– entering a name of the archive or selecting the archive to be replaced and saving it to a flash drive.
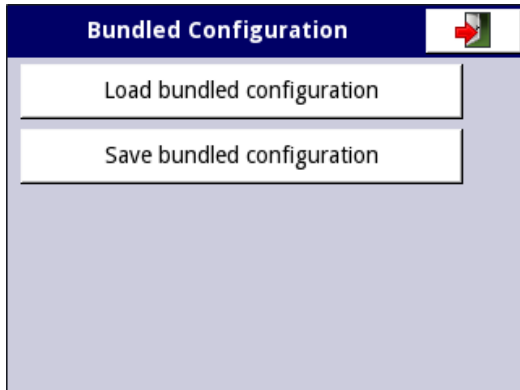
Fig. 7.23. Bundled Configuration Menu

Please note that the file name of the bundled configuration archive is assigned by the user. During the saving process, a window with available files appears, then you can select the file to be replaced or create a new file, giving it a new name by pressing the button with the key symbol in the upper left corner of the screen (see **7.24**).
Then confirm the operation and the data will be saved on the flash drive.
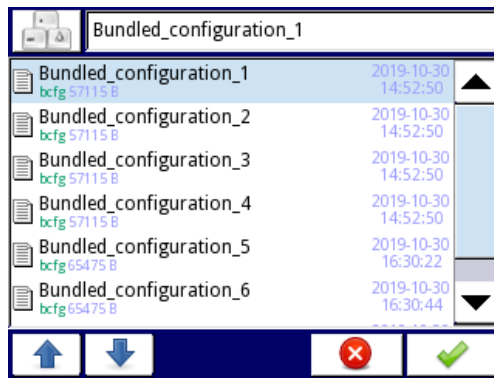
Fig. 7.24. Example of a file selection window to load a file or to name a file to save it

The **Bundled Configuration** file is an archive that collects all the files required for full device configuration and selected by the user, such as:
   • **Configuration files**
   • **Permission files**
   • **SCADALite files**
   • **User characteristics files**
   • **Modbus template files**
   • **Modbus Slave settings**

Figure 7.25 shows an example view of the window for selecting configuration files located in the device (if the **Save bundled configuration** option is selected) or contained in the selected archive of bundled configuration (if the **Load bundled configuration** option is selected).

Files used by the device configuration are selected by default and inactive in the list - the user cannot unselect them.
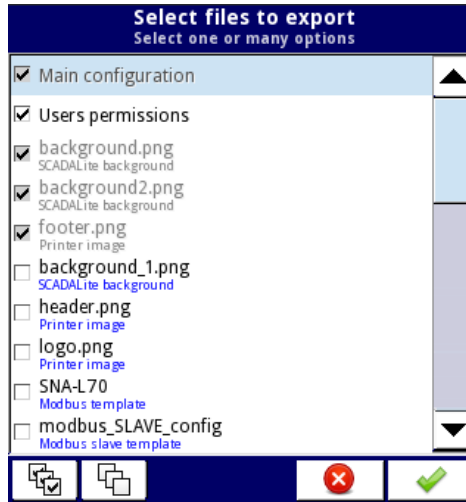
Fig. 7.25. Example of a configuration file selection window. The list shows the file name and/or configuration file type.

Files required by the main configuration but not existing in the archive (in case of loading), or not existing on the device (in case of saving), will be listed in gray with the note "*not exists*".

The **Bundled Configuration** file is an archive in **ZIP** format with extension *.bcfg (for device with display 3,5") or *.bcfg2 (for device with display 5,7") and structure shown in **7.26**.

MyConfiguration.bcfg

*MyConfiguration.**cfg***

*MyConfiguration.**ucfg***

ModbusSlaveTemplates

    *file1.**scfg***

    *file2.**scfg***

ModbusTemplates

    *file1.**mcfg***

    *file2.**mcfg***

SCADALite

    *file1.**png***

    *file2.**png***

    *file3.**png***

UserCharacteristics

    *file1.**csv***
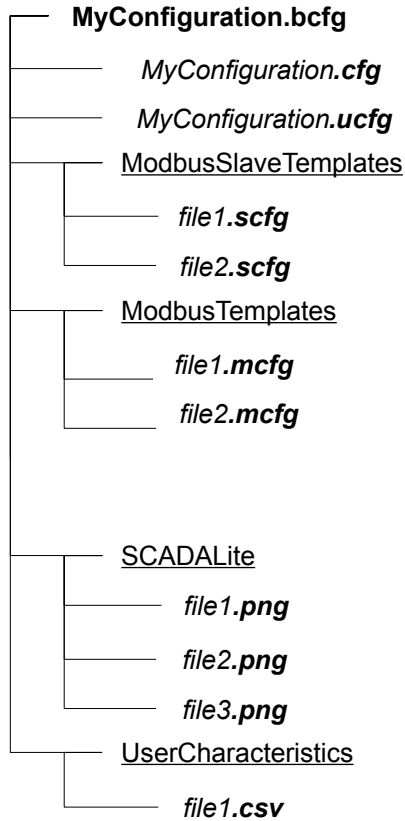
Fig. 7.26. Archive tree structure

       The main archive tree contains the main configuration file (*.cfg or *.cfg2), the user authorization file (*.ucfg) and subdirectories that contain the other configuration components, such as: **Modbus template files, Printer files, SCADALite files, User characteristics Files.**
       There can be only one master configuration file and only one user authorization file in the bundled configuration archive. It is not required for the archive to have all subdirectories. The archive may contain only certain types of configuration files.
       The **Bundled Configuration** archive in **ZIP** format (files with *.zip extension) can be prepared using any archive manager on a PC and changed to *.bcfg (for device with display 3,5") or *.bcfg2 (for device with display 5,7") respectively.

       (i)   If there is more than one main configuration file or more than one user authorization file in the archive, only the first file will be loaded, the alphabetical order decides.

Example:
Task
The task consists in uploading the main configuration and several configuration files to the device using the **Bundled configuration files** menu and using the bundled configuration archive created with a PC.

The device should contain files:
- main configuration file *main_configuration.cfg*
- user authorization file *users_configuration.ucfg*
- background for the SCADALite view: *background_1.png, background_2.png, background_3.png*
- modbus template *modbus.mcfg*
- user characteristics *characteristic_1.csv, characteristic_2.csv*
- slave settings *slave.scfg*
- print images *header.png, footer.png*

Solution:
You should create a bundled configuration archive containing the above-mentioned files. To do this you must:
- Create a folder with a subfolder structure as shown in **7.27**
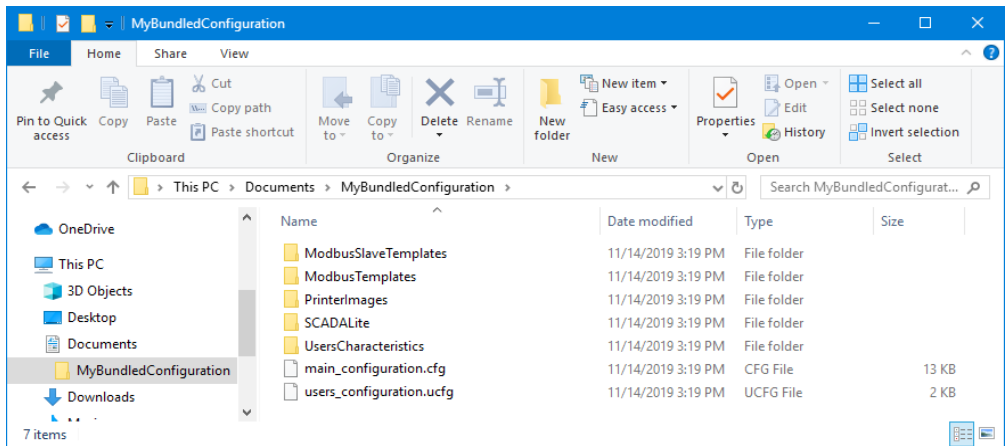
Fig. 7.27. Structure of the bundled configuration archive

- Place the file *main_configuration.cfg* and *users_configuration.ucfg* in the main path of the created folder
- Place the *slave.scfg* file in the **ModbusSlaveTemplates** subfolder
- Place the m*odbus.mcfg* file in the **ModbusTemplates** subfolder
- Place the *background_1.png*, *background_2.png*, *background_3.png* files in the **SCADALite** subfolder
- Place the *characteristic_1.csv* and *characteristic_2.csv* files in the **UsersCharacteristics** subfolder
- Place the *header.png* and *footer.png* files in the **PrinterImages** subfolder
- Pack the contents of the folder prepared in this way into the archive in **ZIP** format,

with the extension *.zip changed to *.bcfg (for device with display 3,5") or *.bcfg2 (for device with display 5,7"), e.g. with the name *BundledConfiguration.bcfg*
In this way, the *BundledConfiguration.bcfg* archive file was created on your computer.

Then copy the *BundledConfiguration.bcfg* archive file to a flash drive, place it in the USB port of the **MultiCon CMC-N16** device and select **MENU→File Management → Bundled configuration files → Load bundled configuration**.
In the next step, select *BundledConfiguration.bcfg* from the list of files available on the flash drive and confirm. In this step, select which files are to be loaded into the device. In this task, select all the files in the archive and confirm. All selected files will be loaded into the device and the device will operate according to the new main configuration settings.

> ⓘ Using the **Bundled Configuration** button allows to transfer the full configuration with all the necessary files. This is the way of transferring configuration recommended by the manufacturer. However, in special cases, can use the option of transferring a single file using one of the buttons described below.

Configuration files:
Another button in the **File management** menu is the **Configuration files** button. By pressing this button, the menu shown in **7.28**, which allows the user to load/save the configuration.
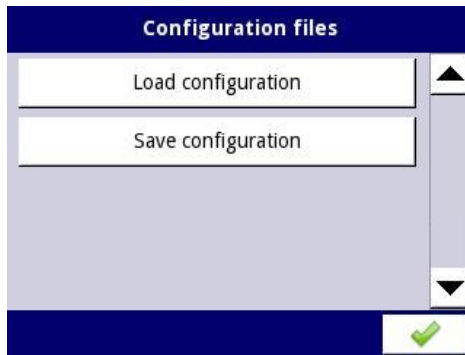
Fig. 7.28. View of the window Configuration files menu

Load/save configuration allows you to load/save the user-defined configuration that contains:
– general settings (see **Chapter 7.8. GENERAL SETTINGS**),
– settings of logical channels, built-in outputs, external outputs,
– Modbus protocol settings,
– *profile/timer settings,*
– regulators settings,
– network settings,
– group settings.

Authorization files:
Authorization files in the **File management** menu is a sub-menu related to multi-level access mode (see **Chapter 7.18. ACCESS OPTIONS – MultiLevel Access FUNCTIONALITY**). Only the Administrator has authorization to manage the authorization

files. If no user is currently logged in, it is necessary to enter the Administrator password before entering this menu (**7.29**).
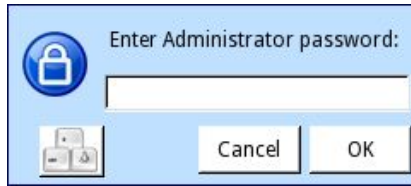
Fig. 7.29. Administrator password window

The **Authorization files** menu gives two buttons to the Administrator's disposal: **Load configuration** and **Save configuration** (**7.30**). They allow to store and transfer saved access settings using an external USB stick. This significantly reduces the time to set up subsequent devices using the same access assignment model. Saved authorization files contain only the configuration of the **Access options** (**Chapter 7.18. ACCESS OPTIONS – MultiLevel Access FUNCTIONALITY**). Configuration files from the **Configuration files** menu do not contain these settings!
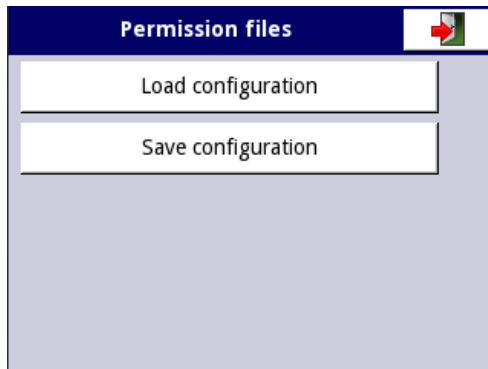
Fig. 7.30. View of the Authorization files menu window

Modbus template files:
Pressing this button opens the menu shown in **7.31**, which allows the user to load/save Modbus templates and slave settings.
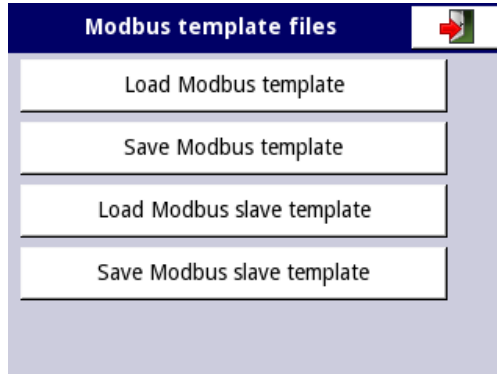
Fig. 7.31. View of the Modbus template files menu window

Load/save Modbus templates allows you to load/save configurations concerning the Modbus MASTER protocol, i.e:
– name,
– configuration of the device channels (the list of inputs and outputs),
– recording blocks configuration (list of blocks) - see **Chapter 7.16.3. Modbus - MASTER mode**.

Having such Modbus templates saved, the connection between **MultiCon CMC-N16** and SLAVE device can be quickly established at any time. You only need to select the appropriate address of the SLAVE device (more about Modbus templates in **Chapter 7.16.3. Modbus - MASTER mode**).

Load/save Modbus slave templates allows to load/save Modbus SLAVE configuration with defined recording blocks of **MultiCon CMC-N16** device, i.e.: recording type (Holding/Input), recording mode (read or write and read), data format (16 bits/32 bits). For Modbus SLAVE configuration see **Chapter 7.16.2. Modbus - SLAVE mode**.

The process of exchanging configuration files, Modbus templates or slave settings between the device and the flash drive starts with connecting the flash drive to the **MultiCon CMC-N16**. Then go to **MENU -> File Management**. Depending on the button selection, the screen will show a list of files to be exchanged:
– for the **configuration file** with *.cfg/.cfg2* extension
– for the **Modbus template file** with *.mcfg* extension
– for the **Modbus slave template file** with *.scfg* extension

    Please note that the file name is given by the user. To save, after pressing the **Save Configuration, Save Modbus Template** or **Save Slave Settings** button, a window with available files appears, which can be named accordingly by pressing the button with the key symbol in the upper left corner of the screen (see **7.32**). Then confirm the recording and the data will be saved on the flash drive. Examples of configuration files are shown in **7.32**.
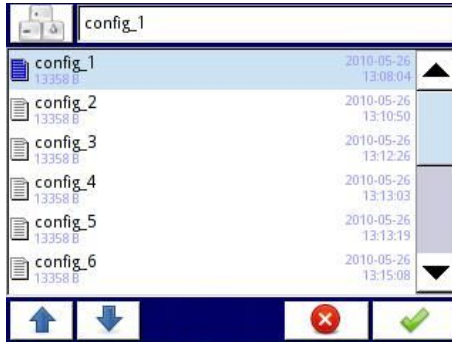
Fig. 7.32. An example of a file selection window

SCADALite files:
        SCADALite files button is visible only when SCADALite license is active on the device. The description of the SCADALite mode can be found in **7.15.5. Groups - SCADALite**.
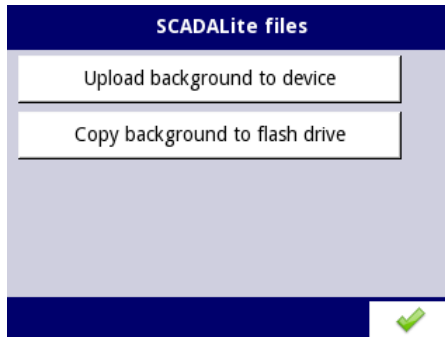
Fig. 7.33. Screen view of SCADALite files

The **SCADAlite files** menu contains the following buttons:
- **Upload background to device** – downloads the image from flash memory and saves it to the device for later use as background for **SCADALite** display mode,
- **Copy background to flash drive** – copy the image from device memory to flash drive.
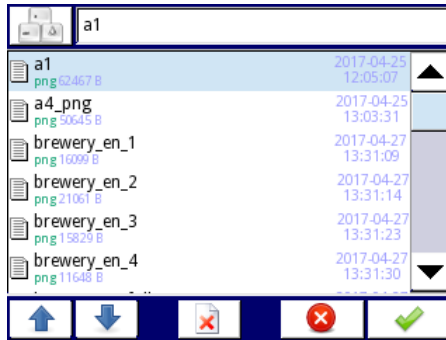
*Fig. 7.34. View of the **Select Background** button window*

The background for the **SCADALite** view should be saved in **.png* format, measuring 320x202 pixels. It can be created in any graphic program, as well as prepared on the basis of real photographs of objects or installations. The file prepared in this way should be uploaded to the device using an external flash memory. When the background dimensions are larger than recommended, the message shown in **7.35** will be displayed.
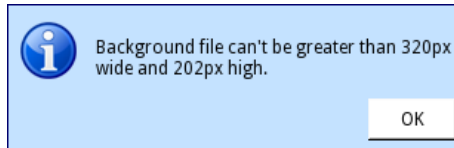
Fig. 7.35 Warning message view

If the image dimensions are larger than recommended, the image will not be loaded, and if the image dimensions are smaller than recommended, the image will be centered on a white background. If the file is prepared in a different format, it will not be recognized by the device.

Figure **7.36** shows screen dimensions for **SCADALite** display mode.

*Fig. 7.36. Dimensions of background image of **SCADALite** display mode*

User characteristics files:
By pressing this button the menu shown in **7.37** opens, which allows the user to load/save the
user characteristics.
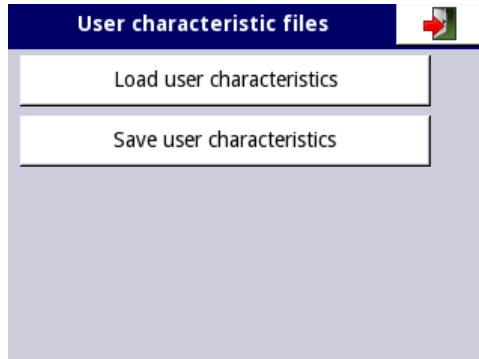
*Fig. 7.37. Example of the **User characteristics files** menu window*

Load/save user characteristics allows to load/save template of user characteristics scaling
from or to external flash memory (see also **Chapter 7.9.1. Logical Channels - General
settings**).

Structure of *.csv* file:
        The file to be uploaded to the device must have a suitable structure so that the
device can download it to the internal memory. The file must consist of two-parameter
columns separated by a field separator and each parameter pair must be on a separate line.
The decimal separator can be in form of a **dot**, then a **comma** is the list separator, and if a
comma is the decimal separator, then a **semicolon** is the list separator.
        In the line, the first parameter is the input value for the transformation (**X** value) and
the second is the output value for the transformation (**Y** value). In addition, it is recommended
that the first line contains a header. It should contain two column names with **X** and **Y** values
in quotation marks and separated by a field separator. It is also acceptable to use the
apostrophes as start and end of the field.

Errors in the file:
        The device validates the contained data before recording the characteristics.
The missing header will be detected by the device and, when saving to the internal memory,
the device will add a default header in the form of "Input"; "Output" without modifying the
source file. The value ranges are then checked (the absolute value of the parameters must be
within the closed range: <10^6, 10^{-6}>), separators (see description above), number of **X** and **Y**
coefficients (maximum 20) and repeating lines (will not be loaded). Each error will be detected
and displayed for the user to correct it.Example of application and example of file content:

Task:
        Let's assume that we have a water tank as in **7.38**. The task is to create a file with
user characteristic, which shows amount of water in the tank depending on the dipping of
the probe. Dimensions shown in the figure are scaled in meters.
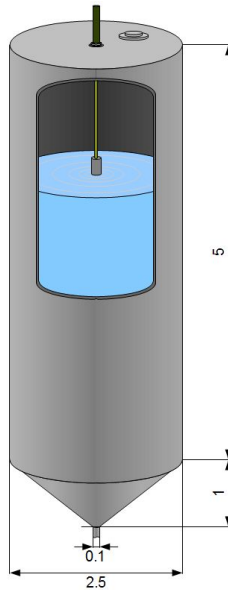
Fig. 7.38. Example of the tank in shape of combined cylinder and cone

Solution:
    Remembering that user characteristic has maximal 20 points, the user has to choose an optimal locations of the probe in the tank, which should be considered in the characteristic. Knowing that the value of a cylinder is changing linearly with its hight, we are calculating the volume of the tank with highest position (0 m of immersion) and in the place where the cylindrical and cone parts of the tank are connected (5 m of immersion). All points of characteristic that are left, divide into equal parts between 5 and 6 meters. Knowing formulas for volume of a cylinder and a truncated cone we can now calculate a volume of water in the tank depending on the probe immersion. The results of these calculations and example of **.csv** file content is shown below:

```
"Probe immersion","Volume of water in the tank"
0,26.248
5,1.704
5.056,1.439
5.111,1.203
5.167,0.995
5.222,0.811
5.278,0.652
5.333,0.516
5.389,0.399
5.444,0.302
5.5,0.222
5.556,0.158
5.611,0.107
5.667,0.069
```

```
5.722,0.041
5.778,0.022
5.833,0.01
5.889,0.003
5.944,0.001
6,0
```

The file prepared in this way can now be uploaded to internal memory of the device. The file first must be saved on a flash drive, then connected to USB socket and upload to the device (MENU → Files management → Configuration files → Load user characteristics). The characteristic can now be used in any logical channel (see also **Chapter 7.9.1. Logical Channels - General settings**).

7.5. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION
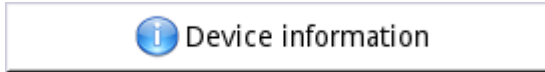
Fig. 7.39. The button which will show information about the device

The **Device information** menu offers basic information about the device and allows the user to enter a licence key for data logging, perform a firmware update of the device, run displaying on the remote screen, and download a user manual to the flash drive.
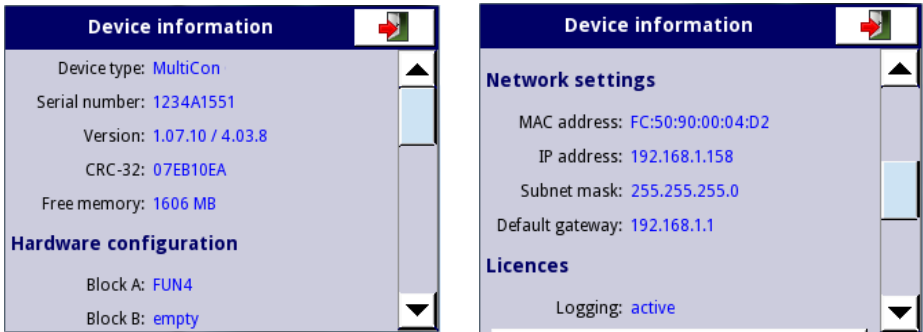
Fig. 7.40. Device Information screen

Pressing the **MENU -> Device information** button (see **7.39**) will show the window (see example window information **7.40**) with basic information about software and hardware on the device, such as:
– type of device,
– serial number,
– version,
– CRC-32,
– available free memory,
– hardware configuration - a list of installed hardware configurations (**number of block: hardware configuration type**)
– network settings,
– active licences.

Enter licence key button (see **7.42**) allows the user to enter a licence key purchased from the manufacturer (or supplier), enabling additional software options which enhance the functionality of the device. After entering and accepting the licence key the device automatically starts up with new software options (e.g. if the licence key for data logging is entered, the text under the **Licences** heading appears as: **Logging**: the period of validity – see **7.42**).

Trial licences (see **7.42**) allows the user to activate a 30-days licence for the specified functionality once if the functionality has never been activated before (30-days or permanent). After selecting and accepting the licence key, the device will automatically start up with new software functionality (i.e. if the user select the licence key for E-mail notification

in the **Device information** in section **Licences** will appear the text „**E-mail notific.: available for 30 days**" - see **7.42**).

   Firmware update button (see **7.42**) allows the user to update the device software. To perform the update:
   – download the latest software version available from the supplier website ,
   – unzip file and copy to a flash drive,
   – plug in the flash drive (some high capacity Pendrives are not supported, best choice is a pendrive with 2GB capacity) - start the update process by pressing the **Firmware update** button (see **7.42**).

Note on the update:
   – do not power off the device or remove the flash drive from the USB port during the update,
   – the update process must run continuously to the end, the user will be informed of the progress throughout the update cycle;
   Attention! The user can not start an unfinished update again because this may damage the device,
   – there can not be more than one update file on a flash drive,
   – update files must reside in the main folder (root of the drive),
   – the update process may take about 5 minutes, depending on the version of the device.

   The requirements for removable flash drives are presented in **Chapter 7.4FILES MANAGEMENT**.

   The **Remote display** button allows the **MultiCon CMC-N16** to communicate with the PC to display the screen of the **MultiCon CMC-N16** on the PC monitor to configure and view the data.

This button is available only for devices with any hardware communication configuration.

Below please find a step by step description of how to configure the **MultiCon CMC-N16** with a PC.

1. Download free "**Xming**" software from
    http://sourceforge.net/projects/xming/files/Xming/6.9.0.31/Xming-6-9-0-31-setup.exe/download
2. Install "**Xming**" on your PC using default settings. During installation enable create a desktop icon for the "**Xming**" option and disable the "**Launch Xming**" option.
3. After installation of "**Xming**" use the right mouse button to open the properties window for the "**Xming**" desktop icon and replace the following text
    :0 -clipboard -multiwindow
    with   **:0 -ac -notrayicon -clipboard -screen 0 320x240+790+400**
    (see Fig. 7.41), then close the properties window.

"...\Xming.exe" **:0 -ac -notrayicon -clipboard -screen 0 320x240+790+400**

Screen number (0÷9)

Shift the Xming window by 790 pixels to the right and 400 pixels down,
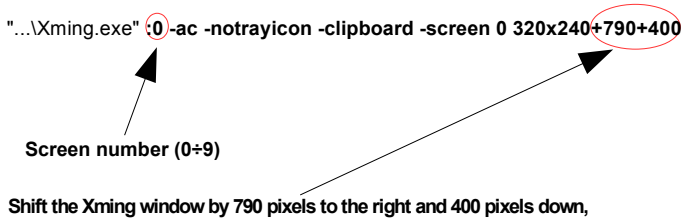
Fig. 7.41 Xming parameters description

The screen number in "**Xming**" should be set the same as the value of the **Screen number** parameter in **Network settings** submenu in the **MultiCon CMC-N16** device (see point **5** below).

4. Next close the window properties and run "**Xming**" using a desktop icon.
5. In "**Network settings**" of the device in the group "**Remote Display**" enter the **IP address** of your computer and "**Screen number=0**", then exit the menu and save the configuration changes.
6. In the "**Device information**" menu press the "**Remote display**" button and confirm switching to a remote display. After a while the device screen will be redirected to your computer screen.

ⓘ The operations described in **5.** and **6.** step may be replaced by initialize connection to the remote screen of **MultiCon CMC-N16** device without the need for physical contact with it. This process is described in **Chapter 8.8.3**.

ⓘ To establish a connection to remote display of the device, port numbers 6000-6009 (depending on screen number) must be unlocked in the software and hardware firewall.

The **Export manual** button allows you to download the user manual on to the flash drive.

The **Service options** button (see **7.42**) allows the user to:

– Delete the current configuration and restore default settings (excepting Access options – see **Chapter 7.18. ACCESS OPTIONS – MultiLevel Access**

FUNCTIONALITY). It is done by passing the password, which is **"RENEW CONFIGURATION"**.

– Calibrate the RTC clock in the device. RTC is a hardware clock which measures time when the device is switched off. This calibration increases precision of the device and should be performed in order to compensate the work temperature effect.
To do this press the **Service options** button and pass the password, which is **"RTC CALIB"**, and then enter the current time and date. After at least two weeks a password should be entered again and once again the current time and date.

– Calibrate the system clock in the device. It is a software clock which is clocked by the CPU and it measures time while the device is switched on. During the calibration a delay between RTC and the system clock is calculated and used to correct the software clock measurements. This calibration should be performed after the RTC calibration is done.
To do this press the **Service options** button and pass the password, which is **"SYSCLK CALIB"** and then the device should be powered for at least 48 hours. After this time pass enter the password again.

– Calibrate the touch screen. In time, there is a possibility that the touch screen will start to react to the user's touch a little of. It manifests itself with slight shift between place of touch and expected place where the device should react. To calibrate the touch screen, connect mouse to a USB port, press the **Service options** button and pass the password, which is **"TS CALIB"**. After confirming a message, a calibration screen will appear. The calibration is consist of touching the screen precise where five consecutively crosses appears.
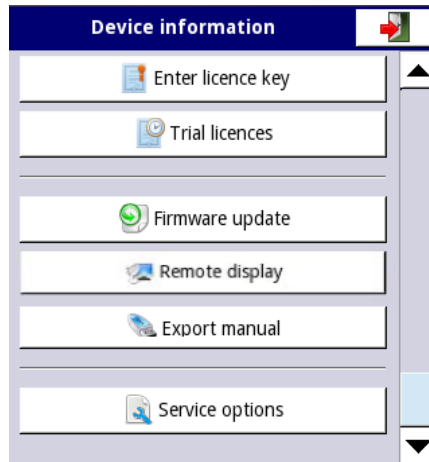
*Fig. 7.42. **Device information** menu (cont.)*

ℹ️ Remote display mode should not be used for normal device work, but only for its configuration.

7.6. DEVICE CONFIGURATION
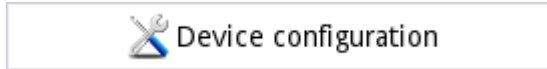
Fig. 7.43. The device configuration menu

The **Device configuration** menu is the main menu of the device that allows the user to configure all the inputs and outputs of the device to measure and control the system.

To prevent accidental or unauthorized change to the settings in the **Device configuration** menu, the user can set the access protection (see **Chapter 7.18. ACCESS OPTIONS – MultiLevel Access FUNCTIONALITY**). If this option is set, before entering each submenu, the user will be requested to enter the password.

More information about selected sub-menus is described in further chapters.

To exit the main menu, press the ⬛ button located in the upper right corner of the screen. Due to the fact that the configuration process takes place in real time, all changes must be confirmed before saving them. In the confirmation window, you can **Save** or **Revert changes** (see **7.45**).

There is a possibility to press the ⬛ button again. In that case the warning message will appear (see **7.46**), and the user must choose to save a new configuration or go to the main screen with an unsaved but active configuration. If the second option is chosen, the icon: ⬛, will be displayed on the information bar, the device will also keep a new configuration in memory and the user can determine if those changes brought a desired effect (see **7.47**). This operation can be repeated any number of times, as long as the new configuration is saved or the old one is reverted.

ⓘ   In case of power failure, changing to a remote or local display, any unsaved settings will be lost, and the last saved configuration will be restored.

Chapter 7.8	CONFIGURATION
	General settings
Chapter 7.9	Logical channels
Chapter 7.10	Built-in inputs
Chapter 7.11	Built-in outputs
Chapter 7.12	External outputs
Chapter 7.13	Profiles/timers
Chapter 7.14	Controllers
Chapter 7.15	Groups
Chapter 7.16	Modbus
Chapter 7.17	Network settings
Chapter 7.18	Access options
Chapter 7.19	E-mail notifications

Fig. 7.44. Main menu selection panel

Configuration changes

Save changes

Revert changes

Fig. 7.45. Save / revert changes window

Fig. 7.46. Warning message about unsaved configuration

Fig. 7.47. Example of unsaved configuration screen view

7.7. CONFIGURATION MENU STRUCTURE

| General settings |
| Input channels |

1

| Basic |
| Language |
| Date and time |
| Time Zone |
| Reg.num. display. |

LCD screen
| Backlight |

Screen saver
| Mode |
| Brightness |

| Display mode |
| Displayed group |
| Display time |
| Add a new view |
| Delete this view |
| Move to pos. |

| Built-in inputs |

2

Start-up view
| Selection method |
| Group |
| Default mdoe |

SCADALite
| Home group |

Automatic view change
| Change mode |
| Setup list |
| Triggering mode |
| Display time |
| Triggering source |
| Change timeout |

| Built-in outputs |

| Name |

Binary output
| Mode |
| Source |
| Alarm state |

Levels
| Level mode |
| Level |
| Low level |
| High level |
| Hysteresis |

Timing
| ON delay |
| OFF delay |
| Name |
| Min. ON time |
| Min. OFF time |

Analog outputs
| Unit |
| Source |

Input levels
| Lower level |
| Upper level |

Output levels
| Lower level |
| Upper level |
| Alarm level |

Creating a new log file
| Time delay |

PID Autotuning coeff.
| Save mode |

CSV Format settings
| Decimal separator |
| List separator |

Remote shutdown
| Mode |
| Triggering source |
| Delay |

| Comm. port |
| Device |
| Output channel |

Binary output
| Mode |
| Source |
| Alarm state |

Levels
| Level mode |
| Level |
| Hysteresis |

Timing
| ON delay |
| OFF delay |
| Min. ON time |
| Min. OFF time |

Analog output
| Source |

Input levels
| Lower level |
| Upper level |

Output levels
| Lower level |
| Upper level |
| Alarm level |

| External outputs |

3

③ ④

Name
Triggering mode
Triggering source
Triggering times
Pause mode
Pause source
Idle value
Section list
Looping
Loop count
Looping source
Return to possition

Months
Days
Week days
Hours
Minutes
Seconds

Duration
Unit
Shape
Final value

Profiles/times

Controllers ⑧

Groups ⑨

DHCP
IP address
Subnet mask
Default gateway
DNS server 1
DNS server 2
Remote display
IP address
Screen number
Web Server
Server port
NTP Server
NTP server address

Modbus ⑤

Network settings

Access options

Access mode
Access password
Access timeout
Users list

Name
Password
Permissions
Access dongle
Assign dongle
Remove dongle

⑥

⑦

5

| Mode |
| Baud rate |
| Format |
| Request timeout |
| Request retials |
| Interval |
| SLAVE devices |
| Address |
| SLAVE settings |

| Device type |
| Load device templ. |
| Device name |

Device templates
| Import from templ. |
| Export to template |

Device channels
| Input list |
| Output list |

Register blocks
| Blocks config.mode |
| Max. block size |
| Block list |
| Avail. input blocks |
| Avail. output blocks |

| Register type |
| Write mode |
| Register number |
| Data format |
| 32-bit reading |
| Ordering |
| Data shift |
| Data mask |

Channel value
| Value register |
| Decimal point |
| Dec.point register |

Channel status '-HI-'
| -HI- state |
| -HI- register |
| -HI- value |

Channel status '-LO-'
| -Lo- state |
| -LO- register |
| -LO- value |

Channel status '-WAIT-'
| -WAIT- state |
| -WAIT- register |
| -WAIT- value |

Channel status '-ERR-'
| -ERR- state |
| -ERR- register |
| -ERR- value |

Device templates
| Import from templ. |
| Export to template |

Device channels
| Output list |
| Avail. output blocks |

| Block type |
| Register size |
| First register |
| Last register |

| Output active |

Channel value
| Control type |
| Output register |

Log. ch.
| Value register |
| Status register |
| Dec.point register |

| Register type |
| Write mode |
| Register number |
| Data format |
| 32-bit reading |
| Ordering |
| Data shift |
| Data mask |

| Register type |
| Write mode |
| Register number |
| Data format |
| 32-bit reading |
| Ordering |
| Data shift |
| Data mask |

6

| Edit buttons |
| General settings |
| Logical channels |
| Built-in inputs |
| Built-in outputs |
| External outputs |
| Profiles/timers |
| Controllers |
| Groups |
| Modbus |
| Network settings |
| Logging files |
| Configuration files |
| E-mail notification |
| Printouts |
| Printer files |
| Enter licence key |
| Firmware update |
| Remote display |
| SCADALite files |
| Trial licences |

7

E-mail notifications		E-mail configuration		Mode
Messages		Sender name
	Attach. name suffix
	Sender address
	Login
	Password
	SMTP server
	SMTP port
	Security
	Retry every [min]

| Mode |
| Recipient(s) |
| Subject |
| Message text |
| Attachment |
| Select channels |
| Select groups |
| Retry when failed |
| Expiry period [h] |
| Trigger channel |
| Triggering mode |

⑨

Group
New log f. – q. access
Create a new log f.
Description
Logging options
Mode
Trigering source
Base period
Base unit
Alternative mode
Alternative source
Alternative period
Alternative unit
Log file counter
Log file auto-creation
Mode
Triggering source
Triggering mode
Display options
Name
Defined mode
Bars
Charts
Line width
Time scale
Background
Rotation direction
Offset
Labels
Channels
Slot 1
Theme
Slot 2
Theme
Slot 3
Theme
Slot 4
Theme
Slot 5
Theme
Slot 6
Theme

SCADALite
SCADALite mode
Select background
Slots
Indicator type
Dimensions [px]
Transparency
Linked group
Group number
Controllers view
Mode
Adjacent group 1
Adjacent group 2
Adjacent group 3

Controller name
Autotuning param.
Start autotuning
Mode
Set point mode
Set point value
Set point source
Feedback source
Start mode
Start source
Start triggering
Timeout
Controller param.
Mode
P coefficient
I coefficient
D coefficient
Differential signal
D Inertia
Dead zone
Controller runn. modes
Mode
Source
Triggering
Fuzzy Logic param.
Mode
Slew rate
Unit
Hysteresis
Controller output
Offset
Low output limit
High output limit
Initial conditions
Init type
Init source
Initial output

⑧

7.8. GENERAL SETTINGS

```
┌──────────┐        ┌──────────────────────┐        ┌────────────────────────┐
│   MENU   │ ━━━▶  │  Device configuration │ ━━━▶  │    General settings    │
└──────────┘        └──────────────────────┘        ├────────────────────────┤
                                                     │    Logical channels    │
                                                     ├────────────────────────┤
                                                     │     Build-in inputs    │
                                                     ├────────────────────────┤
                                                     │    Build-in outputs    │
                                                     ├────────────────────────┤
                                                     │    External outputs    │
                                                     ├────────────────────────┤
                                                     │     Profiles/timers    │
                                                     ├────────────────────────┤
                                                     │       Controllers      │
                                                     ├────────────────────────┤
                                                     │         Groups         │
                                                     ├────────────────────────┤
                                                     │         Modbus         │
                                                     ├────────────────────────┤
                                                     │    Network settings    │
                                                     ├────────────────────────┤
                                                     │     Access options     │
                                                     ├────────────────────────┤
                                                     │   E-mail notifications │
                                                     └────────────────────────┘
```

The **General settings** menu allows to configure user interface display settings, the default screen when the device powers on and automatic view change settings.

The parameters of the **General settings** menu are:
– **Basic** parameter block – this block includes two parameters:
  • **Language** - this parameter allows the user to select the language, available languages are: English, Polish, Spanish, German, Russian, French, Czech, Romanian and Hungarian.
  • **Date and time** - this parameter allows the user to set the current date and time,
  • **Time zone** – the parameter which should contain the time zone where the device works which allows to properly correct date and time in the device,
  • **Reg. num. displaying** - this parameter allows to change the mode of displaying the Register addresses for **Logical channel**
    and **External output** menu
    Two formats are available:
    ◦ **hexadecimal** - the registers are displayed in the hexadecimal format. To indicate that data has a hexadecimal format, the letter '**h**' is added at the end - for example, 12h (0x12),
    ◦ **decimal** - the registers are displayed in the decimal format (without any marker) e.g. 123
– **LCD screen** parameter block:
  • **Backlight** - this parameter allows the user to set the level of the LCD backlight. Available levels are: **20%** (the least backlight), **40%**, **60%**, **80%**, **100%** (the most backlight),
– **Screen saver** parameter block - these parameters can reduce the backlight level of the LCD screen (or make it completely blank) during normal operation, i.e. when the user does not touch the screen for a set time. This block has two parameters:
  • **Mode** - this parameter has the following options:

- ◦ **disabled** - this option turns off screen savers, the LCD screen is illuminated at all times according to parameter settings: **Backlight** (see the above **Screen saver** parameter block)
        - ◦ **1min, 5min, 10min, 30min**,
    - • **Brightness** - this parameter is hidden for **Mode = disabled**, in the other modes (**1min**, **5min**, **10min**, **30min**) this parameter is visible, the user can change the brightness level of the LCD screen after time set in parameter **Mode** elapses. The options are: **0%** (screen blank), **20%**, **40%**, **60%**,
- – **Start-up view** parameter block **-** allows the user to set the initial view of data panel when the device is turned on; this block includes parameters:
    - • **Selection method** – selects the method of setting the start-up view parameters:
        - ◦ **last view** – last displayed group and mode is presented, as just before the device has been turned off,
        - ◦ **specified view** - allows the user to set the start-up view as specified group in right display mode.
    - • **Group** - selects a group displayed at start-up. If the display mode is chosen as **many groups**, the parameter **Group** specify the first of five following groups displayed on the screen (many group mode presents 5 groups in one window). For example, when the user sets Group as Group 8 then the unit will display (starting from the left side): Group 8 → Group 9 → Group 10 → Group 1 → Group 2,
    - • **Default mode** - select the default presentation of data in the displayed group. For possible modes see **Chapter 6.3.3. Data panels**. The default mode presents groups for which no defined mode has been set (see **defined mode** parameter in **Chapter 7.15.3. Groups - Display options**).
- – **SCADALite** parameter block – this parameter block allows the user to set the home group for this display mode, this parameter block contains:
    - • **Home group** – this parameter allows to choice a group, which will be displayed as a home group after clicking on the button **Home** described in **Chapter 7.1. EDIT DIALOGUES**,
- – **Automatic view change** parameter block - this parameter block allows the user to set the display to change every time period. The parameters of this block include:
    - • **Change mode** - this parameter has the following options:
        - ◦ **disabled** - no changes in the display. For this mode the remaining parameters in this block are not visible,
        - ◦ **change modes** - allows the user to automatically change the displayed mode,
        - ◦ **change groups** - allows the user to automatically change a displayed group,
        - ◦ **detailed list** - allows to automatically change displayed views defined by the user,
    - • **Triggering mode** – this parameter has the following options:
        - ◦ **by time** – it enables to change a displayed view at fixed time intervals,
        - ◦ **from log. channel** – changes a displayed view depending on the value in a selected logical channel. If the value in this channel is less than 1, the displayed view number is 1, if the value in this parameter equals or is more than 1 but less than 2 the displayed view number is 2, and so on. The last defined view does not have the upper limit.
    - • **Display time** – this parameter is visible for the **Change mode**: **change modes**, **change groups** – duration (set in seconds) for each screen,
    - • **Triggering source** – this parameter is visible for the **Change mode**:

change modes, **change groups**, and for the **Triggering mode**: **from log. channel**; it allows the user to choose a logical channel from the list, whose value will set up the displaying view,
- **Change timeout** – this parameter determines the time from the device startup or manually change the view to the first automatic view change,
- **Setup list** – this button is visible for **Change mode** parameter setted set as **detailed list**, this parameter is described below in this **Chapter**,

Setup list parameter
After pressing the **Setup list** button the user enters the **View** menu allowing the creation of **1** to **20** views.
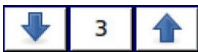
 The arrows placed in the upper right corner of the screen allow you to move to the next view. The middle button allows you to directly select a particular view.

Parameters of **View** menu are:
- **Display mode** - this parameter allows the user to select the presentation of the data in the displayed group (see parameter: **Displayed group**). For possible modes see **Chapter 6.3.3. Data panels** and **Chapter 7.15. GROUPS**,
- **Displayed group** - allows the user to select a group displayed at startup. If the **Display mode** is chosen as **many groups**, the parameter **Display group** selects the first of five following groups displayed on the screen (the mode of many groups presents 5 groups in one window). For example, when setting the parameter to **Display group=Group 8** then the display will show: starting from the left side of: Group 8 -> Group 9 -> Group 10 -> Group 1 -> Group 2,
- **Display time** - this parameter is visible for the **Triggering mode**: **by time**, it sets the duration (in seconds) of the selected view,
- **Add a new view** button - adds a new view to the list
- **Delete this view** button - deletes a selected view from the list,
- **Move to position** - this parameter allows the user to move the current view to the appropriate position,

Example:
The steps to create four views are as follows:
1. In the Change mode parameter select the detailed list,
2. Press the **Setup list** button and enter the **View** menu,
3. Set parameters for the first View,
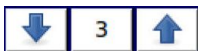
 The arrows placed in the upper right corner of the screen allow you to move to the next view. The middle button allows a direct selection of a particular view.

4. To add or delete further views use the **Add a new view** button or **Delete this view** button, respectively,
5. When the user wants to add a view between the existing views, e.g. between views 2 and 3, the user can choose the two following ways:
   - select the 2nd view (by the arrows in the upper right corner of the screen) and then add a new view by clicking the **Add a new view** button,
   - after adding a new view at the end of the list, the **Move to position** parameter set as **3** value,

6. When finished, the user can see all the defined views by clicking the middle button between the arrows in the upper right corner of the screen.

(i) For a device with a 3.5" display, below functionality is only possible from flash version 1.09. There is no limit for the device with 5.7" display.

– **Creating a new log file** parameter block allows the user to set time delay to creation a new log file. More information about logging parameters see **Chapter 7.14. CONTROLLERS**). The block includes:
   • **Time delay** – this parameter at the time of triggering from logical channel creating a new log file allows to set time delay to creating a new log file. During time delay a message showed in **7.48** is displaing, allowing the user to abort creating a new log file, when creating a large number of log files may occur a delay up to 40 seconds to create the files structure, then the values of recorded channels are save,

Fig. 7.48. Example of a message about the possibility of abort creating a new log file

– **PID Autotuning coefficients** parameter block – contains one parameter (**see Chapter 7.14. CONTROLLERS**),
   • **Save mode** – it takes two options,
      ◦ **ask before save** – after completing the autotuning process, a window with information about new PID coefficients appears, after closing it, a standard configuration save menu appears in which the user can save or reject settings,
      ◦ **automatic save** – after completing the autotuning process, the configuration is automatically saved with new PID coefficients,
– **CSV Format settings** parameter block – contains parameters which are related with creating and sending CSV files,
   • **Decimal separator** – selects a decimal separator in the CSV files for numeric values,
      ◦ **dot** – decimal separator is a dot and a comma will separate all fields in the CSV files,
      ◦ **comma** – decimal separator is a comma and a semicolon will separate all fields in the CSV files,
   • **List separator** – read only parameter. Changes according to **Decimal separator** parameter:
      ◦ **comma** – list separator is comma when **Decimal separator** parameter is

set as **dot**,
- ◦ **semicolon** – list separator is semicolon when **Decimal separator** parameter is set as **comma**,
– **Remote shutdown** parameter block – allows to set the automatic shutdown of the device,
  - • **Mode** – turns on and off shutdown from a logical channel,
    - ◦ **disable** – remote shutdown is off,
    - ◦ **from log. channel** – remote shutdown is on,
  - • **Triggering source** – contains the logical channels list, where the selected one will be the source for the remote shutdown; the shutdown procedure will start each time when the signal in the selected channel will change from value ≤ **0** to value **> 0**,
  - • **Delay** – settable time that will pass before the device will shuts down; this procedure will be stopped if during this time:
    - ◦ the signal in **Triggering source** parameter will drop to ≤ **0** value,
    - ◦ the user will press the **cancel** button on the screen.

When the device starts or when any changes have been made in the **Remote shutdown** parameter block, and the value in the **Triggering source** parameter in the selected channel is greater than 0, the shutdown procedure will not start. To start the shutdown procedure, the value in this channel must change to the value equal or lower than 0 and then to the value greater than 0.

7.9. LOGICAL CHANNELS
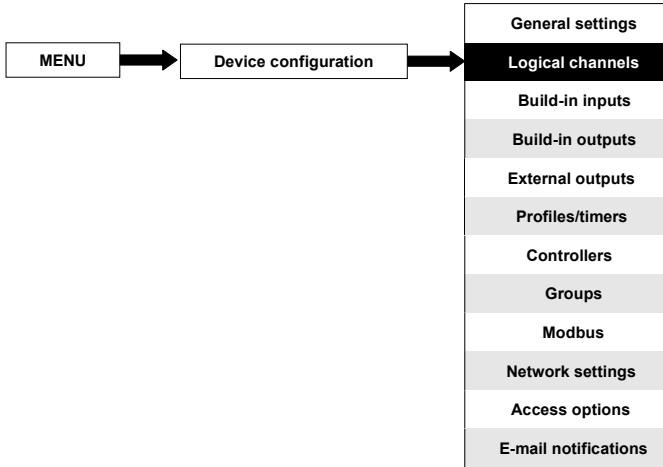
The **Logical channels** menu is used to configure the logical channels. The channels can be treated as input data for outputs, controllers or other **Logical channels** and can be collected into **Groups** for simultaneous display. To see a detailed definition of **Logical Channel** see **Chapter 5.1.1. Logical channels**.

To enter the configuration menu of a particular channel directly, press and hold on the screen over the channel data panel for a few seconds (see option **(1)** in the **6.8**). If   password is set (see **Chapter 7.18. ACCESS OPTIONS – MultiLevel Access FUNCTIONALITY**) then the user has to enter the password before entering the configuration.

7.9.1. Logical Channels - General settings
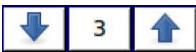
The arrows placed in the upper right corner of the screen allow you to switch between a succession of logical channels. The middle button allows you to select a specific logical channel from the list directly.

The parameters of a logical channel depend on the **Mode** of the logical channel. The **Logical channel** has the following modes:
– **disabled**
– **Hardware input**
– **Hardware output monitor**
– **Modbus**
– **Set point value**
– **Math function**
– **Controller**
– **Profile/timer**
– **Profile/timer (cycle counter)**
– **Profile/timer**
– **Profile/timer (cycle counter)**

– **Data from other channel**

A detailed description of the modes is presented in the following **Chapters**.

The channel for **Mode=disabled** has only one parameter - the **name** of the channel. In other modes the **Logical channels** are **active** and may affect the processing and control data.

The parameters and blocks of parameters common for **active Logical channels**:
– **Name** - it gives the name to the logical channel,
– **Unit** - is related with a data source of the channel,
  • for **Built-in modules** it will automatically use a default **Unit**,
  • for **Mode=Set point value** and **Mode=Controller** the **Unit** can be defined freely, directly in the **Logical Channel** menu,
  • for other modes the **Unit** can be added only using the **Scaling** parameter (see below in this **Chapter** for discussion about the **Scaling** parameter),
– **Mode** - in this parameter the user selects the source of the data for the logical channel. It is possible to select one of ten modes:
  • **disabled**,
  • **Hardware input** - see **Chapter 7.9.2**,
  • **Hardware output monitor** - see **Chapter 7.9.3**,
  • **Modbus** - see **Chapter 7.9.4**,
  • **Set point value** - see **Chapter 7.9.5**,
  • **Math function** - see **Chapter 7.9.6**,
  • **Controller** - see **Chapter 7.9.7**,
  • **Profile / timer** - see **Chapter 7.9.8**,
  • **Profile/timer (cycle counter)** – see **Chapter 7.9.9**,
  • **Data from other channel** – see **Chapter 7.9.10**,
– **Latch** parameter block - allows the user to set the **latch function** which will hold the last value of a channel; this block has the following parameters:
  • **Mode** - this parameter allows the user to choose how to trigger the latch function; there are 2 options:
    ◦ **disabled** - the latch function is disabled,
    ◦ **from logic channel** - the latch function is activated depending on the value of the channel selected in the **Triggering source** parameter,
  • **Triggering source** - this parameter is only visible if the user sets **Mode=from logic channel**; using this parameter the user chooses a logical channel which is the triggering source of the latch function (when the value of the triggering channel is **≤0** the latch is **active**, for the value **>0** the latch is **disabled**),

> During the device restart, the logical channels, which have the latch function enabled, are of the value: '**0**' and on the LCD screen blinking dashes '**----**' appear in place of the value.

– **Processing** parameter block - is used for scaling and filtering data (for explanation see below)

– **Displaying** parameter block - for these parameters the user selects the format and range of the data displayed on the screen. For more information about **Displaying** parameters see below in this **Chapter**.

Processing parameter block
For this block the parameters are:
- **Scaling**,
- **Filter type**,
- **Rounding**,

To enter the scaling menu press the button next to the **Scaling** label. The menu has the following options:
    a) **disabled** – no scaling of the input data,
    b) **linear** – after choosing this option, the following parameter appears:
- **Configure scaling** – the submenu which allows the user to change the unit in this logical channel and rescale the value using two points. This submenu contains the following options:
  - **Input unit** – read only parameter, it displays the unit of the logical channel source,
  - **Output unit** – the unit of rescaled data,
  - **Point 1** block contains the following options:
    - **Input value** – value of point 1 before scaling,
    - **Output value** – value of point 1 after scaling,
  - **Point 2** block contains the following options:
    - **Input value** – value of point 2 before scaling,
    - **Output value** – value of point 2 after scaling,

In the **MultiCon CMC-N16** linear scaling can be performed with the table placed below, but the slope and offset are constant values and they should be determined by the user earlier.

	Input value	Output value
Point 1	x_1	$y_1 = slope \cdot x_1 + offset$
Point 2	x_2	$y_2 = slope \cdot x_2 + offset$

Tab.7.2 The way to calculate the output value

Example:
The task is to rescale temperature from Centigrades, in which all of the **MultiCon CMC-N16** hardware configurations are scaled, to the degrees of Fahrenheit. The formula which should be used for this task is shown below:

$$T_F = 1.8 \cdot T_C + 32$$

where:

T_F – temperature in the degrees of Fahrenheit,
T_C – temperature in Centigrades,
9/5 – slope,
32 – offset,

Let's assume that our temperature range is from -50˚C to 85˚C (but this range is arbitrary). In that case:

– **Scale** parameter set as **linear**,
– enter the **Configure scaling** submenu:
  • in the **Output unit** parameter write: ˚**F**,
  • in the **Input value** parameter for **Point 1** block write: **-50** (which is the low range limit),
  • in the **Output value** parameter for **Point 1** block write: **-58** (according to the formula),
  • in the **Input value** parameter for **Point 2** block write: **85** (which is the low range limit),
  • in the **Output value** parameter for **Point 2** block write: **185** (according to the formula),
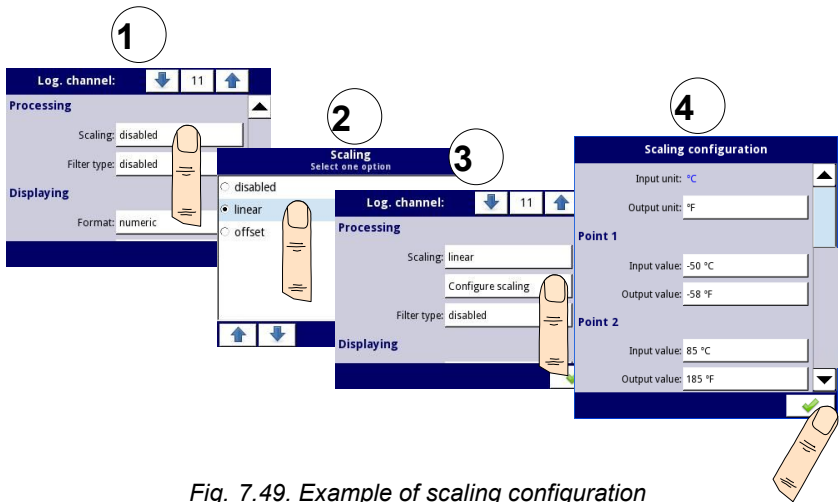
Fig. 7.49. Example of scaling configuration

c) **offset** – after choosing this option, the following parameter appears:
  • **Configure scaling** – the function which allows to add a constant value to the input value. It is obtained through the following formula:

$$y = x + offset$$

In this function, there are the following parameters:
  • **Input unit** – the read only parameter, it displays the unit of the logical channel source,
  • **Value to add** – it defines the offset value,

d) **user char.** - is defined as a set of X-Y points. The number of the points is variable and may be set from 2 to 20 points which create linear segments (see **7.50**). After choosing this option, the following parameter appears:
  • **Configure scaling** – the submenu which allows to define further points of the user characteristic. There are the following parameters:

- ◦ **Import from template** – copies from internal memory of the device earlier downloaded user characteristic template. That template is copied into a logical channel and any changes that are made in this channel are not effecting the source template,
- ◦ **Export to template** – saves in the internal memory a user characteristic template from current logical channel under desired name,
- ◦ **Input unit** – the read only parameter, it displays the unit of the logical channel source,
- ◦ **Output unit** – the unit of rescaled data,
- ◦ **Num. of point** – the read only parameter, it displays the number of points that are currently defined in the characteristic,
- ◦ **Edit points** – the submenu that defines a point in the characteristic:
  - – **Input value** – the point value before scaling,
  - – **Output value** – the point value after scaling,
  - – **Add point** button – adds a new point, behind the currently displayed one,
  - – **Delete point** button – deletes the currently displayed point, the parameter is inactive when the number of points in the characteristic is **2**.
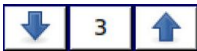
The arrows placed in the upper right corner of the screen allow to switch between points. The middle button allows a direct selection of a particular point from the list.

For 2 points the user characteristic works like a linear process (see subsection **b)**). For more than 2 defined points the user characteristic is a composite of the line characteristics, therefore for the input value **'x'** the user obtains the output value **'y'** which is described by the relationship:

$$y = slope_n \cdot x + offset_n$$

,where **'slope'** and **'offset'** are the coefficients of a segment contained between two points (see **7.50**), and **n = 1, 2 ..** is the number of the segment.

If the input exceeds the extreme **'x'** values of the designated points of P_n, the output value is scaled by the functions defined at the extreme segments.
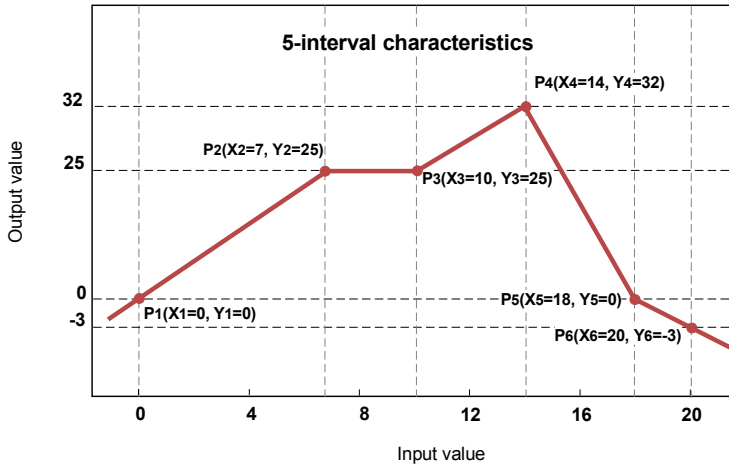
Fig. 7.50. Example of user characteristic

Example:
The steps to create a user characteristic consisting of 6 points (like in **7.50**):
1. Press the button next to the **Scaling** label and select the **user characteristic** option (point **(1)** and **(2)** in **7.51**).
2. Press the **Configure scaling** button and enter the **Scaling configuration** menu (point **(3)** in **7.51**).
3. If you want to create an output **Unit**, which replaces the input unit, or if no unit is defined on the input, press the button next to the **Output Unit** label.
4. Press the **Edit points** button (point **(4)** in **7.51**) and go to the **Edit points** menu.
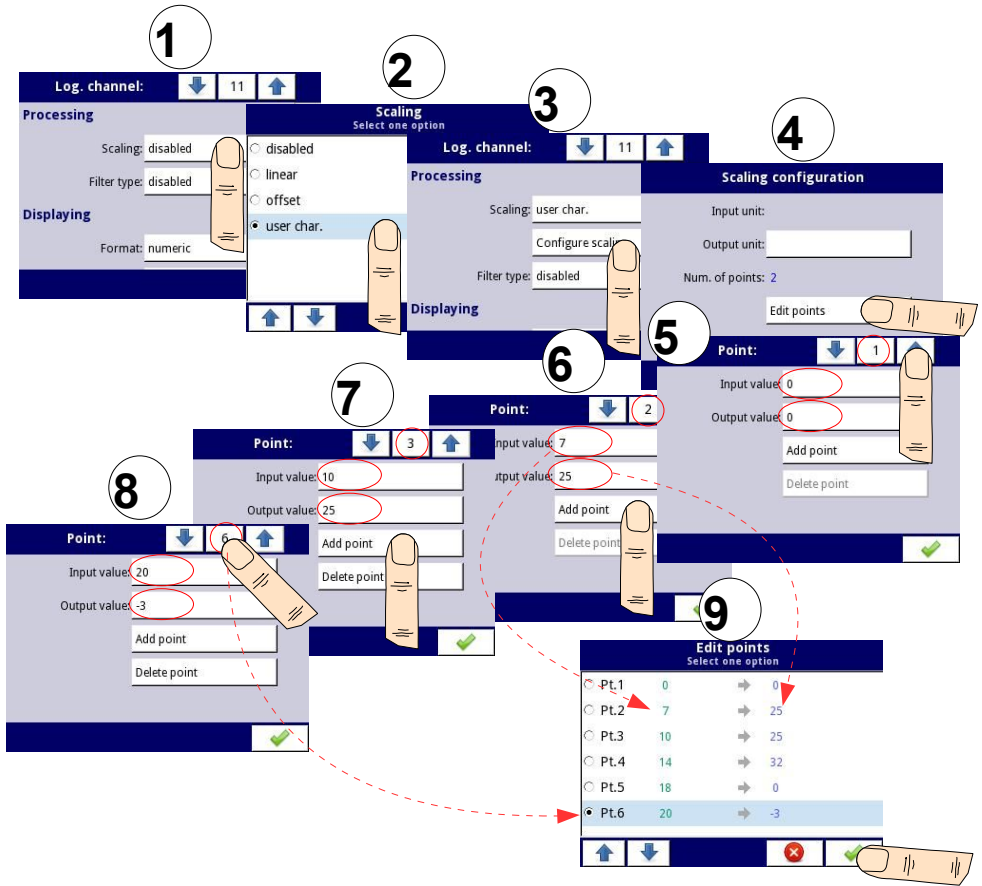
Fig. 7.51. Configuring the user characteristic

For **Point 1** set input and output value (point **(5)** in **7.51**).

    5. Switch to **Point 2** by using the arrow keys and also set the value of input and output there (point **(6)** in **7.51**).

    6. To **add** or **delete** points the user should use the **Add point** button or **Delete point** button, respectively,

    7. When the user wants to add a new point between the existing point e.g. between 5 and 6, select the edit **Point 5** and then add a new point by pressing the **Add point** button.

    8. At the end we check all the points defined by clicking the middle button between the arrows in the upper right corner of the screen (point **(8)** and **(9)** in **7.51**).

Filter type
The **Filter type** parameter has the following options:
–   **disabled** - filtering of the input value is turned off,
–   **smoothing** - this option enables the filter that is expressed by the formula:

$$Y_n = X_n \cdot \left(1 - e^{\frac{-0,1\,sek.}{w}}\right) + Y_{n-1} \cdot e^{\frac{-0,1\,sek.}{w}}$$

where:
n - number of sample, where n = 1, 2, 3 ...,
Y_n - output value for n-th sample,
Y_{n-1} - output value for n-1 sample,
X_n - input value for n-th sample,
w - time constant in seconds, this filter coefficient is defined by the user
from the **Decay constant** parameter (a value of **'0'** for the filter is turned off),
0.1 sec. - sample time.

After selecting the **Smoothing** option in the **Filter type** parameter a new button is available - **Filter configuration** which allows the user to enter a time constant with the **Decay constant** parameter (see the above filter formula).
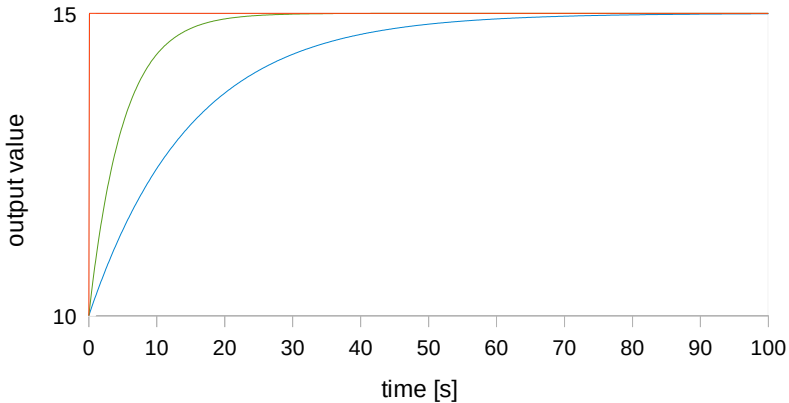
Fig. 7.52. Example of the input signal (red chart), filtered signal for the time constant = 5s (green chart) and the time constant = 15s (blue chart)

–   **peak detect** – the function which allows to detect and visualize peak values in a measured signal. Detection of the peak value can be performed if the measured signal raises and drops (or the opposite) in value, which is settable by the user. Next, the detected value can be held for a period of time settable by user. If during the time when a peak is detected, a new peak is detected, then the peak value is updated and the time of hold is reset. When this time elapses, or no peak is detected, the device will display the current value of the signal (see **7.53**).

After selecting this option, the **Filter conf.** button will appear which contains the configuration parameters of this filter:

- **Mode** – allows to choose the filter work mode,
  - ◦ **peaks** – the function detects the highest values in the logical channel
  - ◦ **valleys** – the function detects the lowest values in the logical channel,
- **Value** – the peak will be detected when the signal value drops or rises by the value in this parameter,
- **Hold time** – the time for which the detected peak will be held, on condition that there will be no new peak detected,
- **Reset mode** – turns on and off the reset of the peak value from the logical channel,
  - ◦ **disabled** – reset form the logical channel is off,
  - ◦ **from log. channel** – allows to select a logical channel in which the value that is higher than 0 will reset the peak detect,
- **Reset source** – it contains the logical channels list, where selected one on will be for the peak detect source of reset,
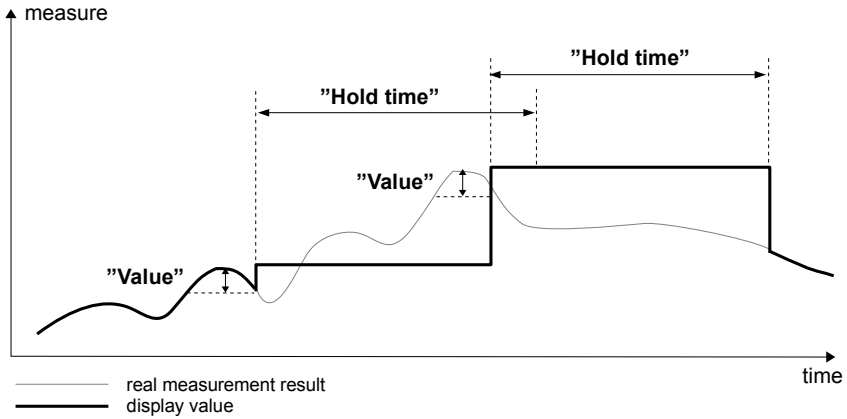
Fig. 7.53. Process of peak detection

Rounding
Rounds numerical values according to **Precision** parameter settings.
The **Rounding** parameter has options:
  – **disabled** – rounding is not active. Numerical values are stored with maximal precision. Only displaying is adjusted to **Precision** parameter settings,
  – **standard** – numerical values are rounded down when digit on less valid position is in range of <0,4>; numerical values are rounded up when digit on less valid position is in range of <5,9>,
  – **floor** – numerical values are rounded always down,
  – **ceil** – numerical values are rounded always up,

Example of **Rounding** function effect:

Input value for **Rounding** function is **3.456789**. **Tab.7.3** shows results of all available rounding options (horizontally) for all values of **Precision** parameter (vertically).

	0	0.0	0.00	0.000	0.0000
standard	3	3.5	3.46	3.457	3.4568
floor	3	3.4	3.45	3.456	3.4567
ceil	4	3.5	3.46	3.457	3.4568

Tab.7.3 Example of **Rounding** function effect

Displaying parameter block:
The constant parameters of the **Displaying** block are:
  – **Format** - the logical channel data formats, which are:
    • **numeric** – value in this channel is displayed as a number,
    • **binary** – only for values: **'0'** for a low state and **'1'** for a high state,
    • **time** – value in this channel is displayed as time. The logical channel in the **time Format** will always consider its value as a number of seconds and base on this and settings in the **Time pattern** parameter, the logical channel will calculate the value to be displayed,
  – **Time pattern** – this parameter occurs only when the **Format** parameter is set as **time**; it allows the user to define the time pattern at his will. The only restrictions are that higher units have to stand before lower units and different units have to be separated with the colon character. It is allowed to create units which contain any number of digits. When the user creates a pattern which is forbidden, the device will detect it and set the pattern as default.
  – **Precision** – this parameter occurs when the **Format** parameter is set as **numeric** or **time**, which specifies the precision to be displayed on the output value (see **7.55**),
  – **Off-state text** – this parameter occurs when the **Format** parameter is set as **binary**, when the input value is **≤ 0** the value is replaced by the text defined by the user, the default settings text is: **OFF**,
  – **On-state text** - this parameter occurs when the **Format** parameter is set as **binary**, when the input value is **>0** the value is replaced by the text defined by the user, the default settings text is: **ON**,

The text of **Off-state** and **On-state** can be:
• text with the black font such as: ALARM, off, OK,
• text using numbers and special characters such as: ALARM_ #12
• text using font color and / or a background color for example: **OFF ON**

• no text, only a rectangle with the selected color - the width of the rectangle on the screen is defined by pressing the **Spacebar** (empty string), and the color of the rectangle is the background color, for example:

– **Digits** - this parameter occurs when the **Format** parameter is set as **numeric**, allowing the user to choose the part of the number that is to be the value of the logical channel displayed on the screen.

Example of a number representing the value of the logical channel:

1	2	5	6	2	8	0	2	2	1	5	8	6	4	.	8	7	3
part 1 of 3						part 2 of 3						part 3 of 3					
						part 1 of 2						part 2 of 2					
												all digits (a limited ability to display up to 5 digits plus a decimal point on the screen)					

– Available options for the **Digits** parameter are:
• **all digits** – display all parts. With this option the number is limited only to the possibilities of the display;
• **part 1 of 2** - display the higher part of the number on the screen,
• **part 2 of 2** – display the lower part of the number on the screen,
• **part 1 of 3** – display the highest part of the number on the screen,
• **part 2 of 3** - display the middle part of the number on the screen,
• **part 3 of 3** - display the lowest part of the number on the screen,

i All channels (2 or 3) showing only a part of the same number, have to be set at the same displaying precision.

For **time Format** in the logical channel only 6 digits, three special signs (colons, dot), but not more than eight signs and minus. Using the **Digits** and **Precision** parameters, the device will display the time from the youngest unit with maintaining all restrictions, see **7.54**.
The minus signs is displayed only before the highest unit of the **Time Pattern**. When the highest unit can not be displayed in the logical channel, the minus sign will not appear and the value in this channel will be considered as a positive value.
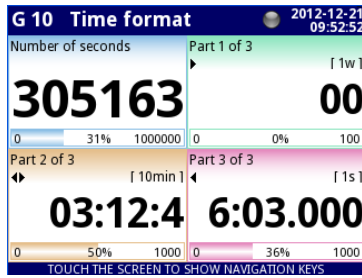
*Fig. 7.54.Example of **time Format** view with **Precision 0.000** and standard **Time Pattern** [ww:dd:hh:mm:ss]*

–   **Graph** low – the minimum range value for graphs, bars, needle dials and percentage bars (see **7.55**),
–   **Graph high** – the maximum range value for graphs, bars, needle dials and percentage bars (see **7.55**).
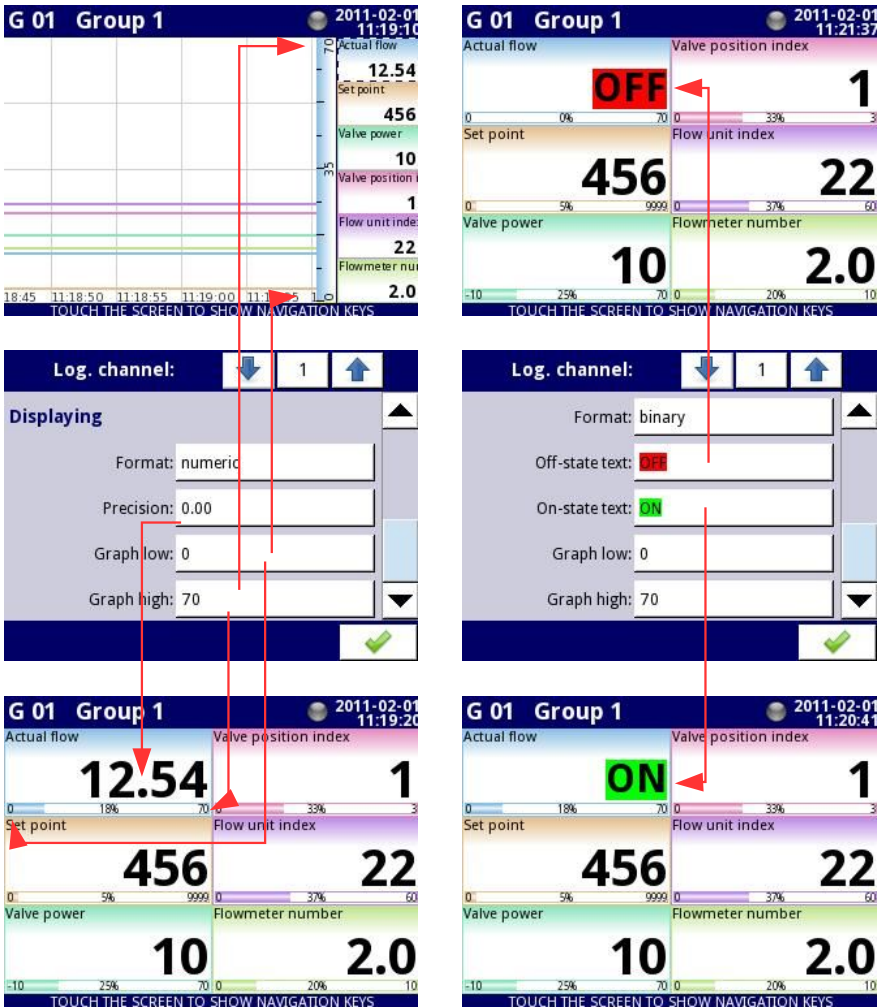
Fig. 7.55. Input channels menu – 2 different kinds of Displaying parameters

To calculate the percentage which will be displayed on the indicator, the device is using the following equation:

$$R = \frac{V - Lo}{Hi - Lo} \cdot 100\%$$

where:

R – result
V – value in logical channel
Lo – Graph low value in logical channel
Hi – Graph high value in logical channel

– **Channel highlight** – a block of parameters, used to change the background color and its mode in the logical channel. The user can set up to three different variants of highlights, depending on the event weight, which should trigger the highlight in this channel.
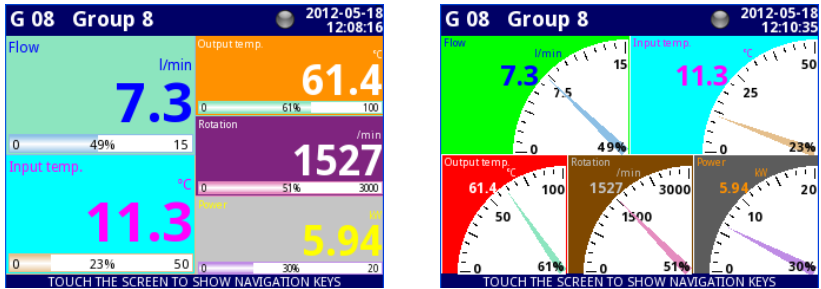
Fig. 7.56 Example of logical channel highlights

In the **Channel highlight** the user can set the following parameters:

- **Mode** – is used to change the way of the highlight displaying. There are three options: disabled, permanent, blinking,
- **Blink period** – it appears when the **Mode** parameter is set on **blinking**. It defines the blink period of the highlight,
- **Trig. hold time** – it is a minimal time during which the highlight variant will be generated, even if the generating signal disappears. However, the user should pay attention to the highlight hierarchy. If the highlight variant with a higher number is active, and the highlight variant with a lower number appears, the one with a lower number will be generated, even if the **Trig. hold time** of the first one does not expire,
- **Trigger** – using this parameter; the user can choose the logical channel which will be the source of triggering for the channel highlight. When in the source channel, there is the value of **≤ 0**, then the highlight is inactive, and when there is the value of **> 0**, then the highlight is active,
- **Alarm level** – in this parameter; the user can choose if the highlight variant is to be active, when the channel selected in the **Trigger** parameter, returns an **Err**, **Hi** or **Lo** alarm:
  ◦ **no highlight** – the selected highlight variant will not be active during the alarm state in the logical channel,
  ◦ **highlight forced** – the selected highlight variant will be active during the alarm state in the logical channel,
- **Color** – in this parameter; the user can set the background color and font color of the highlight variant,

Using the arrows placed in the upper right corner of the screen the user can choose a highlight variant which he wants to configure. The middle button allows a direct selection of a specific variant from the list.
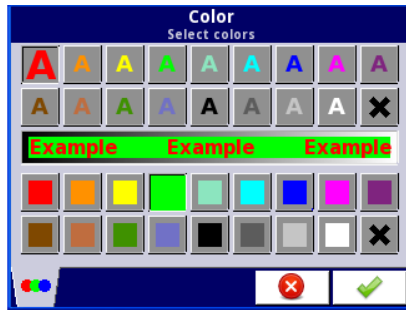
Fig. 7.57 Color edit dialog

Comments regarding the display:
– The precision of the displayed data can be set in the device with any accuracy (up to 4 decimal places), it must be remembered that the resolution and accuracy of external sensors connected to the device is finite, and usually not better than 0.1%.
– The user should pay attention to the highlight hierarchy. If there are two or three variants of highlight generated at once, the one with the lowest number will be displayed.
– The time scale is common for the entire **Group** and can be set in the **Groups** menu (see **Chapter 7.15. GROUPS**).

Examples of Channel highlight configuration.

Example 1:

Task:
  The user's task is to set a blinking highlight on the alarm channel, which will be active when the measured current value is too high or too low.

Solution:
  After connecting all measuring devices and configuring them in **MultiCon CMC-N16**, we can start to configure the highlights.
  To complete the task, we will need to configure 4 logical channels and 1 virtual relay:
a) Channel with measured current value.
b) Channel with overhead alarm threshold .
c) Channel with bottom alarm threshold.
d) Channel with highlight alarm.
e) Virtual relay which will define the proper time to activate the alarm.
Logical channel 1: Measuring the current:
– In **Name** parameter write: **Current value**,
– **Mode** parameter set as: **Hardware input**,
– **Source** parameter set as: **Inp.A1 : Current**,
– **Precision** parameter set as: **0.00** ,
– In **Graph low** parameter write: **0**,
– In **Graph high** parameter write: **22**,
– All other parameters should be set as default.

Logical channel 2: Overhead alarm threshold:
- – In **Name** parameter write: **Overhead alarm threshold**,
- – **Mode** parameter set as: **Set point value**,
- – **Edit button** parameter set as: **enabled**,
- – In **Unit** parameter write: **mA**,
- – **Precision** parameter set as: **0.00** ,
- – In **Graph low** parameter write: **0**,
- – In **Graph high** parameter write: **22**,
- – All other parameters should be set as default.

Logical channel 3: Bottom alarm threshold:
- – In **Name** parameter write: **Bottom alarm threshold**,
- – **Mode** parameter set as: **Set point value**,
- – **Edit button** parameter set as: **enabled**,
- – In **Unit** parameter write: **mA**,
- – **Precision** parameter set as: **0.00** ,
- – In **Graph low** parameter write: **0**,
- – In **Graph high** parameter write: **22**,
- – All other parameters should be set as default.

Logical channel 4: Alarm displaying:
- – In **Name** parameter write: **Alarm state**,
- – **Mode** parameter set as: **Hardware output monitor**,
- – **Source** parameter set as: **Out.V1 : Virtual relay**,
- – **Format** parameter set as: **binary**,
- – In **Off-state text** parameter write: **No alarm**,
- – In **On-state text** parameter write: **ALARM!**,
- – In **Graph low** parameter write: **0**,
- – In **Graph high** parameter write: **22**,
- – Now we go to **Channel highlight** submenu:
  - • **Highlight color** parameter set as: **1**,
  - • **Mode** parameter set as: **blinking**,
  - • In **Blink period** parameter write: **1**,
  - • In **Trig.hold time** parameter write: **1**,
  - • **Trigger** parameter set as logical channel that we are currently configuring (in this task, it will be Logical channel 4 named **Alarm state**),
  - • **Alarm state** parameter set as: **no highlight**,
  - • Now, go to **Color** parameter and set font color (at the top of the screen) and background color (at the bottom of the screen),
  - • For **Highlight color 2** and **3**, set **Mode** parameter as **disabled**.

Virtual relay V1: Defining the proper time to activate the alarm:
- – **Mode** parameter set as: **outside range**,
- – **Source** parameter set as logical channel named **Current value**,
- – **Alarm state** parameter set as: **immed. ON**,
- – **Level mode** parameter set as: **channel**,
- – **Lower level** parameter set as logical channel named **Overhead alarm threshold**,
- – **Upper level** parameter set as logical channel named **Bottom alarm threshold**,
- – All other parameters should be set as default.

<u>Example 2:</u>

<u>*Task:*</u>
        The user's task is to set a blinking highlight on the temperature channel, which will be activated on three different levels.

<u>*Solution:*</u>
        After connecting all measuring devices and configuring them in **MultiCon CMC-N16**, we can start to configure the highlights.
        To complete the task, we will need to configure 7 logical channels:
a)   Channel with first level.
b)   Channel with second level.
c)   Channel with third level.
d)   Channel to inform us when first alarm will appear.
e)   Channel to inform us when second alarm will appear.
f)   Channel to inform us when third alarm will appear.
g)   Channel where the temperature is measured, and the alarms will appear.

Logical channel 1: First level:
    –   In **Name** parameter write: **First level**,
    –   **Mode** parameter set as: **Set point value**,
    –   **Edit button** parameter set as: **enabled**,
    –   In **Unit** parameter write: **°C**,
    –   **Precision** parameter set as: **0.0** ,
    –   In **Graph low** parameter write: **-50**,
    –   In **Graph high** parameter write: **85**,
    –   All other parameters should be set as default.
Logical channel 2: Second level:
    –   In **Name** parameter write: **Second level**,
    –   **Mode** parameter set as: **Set point value**,
    –   **Edit button** parameter set as: **enabled**,
    –   In **Unit** parameter write: **°C**,
    –   **Precision** parameter set as: **0.0** ,
    –   In **Graph low** parameter write: **-50**,
    –   In **Graph high** parameter write: **85**,
    –   All other parameters should be set as default.
Logical channel 3: Third level:
    –   In **Name** parameter write: **Third level**,
    –   **Mode** parameter set as: **Set point value**,
    –   **Edit button** parameter set as: **enabled**,
    –   In **Unit** parameter write: **°C**,
    –   **Precision** parameter set as: **0.0** ,
    –   In **Graph low** parameter write: **-50**,
    –   In **Graph high** parameter write: **85**,
    –   All other parameters should be set as default.
Logical channel 4: Calculating the time of the first alarm,
    –   In **Name** parameter write: **Calc. First level**,
    –   **Mode** parameter set as: **Math function**,
    –   Click on the **Function** button,
        •   **Function** parameter set as: **X – Y**,
        •   **Source X** parameter set as logical channel where the temperature

       measurement is shown,
- • **Type of source Y** parameter set as: **channel**,
- • **Source Y** parameter set as logical channel named as **First level**,
– **Precision** parameter set as **0.0** ,
– **Graph low** parameter set as: **-50**,
– **Graph high** parameter set as: **85**,
– All other parameters should be set as default.

Logical channel 5: Calculating the time of the second alarm,
- – In **Name** parameter write: **Calc. Second level**,
- – **Mode** parameter set as: **Math function**,
- – Click on the **Function** button,
  - • **Function** parameter set as: **X – Y**,
  - • **Source X** parameter set as logical channel where the temperature measurement is shown,
  - • **Type of source Y** parameter set as: **channel**,
  - • **Source Y parameter** set as logical channel named as **Second level**,
- – **Precision** parameter set as **0.0** ,
- – **Graph low** parameter set as: **-50**,
- – **Graph high** parameter set as: **85**,
- – All other parameters should be set as default.

Logical channel 6: Calculating the time of the third alarm,
- – In **Name** parameter write: **Calc. Third level**,
- – **Mode** parameter set as: **Math function**,
- – Click on the **Function** button,
  - • **Function** parameter set as: **X – Y**,
  - • **Source X** parameter set as logical channel where the temperature measurement is shown,
  - • **Type of source Y** parameter set as: **channel**,
  - • **Source Y** parameter set as logical channel named as **Third level**,
- – **Precision** parameter set as **0.0** ,
- – **Graph low** parameter set as: **-50**,
- – **Graph high** parameter set as: **85**,
- – All other parameters should be set as default.

Logical channel 7: Temperature measurement:
- – In **Name** parameter write: **Temperature**,
- – **Mode** parameter set as: **Hardware input**,
- – **Source** parameter set as: **Inp.A4 : Intern.temp**,
- – **Precision** parameter set as: **0.0** ,
- – In **Graph low** parameter write: **-50**,
- – In **Graph high** parameter write: **85**,
- – Now go to **Channel highlight** submenu:
  - • **Highlight color** parameter set as: **1**,
    - ◦ **Mode** parameter set as: **permanent**,
    - ◦ In **Trig.hold time** parameter write: **5**,
    - ◦ **Trigger** parameter set as logical channel named **Calc. Third level**,
    - ◦ **Alarm state** parameter set as: **no highlight**,
    - ◦ Now, go to **Color** parameter and set font color (at the top of the screen) and background color (at the bottom of the screen),
  - • **Highlight color** parameter set as: **2**,
    - ◦ **Mode** parameter set as: **blinking**,
    - ◦ In **Blink period** parameter write: **3**,

- ◦ In **Trig.hold** time parameter write: **3**,
- ◦ **Trigger** parameter set as logical channel named **Calc. Second level**,
- ◦ **Alarm state** parameter set as: **no highlight**,
- ◦ Now, go to **Color** parameter and set font color (at the top of the screen) and background color (at the bottom of the screen),
- • **Highlight color** parameter set as: **3**,
  - ◦ **Mode** parameter set as: **blinking**,
  - ◦ In **Blink period** parameter write: **1**,
  - ◦ In **Trig.hold time** parameter write: **1**,
  - ◦ **Trigger** parameter set as logical channel named **Calc. First level,**
  - ◦ **Alarm state** parameter set as: **no highlight**,
  - ◦ Now, go to **Color** parameter and set font color (at the top of the screen) and background color (at the bottom of the screen),
- – All other parameters should be set as default.

7.9.2. Logical Channels - Hardware input mode
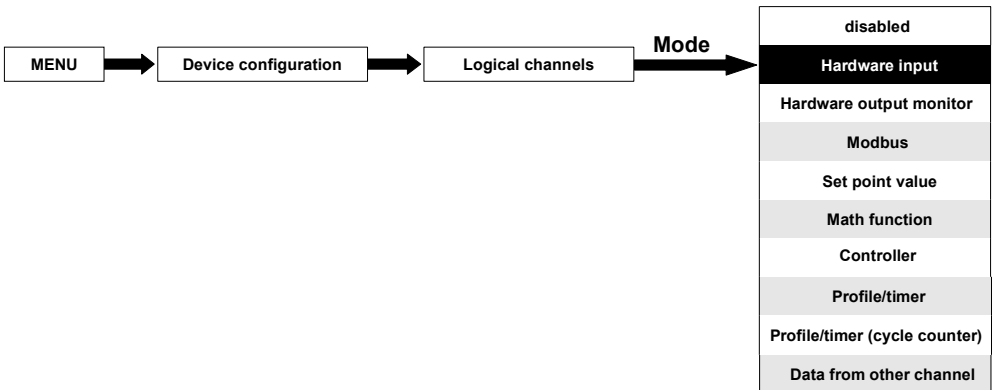
This mode allows the user to measure data from installed hardware input configurations which can be displayed, and/or processed in any other logical channels (e.g. by a mathematical function or virtual relay) or it can be the data source for controlling outputs.

The **Logical channels** parameters in the **Hardware inpu**t mode are:
- – **Name** - it gives the name to the logical channel,
- – **Unit** - for **Built-in** hardware configurations it will automatically use the default **Unit**, to change the unit use the **Scaling** parameter in the **Processing** parameter block,
- – **Mode=Hardware input** - in this parameter the user can select the source type of the data for the logical channel,
- – **Source** - in this parameter the user selects the source of the data from the hardware input list for the logical channel (see below in this **Chapter**),
- – **Configure source** button - after pressing this button the user can change the source configuration, e.g. the range of the input value (see below in this **Chapter**), another method to change the source configuration is to use the **Built-in inputs** menu (see **Chapter 7.10. BUILT-IN INPUTS**),

– **Latch** parameter block - allows the user to set the **latch function** which will hold the last value of a channel (discussed in **Chapter 7.9.1. Logical Channels - General settings**),
– **Processing** parameter block - is used for scaling and filtering data (discussed in **Chapter 7.9.1. Logical Channels - General settings**),
– **Displaying** parameter block - these parameters allow the user to select the format and range of the data displayed on the screen (discussed in **Chapter 7.9.1. Logical Channels - General settings**),

Source parameter in **Hardware input** mode

After pressing the **Source** button a list of available hardware inputs appears. The selected option will be the source of the data for this logical channel. A sample list of available hardware inputs for a device with only one hardware input configuration **FUN2** (see **Appendices 8.2 FUN2, FUN4 - UNIVERSAL HARDWARE INPUT CONFIGURATIONS WITH FLOWMETER MODE**) is shown in **7.58**.
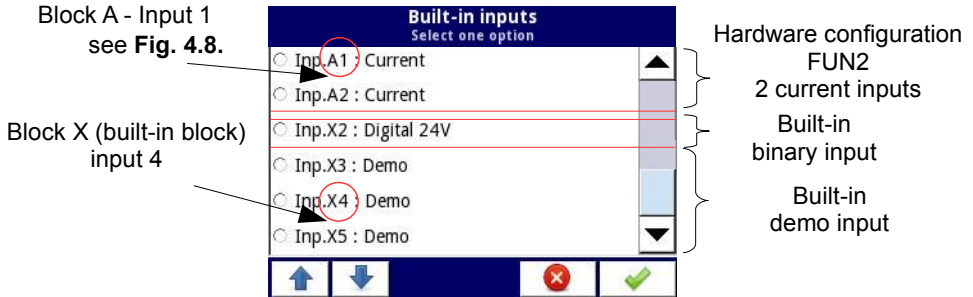
Fig. 7.58. The view of a sample list of available hardware inputs for a device

ⓘ In the device there are two methods to change the source configuration of the **Built-in inputs**:
   • using the **Configure source** button in the **Logical channel** menu in the **Hardware input** mode,
   • using the **Built-in inputs** menu.

The **Source** for the **Hardware input** mode can be (in the same order as the list in the device - see **7.58**):
 a) installed **hardware input configurations** in the appropriate blocks A, B or C (see **Fig. 4.9.**) - the list of currently available configurations is on the website,

A description of the hardware input configurations parameters is shown in **Chapter 7.10.2Built-in inputs - Hardware input configurations**.

The following steps change the **Source configuration** for the sample of 3 hardware configurations shown in **7.59**:
   – Step **(1)** - selection of the **Source** for the channel in the **Hardware input** mode, for example: **Inp.A1:Current**, next press the **Configure source** button to enter the **Source configuration** menu,
   – Step **(2)** - press the **Mode** button to change the range of the current input,

– Step **(3)** - choose from the list of available options for the signal range - for example: **Current 0-20mA** (for the current hardware configuration).
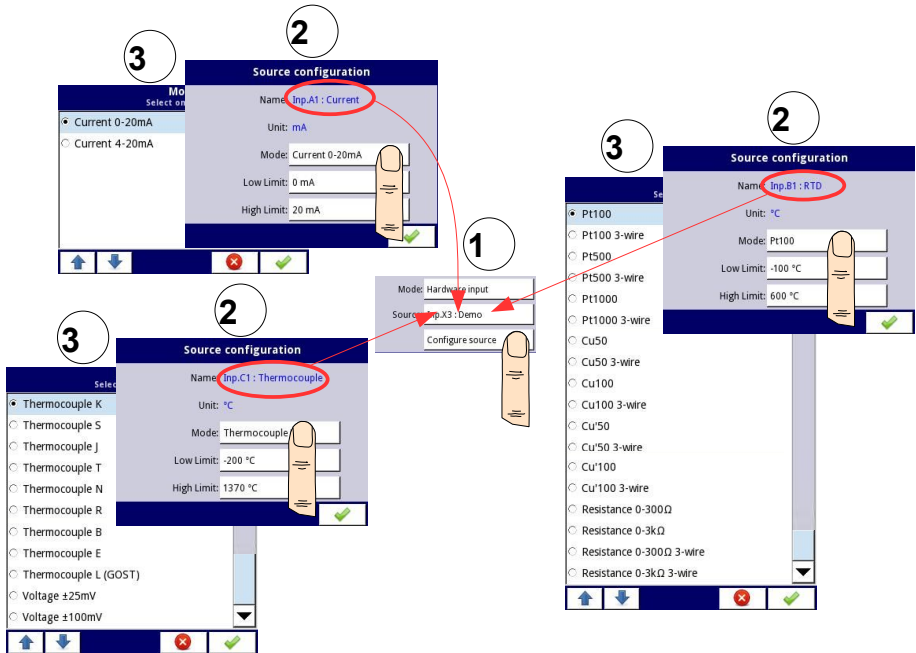
Fig. 7.59. Change source configuration for different types of hardware configurations

b) built-in digital input is always designated as **Inp.X2: Digital 24V**

The device has a built-in digital input which can be used for example as a switch for a process. The descriptions of the parameters is shown in **Chapter 7.10.3. Built-in inputs - Binary input Inp.X2 : Digital 24V**.

c) built-in **Demo** input numbered **Inp.X3:Demo, Inp.X4: Demo, Inp.X5: Demo**

The device has 3 built-in simulation **Demo** inputs which can be defined by the user. The descriptions of the parameters is presented in **Chapter 7.10.4. Built-in inputs - Demo input numbered X3, X4, X5**.

i	The user should remember that if the **Hardware input** is displayed in more than one logical channel and if the user changes the parameters in one of these channels, that will result in the same changes in all the others.

7.9.3. Logical Channels - Hardware output monitor mode

disabled
Hardware input
Hardware output monitor
Modbus
Set point value
Math function
Controller
Profile/timer
Profile/timer (cycle counter)
Data from other channel

MENU ▶ Device configuration ▶ Logical channels ▶ **Mode**
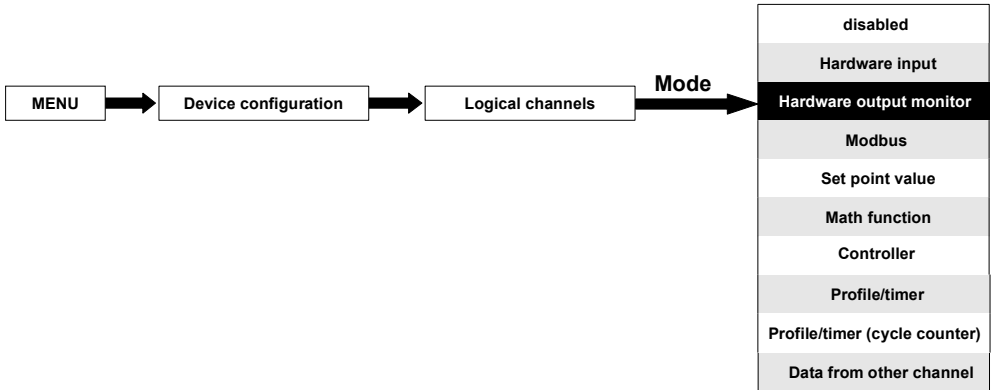
This mode allows the user to display data from the hardware output configurations, processed in any other logical channels (e.g. by a mathematical function or virtual relay) or it can be the data source for controlling another output.

The configuration view of a **Logical channel** in the **Hardware output monitor** mode is shown in **7.60**.
The parameters of a **Logical channel** in the **Hardware output monitor** mode are:
   – **Name** - it gives the name to the logical channel,
   – **Unit** - for **Built-in modules** it will automatically default to the unit of the hardware configuration. To change the unit use the **Scaling** parameter in the **Processing** parameter block,
   – **Mode=Hardware output monitor** - in this parameter the user selects the source of data for the logical channel,
   – **Source** - in this parameter the user selects the source of data from the available built-in outputs list for the logical channel (see below in this **Chapter**),

   – **Latch** parameter block - allows the user to set the **latch function** which will hold the last value of a channel (discussed in **Chapter 7.9.1. Logical Channels - General settings**),

   – **Processing** parameter block - is used for scaling and filtering data (discussed in **Chapter 7.9.1. Logical Channels - General settings**),

   – **Displaying** parameter block - these parameters allow the user to select the format and range of data displayed on the screen (discussed in **Chapter 7.9.1. Logical Channels - General settings**),
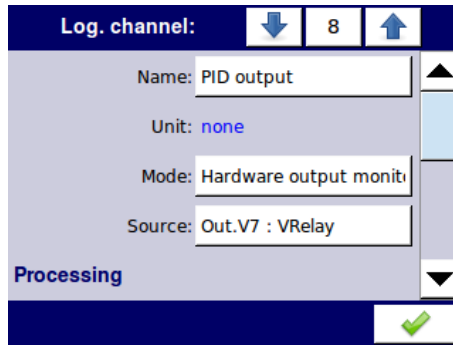
*Fig. 7.60. Input channels menu – parameters specific for **Hardware Output Monitor** mode*

<u>**Source** parameter in the **Hardware output monitor** mode</u>
After pressing **Source**, a list of available hardware outputs appears. The selected option will be the source of data for that logical channel. An example list of available hardware outputs for the device with only one output hardware configuration such as **R41** - 4 relay outputs (see **Appendices 8.4. R21, R41, COP2, COP3, OC2, OC4, R21COP2, COP2OC2, R21OC2 – HARDWARE OUTPUT CONFIGURATIONS**) is shown in **7.61**.

Source for the **Hardware output monitor** mode can be (in the same order as in the list in the device - see **7.61**):
  – installed hardware output configurations in the respective blocks A, B or C (see **7.11BUILT-IN OUTPUTS**) -a list of modules currently available is on the website; more about the hardware output configurations in **Chapter 7.11BUILT-IN OUTPUTS**,
  – built-in **Sound signal** output is always marked as **Out.X1: Sound signal** - more about the **Sound** outputs is in **Chapter 7.11BUILT-IN OUTPUTS**,
  – built-in Virtual relays marked as Out. V1 - V16 - more about Virtual relay in Chapter **7.11BUILT-IN OUTPUTS**.
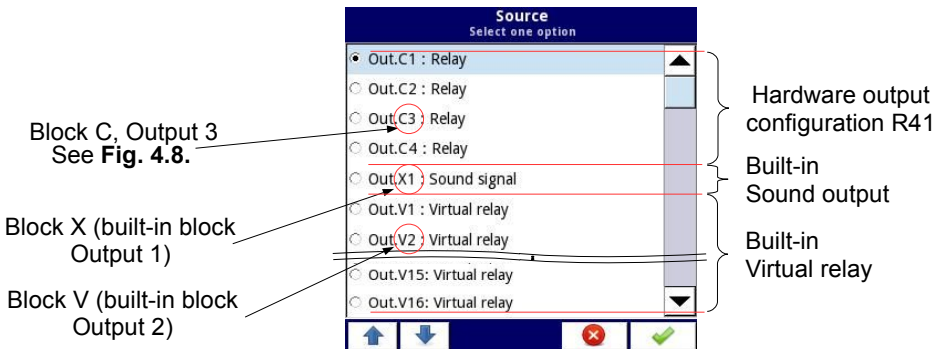
Fig. 7.61. Sample list of available hardware outputs

7.9.4. Logical Channels - Modbus mode
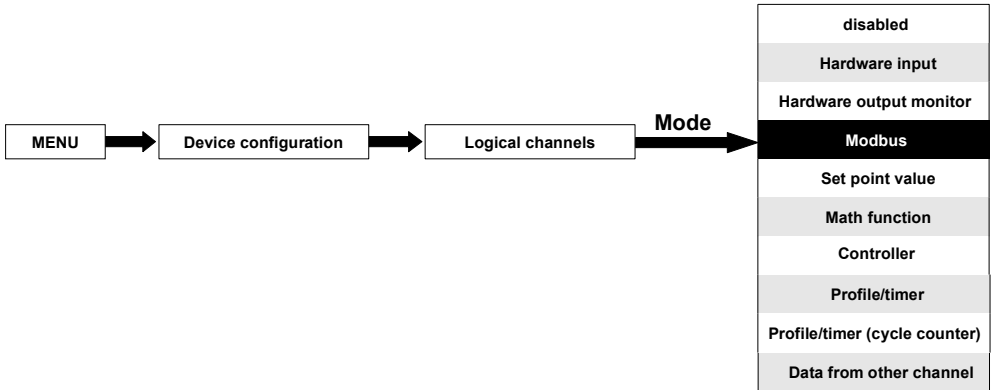
Modbus mode allows the user to configure the logical channel to:
- read/write the data from/to a SLAVE device transmitted over RS-485 Modbus RTU (**MultiCon CMC-N16** device in **MASTER** mode),
- read/write the data from/to the logical channel of **MultiCon CMC-N16** device transmitted over RS-485 Modbus RTU (**MultiCon CMC-N16** device in SLAVE mode) or by Ethernet port (Modbus TCP/IP, **MultiCon CMC-N16** device in **SLAVE** mode)

This data stored in the logical channel in the **Modbus** mode can be displayed on the screen, processed by other logical channels (e.g. by a mathematical function or virtual relay) or it can be the data source for controlling an output.
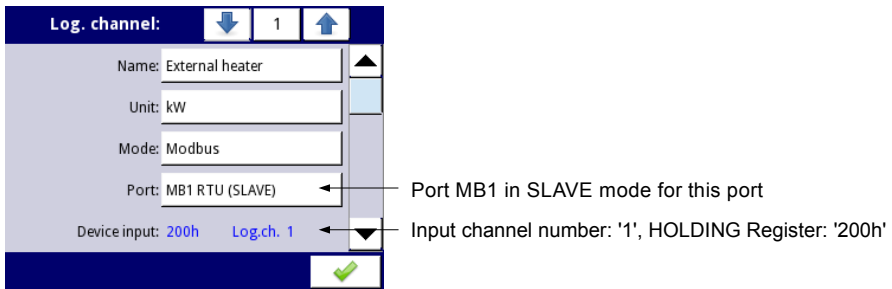
*Fig. 7.62. Sample Logical channel setting in **Modbus mode***

The parameters of the **Logical channel** in the **Modbus** mode are (see **7.62**):
- **Name** - it gives the name to the logical channel,
- **Unit** - to create a unit use the **Scaling** parameter in the **Processing** parameter block (more information see **Chapter 7.9.1. Logical Channels - General settings**),
- **Mode=Modbus** - in this parameter the user selects the source type of data for the logical channel,
- **Port** - this parameter allows the user to select a serial/ethernet port from the list. The **MultiCon CMC-N16** device has one built-in RS-485 port.
  The number of serial ports can be increased if in the device is a hardware

communication configuration in block D with Ethernet port and additional RS-485/RS-232 port, which allows the advanced Multi-Modbus system (see **7.63**).

- **Slave device** - this parameter is only visible if the selected option in the **Port** parameter is in the **Modbus MASTER** mode (see **Chapter 7.16.3. Modbus - MASTER mode**). Using this parameter the user can select the SLAVE device from the list defined in the **Modbus** menu to exchange the data with it,
- **Device input** - this parameter depends on the mode of the port
  - for the **Modbus MASTER** mode (see **Chapter 7.16.3**), using this parameter the user can select the read register of the SLAVE device from the list defined in the **Modbus** menu,
  - for the **Modbus SLAVE** mode (see **Chapter 7.16.2**), next to the **Device input** label there is the logical channel number and register number assigned to this logical channel displayed,
- **Value restoring** - this parameter occurs only in the **Modbus SLAVE** mode and means that the value last entered into this channel will be restored after restarting the device. The new value is saved not more often than every 30 minutes or while safe shutdown.
- **Latch** parameter block - allows the user to set the **latch function** which will hold the last value of a channel (discussed in **Chapter 7.9.1. Logical Channels - General settings**),
- **Processing** parameter block - is used for scaling and filtering data (discussed in **Chapter 7.9.1. Logical Channels - General settings**),
- **Displaying** parameter block - these parameters allow the user to select the format and range of data displayed on the screen (discussed in **Chapter 7.9.1. Logical Channels - General settings**),
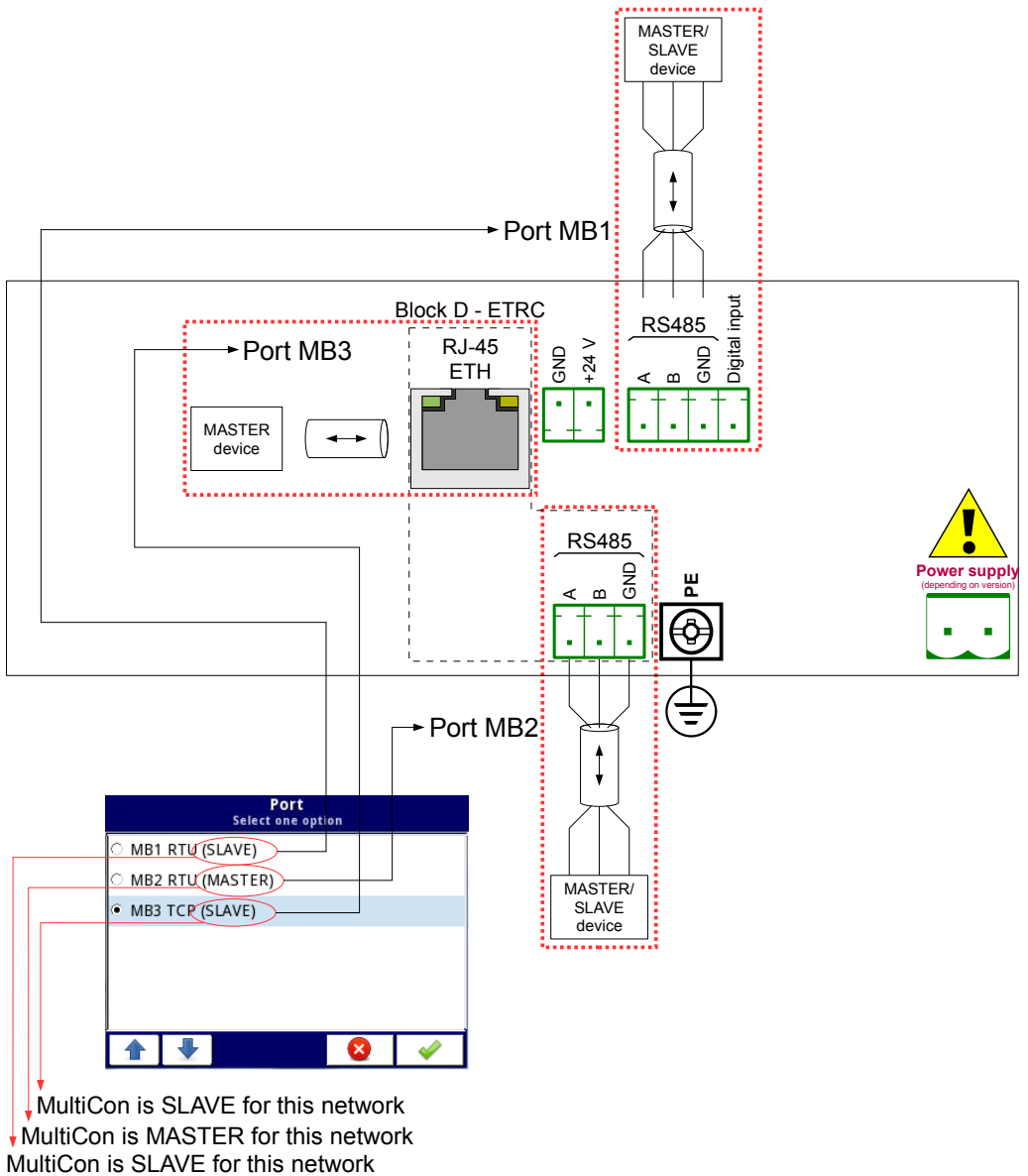
Fig. 7.63. Sample list of available Modbus ports
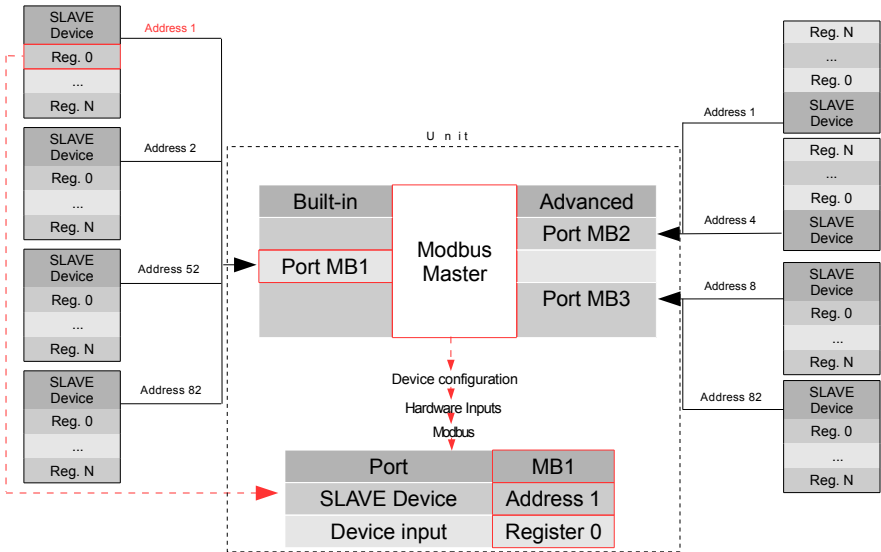
Fig. 7.64. An example of configuration block diagram
of the Modbus protocol implemented in the device

ⓘ   Logical channels in Modbus mode reading registers from Slave devices that are not connected, will return an error and instead of the value will display the state **-ERR-**.

For more information about the Modbus protocol implemented in the **MultiCon CMC-N16** see **Chapter 7.16. MODBUS**.

7.9.5. Logical Channels - Set point value mode

disabled
Hardware input
Hardware output monitor
Modbus
Set point value
Math function
Controller
Profile/timer
Profile/timer (cycle counter)
Data from other channel

MENU → Device configuration → Logical channels → **Mode**
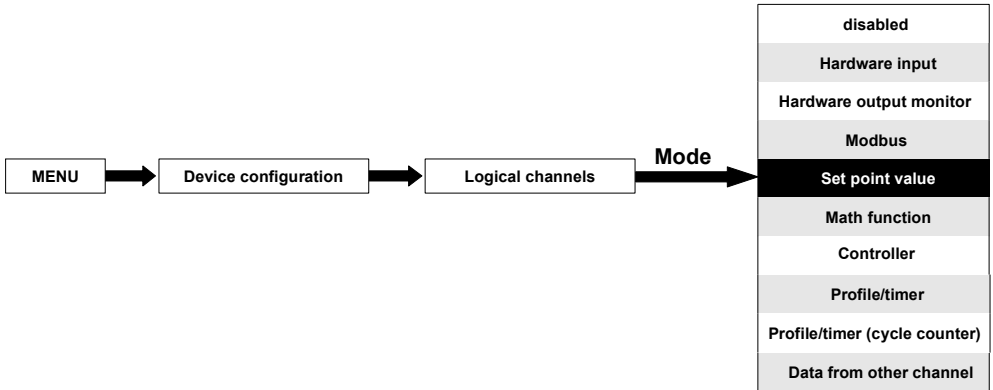
In the **Set point value** mode the user defines a constant value which can be displayed on the screen, used for calculations in other channels (e.g. in math functions, controllers etc.) or as a data source for built-in or external outputs which will control any object.

The parameters of the **Logical channels** in the **Set point value** mode should be:
– **Name** - it gives the name to the logical channel,
– **Mode=Set point value** - in this parameter the user selects a source type of data for the logical channel,
– **Unit** - this parameter allows the user to define any unit,
– **Set point value** - this parameter is visible only if **Edit button=disabled**, after pressing the button next to the **Set point value** parameter appears in the window allowing to enter a value (see **7.8**), this value will be a source of data for this Logical channel.
– **Edit button** - this parameter allows you to activate the button in the panel data (see **7.65**), this parameter has two options:
  • **disabled** - the button is disabled on the display, in this case the data source of the logical channel will be the value set in the **Set point value** parameter, (see **7.66**),
  • **enabled** - this button is active, in this case the data source of the logical channel will be the value set after pressing the button, but the initial value is the number written in the **Set point value** parameter, before setting the **Edit button** parameter to **enabled**,
– **Button mode** – the parameter is visible only in case when the **Edit button** parameter is set to **enabled**, and the **Format** parameter in the **Displaying** parameter block is set to **binary**, it has two options:
  • **bistable** – pressing and releasing the button will cause a permanent change of state,
  • **monostable** - pressing and releasing the button will switch from **OFF** state to **ON** state for 0.1 seconds, and after that time it will switch to **OFF** state again,

– **Latch** parameter block - allows the user to set the **latch function** which will hold the last value of a channel (discussed in **Chapter 7.9.1. Logical Channels - General settings**),

– **Displaying** parameter block - for these parameters the user selects the format and range of data to be displayed on the screen (discussed in **Chapter 7.9.1. Logical Channels - General settings**),
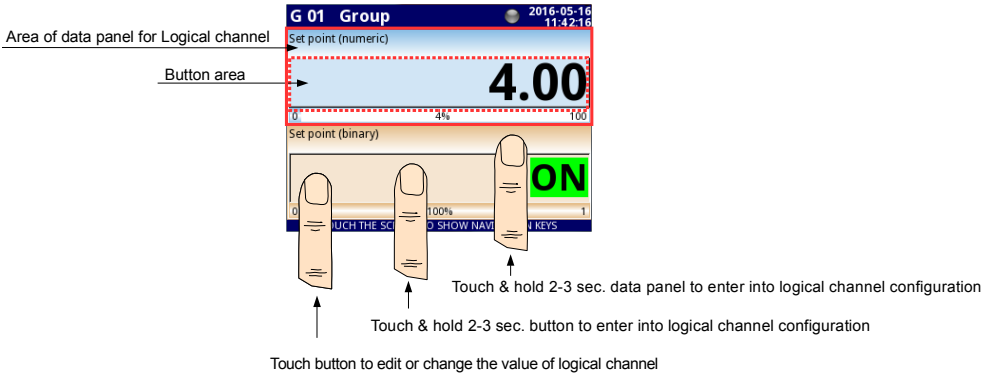

*Fig. 7.65. Data panels for Logical channel in **Set point value** mode*
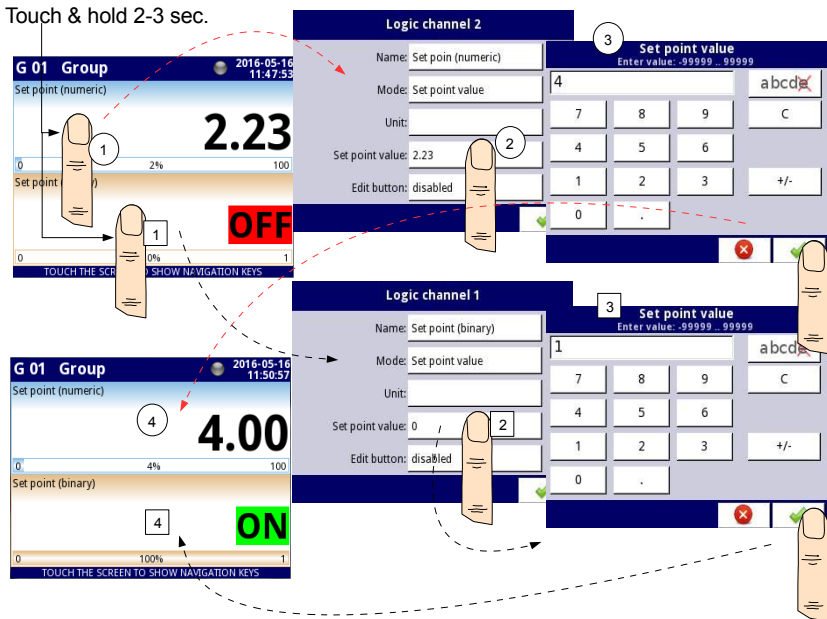
Fig. 7.66. Configuration of the value for Logical channel in Set point value for the disabled button

Manual operation of a button in the data panel

The action button in the data panel depends on the parameters from the **Latch** block (description of the **Latch** parameter block in **Chapter 7.9.1. Logical Channels - General**

settings) and on the **Format** parameter in the **Displaying** parameter block (description of the **Displaying** parameter block in **Chapter 7.9.1. Logical Channels - General settings**).

a) **for disabled latch function**, see **7.67**
For format:

– **numeric** and **time** – after pressing the button an edit window appears allowing to enter the value (see **7.8**), this value will be a source of data for that Logical channel,
– **binary** - pressing the button causes a switch between the states (**'0'** and **'1'**) display in the data panel in accordance with the text states set in the parameters: **Off-state text** (channel value = **'0'**) and **On-state text** (channel value = **'1'**) in the **Displaying** parameter block,

b) **for enabled latch function**
For format:

– **numeric** and **time** – after pressing the button an edit window appears allowing to enter the value (see **7.8**), which enters the new value that isn't a data source for this channel but the value stored at the time of activation of the latch function, in the edit window the value of the button still appears a new value which will be the data source for that logical channel by disabling the latch function,
– **binary** - pressing the button does not switch between the states (**'0'** and **'1'**) displayed in the data panel in accordance with the text states set in the parameters: **Off-state text** (channel value = **'0'**) and **On-state text** (channel value = **'1'**) in the **Displaying** parameter block, but with each new press of the button the state of the button is stored in the buffer and set the value of the logical channel with the currently stored state when the latch function is disabled again.
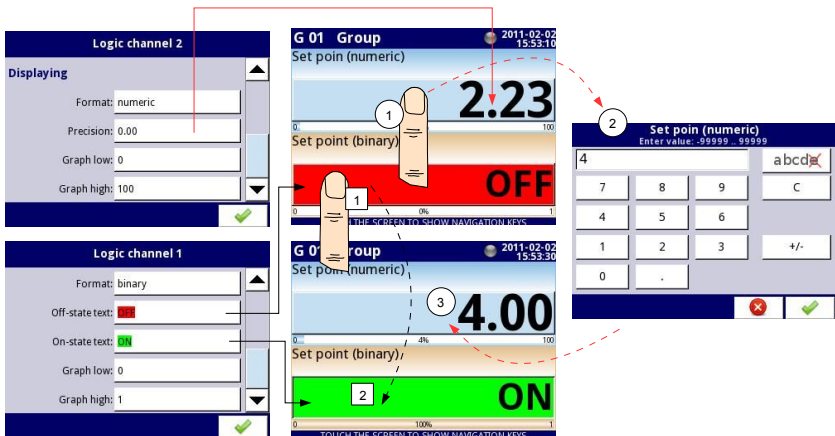
Fig. 7.67. Manual operation of the button for numeric and binary format

7.9.6. Logical Channels - Math function mode
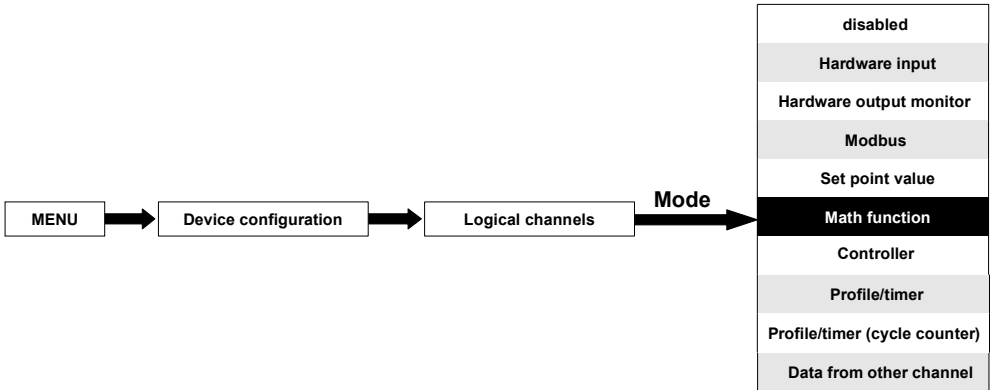
This mode allows the user to define the **Math** function to be displayed on the screen, process this data in any other logical channels or it can be the data source for output to the control and settings of any object.
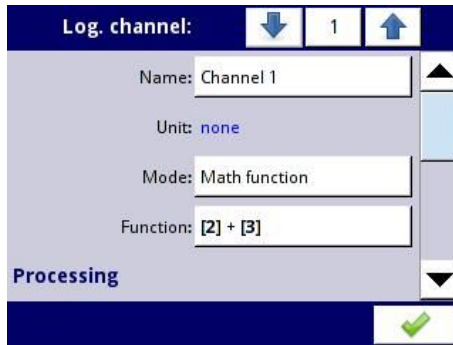
Fig. 7.68. Input channels menu - parameters specific for Math function mode

The **MultiCon CMC-N16** has mathematical functions which increase the functionality and the range of the applications of the device. **7.68** presents the parameters of the **Input channel** into the **Math function** mode. This mode allows the user to display channel value, process this data in any other logical the channel or it can be the data source for an output to control and set any object.

The parameters of the **Logical channel** in the **Math function** mode are:
   – **Name** - it gives the name to the logical channel,
   – **Unit** - to create a unit use the **Scaling** parameter (for more information see **Chapter 7.9.1. Logical Channels - General settings**),
   – **Mode=Math function** – in this parameter the user selects a source type of data for the logical channel,
   – **Function** - this parameter allows the user to select a math function from the list, for more information see below in this **Chapter**,

– **Latch** parameter block - allows the user to set the **latch function** which will hold the last value of a channel (discussed in **Chapter 7.9.1. Logical Channels - General settings**),
– **Processing** parameter block - is used for data scaling and filtering of data (discussed in **Chapter 7.9.1. Logical Channels - General settings**),
– **Displaying** parameter block - for these parameters the user selects the format and range of data displayed on the screen (discussed in **Chapter 7.9.1. Logical Channels - General settings**),

<u>**Function** parameter in the **Math function** mode</u>

Basic math functions implemented into the device are: addition, subtraction, multiplication and division. The unit allows the operation of logic functions, trigonometric, array operations, determining the arithmetic mean, finding maximum and minimum values, and many other functions that are discussed in the table below. After pressing the button next to the **Function** label go to the math function menu. This menu consists of the following parameters (Note! Not all parameters are available for each function):

– **Function** – available for all math functions, by pressing the button next to this label a list of available math functions appears, from which the user can select the appropriate function,
– **Type of source X** – available for certain math functions. Available types are:
  • **channel** – means that the **'X'** source will be a logical channel selected from the list in the **Source X** parameter,
  • **value** – means that the **'X'** source will be a constant value entered in the **Source X** parameter,
– **Source X** – available for certain math functions. Depends of the **Type of source X** parameter. This parameter allows the user to:
  • select a logical channel from the list (**Type of source Y=channel**),
  • or enter a set point value (**Type of source Y=value**),
– **X error handling** – available for some math functions, depending on this parameter, the user can set the following options:
  • **errors forwarded to result** – when the result of the selected channel (**Source X**) is a state: Error, Hi, Lo, or undefined, then the output received states: **Err**, **Hi** or **Lo**,
  • **skip erroneous channels** - means that these channels, which result in a status of **Error**, **Hi**, **Lo** are ignored in the calculation of a selected math function,
– **Init. value mode** – available for some math functions, depending on this parameter the user can set the following options:
  • **channel** – means that the initial value will be loaded from the selected logical channel,
  • **value** – means that the initial value will be a constant value entered in the **Initial value** parameter,
– **Init. value source** – available for certain math functions, selects a logical channel from the list which will be the initial value source for the function,
– **Initial value** – available for certain math functions, a constant value which will be the initial value for the function after its reset or device start,
– **Max. value mode** – available for some math functions, depending on this parameter the user can set the following options:
  • **channel** – means that the maximal value will be determined by the selected logical channel,

- • **value** – means that the maximal value will be a constant value entered in the **Maximal value** parameter,
- – **Max. value source** – available for certain math functions, selects a logical channel from the list which will be the maximal value source for the function,
- – **Maximal value** – available for certain math functions, a constant value which will be the maximal value for the function,
- – **Averaging time** – available for **Function=avarege of X**, determines the logical channel amount of samples which fits in this time and will be averaged; after this time the function enters a saturated state,
- – **Reset now** – the button available for certain math functions, asynchronously resets the function value,
- – **Reset mode** – available for some math functions, depending on this parameter the user can set the following options:
    - • **disabled** – means that the function will stop at this value after reaching the maximal value,
    - • **automatic** – means that the function will automatically reset itself and return to the initial value after reaching the maximal value,
    - • **from log. channel** – means that the function reset is triggered from the selected logical channel,
- – **Reset source** – available for some math functions, selects a logical channel which will be a reset source for the function,
- – **Reset triggering** – available for some math functions, allows to select a triggering method from the logical channel, the user can set the following options:
    - • **high level** – function resets when resetting channel is in high state,
    - • **low level** – function resets when resetting channel is in low state,
    - • **rising edge** – function resets when rising edge occurs in resetting channel,
    - • **falling edge** – function resets when falling edge occurs in resetting channel,
- – **J input mode** – available for **Function=flip flop**, the user can set the following options:
    - • **always 0** – flip flop J input is always in logical state '0',
    - • **always 1** – flip flop J input is always in logical state '1',
    - • **from log. channel** – means that flip-flop J input source will be the logical state value of the selected logical channel,
    - • **from channel (negation)** – means that flip-flop J input source will be the negated logical state value of the selected logical channel,
- – **J input source** – available for **Function=flip flop**, selects a logical channel which will be the source for the flip-flop J input,
- – **K input mode** – available for **Function=flip flop**, the user can set the following options:
    - • **always 0** – flip flop K input is always in logical state '0',
    - • **always 1** – flip flop K input is always in logical state '1',
    - • **from log. channel** – means that the flip-flop K input source will be the the logical state value of the selected logical channel,
    - • **from channel (negation)** – means that the flip-flop K input source will be negated logical state value of the selected logical channel,
- – **K input source** – available for **Function=flip flop**, selects a logical channel which will be the source for the flip-flop K input,
- – **Clock source** – available for **Function=flip flop** but unavailable if the parameters **J input mode** and **K input mode** are both set as always 0, allows to select a logical channel which will be the flip-flop clock input source,
- – **Clock triggering** – available for **Function=flip flop**, the user can set the following

options:
- • **high level** – flip-flop triggers when the parameter channel selected in the **Clock source** is in the high logical state,
- • **low level** – flip-flop triggers when the parameter channel selected in the **Clock source** is in the low logical state,
- • **rising edge** – flip-flop triggers when the rising edge occurs in the parameter channel selected in the **Clock source**,
- • **falling edge** – flip-flop triggers when the falling edge occurs in the parameter channel selected in the **Clock source**,
– **R input mode** – available for **Function=flip flop**, the user can set the following options:
- • **disabled** – resetting flip-flop R input in inactive,
- • **from log. channel** – means that the flip-flop R input source will be the logical state of the selected logical channel,
- • **from channel (negation)** – means that the flip-flop R input source will be the negated logical state of the selected logical channel,
– **R input source** – available for **Function=flip flop**, selects a logical channel which will be the flip-flop R input source,
– **S input mode** – available for **Function=flip flop**, the user can set the following options:
- • **disabled** – setting flip-flop S input in inactive,
- • **from log. channel** – means that the flip-flop S input source will be the logical state of the selected logical channel,
- • **from channel (negation)** – means that the flip-flop S input source will be the negated logical state of the selected logical channel,
– **S input source** – available for **Function=flip flop**, selects a logical channel which will be the flip-flop S input source,
– **Type of source Y** - available for certain math functions. Available types are:
- • **channel** - meaning that the **'Y'** source will be a logical channel selected from the list in the **Source Y** parameter,
- • **value** - means that the **'Y'** source will be a constant value entered in the **Source Y** parameter,
– **Source Y** - available for certain math functions. Depends on the **Type of source Y** parameter. This parameter allows the user to:
- • select a logical channel from the list (**Type of source Y=channel**),
- • or enter the Set point value (**Type of source Y=value**),
– **Unit** - available for certain trigonometric functions. Allows an option to be selected:
- • **degree**
- • **radian**

Example of the logical channel configuration in the Math function mode:

Steps to configure the function summarizing the values of logical channels: 1, 3, 4, 5 are shown in **7.69**. If any logical channel in sin the **Error** state or the value exceeds the range of a logical channel (Hi, Lo) then the output is in the same state. The following steps are:

(1) - Select the **Math function** mode in the appropriate logical channel (e.g. 14). Press the button next to the **Function** label to enter the **Function** menu,
(2) - Enter the **Function** sub-menu by pressing the button next to the **Function** label,
(3) - Choose a function from the list, in this case: **Mean X[i]**,

(4) - Press the button next to the **Source X** label to enter to the channel list,
(5) - Choose a logical channel from the list, in this case **1**, **3**, **4**, **5**,
(6) - Set the **X error handling** parameter to **errors forwarded to result**,
(7) - After configuring we get a description of the function next to the **Function** label: **Mean [1, 3, 4, 5]**.
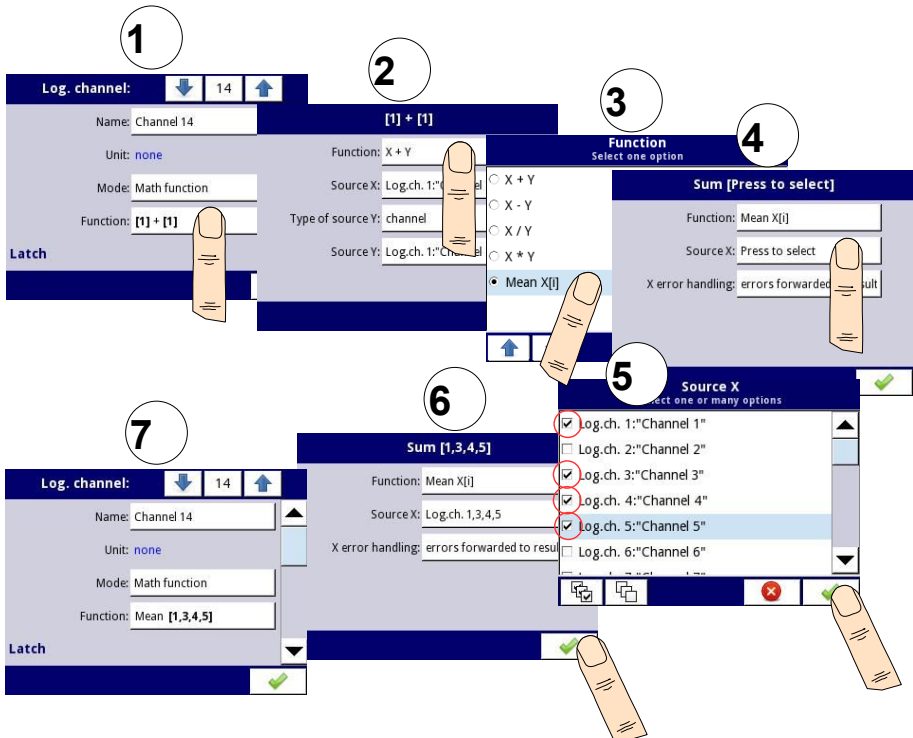
*Fig. 7.69. Sample configuration of logical channel in **Math function** mode*

Math function	Description	Example
X+Y	The sum of **Source X** and **Source Y** value	**[1] + [2]** – The sum of channel 1 and 2
X-Y	The subtraction of **Source X** and **Source Y** value	**[1] - [2]** – The subtraction of channel 1 and 2
X/Y	The ratio of **Source X** and **Source Y** value	**[1] / [2]** – The ratio of channel 1 to channel 2
X*Y	The product of **Source X** and **Source Y** value	**[1] * [2]** – The product of channel 1 and channel 2
(X>0) AND (Y>0)	Logical AND	**[1] AND [2]** - result = 1, when the value of channel 1 and 2 is greater than 0
(X>0) OR (Y>0)	Logical OR	**[1] OR [2]** – result = 1, when the value of channel 1 or/and 2 is greater than 0
(X>0) XOR (Y>0)	Logical XOR	**[1] XOR [2]** – result = 1, when the value of one channel is greater than 0 and the value of second channel is **≤ 0.** When both channels have values **≤ 0** or when both channels have values >0 then the result is 0.
SUM X[i]	The sum of selected channels	**SUM[1,2,3,4]** – the result is the sum of channels 1, 2, 3, 4
MEAN X[i]	The mean value of the selected channels.	**MEAN[1,2,3,4]** – the result is the arithmetic mean value of channels 1, 2, 3, 4
MULT X[i]	The product of the value of the selected channels.	**MULT[1,2,3,4]** - the result is the product of channels 1, 2, 3, 4
MIN X[i]	The smallest value of selected value of the selected channels	**MIN[1,2,3,4]** - the result is the lowest value of the selected channels 1, 2, 3, 4
MAX X[i]	The largest value of selected value of the selected channels	**MAX[1,2,3,4]** - the result is the highest value of the selected channels 1, 2, 3, 4

Math function	Description	Example
ANY X[i]>Y	The result = 1 if the value of any set of selected channels is greater than the value of the channel or constant value Y	**ANY[1,2,3,4]>[5]** – if the value of any set of channel 1, 2, 3, 4 is greater than the value of channel 5 then the result is 1, otherwise it will be 0
ALL X[i]>Y	The result = 1 if all values of selected channels are greater than the value of the channel or constant value Y	**ALL[1,2,3,4]>[5]** – the result is 1 if all values of channels 1, 2, 3, 4 are greater than the value of channel 5
ANY X[i]<Y	The result = 1 if the value of any set of selected channels is less than the value of the channel or constant value Y	**ANY[1,2,3,4]<[5]** - if the value of any set of channel 1, 2, 3, 4 is less than the value of channel 5 then the result is 1, otherwise it will be 0
ALL X[i]<Y	The result = 1 if all values of selected channels are smaller than the value of the channel or constant value Y	**ALL[1,2,3,4]>[5]** – the result is 1 if all values of channels 1, 2, 3, 4 are less than the value of channel 5
X[i] selected by Y	The result is a value of channel from list of channel X which selected by the value of channel Y	**[1,2,3,4]selected by[5]** - by the value of channel 5 is selected appropriate value from channels 1, 2, 3, 4 (for value ≤ **0** of channel 5 will be selected value of channel 1; for value (0,1> of channel 5 -> value of channel 2; for value (1,2> of channel 5 -> value of channel 3; for value >2 of channel 5 -> value of channel 4). (see example *7.9.11.5*, *Chapter 7.9.11*)
sin(X)	Sine of **Source X** value	**sin([17])** - sine value of channel 17
arcsin(X)	Arcsine of **Source X** value	**arcsin([8])** - arcsine value of channel 8
cos(X)	Cosine of **Source X** value	**cos([4])** - cosine value of channel 4
arccos(X)	Arccosine of **Source X** value	**arccos([1])** - arccosine value of channel 1
tan(X)	Tangent of **Source X** value	**tan([2])** - tangent value of channel 2
arctan(X)	Arctangent of **Source X** value	**arctan([4])** - arctangent value of channel 4

Math function	Description	Example
cot(X)	Cotangent of **Source X** value	**cot([10])** - cotangent value of channel 10
arccot(X)	Arccotangent of **Source X** value	**arccot([3])** - arccotangent value of channel 3
X^Y	Exponentiation - involving two numbers, the base of **Source X** value and the exponent of **Source Y** value	**[1]$^{[2]}$** - value of channel 1 to the power of channel 2 value
$\log_Y(X)$	Logarithm of **Source X** value with respect to base of **Source Y** value	**log$_{[2]}$([4])** - logarithm of channel 4 with respect to base of channel 2
min(X) reset by Y	Minimal value of selected channel **X** if value of channel **Y≤0**, or current value of **X** (reset) if value of **Y>0**	**min([1]) reset by [4]** - minimal value of channel 1 reset by channel 4
max(X) reset by Y	Maximal value of selected channel **X** if value of channel **Y≤0**, or current value of **X** (reset) if value of **Y>0**	**max([1]) reset by [4]** - maximal value of channel 1 reset by channel 4
derivative of X	Derivative of **Source X** parameter value	**derivative of [7]** – derivative of value in channel 7
integral of X	Integral of **Source X** parameter value	**integral of [3]** – integral of value in channel 3
count X pulses	Counter of rising edges in **Source X** parameter	**count [11] pulses** – channel 11 pulses counter
flip flop	**JK** type flip-flop with clock input and asynchronous **R** and **S** inputs	**flip-flop**
average of X	Current average of **Source X** parameter value with saturation and without forgetting	**average of [1]** - average of value in channel 1

Tab. 7.4 Available math functions

ⓘ In the device the values of the logical channel **≤ 0** are interpreted as a **'0'** logic, and the values of the logical channel **> 0** as a **'1'** logic.

ⓘ The user should pay attention to the fact that some mathematical functions have limitations. Those limitations are listed below:

Function **X/Y**:
- If **Y** equals **0**, then logical channel will be in **-Err-** state;
  putting it in a different way:
    Y == 0 ⇒ ERROR
Function **arcsin(X)**:
- If **absolute value** of **X** is greater than **1**, then logical channel will be in **-Err-**

state; putting it in a different way:

$$abs(X) > 1.0 \Rightarrow ERROR$$

Function **arccos(X)**:
- If **absolute value** of **X** is greater than **1**, then logical channel will be in **-Err**- state; putting it in a different way:

$$abs(X) > 1.0 \Rightarrow ERROR$$

Function **tan(X)**:
- If **absolute value** of **X** (in radians), minus **k** multiplied by **π** is lower than **one hundred millionth**, then logical channel will be in **-Err-** state; putting it in a different way:

$$abs(X[rad]) - k*\pi < 1.0e\text{-}8, \, k \in N \Rightarrow ERROR$$

Function **arctan(X)**:
- If **absolute value** of **X** (in radians) is lower than **one hundred millionth**, then the function result will be **0**; putting it in a different way:

$$abs(X[rad]) < 1e\text{-}8 \Rightarrow arctan(X) = 0$$

- If **absolute value** of **X** (in radians) is greater than **one hundred millions,** then the function result will be **π** divided by **2,** multiplied by **sign of X**; putting it in a different way:

$$abs(X[rad]) > 1e8 \Rightarrow arctan(X) = PI/2 * sign(X)$$

Function **cot(X)**:
- If **absolute value** of **X** (in radians), minus **k** multiplied by **π** is lower than **one hundred millionth**, then logical channel will be in **-Err-** state; putting it in a different way:

$$abs(X[rad]) - k*\pi < 1.0e\text{-}8, \, k \in N \Rightarrow ERROR$$

Function **arcctg(X)**:
- If **absolute values** of **X** (in radians) is lower than **one hundred millionth**, then the function result will be **π** divided by **2**, multiplied by **sign of X**; putting it in a different way:

$$X \, [rad] < 1e\text{-}8 \Rightarrow arcctg(X) = PI/2 * sign(X)$$

- If **absolute value** of **X** (in radians) is greater than **one hundred millions**, then the function result will be **0**; putting it in a different way:

$$abs(X[rad]) > 1e8 \Rightarrow arcctg(X) = 0$$

Function **X^Y**:
- If **X** equals **0**, and **Y** is lower than **0**, then logical channel will be in **-Err-** state; putting it in a different way:

$$X==0 \wedge Y < 0 \Rightarrow ERROR$$

- If **X** is lower than 0, and **Y** is not **integer**, then logical channel will be in **-Err-** state; putting it in a different way:

$$X < 0 \wedge Y \notin C \Rightarrow ERROR$$

Function **log_y(X)**:
- If **X** or **Y** is lower than **1.40129846*10^{-45}**, then logical channel will be in **-Err-** state; putting it in a different way:

$$X < 1.40129846e\text{-}45 \vee Y < 1.40129846e\text{-}45 \Rightarrow ERROR$$

- If **absolute value** of **Y** minus **1** is lower than **1.19209290*10^{-7}**, then logical channel will be in **-Err-** state; putting it in a different way:

$$abs(Y - 1) < 1.19209290e\text{-}7 \Rightarrow ERROR$$

7.9.7. Logical Channels - Controller mode
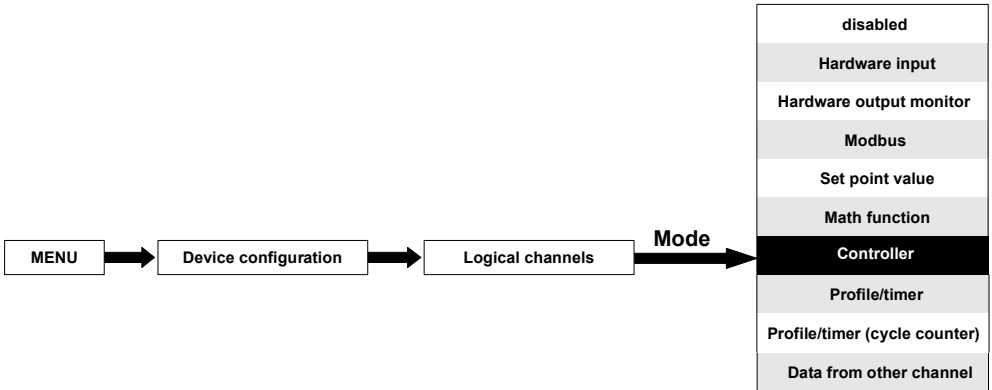
This mode allows the user to set up the PID control loop which may control the objects and Controller output signal for display on the screen, processed in any other logical channels or it can be the data source for output to the control and settings of any object. To create the **controlling process** the **Logical Channel** should be set in the **Controller** mode as in **7.70**.

Fig. 7.70. Input channel configuration in Controller mode

The parameters of the **Logical channel** in the **Controller** mode are:
– **Name** - it gives the name to the logical channel,
– **Mode=Controller** - in this parameter the user selects the source of data for the logical channel,
– **Unit** - this parameter allows the user to define the unit,
– **Controller number** - this parameter allows the user to select a controller from the list (1÷8). The user must configure the selected controller in the **Controllers** menu (overview and configuration parameters that define the controllers can be found in **Chapter 7.14. CONTROLLERS**).
– **Set point channel** - this parameter allows the user to select a logical channel with the set point value (see **7.71**), the **Set point channel** defines the input data for process control,
– **Feedback channel** - this parameter allows the user to select a channel with data

returned from the control system (see **7.71**), the **Feedback channel** defines the input data for process control,
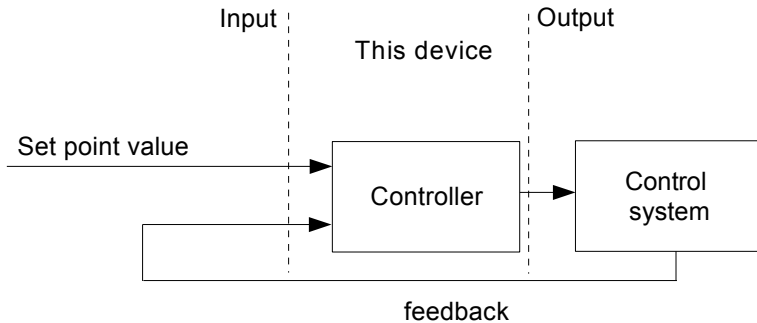
Fig. 7.71. The overall structure of control system implemented in the device

– **Latch** parameter block - allows the user to set the **latch function** which will hold the last value of a channel (discussed in **Chapter 7.9.1. Logical Channels - General settings**),

– **Processing** parameter block - is used for scaling and filtering data (discussed in **Chapter 7.9.1. Logical Channels - General settings**),

– **Displaying** parameter block - for these parameters the user selects the format and range of data displayed on the screen (discussed in **Chapter 7.9.1. Logical Channels - General settings**).

The user must configure the controller parameters in the **Controllers** menu before using this controller to control a real object (To read more about **Controllers profiles** see **Chapter 7.14. CONTROLLERS**).

7.9.8. Logical Channels - Profile/timer mode

disabled
Hardware input
Hardware output monitor
Modbus
Set point value
Math function
Controller
Profile/timer
Profile/timer (cycle counter)
Data from other channel

MENU → **Device configuration** → **Logical channels** → **Mode**
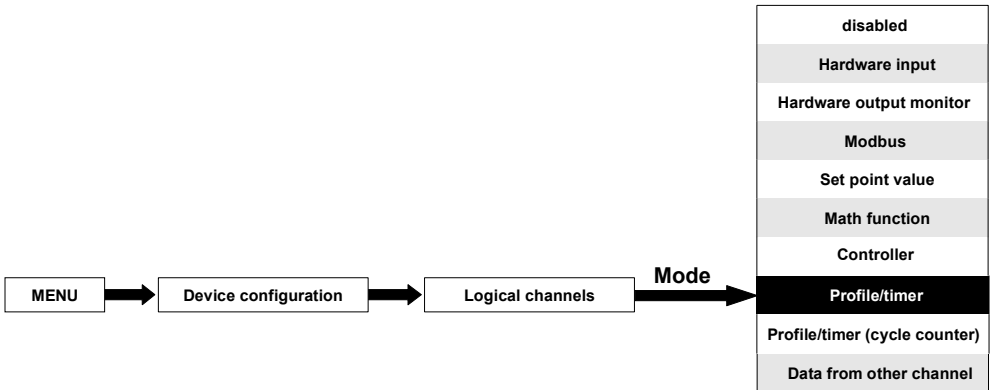
This mode allows the user to set up an appropriate Profile/timer defined in the **Profiles/timers** menu which may generate the signal defined by the user to display on the screen, processed in any other logical channels or it can be the data source for output to the control and settings of any object. To create the **Profile/timer** the **Logical Channel** should be set in the **Profile/timer** mode as in **7.72**.

Fig. 7.72. View of the input channel configuration in Profile/timer mode

The parameters of the **Logical channel** in the **Profile/timer** mode should be:
- **Name** - it gives the name to the logical channel,
- **Unit** - to create a unit use the **Scaling** parameter in the **Processing** parameter block (for more information see **Chapter 7.9.1. Logical Channels - General settings**),
- **Mode=Profile/timer** - in this parameter the user selects the source of data for the logical channel,
- **Source** - this parameter allows the user to select a Profile/timer from the list (**8 Profiles/timers** are implemented in the device), whose selected option will be the data source for this logical channel. The user must configure the selected Profile/timer by pressing the **Configure source** button or enter the **Profiles/timers** menu (overview and configuration parameters that define the Profiles/timers can be found in **Chapter 7.13. PROFILES/TIMERS**),

– **Configure source** - this button allows the user to configure the **Profile/timer** selected in the **Source** parameter. For more information see below in this **Chapter**,
– **Latch** parameter block - allows the user to set the **latch function** which will hold the last value of a channel (discussed in **Chapter 7.9.1. Logical Channels - General settings**),

– **Processing** parameter block - is used for scaling and filtering data (discussed in **Chapter 7.9.1. Logical Channels - General settings**),

– **Displaying** parameter block - for these parameters the user selects the format and range of the data displayed on the screen (discussed in **Chapter 7.9.1. Logical Channels - General settings**).

The user must configure the Profile/timer parameters in the **Profiles/timers** menu before using this Profile/timer to control a real object (To read more about **Profiles/timers** profiles see **Chapter 7.13. PROFILES/TIMERS**).

Configure source button in **Profile/timer** menu
There are two methods for configuring Profiles/timers:
– through the **Profiles/timers** menu, following the steps: **MAIN->Device configuration-> Profiles/timers** (see **Chapter 7.13. PROFILES/TIMERS**),
– directly in the **Logical channel** in the **Profile/timer** mode by pressing the **Configure source** button.

The view of the profile/timer configuration window in both cases is the same. Overview of setting the profile/timer setting is presented in **Chapter 7.13. PROFILES/TIMERS**.

7.9.9. Logical Channels - Profile/timer (cycle counter) mode

disabled
Hardware input
Hardware output monitor
Modbus
Set point value
Math function
Controller
Profile/timer
Profile/timer (cycle counter)
Data from other channel

MENU ➡ **Device configuration** ➡ **Logical channels** ➡ **Mode**

This mode is similar to the **Profile/timer** mode but instead of using the signal generated by the Profile/timer, the number of cycles elapsed by the profile is transferred to the logical channel.

> ℹ The cycle counter is generated depending on the **Looping** parameter value, i.e. when **Looping=disabled** the cycle counter equals 0.

The parameters of the **Logical channel** in the **Profile/timer (cycle counter)** mode should be:
  – **Name** - it gives the name to the logical channel,
  – **Unit** - for **Built-in modules** it will automatically use the default **Unit**, to change the unit use the **Scaling** parameter in the **Processing** parameter block, to create a unit use the **Scaling** parameter (for more information see **Chapter 7.9.1. Logical Channels - General settings**),
  – **Mode=Profile/timer** - in this parameter the user selects the source of data for the logical channel,
  – **Source** - this parameter allows the user to select a Profile/timer from the list (**8 Profiles/timers** are implemented in the device), the selected option will be the data source for this logical channel.
  – **Configure source** - this button allows the user to configure the **Profile/timer** selected in the **Source** parameter. For more information see below in this **Chapter**,
  – **Latch** parameter block - allows the user to set the **latch function** which will hold the last value of a channel (discussed in **Chapter 7.9.1. Logical Channels - General settings**),
  – **Processing** parameter block - is used for scaling and filtering data (discussed in **Chapter 7.9.1. Logical Channels - General settings**),
  – **Displaying** parameter block - for these parameters the user selects the format and range of the data displayed on the screen (discussed in **Chapter 7.9.1. Logical Channels - General settings**),

> ℹ Before selecting or after selecting the **Prodfile/timer** from the list inside the Logical channel, the user must configure the selected Profile/timer by pressing the **Configure source** button or entering the **Profiles/timers** menu,

Configure source button in **Profile/timer (cycle counter)** menu
There are two methods for configuring Profiles/timers:
- through the **Profiles/timers** menu, take the following steps: **MAIN->Device configuration->Profiles/timers** (see **Chapter 7.13. PROFILES/TIMERS**),
- directly in the **Logical channel** in the **Profile/timer** mode by pressing the **Configure source** button.

The configuration of the profile/timer in both cases is the same. The overview of the profile/timer settings is presented in **Chapter 7.13. PROFILES/TIMERS**.

7.9.10. Logical Channels - Data from other channel mode

	disabled
	Hardware input
	Hardware output monitor
	Modbus
	Set point value
	Math function
	Controller
	Profile/timer
	Profile/timer (cycle counter)

Mode

MENU	➡	Device configuration	➡	Logical channels	➡	Data from other channel
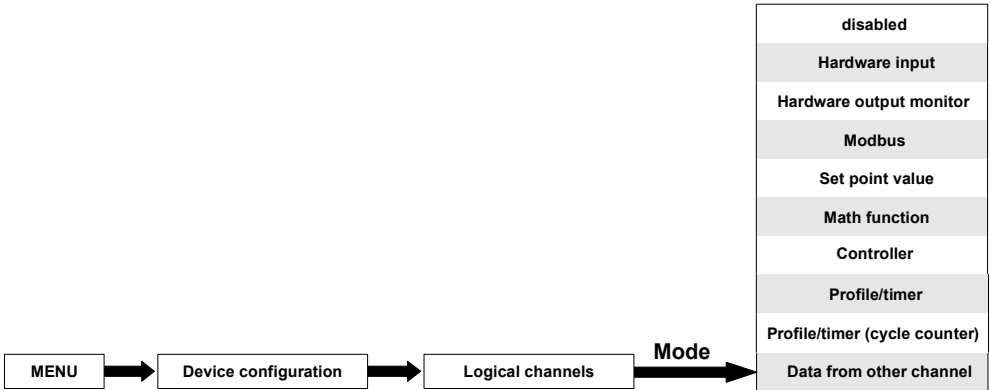
This mode has 2 applications. We can use this mode to:
- copy a logical channel,
- read the value from a source which has more than one quantity of the output value (e.g. **FUN4** hardware input configuration has 2 output quantities: flow and total flow) to be able to display all the possible quantities defined in the logical channel,

The parameters of the **Logical channel** in the **Data from other channel** mode should be:
- **Name** - it gives the name to the logical channel,
- **Unit** - the unit downloaded from the source logical channel; to change the unit use the **Scaling** parameter in the **Processing** parameter block (for more information see **Chapter 7.9.1. Logical Channels - General settings**),
- **Mode= Data from other channel** - in this parameter the user selects the source of data for the logical channel,
- **Source** - this parameter allows the user to select a logical channel from the list, the selected option will be the data source for this logical channel,
- **Quantity** – some hardware configurations can return more than one measured quantity (e.g. **FUN4** hardware input configuration has 2 output elements: flow and total flow). This parameter allows to choose one to display.
- **Latch** parameter block - allows the user to set the **latch function** which will hold the last value of a channel (discussed in **Chapter 7.9.1. Logical Channels - General settings**),
- **Processing** parameter block - is used for scaling and filtering data (discussed in

Chapter 7.9.1. Logical Channels - General settings),
– **Displaying** parameter block - for these parameters the user selects the format and range of data displayed on the screen (discussed in **Chapter 7.9.1. Logical Channels - General settings**).

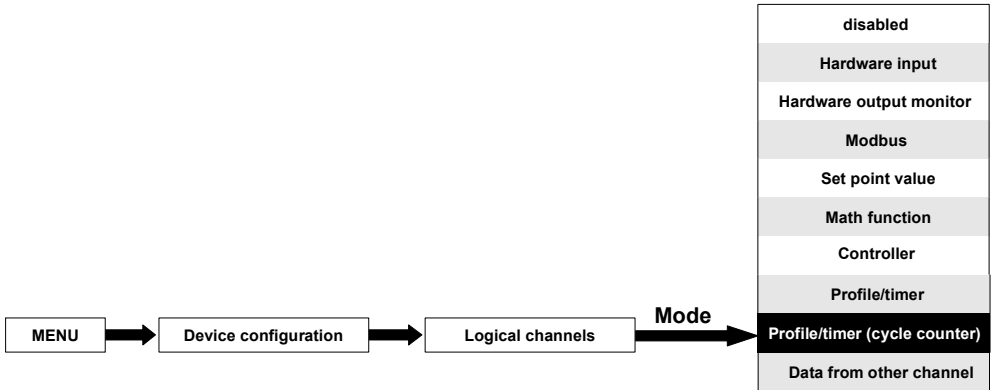## 7.9.11. Examples of Logical Channels configuration

7.9.11.1. Application of the Logical channel in the Hardware input mode for the FUN2 hardware configuration

See also: **Appendices 8.2 FUN2, FUN4 - UNIVERSAL HARDWARE INPUT CONFIGURATIONS WITH FLOWMETER MODE** *and 7.9.6 Logical Channels - Math function mode*.

Task:
This example shows how to measure voltage and current and calculate the power consumption of the heater. Voltage, current and power of the heater must be displayed in one window in the needle dials mode. Additionally in this example the user must use external converters to measure voltage and current exceed the range of the **FUN2** hardware input configuration.

Solution:
First configure the device as described below, then connect the heater to converters and then to the device (in accordance with **7.73**).
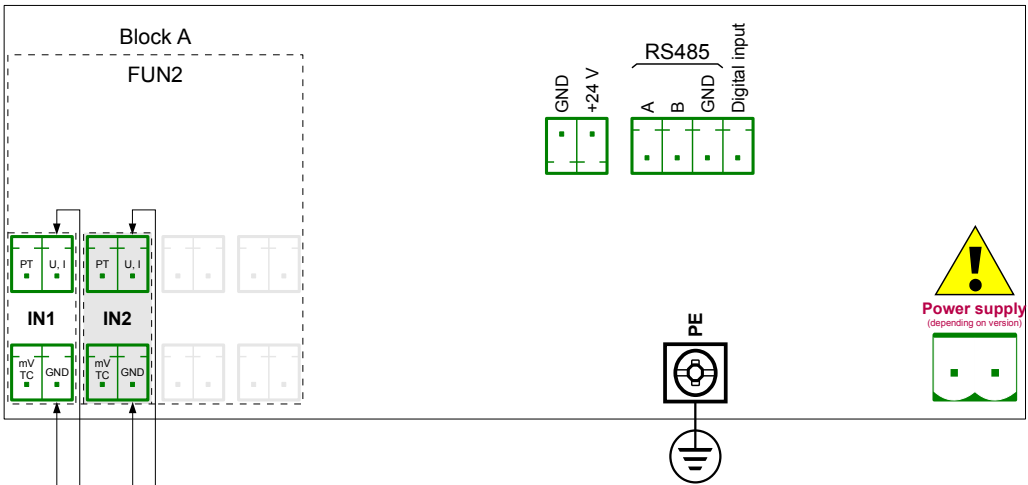
*Fig. 7.73. Schematic diagram for the **FUN2** hardware configuration*

In the next step you need to define three logical channels: for current, voltage and power. To do this:
a) For current,
- – touch the screen and press the **Menu** button,
- – press the **Device configuration** button,
- – enter the **Logical channels** menu,
- – using the arrows in the top navigation bar, select any **Logical channel** such as 20,
- – in the **Name** parameter write **Current**,
- – **Mode** parameter set as **Hardware input**,
- – **Source** parameter set as **Inp. A1: Universal** (see the connection in **7.73**),
- – press the **Source configuration** button to set the source configuration options:
  - • **Mode** parameter set as **0-20mA** current range, because this current range comes from the current converter,
  - • **Low Limit** parameter set as **0mA**,
  - • **High Limit** parameter set as **20mA**,
- – **Scaling** parameter set as **linear**,
- – press the **Configure scaling** button,
  - • in **Output unit** parameter write **A**,
  - • in **Input value** parameter in the **Point 1** section write **0mA**,
  - • in **Output value** parameter in the **Point 1** section write **0A**,
  - • in **Input value** parameter in the **Point 2** section write **20mA**,
  - • in **Output value** parameter in the **Point 2** section write **5A**,
- – for **Display** block parameters,

- • **Format** parameter set as **numeric**,
- • **Precision** parameter set as **0.0**,
- • in **Graph low** parameter write **0A**,
- • in **Graph high** parameter write **5A**,
- – all the other parameters should be on default settings.

The configuration steps are shown in **7.74**.

b) For voltage,
- – using the arrows in the top navigation bar, select any **logical channel** such as 21,
- – in the **Name** parameter write **Voltage"**,
- – **Mode** parameter set as **Hardware input**,
- – **Source** parameter set as **Inp. A2: Universal** (see the connection in **7.73**),
- – press the **Source configuration** button to set the source configuration options:
    - • **Mode** parameter set as **0-10V**, because this voltage range comes from the voltage converter,
    - • **Low Limit** parameter set as **0V**,
    - • **High Limit** parameter set as **10V**,
- – **Scale** parameter set as **linear**,
- – press the **Configure scaling** button,
    - • in **Output unit** parameter write **V**,
    - • in **Input value** parameter in the **Point 1** section write **0V**,
    - • in **Output value** parameter in the **Point 1** section write **0V**,
    - • in **Input value** parameter in the **Point 2** section write **10V**,
    - • in **Output value** parameter in the **Point 2** section write **400V**,
- – for **Displaying** block parameters,
    - • Format parameter set as **numeric**,
    - • **Precision** parameter set as **0.0**,
    - • in **Graph low** parameter write **0V**,
    - • in **Graph high** parameter write **400V**,
- – all the other parameters should be on default settings.
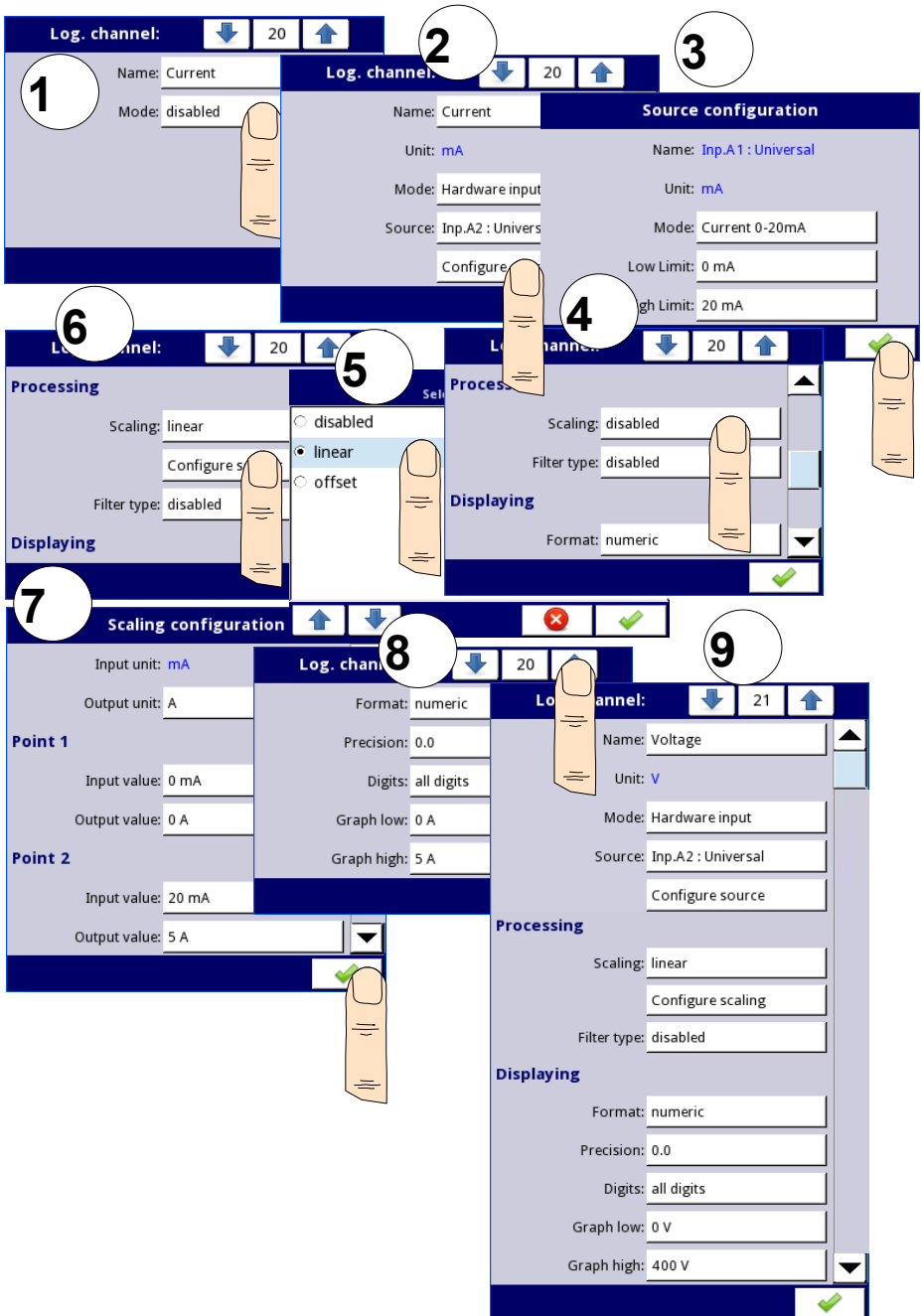
Fig. 7.74. Current and voltage source configuration example

c) For power,
   – using the arrows in the top navigation bar, select any **logical channel** such as 22,
   – in **Name** parameter write **Power**,
   – **Mode** parameter set as **Math function**,
   – press the button next to the **Function** label to select the appropriate function that allows the calculation of the heater power P = I * U,
     • **Function** parameter set as **X*Y**,
     • **Type of source X** parameter set as **channel**,
     • **Source X** parameter set as **logical channel 20**, which is defined as the reading of the heater current,
     • **Type of source Y** parameter set as **channel**,
     • **Source Y** parameter set as **logical channel 21**, which is defined as the reading of the heater voltage,
   – **Scale** parameter set as **linear**,
   – press the **Configure scaling** button,
     • in **Output unit** parameter write **kW**,
     • in **Input value** parameter in the **Point 1** section write **0**,
     • in **Output value** parameter in the **Point 1** section write **0kW**,
     • in **Input value** parameter in the **Point 2** section write **1000**,
     • in **Output value** parameter in the **Point 2** section write **1kW**,
   – for **Displaying** block parameters:
     • **Format** parameter set as **numeric**,
     • **Precision** parameter set as **0.00**,
     • in **Graph low** parameter write **0kW**,
     • in **Graph high** parameter write **2kW**, (we assume that the value of the heater power consumption exceeds 2kW=5A*400V),
   – all the other parameters should be on default settings.
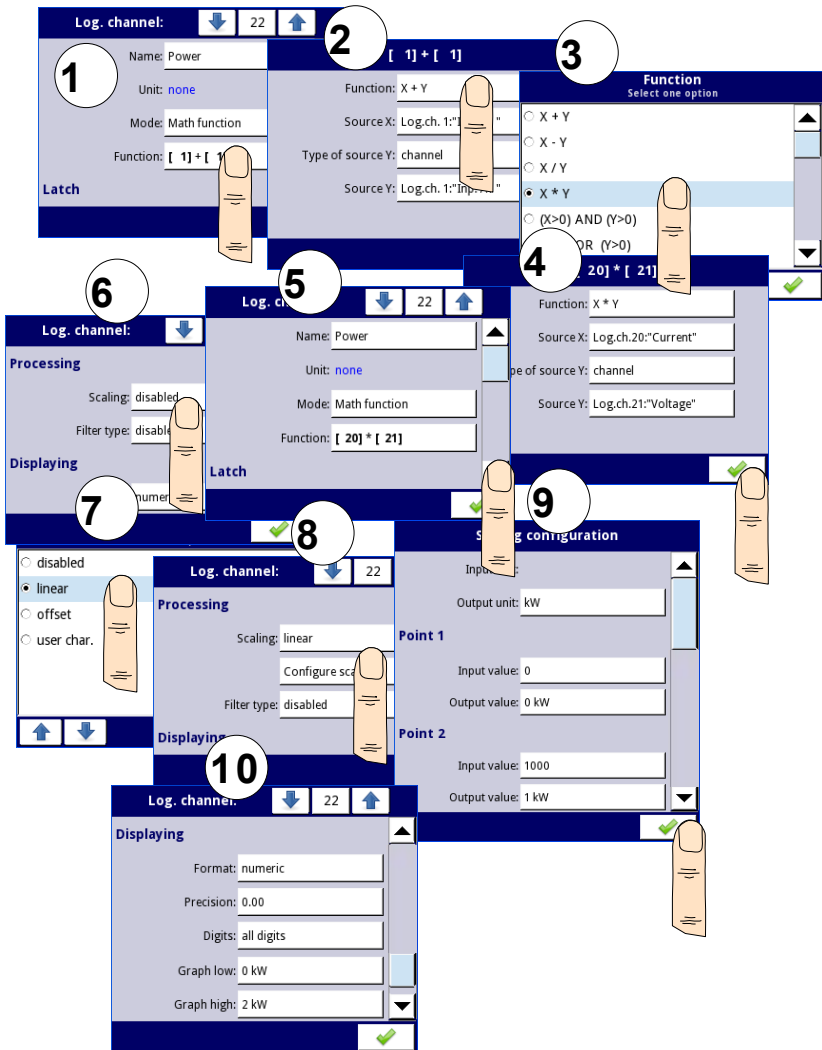
Fig. 7.75.Math function configuration example

In the next step you need to define **Group** to set up logical channels to view them in one window on the screen. To do this:
– enter the **Group** menu and turn it on if it's off,
– in **Name** parameter write **Heater**,
– for **Channels** block parameters:
  • **Slot 1** parameter set as **logical channel 22** called the **Power**,
  • **Slot 2** parameter set as **logical channel 21** called the **Voltage**,

- • **Slot 3** parameter set as **logical channel 20** called the **Current**,
- • other parameters in this block - **Slot 4, 5, 6**, set as **disabled**, because we want to show only three logical channels,
– all the other parameters should be on default settings.

Fig. 7.76.Example of Group configuration

The last step is to define the **Initial view** (see **7.77**) displayed on the LCD screen

when the device starts. The **Initial view** has to display a defined group of logical channels: power, voltage and current in-needle mode. To do this:
– enter the **General settings** menu,
– **Basic** block parameters left unchanged unless the indicated date and time were set incorrectly and should be improved,
– **Backlight** parameter set as a value corresponding to the user, for example, the value of **80%**,
– the **Screen saver** block parameters set to the disabled mode, as in this example we want to display all the time **Backlight** with the same clarity,
– for **Initial view** block parameters:
    • **Display mode** parameter set as **needles**,
    • **Displayed group** parameter set as **Group 1: Heater**,
– for **Automatic view change** block parameters:
    • **Change mode** parameter set to disabled, as in this example we want to all the time display only **Group 1** in the same needle mode,
– after setting all parameters exit the **General settings** menu.

Fig. 7.77 An example of Initial view configuration
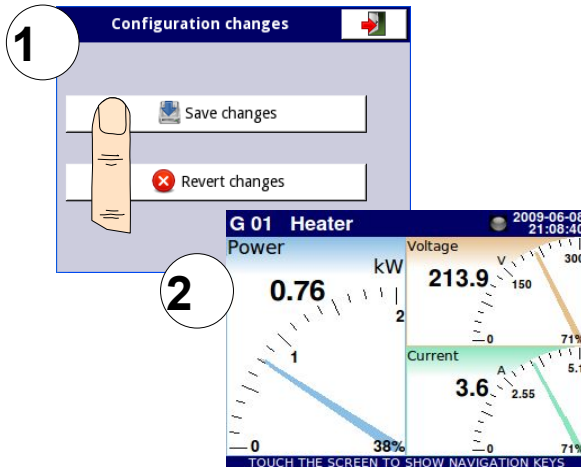
Fig. 7.78 Main page view after saving configuration

Finally, you must exit the menu by pressing the **Save changes** button. The result of the changes will be visible after a call to the first measurement and a manual switch for displaying Group 1 and needles mode (see **Chapter 6.3.2. Navigation bar**). An example of a window view for 3 logical channels is shown in **7.78**.

7.9.11.2. *Application of Logical channel in the* **Hardware input** *mode for FUN4 hardware configuration*

See also: **Appendices** *8.2 FUN2, FUN4 - UNIVERSAL HARDWARE INPUT CONFIGURATIONS WITH FLOWMETER MODE) and 7.9.6 Logical Channels - Math function mode*.

Task:
The task is to measure four temperatures in the production hall by using a K-type thermocouple. If all temperatures are greater than the preset value of **140 ° C**, in a separate channel (alarm condition) the inscription **-HI-** is to be displayed, in other cases **-LO-** is displayed. All the logical channels and the channel with the temperature alarm condition will be displayed in one window in the values mode.

Solution:
Before measuring temperature the thermocouple sensors should be connected to the device. An example of the connection is shown in **7.79**.
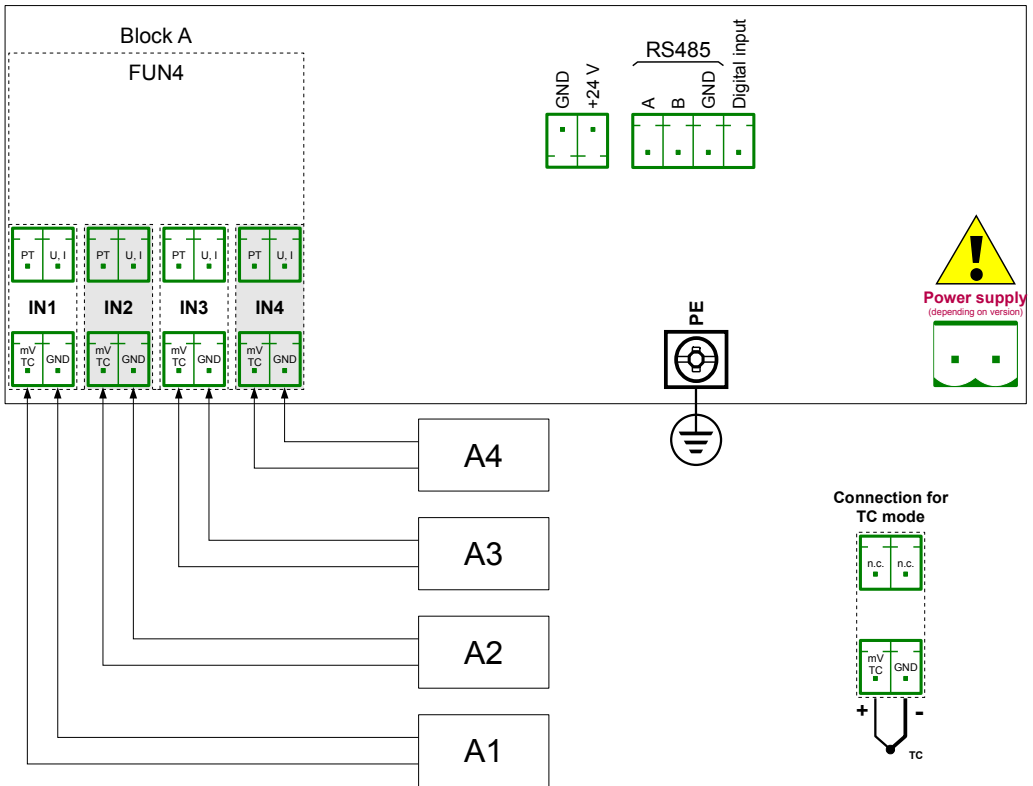
*Fig. 7.79. Schematic diagram for the **FUN4** hardware configuration*

In the first step you need to define five logical channels respectively for four temperatures in the hall of the symbols A1, A2, A3, A4, and the temperature alarm condition. To do this:

a) For the temperature reading of sensor 1 with the symbol A1:
- touch the screen and press the **Menu** button,
- press the **Device configuration** button,
- enter the **Logical channels** menu,
- using the arrows in the top navigation bar, select any **logical channel** (such as 1),
- in the **Name** parameter write **Temperature A1**,
- **Mode** parameter set as **Hardware input**,
- Source parameter set as **Inp. A1: Universal** (see the connection in **7.79**),
- press the **Source configuration** button to set the source configuration options,
    - **Mode** parameter set as **Thermocouple K**,
    - in **Low Limit** parameter write **-200°C**,
    - in **High Limit** parameter write **1370°C**,
- for **Display** block parameters:
    - **Format** parameter set as **numeric**,

- **Precision** parameter set as **0**,
- in **Graph low** parameter write **-200°C**,
- in **Graph high** parameter write **1370°C**,

b) For the temperature reading of sensor 2 with the symbol A2:
- Using the arrows in the top navigation bar, select any **Logical channel** (except for 1) for example 2,
- in the **Name** parameter write **Temperature A2**,
- **Mode** parameter set as **hardware input**,
- **Source** parameter set as **Source - Inp. A2: Universal** (see **7.79**),
- Other parameters are set as the sensor 1.

c) For the temperature reading of sensor 3 with the symbol A3:
- Using the arrows in the top navigation bar, select any **Logical channel** (except for 1 and 2) for example 3,
- in the **Name** parameter write **Temperature A3**,
- Mode parameter set as **hardware input**,
- Source parameter set as **Source - Inp. A3: Universal** (see **7.79**),
- Other parameters are set as the sensor 1.

d) For the temperature reading of sensor 4 with the symbol A4:
- Using the arrows in the top navigation bar, select any **Logical channel** (except for 1, 2 and 3) for example 4,
- in the **Name** parameter write **Temperature A4**,
- Mode parameter set as **hardware input**,
- Source parameter set as **Source - Inp. A4: Universal** (see **7.79**),
- Other parameters are set as the sensor 1.

e) For the alarm state,
- using the arrows in the top navigation bar, select any **Logical channel** (except for 1, 2, 3 and 4) e.g. 5,
- in the **Name** parameter write **Alarm**,
- Mode parameter set as **Math function**,
- enter the **Function** submenu to select the appropriate function that allows to calculate the alarm condition,
  - **Function** parameter set as **All X[i] > Y**,
  - in **Source X** parameter select logical channels 1, 2, 3, 4 which we defined as a measure of temperature,
  - **X error handling** parameter set as **errors forwarded to result**,
  - **Type of source Y** parameter set as **value**,
  - in **Source Y** parameter write **140**,
- for **Displaying** block parameters:
  - Format parameter set as **binary**,
  - in **Off-state text** write **LO** with blue font color,
  - in On-state text write **HI** with red font color,
  - in **Graph low** write **0**,
  - in **Graph high** write **1**,
- exit the **Logical channels** menu.

In the next step you need to define a **Group** of set up logical channels to view them in one window on the screen. To do this:

– enter the **Group** menu, enable the Group 1 if it is off,
– in the **Name** write **Temp.Measurment**,
– for **Channels** block parameters:
   • **Slot 1** parameter set as **logical channel 1** called the **Temperature A1**,
   • **Slot 2** parameter set as **logical channel 2** called the **Temperature A2**,
   • **Slot 3** parameter set as **logical channel 3** called the **Temperature A3**,
   • **Slot 4** parameter set as **logical channel 4** called the **Temperature A4**,
   • **Slot 5** parameter set as **logical channel 5** called the **Alarm state**,
   • **Slot 6** parameter set as **disabled**,
– all the other parameters should be on default settings.

The last step is to define the **Initial view** displayed on the LCD screen when the device starts. The **Initial view** has to display a defined group of logical channels: 1, 2, 3, 4, 5 in the value mode. To do this:
– enter the **General settings** menu,
– **Basic** block parameters left unchanged unless the indicated date and time were set incorrectly and should be improved,
– for **LCD screen** block parameters:
   • **Backlight** parameter set to the value corresponding to the user, for example **80%**,
– for **Screen saver** block parameters:
   • **Mode** parameter set as **disabled**, as in this example we want to display all the time **Backlight** with the same clarity,
– for **Initial view** block parameters:
   • **Display mode** parameter set as **value**,
   • **Displayed group** parameter set as **Group 1: Temp.Measurement**,
– for the **Automatic view change** block parameters:
   • **Change mode** parameter set to disabled, as in this example we want to all the time display only **Group 1** in the same needle mode,
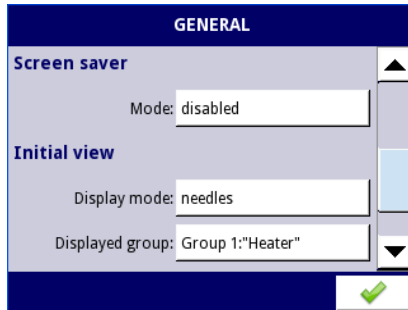– after setting all parameters exit the General settings menu.

Finally, you must exit the menu by pressing the **Save changes** button. The result of the changes will be visible after a call to the first measurement and manual switch for displaying Group 1. An example of a window view for 5 logical channels is shown in **7.80**.
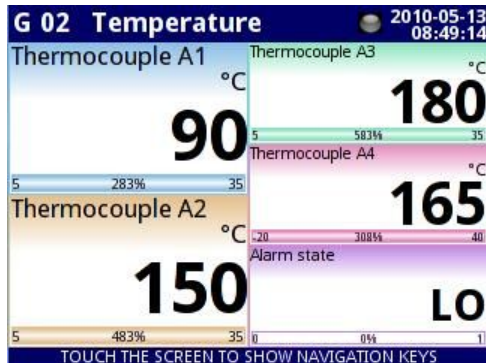
Fig. 7.80. The proposition of Example 2 result

7.9.11.3. Application of Logical channel in the Hardware input mode for FUN4 hardware configuration in RTD mode

*See also: **Appendices 8.2 FUN2, FUN4 - UNIVERSAL HARDWARE INPUT CONFIGURATIONS WITH FLOWMETER MODE**.*

Task:
       The **MultiCon CMC-N16** with the built-in **FUN4** hardware configuration can simultaneously measure temperature at four points. Suppose we want to monitor the temperature in the office, warehouse, basement and corridor. To monitor the temperature in the first two channels, we will use the PT1000 sensors and 3-wire connections, and for the rest PT100 and also 3-wire connections. We want to read the temperature from 4 sensors in one window in the vertical bars mode. In addition, we want to log the temperature every 1 minute (for more details about logging see **Chapter 7.15.2. Groups - Logging**).

Solution:
       Before measuring the temperature the user should configure the device and then connect the sensors as shown in **7.81**.

Fig. 7.81. Schematic diagram for the FUN4 hardware configuration

In the first step you need to configure **Logical channels**.
To read the temperature from sensor 1 (Office):
- touch the screen and press the **Menu** button,
- press the **Device configuration** button,
- enter the **Logical channels** menu,
- using the arrows in the top navigation bar, select any **logical channel** such as 30,
- in the **Name** parameter write **Office**,
- **Mode** parameter set as **Hardware input**,
- **Source** parameter set as **Inp. B1: Universal** (see the connection in **7.81**),
- enter the **Source configuration** submenu:
  - **Mode** parameter set as **Pt1000 3-wire**,
  - in **Low Limit** parameter write **-100°C**,
  - in **High Limit** parameter write **600°C**,
- for **Display** block parameters:
  - **Format** parameter set as **numeric**,
  - **Precision** parameter set as **0.0**, because Pt1000 sensor precision allows to display its value with one decimal place,
  - in **Graph low** parameter write **15°C**,
  - in **Graph high** parameter write **30°C**.

Other channels set in the same way:
    Logical channel 31: **Warehouse**,
    Logical channel 32: **Basement**, in the **Source configuration** set the **Mode** parameter as PT100,
    Logical channel 33: **Corridor** ,in the **Source configuration** set the **Mode** parameter as PT100.

In the next step you need to define a **Group** to set up logical channels to view them in one window on the screen. To do this:
– enter the **Group** menu to enable the e.g. **Group 2** if it is off,
– for **Display options** block parameters:
  • in **Name** parameter write **Workplace**,
  • **Bars** parameter set as **vertical**,
– for **Channels** block parameters:
  • **Slot 1** parameter set as **Logical channel 30** called the **Office**,
  • **Slot 2** parameter set as **Logical channel 31** called the **Warehouse**,
  • **Slot 3** parameter set as **empty**,
  • **Slot 4** parameter set as **Logical channel 32** called the **Basement**,
  • **Slot 5** parameter set as **Logical channel 33** called the **Corridor**,
  • **Slot 6** parameter set as **empty**,
– for **Logging options** block parameters (this block of parameters appears if you have a licence for logging):
  • **Mode** parameter set as **always**, we want to register at all times until we change the configuration to stop the registration,
  • in **Description** parameter write **Measure of Workplace temperature**, (it can be any text),
  • **Base period** parameter set as **1**,
  • **Base unit** parameter set as **minute**, sampling a group of channels at 1 minute,
  • Alternative mode parameter set as disabled.

The last step is to define the Initial view displayed on the LCD screen when the device starts. The initial view has to display a defined group of logical channels: 30, 31, 32, 33 in the bars mode. To do this:
– enter the **General settings** menu,
– for **Initial view** block parameters:
  • **Display mode** parameter set as **bars**,
  • **Displayed group** parameter set as **Group 2: Workplace**.

Finally, you must exit the menu by pressing the **Save changes** button. The result of the changes will be visible after a call to the first measurement and a manual switch for displaying Group 2. An example of the window view for 4 logical channels is shown in **7.82**.
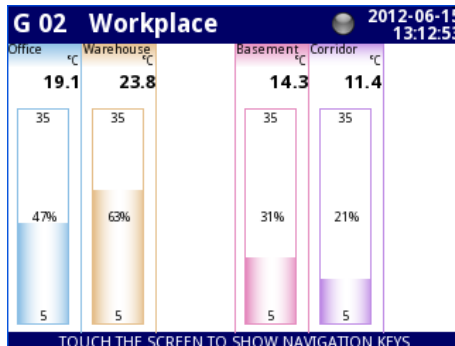
Fig. 7.82. Logical channels configuration - Example 3

7.9.11.4. Application of Logical channel in the Modbus mode

See also: **Chapter 7.16. MODBUS**.

Task:
The task is to read the temperature of two **SLAVE** devices from the **register 1**, which has a physical address assigned: **'1 '** and **'8'**.

Solution:
First, configure the device in the MASTER mode (like in **7.16.4.1. Input configuration of Modbus protocol in MASTER mode**), then configure it as follows and connect the SLAVE device as shown in **Fig. 7.83**.
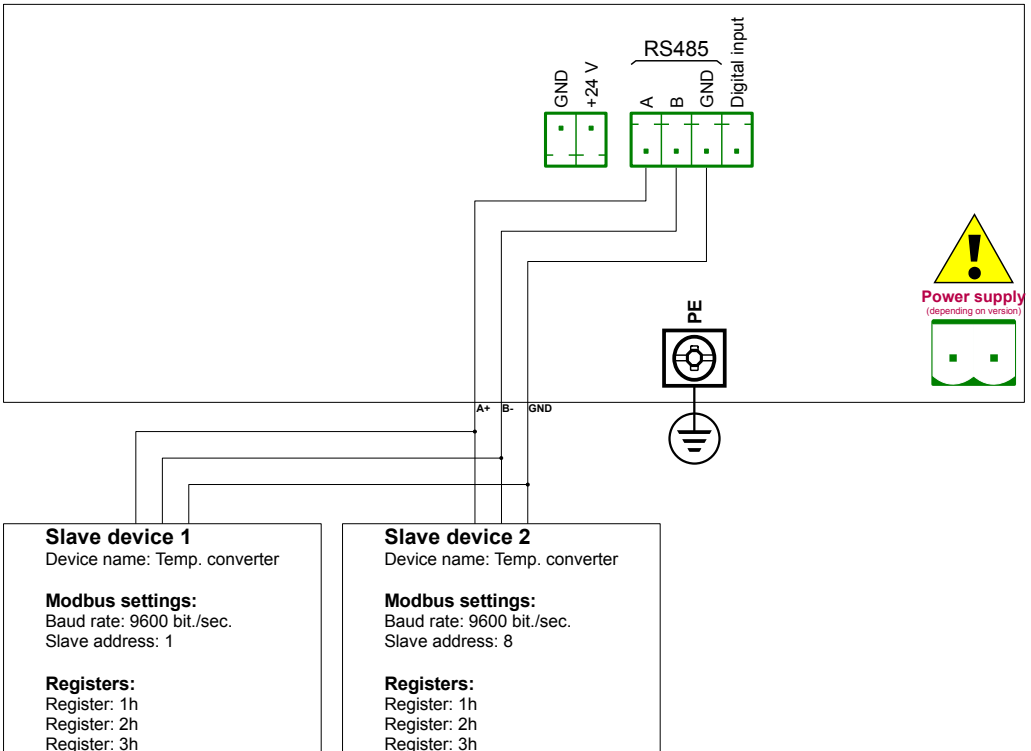
Fig. 7.83. Schematic diagram for SLAVE devices connected to MB1 port

In the first step you need to configure **Logical channels** which will read data from Slave devices. To do this:
 – touch the screen and press the **Menu** button,
 – press the **Device configuration** button,
 – enter the **Logical channels** menu,
 – using the arrows in the top navigation bar, select any **logical channel** such as 1,
 – in the **Name** parameter write **Temperature 1**,

– **Mode** parameter set as **Modbus**,
– **Port** parameter set as **MB1 (MASTER)**,
– **SLAVE devices** parameter set as **Addr.1:"Temp. converter"**,
– **Device input** parameter set as **Inp.1:HR 1h,b0-15** that reads **Register 1h** from **SLAVE 1** device,
– **Scale** parameter set as **linear**; in this example we don't need scaling but it is the only way to set the unit displayed on the screen,
– enter the **Configure scaling** submenu:
  • in **Output unit** write **°C**,
  • in **Input value** parameter in the **Point 1** section write **0**,
  • in **Output value** parameter in the **Point 1** section write **0°C**,
  • in **Input value** parameter in the **Point 2** section write **20**,
  • in **Output value** parameter in the **Point 2** section write **20°C**,
– for **Display** block parameters:
  • **Format** parameter set as **numeric**,
  • **Precision** parameter set as **0.0**,
  • in **Graph low** write **10°C**,
  • in **Graph high** write **50°C**,
– using the arrows in the top navigation bar, select any **Logical channel** such as 2,
– in the **Name** parameter write **Temperature 2**,
– **Mode** parameter set as **Modbus**,
– **Port** parameter set as **MB1 (MASTER)**,
– **SLAVE device** parameter set as **Addr.8:"Temp. converter"**,
– **Device input** parameter set as **Inp.2:HR 1h,b0-15** that reads **Register 1h** from **SLAVE 2** device,
– all the other parameters should be set like in the previously configured logical channel,

In the next step you need to define a **Group** to set up logical channels to view them in one window on the screen. To do this:
– enter the **Group** menu to enable the **Group 1** if it is off,
– in the **Name** parameter write **Temperature**,
– for **Channels** block parameters:
  • **Slot 1** parameter set as **logical channel 1** called the **Temperature 1**,
  • **Slot 2** parameter set as **logical channel 2** called the **Temperature 2**,
  • other parameters in this block – **Slot 3÷6**, set as **disabled**, because we want to show only two logical channels,
– all the other parameters should be on default settings,

The last step is to define the **Initial view** displayed on the LCD screen when the device starts. The initial View has to display a defined Group of logical channels: 1, 2 in the values mode. To do this:
– enter the **General settings** menu,
– for **Initial view** block parameters,
  • **Display mode** parameter set as **values**,
  • **Displayed group** parameter set as **Group 1: Temperature**,

After the whole configuration exit the menu by pressing **Save changes**, the result will be visible as soon as the first measurement is done.

7.9.11.5. Application in Math function mode

See also: **Chapter 7.9.6. Logical Channels - Math function mode**.

Task:
   The task is to create a logical channel with the function returns the value of a logical channel (**X [i]**) selected by the other logical channel (**Y**).
   Suppose we have defined four logical channels 1, 5, 8, 12 in the **Set point** mode. However, in the logical channel 2 there will be the value derived from the temperature sensor. In the logical channel 3 create the math function **X [i] selected by the Y**. Using the value of channel 2 we select the value from channels 1, 5, 8, 12.
The result of this function is placed in **Tab. 7.5**:

Y value	Value of logical channel 3
Y≤0	Value of logical channel 1
0<Y≤1	Value of logical channel 5
1<Y≤2	Value of logical channel 8
Y>2	Value of logical channel 12

*Tab. 7.5 Example of configuring function **X [i] selected by Y***

X[i]

Logical channel 1

Logical channel 5

Logical channel 8

Logical channel 12

Output

Logical channel 3

Y

Logical channel 2

Select Logical channel
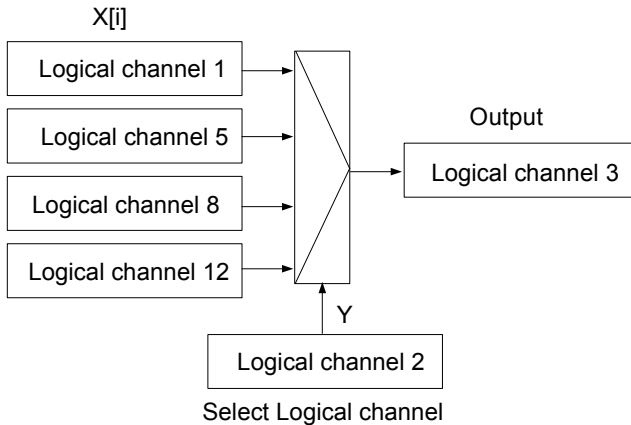
*Fig. 7.84. Block diagram of mathematical function **X [i] selected by Y***

Solution:
We define logical channels in the **Set point value** mode:
– touch the screen and press the **Menu** button,
– press the **Device configuration** button,
– enter the **Logical channels** menu,
– using the arrows in the top navigation bar, select any **logical channel** such as 1,
– in the **Name** parameter write **1**,
– **Mode** parameter set as **Set point value**,

- – in the **Set point value** parameter write **50**,
- – for **Displaying** block parameters:
  - • **Format** parameter set as **numeric**,
  - • **Precision** parameter set as **0**,
  - • in **Graph low** write **0**,
  - • in **Graph high** write **50**.

Other logical channels set the same way:
>   logical channel 5: Name 5, Set point value: 30,
>   logical channel 8: Name 8, Set point value: 80,
>   logical channel 12: Name 12, Set point value: 5.

For the Logical channel in the **Math function** mode:
- – using the arrows in the top navigation bar, select **logical channel 1**,
- – in the **Name** parameter write **Function**,
- – **Mode** parameter set as **Math function**,
- – enter the **Function** submenu to select the appropriate function that allows the calculation of the mean value,
  - • Function parameter set as **X[i] selected by Y**,
  - • in **Source X** parameter select **logical channels: 1**, **5**, **8, 12**, which we defined as constant values,
  - • in **Source Y** parameter select **logical channel 2**,
- – for **Displaying** block parameters:
  - • **Format** parameter set as **numeric**,
  - • **Precision** parameter set as **0**,
  - • in **Graph low** parameter write **0**,
  - • in **Graph high** parameter write **80**.

7.9.11.6. Application of Logical channel in the Controller mode

See also: **Chapter 7.9.7. Logical Channels - Controller mode**.

Task:
   The task is to configure the Controller to control (via RS-485) the Heater to get a constant object temperature. The Controller uses a logical channel in the Set point value Mode (temperature 85ºC) and the feedback signal comes from the Pt100 sensor.

Solution:
   Before measuring and regulating the temperature, the user should configure the device as follows and then connect the RTD sensor and the Current converter to the device. An example of the connection is shown in **7.85**.

Fig. 7.85. Schematic diagram for the hardware configuration FUN4 and MB1 Modbus port

The first step is to configure appropriate parameters for the **Controllers** menu according to **7.14.2.1. Application of the Controllers**. Next, configure logical channels as follows:

For demanding object temperature value:
- touch the screen and press the **Menu** button,
- press the **Device configuration** button,
- enter the **Logical channels** menu,
- using the arrows in the top navigation bar, select any **logical channel** such as 1,
- in the **Name** parameter write **Setpoint**,
- **Mode** parameter set as **Set point value**,
- in **Unit** parameter write **°C**,
- in **Set point value** write **85°C**,
- for **Displaying** block parameters:
  - **Format** parameter set as **numeric**,
  - **Precision** parameter set as **0**,
  - in **Graph low** parameter write **0°C**,
  - in **Graph high** parameter write **90°C**.

For reading temperature from the PT100 sensor,
- touch the screen and press the **Menu** button,
- press the **Device configuration** button,
- enter the **Logical channels** menu,
- using the arrows in the top navigation bar, select any **logical channel** such as 2,
- in the **Name** parameter write **Feedback**,
- Mode parameter set as Hardware input,
- **Source** parameters set as **Inp.A4: Universal** (see **7.85**),
- enter the **Configure source** submenu:
  - **Mode** parameter set as **Pt100**,
  - in **Low Limit** parameter write **-100°C**,
  - in **High Limit** write **600°C**,
- for **Displaying** block parameters:
  - **Format** parameter set as **numeric**,
  - **Precision** parameters set as **0.0** ,
  - in **Graph low** write **0°C**,

- • in **Graph high** write **100°C**.

For channel settings in the Controller mode:
- – touch the screen and press the **Menu** button,
- – press the **Device configuration** button,
- – enter the **Logical channels** menu,
- – using the arrows in the top navigation bar, select any **logical channel** such as 32,
- – in the **Name** parameter write **Controller**,
- – **Mode** parameter set as **Controller**,
- – in **Unit** parameters write **mA**,
- – **Controller** number parameter set as **Controller 1**,
- – **Set point channel** parameter set as logical channel 1 named **Setpoint**,
- – **Feedback channel** parameters set as logical channel 2 named **Feedback**,
- – for **Displaying** block parameters:
  - • **Format** parameter set as **numeric**,
  - • **Precision** parameters set as **0.00** , because with this precision we can send a signal from the controller to the Current converter (see **7.85**),
  - • in **Graph low** write **0**,
  - • in **Graph high** write **10**.

For presentation of data from the above example, it is recommended to use special display mode Controllers view (see **Chapter 7.15.6. Groups – Controllers view**).

For the external output configuration (sending via RS-485) see **Chapter 7.12.4. Examples of external output configurations**.

7.9.11.7. Application of Logical channel in the Profile/timer mode

See also: **Chapter 7.13. PROFILES/TIMERS**.

Task:
      Create the Profile/timer in the logical channel 1 in the edge (once) triggering mode. The trigger source is the logical channel 2 which is connected to the current input from hardware configuration FUN4. The profile consists of 4 sections: 1. ramp from 0 to 10 in 5 seconds, 2. constant value 8 in 2 seconds, 3. ramp from 8 to 4 in 3 seconds and 4. constant value 4 in 1 seconds. The idle value is 0 and looping is disabled.

Solution:
      First configure the device as follows. Then connect the current converter to the UI4 module as it is shown in **Fig. 7.73** (see also **Appendices 8.2. FUN2, FUN4 - UNIVERSAL HARDWARE INPUT CONFIGURATIONS WITH FLOWMETER MODE**).
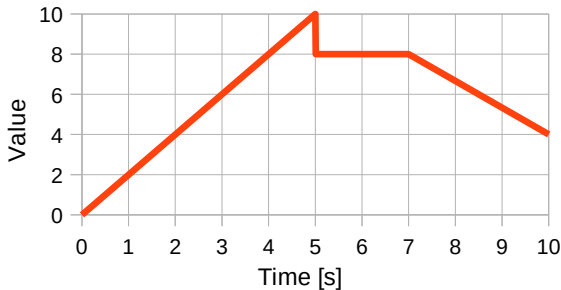
Fig. 7.86. Example of Profile waveform

For the logical channel in the *Profile/timer* mode:
– touch the screen and press the **Menu** button,
– press the **Device configuration** button,
– enter the **Logical channels** menu,
– using the arrows in the top navigation bar, select any **logical channel** such as 1,
– in the **Name** parameter write **Profile**,
– **Mode** parameter set as **Profile/timer**,
– **Source** parameter set as **P/T1:"Profile 1"**,
– press the **Configure source** button,
   • **Mode** parameter set as **edge (once)**,
   • **Triggering source** parameter set as **Log.ch.2**,
   • in the **Idle value** parameter write **0**,
   • press the **Section list** button,
      ◦ press the green button at the bottom left side of the screen,
      ◦ **Section num.** parameter should be set as **1**,
         – in **Duration** parameter write **0**,
         – **Unit** parameter set as **second**,
         – **Shape** parameter set as **slope**,
         – in the **Final value** parameter write **10**,
      ◦ press the green button at the bottom left side of the screen,
      ◦ **Section num.** parameter should be set as **2**,

- – in the **Duration** parameter write **2**,
- – **Unit** parameter set as **second**,
- – **Shape** parameter set as **const.value**,
- – in the **Final value** parameter write **8**,
- ◦ press the green button at the bottom left side of the screen,
- ◦ **Section num.** parameter should be set as **3**,
    - – in the **Duration** parameter write **3**,
    - – **Unit** parameter set as **second**,
    - – **Shape** parameter set as **slope**,
    - – in the **Final value** parameter write **4**,
- ◦ press the green button at the bottom left side of the screen,
- ◦ **Section num.** parameter should be set as **4**,
    - – in the **Duration** parameter write **1**,
    - – **Unit** parameter set as **second**,
    - – **Shape** parameter set as **const.value**,
    - – in the **Final value** parameter write **4**,
- • **Looping** parameter set as **disabled**,
- – for **Displaying** block parameters:
    - • **Format** parameter set as **numeric**,
    - • **Precision** parameter set as **0.0** ,
    - • in **Graph low** write **0**,
    - • in **Graph high** write **10.**

Configuration of logical channel 2 (Triggering source) in the **Hardware input** mode for the **current input** is shown in **7.9.11.1. Application of the Logical channel in the Hardware input mode for the FUN2 hardware configuration** (for this mode we configure logical channel 2 in this task).

Another method is shown in **7.13.4.1. Application of the Profiles/timers**.

7.9.11.8. Application of Logical channel in the Hardware input mode and Data from other channel for hardware configuration DU2 in flow mode

*See also: **Chapter 8.3 DU2, D4 – OPTOISOLATED HARDWARE DIGITAL CONFIGURATIONS.***

Task:
The task is to connect the PNP type sensor, read the measurement from the flowmeter hardware input of hardware configuration DU2 in flow mode and to display the current **Flow** and **Total flow**. The sensor gives 512 pulses per liter of liquid.

Solution:
Before configuring the device according to **7.87**, configure the device as it is shown below.
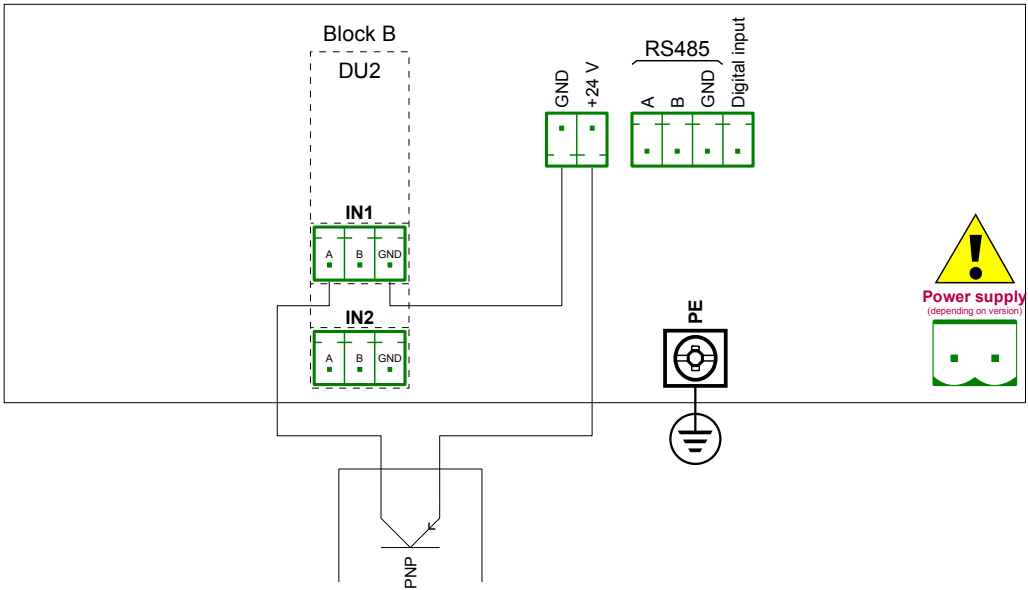
Fig. 7.87. hardware configuration DU2 and PNP type sensor connection scheme

For the logical channel in the *Hardware input* mode:
- touch the screen and press the **Menu** button,
- press the **Device configuration** button,
- enter the **Logical channels** menu,
- using the arrows in the top navigation bar, select any **logical channel**, in this example it will be channel 1,
  - in the **Name** parameter write **Measurement**,
  - **Mode** parameter set as **Hardware input**,
  - **Source** parameter set as **Inp.A3 : Tacho**,
  - press the **Configure source** button,
    - ◦ in the **Base unit** parameter write **dm³**,
    - ◦ **Mode** parameter set as **Flow/Tacho (1/sec.)**,
    - ◦ in **Low Limit** parameter write **-50000 1/sec.**,
    - ◦ in **High Limit** parameter write **50000 1/sec.**,
    - ◦ **Counting mode** parameter set as **with direction input**,
    - ◦ **Filter** parameter set as **disable**,
    - ◦ **Zero time** parameter set as **0.1 sec.**,
    - ◦ press the **Totalizer** button,
      - **Reset mode** parameter set as **from log. channel**,
      - **Reset source** parameter set as **Logical channel 3**,
      - **Hold mode** parameter set as **from log. channel**,
      - **Hold source** parameter set as **Logical channel 4**,
      - **Counting direction** parameter set as **from log. channel**,
      - **Direction source** parameter set as **Logical channel 5**,
  - **Scaling** parameter set as **linear**,
  - press the **Configure scaling** button,

      ◦   in **Output unit** parameter write **dm³/s**,
      ◦   in **Input value** in **Point 1** label write **0 1/sec.**,
      ◦   in **Output value** in **Point 1** label write **0 dm³/s**,
      ◦   in **Input value** in **Point 2** label write **512 1/sec.**,
      ◦   in **Output value** in **Point 2** label write **1 dm³/s**,
- **Format** parameter set as **numeric**,
- **Precision** parameter set as **0.00**,
- in **Graph low** parameter write 0 **dm³/s**,
- in **Graph high** parameter write **100 dm³/s**,

To display the Total flow:
- using the arrows in the top navigation bar, select **logical channel 2**,
  - in the **Name** parameter write **Total flow**,
  - **Mode** parameter set as **Data from other channel**,
  - **Source** parameter set as **Log.ch.1:"Measurement"**,
  - **Quantity** parameter set as **totalizer**,
  - **Precision** parameter set as **0.00**,
  - in **Graph low** parameter write **0 dm³**,
  - in **Graph high** parameter write **1000 dm³**,
- using the arrows in the top navigation bar, select **logical channel 3**,
  - in the **Name** parameter write **Reset**,
  - **Mode** parameter set as **Set point value**,
  - **Edit button** parameter set as **enabled**,
  - **Format** parameter set as **binary**,
- using the arrows in the top navigation bar, select **logical channel 4**,
  - in the **Name** parameter write **Strobe**,
  - **Mode** parameter set as **Set point value**,
  - **Edit button** parameter set as **enabled**,
  - **Format** parameter set as **binary**,
- using the arrows in the top navigation bar, select **logical channel 5**,
  - in the **Name** parameter write **Direction**,
  - **Mode** parameter set as **Set point value**,
  - **Edit button** parameter set as **enabled**,
  - **Format** parameter set as **binary**,
- all the other parameters should be set on default settings.

7.10. BUILT-IN INPUTS

General settings
Logical channels
Build-in inputs
Build-in outputs
External outputs
Profiles/timers
Controllers
Groups
Modbus
Network settings
Access options
E-mail notifications

MENU → Device configuration →
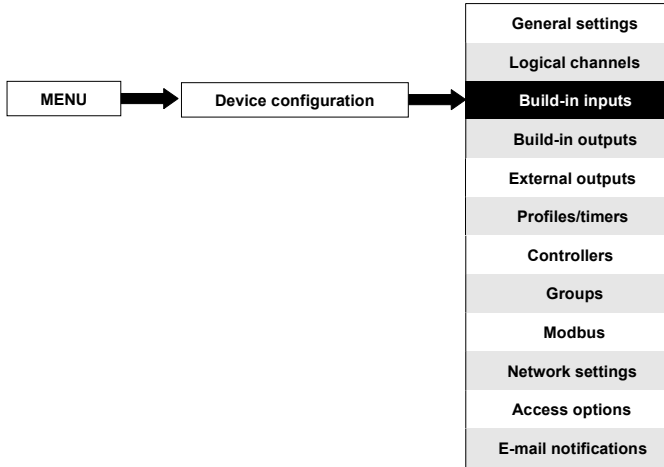
The **Built-in inputs** menu is directly related to the available inputs installed in the device. The basic version includes the following inputs:

- built-in **Digital input** is always marked as **Inp.X2:Digital 24V** - more about it see **Chapter 7.10.3. Built-in inputs - Binary input Inp.X2 : Digital 24V**,
- 3 built-in **Demo** inputs marked as **Inp.X3 : Demo**, **Inp.X4 : Demo**, **Inp.X5 : Demo** - more about them see **Chapter 7.10.4. Built-in inputs - Demo input numbered X3, X4, X5**,

Depending on the customer's needs, hardware input configurations (a description of available hardware input configurations is provided in **Chapter 8** and the producer's website) can be installed in respective blocks A, B or C (location of block see **4.10**).

7.10.1. Built-in inputs - General settings

4 Built-in inputs (**Digital input** and **3 Demo inputs**) are available in the device in the basic configuration and also the input hardware configurations installed in the device depending on the customer's needs. Configured inputs can be used to control any process or can be used by any logical channel. Switch to the **Hardware input** mode in order to visualize the result, or use for further processing.
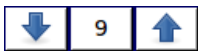
The arrows placed in the upper right corner of the screen allow you to switch between built-in inputs. The middle button allows you to directly select a specific built-in input from the list.

A common parameter for all **Built-in inputs** is **Name** – each input has its own unique name and it can not be changed – for description of this parameter see **7.88**.
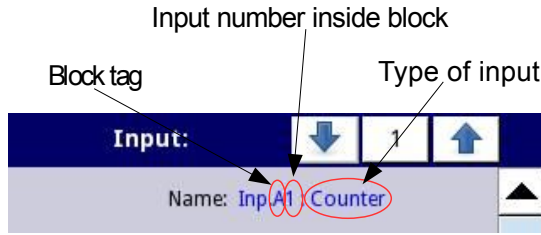
*Fig. 7.88. Description of **Name** parameter in **Built-in inputs** menu*

Other parameters of the **Built-in inputs** menu depend on the hardware configurations installed in the device.

> (i) The list of installed built-in hardware input configurations (block tag and type of hardware configuration) is in the **Device information** menu in the **Hardware configuration** block (see **7.40**),
>
> In the device there are two ways to configure the hardware input:
> - using the **Configure source** button in the **Logical channels** menu in the **Hardware input** mode,
> - directly using the **Built-in inputs** menu.

The **Built-in inputs** can be (in the same order as the list in the device - see **7.89**):
a) installed **input modules** in the appropriate blocks A or B (see **4.10**) - the list of currently available hardware configurations is on the website,

b) built-in **digital input** is always designated as **Inp.X2: Digital 24V** (see **Chapter 7.10.3. Built-in inputs - Binary input Inp.X2 : Digital 24V**),

c) built-in **Demo inputs** numbered **X3**, **X4**, **X5**, (see **Chapter 7.10.4. Built-in inputs - Demo input numbered X3, X4, X5**).
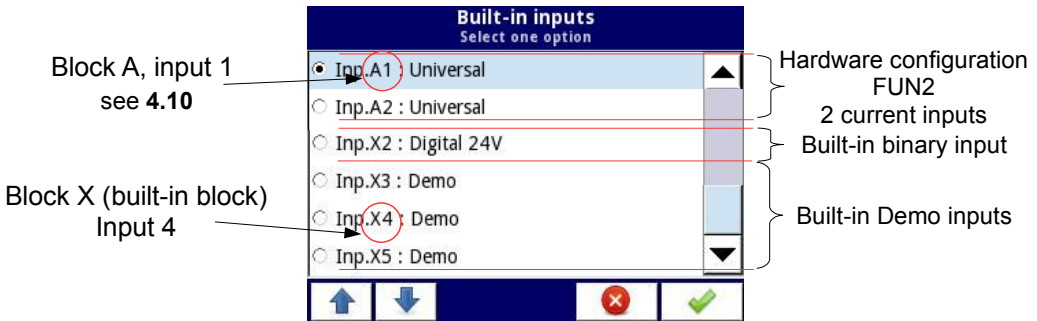
Fig. 7.89. The view of a sample list of available built-in inputs for a device

7.10.2. Built-in inputs - *Hardware input configurations*

A short description of the physical input configuration is shown in **4.10÷4.12** and is dependent on specific measurement hardware configurations. In the **Built-in inputs** menu for the hardware configuration the user can:
– change the ranges covered (depending on hardware configuration), see **8**,
– change the connection method - it depends on the hardware configuration (see **8**), e.g. in the RTD mode of the FUN4 hardware configuration the user can select 2 and 3-wire connections,
– change the type of the input signal reading - depending on the hardware configuration, e.g. the FUN2 hardware configuration can read temperature and voltage,
– change operation of the hardware configuration – e.g. in the flow meters hardware configuration the user can set the **Counting mode** to "with direction input" or to "quad 1" or "quad 4".,

Parameters common for the built-in inputs hardware configurations:
– **Name** - each built-in input already has a name given by the device and the user cannot change it, for the description of the **Name** parameter see **7.88**,
– **Unit** – it is related to the channel source data, for built-in hardware configurations the default unit appears automatically,
– **Mode** – sets the type of measurement; depending on the hardware configuration it sets the measurement range or the way of measurement,
– **Low Limit** – below this value in the logical channel, the low alarm state appears, displayed as **Lo**,
– **High Limit** – above this value in the logical channel, the high alarm state appears, displayed as **Hi**,

Other parameters of the **Built-in inputs** menu depend on the hardware configurations installed in the device.

7.10.3. Built-in inputs - *Binary input Inp.X2 : Digital 24V*

The device has a built-in digital input which can be used for example as a switch for a process. Specifications of the digital input are included in **Chapter 3. TECHNICAL DATA**. This digital input has the following levels:

	input voltage [V]		digital input
	min	**max**	
low level	0	1	0
prohibited level	>1	<8	x
high level	8	24	1

*Tab. 7.6 Voltage levels for **Binary input Inp.X2 : Digital 24V***

The **Binary input** has 2 parameters:
– **Name** – each built-in input already has a name given by the device and the user cannot change it, for the description of the **Name** parameter see **7.88**,

- **Filter time** – in which the user can change the filter time from 0 to 1000 seconds. Filtering is disabled (0 sec.) by default. This parameter determines how quickly the input can change as noticed by the device. Input changes which happen quicker than the value in the Filter time parameter will be ignored. Filtration can be used if:
  - contact bounce occurs when switching,
  - user deliberately wants to reduce the maximum frequency of the input.

7.10.4. Built-in inputs - Demo input numbered X3, X4, X5

The device has 3 built-in simulation **Demo** inputs which can be defined by the user. The configuration contains the following parameters:
- **Mode** – defines the shape of the generated signal (rectangle, triangle, sine),
- **Unit** - any user-defined unit,
- **Low limit** - the value below which there is a low state at the output displayed as status **'-Lo-'**,
- **High limit** - the value above which there is a high state at the output displayed as status **'-Hi-'**,
- **Minimum simulated value** – the lowest generated signal level,
- **Maximum simulated value** – the highest generated signal level,
- **Period** - duration (in seconds) of one cycle of the Demo signal selected in the Mode parameter,
- **High state time** - this parameter is displayed only for the **rectangle** mode; it defines how long (in seconds) the generated signal will be in the **Maximum simulated value** state,
- **Rise time** - this parameter is displayed only for the **triangle** mode; it defines how long (in seconds) the generated signal will be rising from **Minimum simulated value** to **Maximum simulated value**,

ⓘ The user should remember that the defined Demo input, which was edited from any logical channel, will change its settings in all the channel where it occurs.

Example configuration of a **Demo** for sinus mode shown in **7.90**.
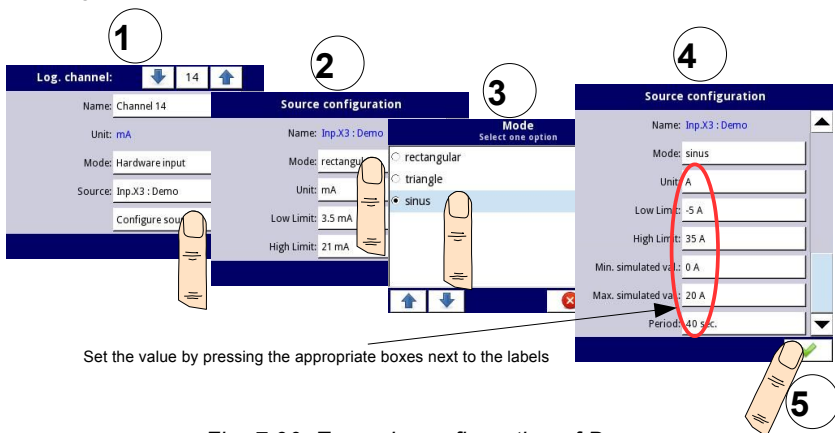
Fig. 7.90. Example configuration of **Demo**

7.10.5. Built-in inputs - Hardware configurations

7.10.5.1. FUN2 and FUN4 hardware configurations

FUN2 and **FUN4** hardware configurations are equipped with two or four universal inputs with flowmeter mode, see **Appendices 8.2. FUN2, FUN4 - UNIVERSAL HARDWARE INPUT CONFIGURATIONS WITH FLOWMETER MODE**. Using them it is possible to make independent measurements in each channel. They can measure: voltage, current, temperature (using resistance sensors and thermocouples) and flow.

The FUN hardware configurations parameters are:
– **Name** – each built-in input already has a name given by the device and the user cannot change it, for the description of the **Name** parameter see **Fig. 7.88**.
– **Unit** – parameter which displays the measurement unit, depending on **Mode** parameter settings,
– **Mode** – defines the measurement quantity, method and range,
– **Low limit** – defines the measurement level below the **Lo** state will be displayed in the logical channel,
– **High limit** – defines the measurement level above which the **Hi** state will be displayed in the logical channel,
– **Connectors compensation** – allows to compensate errors resulting from bad sensor readings. This menu occurs only if Mode parameter is set as Thermocouple,
  • **Compensation mode** – allows to select parameter which will be used in compensation process,
    ◦ **disabled** – compensations is not active,
    ◦ **user** – allows to compensate offset of the sensor,
      – **compensation** – allows to manually correct offset of the sensor, entered value will be added or subtracted from the sensor measured value,
    ◦ **automatic** – allows to compensate the difference between real temperature and these measured be the sensor,
      – **Real temp.** – parameter where the user enters temperature near thermocouple sensor measured by more reliable thermometer,
      – **Compensate** – button which activates the compensation, value in **Real temp.** parameter since that moment will be used for compensation,
      – **Compensation** – read only parameter; displays voltage value calculated by the device to compensate temperature measurement,

The user should remember to set **Connectors compensation** again parameter after changing thermocouple.

7.10.5.2. D4 hardware configuration

D4 hardware configuration has four binary inputs, which can be used individually or in group of four inputs (see **Appendices 8.3.DU2, D4 – OPTOISOLATED HARDWARE DIGITAL CONFIGURATIONS**). Measurement results are presented in decimal system.

The **D4** hardware configuration parameters are:
– **Name** – each built-in input already has a name given by the device and the user cannot change it, for the description of the **Name** parameter see **Fig. 7.88**,

– **Filter time** – defines minimal time that has to elapse from last input state change, if this change wants to be noticed; Each input has the possibility to sets its own Filter time, even if this input is part of another, different set of this parameter do not disturbs their work,

7.10.5.3. DU2 hardware configuration

DU2 hardware configuration is equipped in two universal counters inputs. Each input can work as a digital counter, digital flowmeter or two digital inputs (see **Appendices 8.3. DU2, D4 – OPTOISOLATED HARDWARE DIGITAL CONFIGURATIONS**). Each counter can be configured individually and is equipped with two counting inputs: **A** and **B**.

Counters parameters are:
– **Name** – each built-in input already has a name given by the device and user cannot change it, for description of **Name** parameter see **Fig. 7.88**,
– **Unit** – read only parameter, it is **Pulses**,
– **Low Limit** – value determining amount of pulses, bellow which in logical channel will be displayed **Lo** state,
– **High Limit** – value determining amount of pulses, bellow which in logical channel will be displayed **Hi** state,
– **Function mode** – specifies the pulse counting:
  • **A + B** – every rising edge on any input increments counter value,
  • **A - B** – every rising edge on input 1 increments counter value and every rising edge on input 2 decrements counter value,
  • **quad 1** – counter becomes an encoder element used to measure pulses on inputs using quad method. Direction of counter value changes depends on input 2 state (0 - down, 1 – up), and counting take place when on input 1 rising edge appears,
  • **quad 4** – counter becomes an encoder element used to measure pulses on inputs using quad method. Direction of counter value changes depends on input 1 (2) state (0 - down, 1 – up), and counting take place when on input 2 (1) rising edge appears. Thanks to this method user gains four times higher measurement resolution,
– **Reset now** – button which allows user to manually reset the counter,
– **Reset mode** – it allows to choose an additional resets for the counter, it has following parameters:
  • **disable** – no additional resets,
  • **from log. channel** – an additional reset is logical channel selected from list in **Reset source** parameter, it is activated by high state,
  • **external** – an additional reset is **Res.** input, activated by high state,
  • **both** – both of options shown above are active,
– **Reset source** – this parameter is visible when **Reset mode** is set as **from log. channel** or **both**, it contains a list of logical channels and the chosen one will reset the counter when its value is higher than **0**,
– **Prog. inp. mode** - specifies counter reaction on higher than **0** value appearance on **Prog. input**; it has following parameters:
  • **disable** – **Prog.** input is inactive, any changes on this input will be ignored,
  • **direction** – it allows to change mode or counting direction. When **Function mode** parameter is set as:
    ◦ **A + B** – active **Prog.** input changes mode to **- (A + B)**,
    ◦ **A - B** – active **Prog.** input changes mode to **B - A**,

- ◦ **quad 1** – active **Prog.** input changes counting direction,
- ◦ **quad 4** – active **Prog.** input changes counting direction,
- • **inhibit** – it allows to temporary stop pulses counting,
- – **Filter** – sets filter on counter inputs which allows to getting rid of negative effects of signal bounces; it has following parameters:
  - • **disable** – filter is inactive, recommended only in situation when to counter input is connected electronic input of other device,
  - • **10 ÷ 90** – filter is active; setting one of frequencies causes that input signals which duration corresponds to half of the selected frequency, will be considered as a unwanted signal bounces and ignored. Using this option is recommended when to counter input is connected contact output of the other device,

7.11. BUILT-IN OUTPUTS

			General settings
			Logical channels
			Build-in inputs
MENU	➡	Device configuration ➡	**Build-in outputs**
			External outputs
			Profiles/timers
			Controllers
			Groups
			Modbus
			Network settings
			Access options
			E-mail notifications
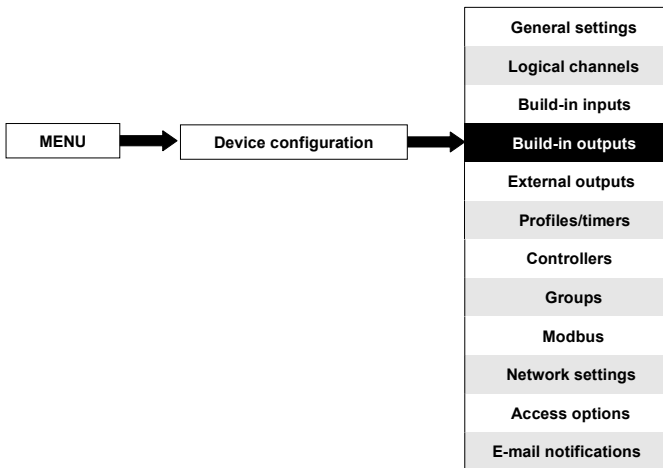
The **Build-in outputs** menu is directly related to the available outputs installed in the device. The basic version includes the following outputs:
- – built-in **Sound signal** output is always marked as **Out.X1: Sound signal** - more about the **Sound signal** output see **Chapter 7.11.2. Built-in outputs - Relays, Sound signal, Virtual relays**,
- – 16 built-in **Virtual relays** marked as **Out.V1: Virtual relay ÷ Out.V16: Virtual relay** - more about **Virtual relays** see **Chapter 7.11.2. Built-in outputs - Relays, Sound signal, Virtual relays**,

Depending on the customer's needs the hardware output configurations (description of available hardware output configurations is provided in **Chapter 8** and the producer's website) can be installed in block C (location of block see **4.8**).

7.11.1. Built-in outputs - General settings

17 Built-in outputs (**Sound signal** and **16 Virtual relays**) are available in the device in the basic configuration and hardware output configurations installed in the device depending on the customer's needs. A configured output can be used to control any process or can be used by any logical channel switched to the **Hardware output monitor** mode in order to visualize the result, or used for further processing of the output data.

The arrows placed in the upper right corner of the screen allow you to switch between built-in outputs. The middle button allows you to select a specific built-in output from the list directly.

Parameters common for **Built-in outputs**:
–   **Name** - each output already has a name given by the device and the user cannot change it, for the description of the **Name** parameter see **7.91**,
–   **Source** – this parameter contains a list of logical channels (up to 60), where the selected logical channel will be a data source for this built-in output (see **7.92**).
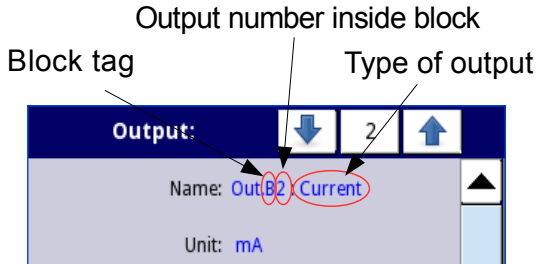
*Fig. 7.91. Description of **Name** parameter in **Built-in outputs** menu*
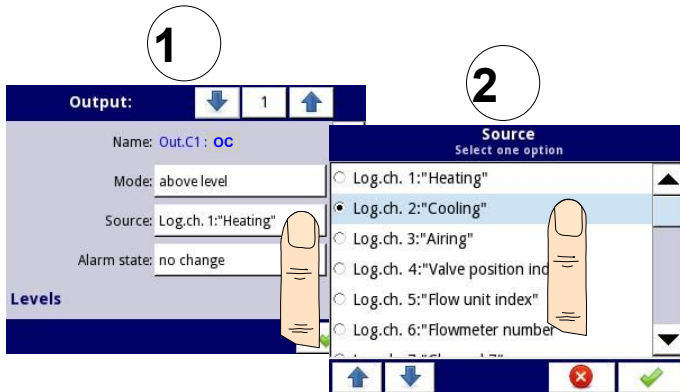
Fig. 7.92. Data source settings for built-in output (for OC hardware configuration)

The list of built-in hardware output configurations (block tag and type of hardware

configuration) is locate in the **Hardware configuration** parameters block in the **Device information** menu (see **7.40** in **Chapter 7.5. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION**).
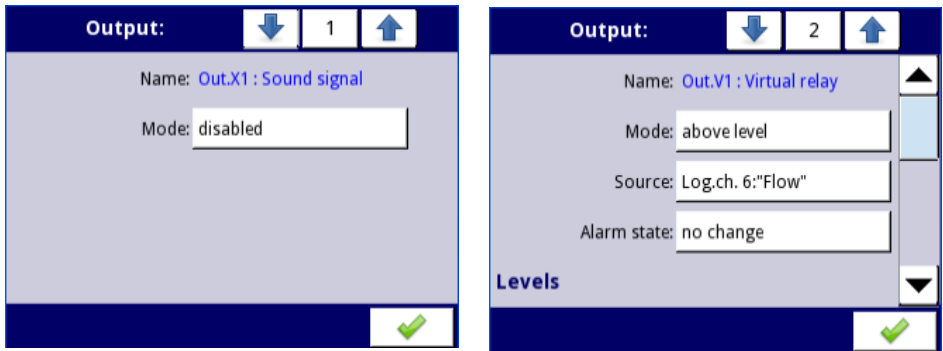
Fig. 7.93. Main settings of disabled (left) and enabled (right) output

State of physical outputs can be used as a source for **Logical Channels** (for details see **Chapter 7.9.3. Logical Channels - Hardware output monitor mode**).

7.11.2. Built-in outputs - Relays, Sound signal, Virtual relays

The parameters of **Built-in outputs** for: Relays, Sound signal, Virtual relays are:
- **Name** - each output already has a name given by the device and the user cannot change it, for the description of the **Name** parameter see **7.91**,
- **Mode** - this parameter allows the user to select the method of operation of the output, the **Mode** parameter has the following options (see **7.94** and **7.95**):
  • **disabled** - the built-in output is inactive,
  • **above level** - the result is a high state when the input data (see the the **Source** parameter) is above the level (see the **Level** parameter block), otherwise the output is low state,
  • **below level** - the result is a high state when the input data (see the **Source** parameter) is below the level (see the **Level** parameter block), otherwise the output is low state,
  • **inside range** - the result is a high state when the input data (see the **Source** parameter) will be within the range (see the **Level** parameter block), otherwise the output is low state,
  • **outside range** - the result is a high state when the input data (see the **Source** parameter) will be out of the range (see the **Level** parameter block), otherwise the output is low state,
- **Source** - this parameter contains a list of logical channels (up to 60), where the selected logical channel will be a data source for this built-in output (see **7.92**).
- **Alarm state** – this parameter allows to choose a type of output reaction in case of the alarm state appearance; The **Alarm state** is when the value of a Logical channel in which the data source for a built-in output returns **-Err-** state or the state of the exceeding range: low **-Lo-** state and high **-Hi-** state; There are the following options

in this parameter:
- **no change** - means that at the time of an alarm state there is no change in the output,
- **immediate OFF** - means that in times of an alarm state the device immediately switches the output to low state,
- **immediate ON** - means that in times of an alarm state the device immediately switches the output to high state,
- **timed OFF** - means that in times of an alarm state the device switches the output to low state after a time delay set in the **Timing** parameter block,
- **timed ON** - means that in times of an alarm state the device switches the output to low state after a time delay set in the **Timing** parameter block,
  - **Levels** block parameter - these parameters allow the user to set a range of changes of the output depending on the input signal, discussed below in this **Chapter**,
  - **Timing** parameter block - these parameters allow the user to set a delay time change of the output state and minimum duration of the output state, discussed below in this **Chapter**,

For **Built-in outputs**: Relays, Sound signal and Virtual relays, the **low state** is value **'0'** and the **high state** is value **'1'**.

Levels parameter block (see **7.94**, **7.95** and **7.96**)
These parameters depend on the **Mode** parameter. The parameters are:
- **Level mode** – has 2 options:
  - **value** – means that the level will be defined as a constant value,
  - **channel** – means that the level will be defined as a logical channel,
- **Level** – this parameter defines a constant signal level exceeding which will cause an output state change (for **Level mode**: **value**), or allows to choose a logical channel whose current value will be the threshold for output state changes (for **Level mode**: **channel**); occurs for **Mode**:
  - **above level** – if **Source** value is higher than **Level** value, a high state appears on the output,
  - **below level** – if **Source** value is lower than **Level** value, a low state appears on the output ,
- **Lower level** and **Upper level** - these parameters define the range at which output state changes occur (for **Level mode**: **value**), or they allow to choose a logical channel whose current value will be the range for output state changes (for **Level mode**: **channel**); occurs for **Mode**:
  - **inside range** – if the input data is within the defined range, a high state appears at the output,
  - **outside range** – if the input data is outside the defined range, a high state appears at the output,
- **Hysteresis** – by defining this parameter the user can move the level for changes in the output state (see **7.94** and **7.95**),
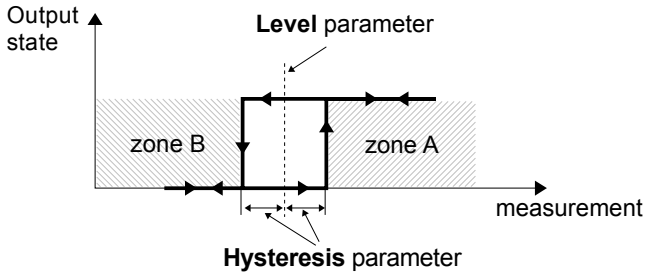
Fig. 7.94. One threshold control of the virtual relays
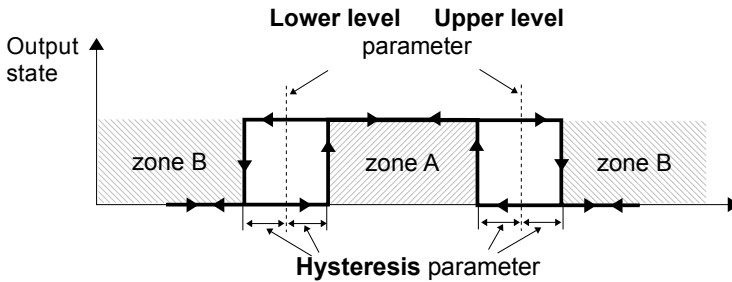
Fig. 7.95. Two threshold control of the virtual relays
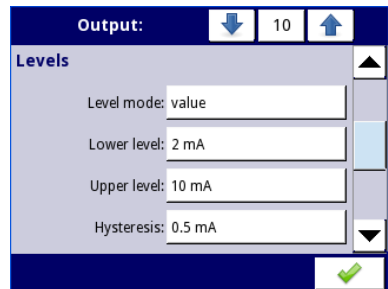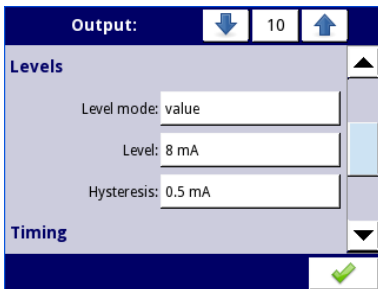
*Fig. 7.96. **Levels** settings for **Mode: above** and **below level** (left) and **inside range** and **outside range** (right)*

Timing parameter block (see **7.99**)
The parameters of this block include:
 – **ON delay** - this parameter allows to set the time that must pass from the time of exceeds **Level** value until the output switches from low to high state (see **7.97**),
 – **OFF delay** - this parameter allows the user to set the time that must pass from the time of exceeds **Level** value until the output switches from high to low state (see **7.97**),
 – **Minimum ON time** - the minimum duration of a high state (if the output switches to high state the low state will occur after this time) - see **7.98**,

– **Minimum OFF time** - the minimum duration of a low state (if the output switches to low state the high state will occur after this time) - see **7.98**,
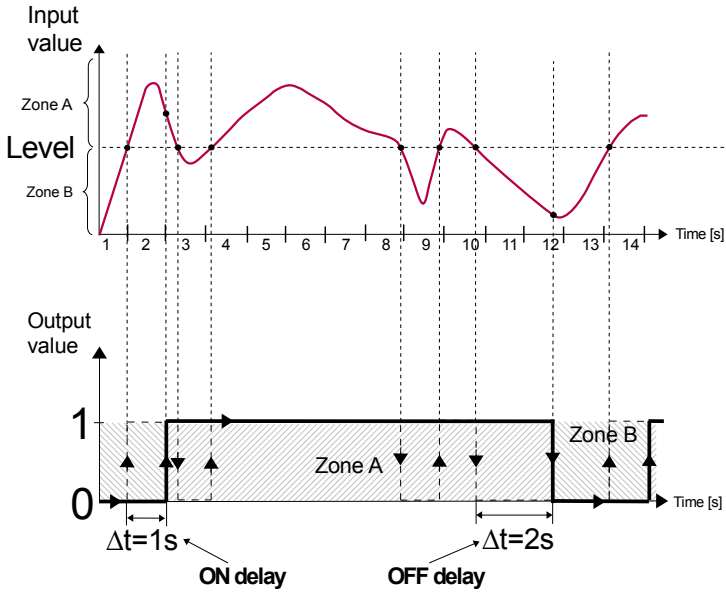
Fig. 7.97. Principle of virtual relay operation for sample timing settings:
***On delay**=1 sec., **OFF delay**=2 sec.*
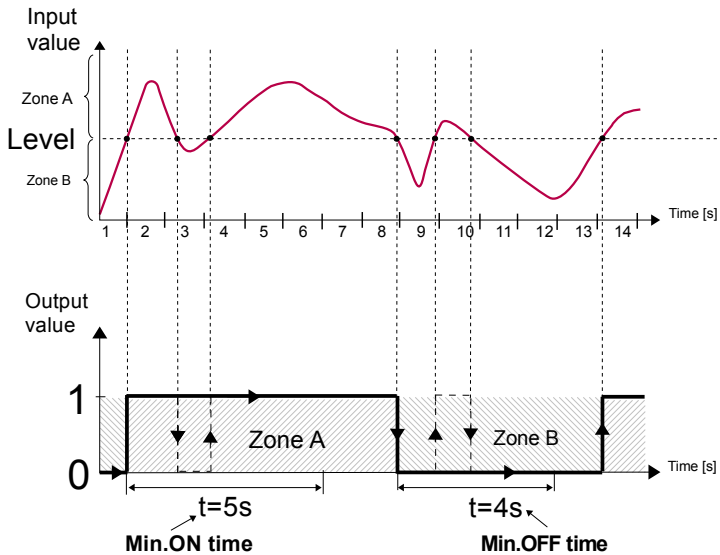
Fig. 7.98. Principle of relay output operation for sample timing settings:
***Min. ON time**=5 sec., **Min. OFF time**=4 sec.*
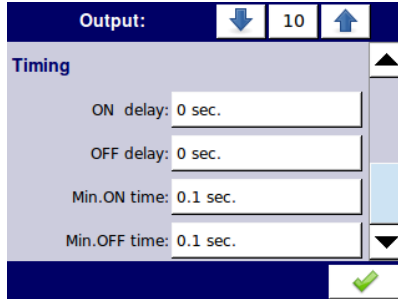
Fig. 7.99. Timing settings for different modes

7.11.3. Built-in outputs - OC-type relays

Technical specifications of OC-type relay can be found in **Appendices 8.4 R21, R41, COP2, COP3, OC2, OC4, R21COP2, COP2OC2, R21OC2 – HARDWARE OUTPUT CONFIGURATIONS**.

Built-in outputs parameters in **PWM** mode are:
– **Name** – each output already has a name given by the device and the user cannot change it - see **7.91**,
– **Source** – this parameter contains a **Logical channels** list, where the selected logical channel will be a data source for this built-in output (see **7.92**),

– **Levels** parameter block - these parameters allow the user to set the range of the input signal which will change the duty cycle of the output signal, discussed below in this **Chapter**,

– **Timing** parameter block - these parameters allow the user to set the timing parameters of the output signal, discussed below in this chapter.

Levels parameter block
The parameters are:

– **Lower level** and **Upper level** - by setting these parameters the range is defined within which the change duty cycle of pulse depending on the source signal; below this range the signal is zero (zero width, zero duty cycle) and above this range the signal is completely filled (fully on of the cycle period) (see **7.100**),

– **Alarm level** - when the data source for the built-in output returns the **Error** state or the range being exceeded: the low **-Lo-** state and high **-Hi-** state, the **Alarm level** parameter for the PULSE mode allows setting the duty cycle of the output signal in time of an alarm state according to the parameters of **Lower level** and **Upper level.**

The duty cycle **D** is defined as the ratio between the pulse duration **t** and the period **T** of a rectangular waveform:

duty cycle $$D = \frac{t}{T}$$ where:

t - is the duration that the function is active,
T - is the period of the function.
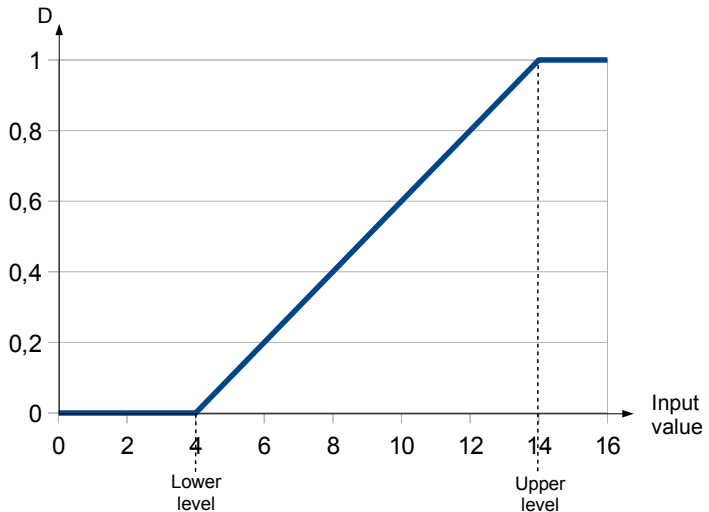
Fig. 7.100. Normalized duration (t_n) of the output signal in PULSE mode for parameters:
***Lower level**=4, **Upper level**=14 (input signal lower than 4 has t_n=0, higher than 14 has t_n=1)*

Timing parameter block
The parameters of this block include:
- **Period** - the duration of one cycle of output pulse (the minimum value is 0.1 seconds),
- **Minimum ON time** - minimum duration of a high state (after switching to a high state, switching back is possible after **Minimum On time**), see **7.101**,
- **Minimum OFF time** - minimum duration of a low state (after switching to a low state, switching back is possible after **Minimum OFF time**), see **7.101**.

> **i** **Minimum ON time** and **Minimum OFF time** parameters shouldn't be set to a value equal to or greater than the **Period** parameter.
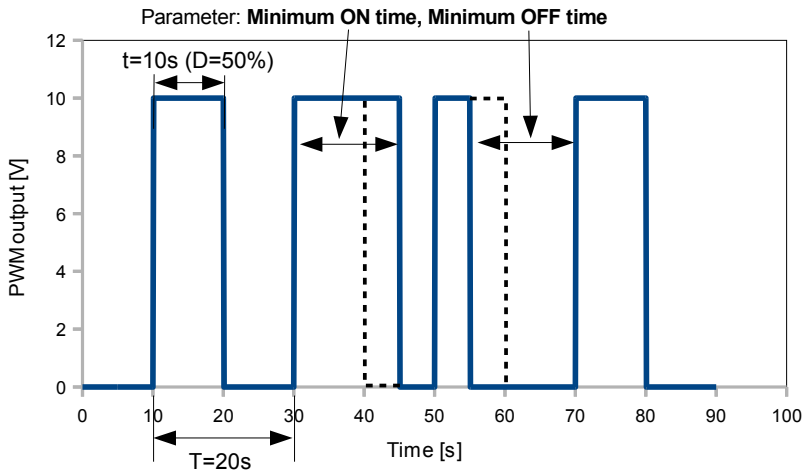
Fig. 7.101. Sample of output signal of OC-type relay in PULSE mode

7.11.4. Built-in outputs - Current output
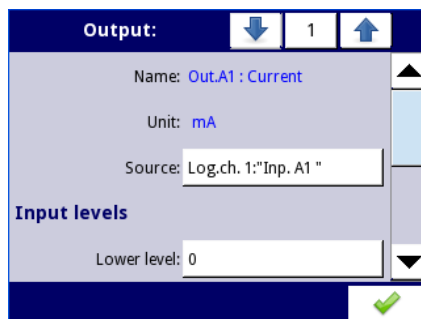
Fig. 7.102. Menu of the Passive current output

The parameters of the built-in output for the **Current outputs** hardware output configurations (**COP2**, **COP4**, **R21COP2**, **COP2OC2**) are:
  – **Name** – each output already has a name given by the device and the user cannot change it - see **7.91**,
  – **Unit** – defined permanently, it occurs only for the Current output,
  – **Source** – this parameter contains a Logical channels list where the selected logical channel will be a data source for this built-in output, data from this source is calculated according to the parameters gathered in the the **Input levels** and **Output levels** blocks,
  – **Input levels** parameter block - determines the range of a data source for this built-in output, this block has the following parameters:
    • **Lower level** and **Upper level** - these parameters limit the range of the input signal selected in the **Source** parameter, below this range the input signal will have a **Lower level** value and above this range the signal will have an **Upper level** value,
  – **Output levels** parameter block - determines the range of an output value, this block has the following parameters:
    • **Lower level** and **Upper level** - these parameters limit the range of the output signal, below this range the input signal will have a **Lower level** value and above this range the signal will have an **Upper level** value, the relationship between Lower and Upper levels is linear and it is limited by a defined range, see **7.104**,
    • **Alarm level** – this parameter defines the output value, when an alarm state appears, but with maintaining a hardware output limitation (**3÷22mA**). An alarm state is when the source logical channel returns **-Err-** or the state of exceeding range: the low **-Lo-** state and high **-Hi-** state.

Lower level and **Upper level** parameter describes the transfer equation (linear) - see **7.103**. **Lower level** of the output defines the current which can be generated when the value of the input signal equals the **Lower level**. **Upper level** of the output defines the current which can be generated when the value of the input signal equals the **Upper level**.
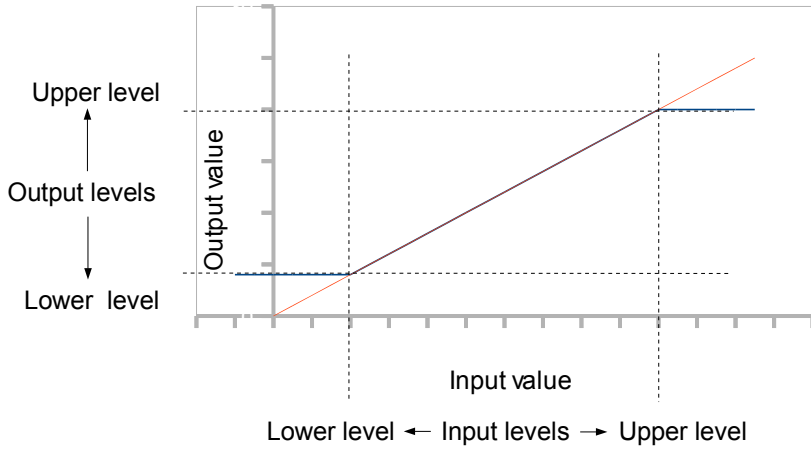
*Fig. 7.103. Input (red)-output (blue) characteristic of signal in **Current output** module*
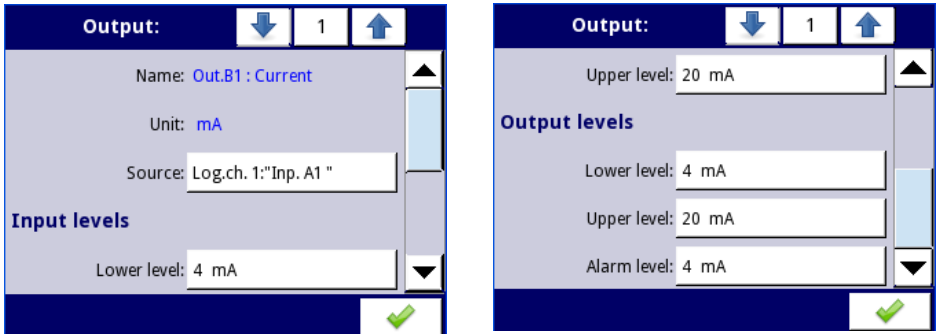
*Fig. 7.104. Parameters of the **Passive current** output*

7.11.5. Examples of build-in output configurations

7.11.5.1. *Application of the output for R21 hardware configuration*

See also: Chapter **8.4.R21, R41, COP2, COP3, OC2, OC4, R21COP2, COP2OC2, R21OC2 – HARDWARE OUTPUT CONFIGURATIONS**. *and* **Chapter 8.2 FUN2, FUN4 - UNIVERSAL HARDWARE INPUT CONFIGURATIONS WITH FLOWMETER MODE**.

Task:
        Let's say that we would like to control temperature (around 30°C) in some room by switching ON and OFF an electric heater. The hysteresis should be 5°C. The temperature is measured using a PT100 sensor and the **FUN4** hardware input configuration, and let the heater is controlled using an internal 1A relay (**R21** hardware output module).

Solution:
        First configure the device as described below, then connect the object and power supply as shown in **7.105**.
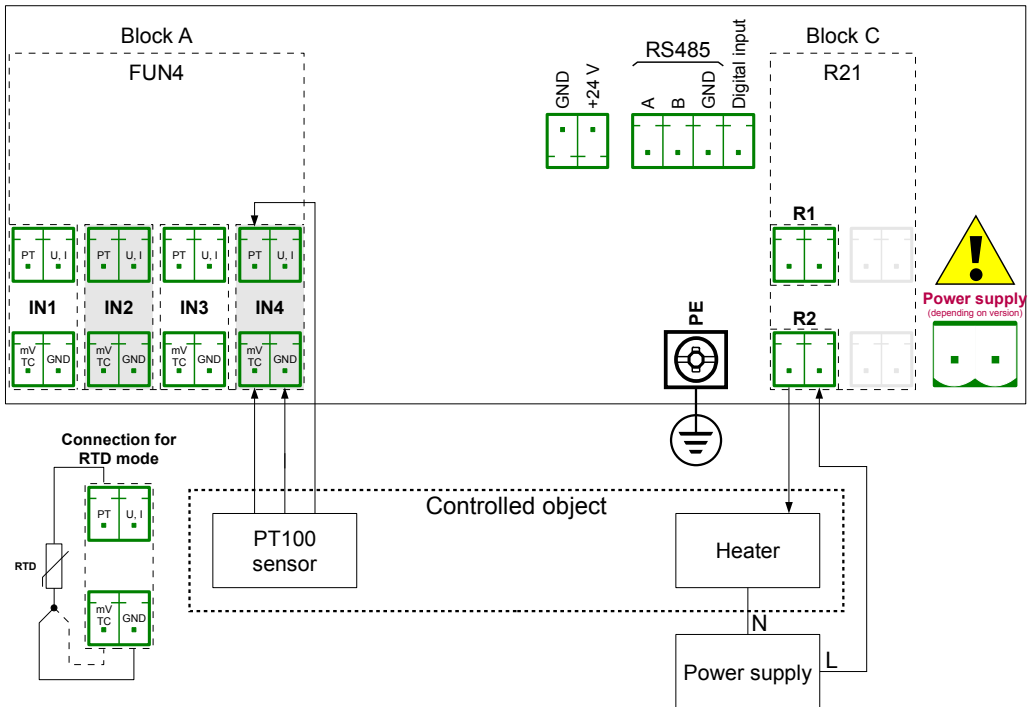
Fig. 7.105.Schematic diagram for the FUN4 hardware configuration in RTD mode and output hardware configuration

        To control temperature in the object shown in **7.105**, it is necessary to configure one **Logical channel** and **Relay output**.

For reading PT100 sensor temperature:
– touch the screen and press the **Menu** button,
– press the **Device configuration** button,
– enter the **Logical channels** menu,
– using the arrows in the top navigation bar, select any **Logical channel** such as **1**,
– in the **Name** parameter write **Feedback**,
– **Mode** parameter set as **Hardware input**,
– **Source** parameter set as **Inp.A4 : Universal** (see **7.105**),
– enter the **Configure source** submenu:
   • **Mode** parameter set as **Pt100** (3-wire),
   • in **Low limit** parameter write **-100°C**,
   • in **High limit** parameter write **600°C,**
– for **Displaying** parameter block:
   • **Format** parameter set as **numeric**,
   • **Precision** parameter set as **0.0**,
   • in **Graph low** parameter write **0°C**,
   • in **Graph high** parameter write **100°C**.

For Relay output configuration:
– touch the screen and press the **Menu** button,
– press the **Device configuration** button,
– enter the **Built-in outputs** menu,
– using the arrows in the top navigation bar, select a relay output which is connected to the heater. For this example it is the output named **Out.C2 : Relay** (see **7.105**),
– **Mode** parameter set as **below level** because we want the relay to switch on when the temperature is lower than 30°C,
– **Source** parameter set as **Log.ch.1:"Feedback"**, which returns Pt100 sensor temperature,
– **Alarm state** parameter set as **immed. OFF**,
– for **Levels** parameter block:
   • **Level mode** parameter set as **value**,
   • in **Level** parameter write **30°C**,
   • in **Hysteresis** parameter write **5°C**, the relay will switch on below 25°C and switch off above 35°C,
– all the other parameters should be on default settings.

7.11.5.2. Application of output for COP2 hardware output configuration

See also: Chapter 8.4.R21, R41, COP2, COP3, OC2, OC4, R21COP2, COP2OC2, R21OC2 – HARDWARE OUTPUT CONFIGURATIONS.

Task:
      Assume that the Logical channel 1 indicates the pressure in the range 100÷500 bars, and its result should be regenerated to the current output in the corresponding range 4÷20mA.

Solution:
For Current output configuration:
- touch the screen and press the **Menu** button,
- press the **Device configuration** button,
- enter the **Built-in outputs** menu,
- using the arrows in the top navigation bar select a current output on which we want to generate current in a demanded range, e.g. **Out.C1 : Current**,
- **Source** parameter set as **Logical channel 1**, which returns the pressure value in the range 100÷500 bars.
- for **Input levels** parameter block:
  - in **Lower level** write **100 bar**,
  - in **Upper level** write **500 bar**,
- for **Output levels** parameter block:
  - in **Lower level** parameter write **4 mA**,
  - in **Upper level** parameter write **20 mA**,
  - in **Alarm level** parameter write **4 mA**, which means that when the Logical channel 1 is in the alarm state (**Err**, **Hi** or **Lo**), then on the current output there will be 4 mA.

      The user should remember that the Current output is passive, which means it needs to be powered from a current loop. **7.106** shows how to do it. The output polarization in IO modules is irrelevant.
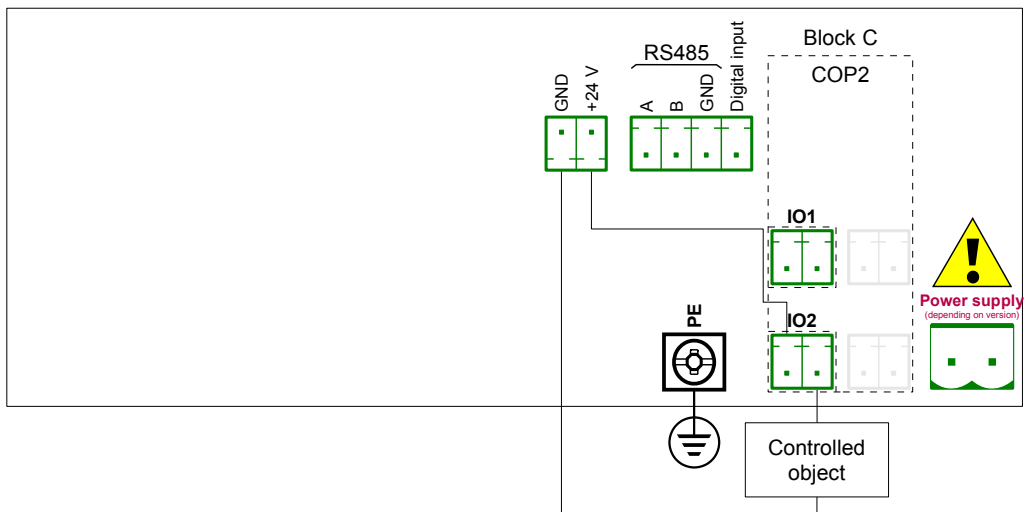
Fig. 7.106. Passive current output connection e*xample*

7.12. EXTERNAL OUTPUTS

General settings
Logical channels
Build-in inputs
Build-in outputs
External outputs
Profiles/timers
Controllers
Groups
Modbus
Network settings
Access options
E-mail notifications

MENU ➡ **Device configuration** ➡
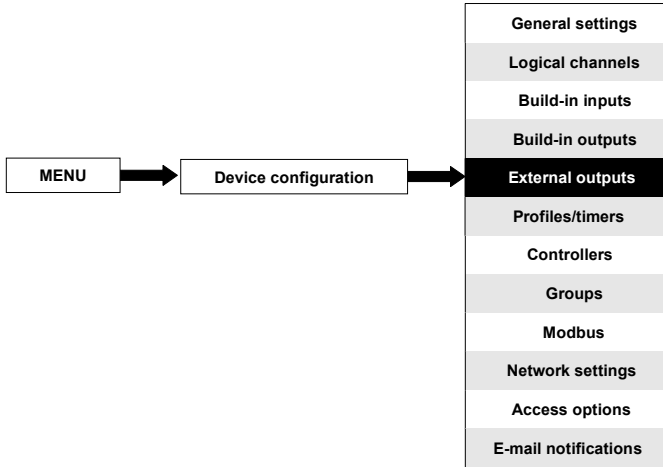
The **External outputs** menu is related to sending the date to a SLAVE device using the Modbus communication protocol. In this menu it is determined what data will be send to a SLAVE device while the configuration of the Modbus in the Master mode (for example baud rate, define the SLAVE device, active output register list and etc.) is defined in the **Modbus** menu (see **Chapter 7.16.3. Modbus - MASTER mode**).

7.12.1. External outputs - General settings

In the device there are as many external outputs as will be defined in the **Modbus** menu   (see **Chapter 7.16.3. Modbus - MASTER mode**). In case when the external outputs are not defined or inactive then in the **External outputs** menu there is an empty list.
External outputs can have a control type (control type setting, see **Chapter 7.16.3.2. Modbus MASTER - Device channels parameter block**):
   – **as a relay**,
   – **as a linear output**,

The view of the **External outputs** menu is created for two types of control: as a relay (digital output) and a linear output (analogue output) is shown in **7.107** and **7.108**, respectively.
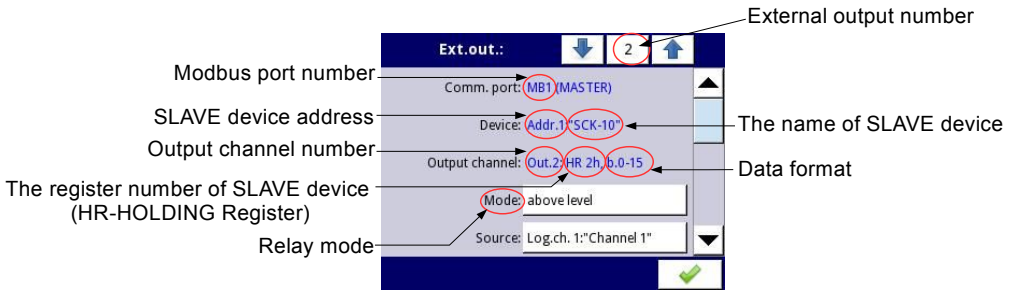
*Fig. 7.107. View of **External outputs** menu for 'as a relay' type control*
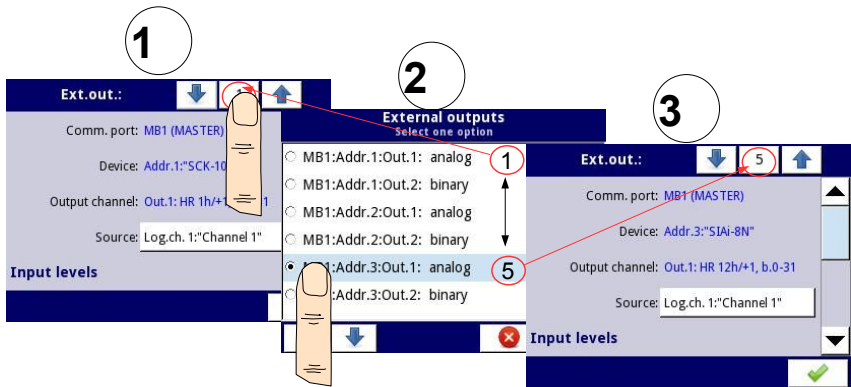
*Fig. 7.108. View of **External outputs** menu for 'as a linear output' type control*
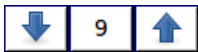
 The arrows placed in the upper right corner of the screen let you switch between a succession of external outputs. The middle button allows a direct selection of a specific external output from the list.

The parameters that are common for the **External** outputs (see **7.107** and **7.108**) are:

– **Communication port** - this parameter is read only in the **External outputs** menu, and indicates the Modbus port number, a description of Modbus ports configuration and the view of the Modbus port connectors is located in **Chapter 7.16. MODBUS**,

– **Device** - this parameter is read only in the **External outputs** menu, it shows the address and name of the SLAVE device configured in the Modbus menu (**Chapter 7.16.3. Modbus - MASTER mode**),

– **Output channel** - this parameter is read only in the **External outputs** menu, it shows the output channel number, type of register and data format configured in the Modbus menu (each SLAVE device defined in a specific address has its individually numbered output list),

– **Source** – this parameter allows to choose a source for an external output from the logical channel list (see **7.109**),
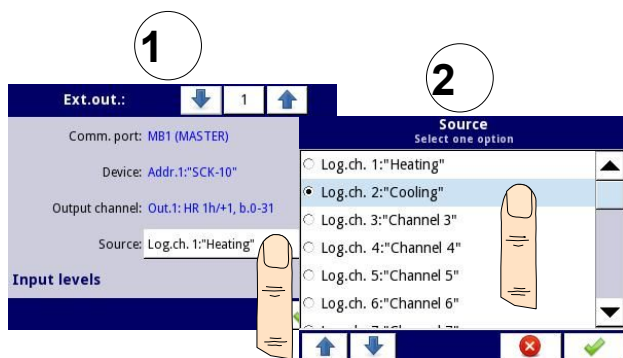
*Fig. 7.109. Sample selection of **Source** for External output*

7.12.2. External outputs - Control type: as a relay

ⓘ For **External outputs** in "**as a relay**" Control type, a **low state** is **0** and a **high state** is the **maximal** value (for 16-bit format is the value **65535**).

The parameters of **External outputs** in the Control type **as a relay** are (see **7.107**):
–   **Comm.port** - this parameter is read only in the **External outputs** menu, it is the Modbus port number for which the Modbus external output is configured; a description of the Modbus ports configuration and indication of the Modbus port connectors is located in **Chapter 7.16. MODBUS**,
–   **Device** - this parameter is read only in the **External outputs** menu, it shows the address and name of the SLAVE device configured in the **Modbus** menu (**Chapter 7.16.3. Modbus - MASTER mode**),
–   **Output channel** - this parameter is read only in the **External outputs** menu, it shows the output channel number, type of register and data format configured in the Modbus menu (each SLAVE device as defined in a specific address, has its output list individually numbered),
–   **Mode** – this parameter allows the user to select the method of the external output operation depending on the source signal (**Source** parameter); it has the following options (see **Fig. 7.94** and **Fig. 7.95**):
  •   **disabled** – this external output is inactive,
  •   **above level** - the result is a high state when the input data (see **Source** parameter) is above the level (see **Level** parameter block),
  •   **below level** - the result is a high state when the input data (see **Source** parameter) is below the level (see **Level** parameter block),
  •   **inside range** - the result is a high state when the input data (see **Source** parameter) is within the range (see **Level** parameter block),
  •   **outside range** - the result is a high state when the input data (see **Source** parameter) is out of the range (see **Level** parameter block),
–   **Source** – this parameter contains the Logical channels list where the selected logical channel will be a data source for the external output,
–   **Alarm state** – it allows the user to choose the external output reaction to an alarm state. It occurs when the logical channel which is the source for the external output, returns **Err**, **Hi** or **Lo** value. This parameter has the following options:
  •   **no change** - means that at the time of an alarm state there is no change on the

output,
- • **immed. OFF** - means that in times of an alarm state the device immediately switches the output to low state,
- • **immed. ON** - means that in times of an alarm state the device immediately switches the output to high state,
- • **timed OFF** - means that in times of an alarm state the device switches the output to low state after a time delay set in the **Timing** parameter block,
- • **timed ON** - means that in times of an alarm state the device switches the output to low state after a time delay set in the **Timing** parameter block,
- – **Levels** block parameter - these parameters allow the user to set the range of changes of the output depending on the input signal, discussed below in this **Chapter**,
- – **Timing** parameter block - these parameters allow the user to set the delay time change the output state and minimum duration of the output state, discussed below in this **Chapter**.

Levels parameter block (see **Fig. 7.94**, **Fig. 7.95** and **Fig. 7.96**)

These parameters depend on the **Mode** parameter. The options are:
- – **Level** - this parameter defines the source signal level at which the output switches the state (from low to high state or from high to low state), occurs for the mode:
  - • **above level** - above the level we get a high state at the output,
  - • **below level** - below the level we get a high state at the output,
- – **Lower level and Upper level** - these parameters define the range at which a switch the output state (from low to high state or vice versa from high to low state), occurs for the mode:
  - • **inside range** - if the input data is within the defined range at the output we get a high state,
  - • **outside range** - if the input data is outside the defined range at the output we get a high state,
- – **Hysteresis** – this parameter defines the offset for a switch state on the output,

Timing parameter block (see **Fig. 7.99**)

The parameters of this block include:
- – **ON delay** – it is the time that must pass from the time of a switch request to the actual switch from low state to high state (see **Fig. 7.97**),
- – **OFF delay** - it is the time that must pass from the time of a switch request to the actual switch from high state to low state (see **Fig. 7.97**),
- – **Min.ON time** - the minimum duration of a high state (if the output switches to high state the low state will occur not faster than **Minimum ON time**), see **Fig. 7.98**,
- – **Min.OFF time** - the minimum duration of a low state (if the output switches to low state the high state will occur not faster than **Minimum OFF time**), see **Fig. 7.98**,

ⓘ **Note!** If the external output is active (see the parameter **Output active=yes** in the **Chapter 7.16.3.2. Modbus MASTER - Device channels parameter block**) for **Mode=disabled** the device sends the '0' value to a Slave device.

7.12.3. External outputs - Control type: as a linear output

The parameters of **External outputs** in the type of control **as a linear output** are as follows (see **7.108**):

– **Comm.port** - this parameter is read only in the **External outputs** menu, it is the Modbus port number for which the Modbus external output is configured; a description of the Modbus ports configuration and indication of the Modbus port connectors is located in **Chapter 7.16. MODBUS**,

– **Device** - this parameter is read only in the **External outputs** menu, it shows the address and name of the SLAVE device configured in the **Modbus** menu (**Chapter 7.16.3. Modbus - MASTER mode**),

– **Output channel** - this parameter is read only in the **External outputs** menu, it shows the output channel number, type of register and data format configured in the Modbus menu (each SLAVE device as defined in a specific address, has its output list individually numbered),

– **Source** - this parameter contains the Logical channels list where the selected logical channel will be a data source for the external output. Data from this source is calculated according to the parameters in the **Input levels** block and **Output levels** block,

– **Input levels** parameter block – data entered here has the same unit as a selected logical channel (source). For this block, the parameters are:
   • **Lower level** – it sets a lower limit value below which the source input signal will equal this value,
   • **Upper level** - it sets an upper limit value above which the source input signal will equal this value,

– **Output levels** parameter block – this block has the following parameters:
   • **Lower level** – it sets a lower limit output signal value below which the output signal will equal this value,
   • **Upper level** – it sets an upper limit output signal value below which the output signal will equal this value,
   • **Alarm level** – it sets an output value which has to appear when an alarm state appears. It occurs when the logical channel which is the source for the external output returns the **Err**, **Hi** or **Lo** value.

The user should remember that the relationship between input and output levels are described by the linear transmission equation, and it is limited by the range, (see **7.110**).
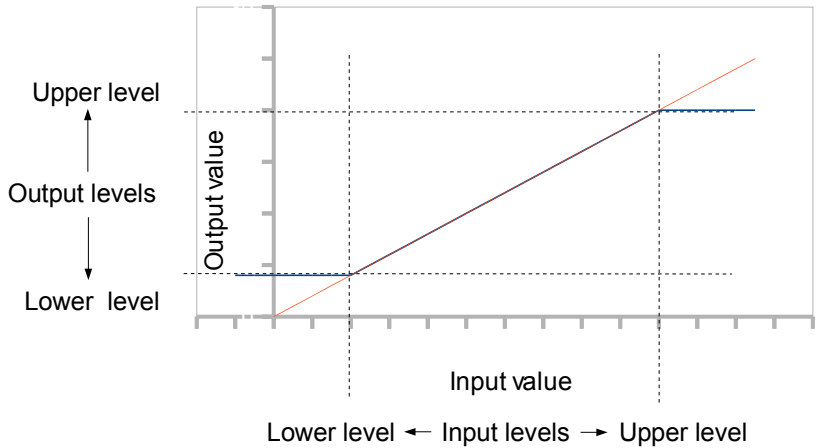
*Fig. 7.110. The relation between the input and output for **External output***

Note! If the external output is active (see the parameter **Output active=yes** in the **Chapter 7.16.3.2. Modbus MASTER - Device channels parameter block**) for an undefined **Source** parameter (e.g. a Logical channel in the **Mode=disabled**) or for an undefined **Input** and **Output levels** parameter block (every parameter has value 0) the **MultiCon CMC-N16** sends the **'0'** value to a Slave device.

7.12.4. Examples of external output configurations

During the **External output** configuration the user should terminate communication between the **MultiCon CMC-N16** and the SLAVE device.

7.12.4.1. Application of external output for protocol Modbus in the MASTER mode

See also: **Chapter 7.16.3. Modbus - MASTER mode**.

Task:

The task is to send data located in Logical channel 1 to a SLAVE device (address 5) without scaling and restrictions.
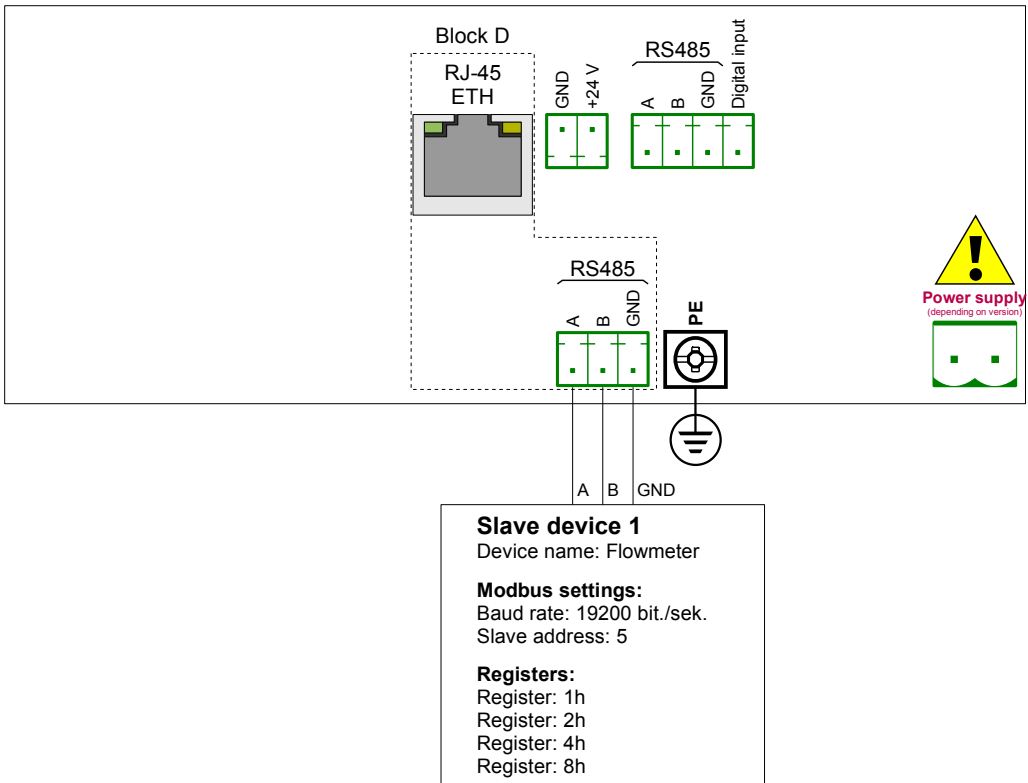
Fig. 7.111. Connection diagram for Modbus MB2

<u>*Solution:*</u>
First we need to configure settings in the Modbus menu as it is shown in
7.16.4.2. Configuration of the Modbus Input in the MASTER mode

Next we configure logical channels used to set values, which will be sent by RS-485 to the output. To do this:
– touch the screen and press the **Menu** button,
– press the **Device configuration** button,
– enter the **Logical channels** menu,
– using the arrows in the top navigation bar, select any **Logical channel** such as 1,
– in the **Name** parameter write **Set point value**,
– **Mode** parameter set as **Set point value**,
– in the **Set point** value parameter write **50**,
– for **Displaying** block parameters:
  • **Format** parameter set as **numeric**,
  • **Precision** parameter set as **0**,
  • in **Graph low** parameter write **0**,
  • in **Graph high** parameter write **65535.**

Next two logical channels set similarly:
- **Logical channel 2**:
  - in **Name** parameter write **Valve position**,
  - in **Set point** value write **1**,
- **Logical channel 3**:
  - in **Name** parameter write **Alarm state**,
  - in **Set point** parameter write **0**,
- all the other parameters should be on default settings.

Define external outputs. To do this:
- touch the screen and press the **Menu** button,
- press the **Device configuration** button,
- enter the **External outputs** menu,
- using the arrows in the top navigation bar, select **Ext.out.:1**, (it is a linear output),
- there are three constant parameters which are consistent **Modbus** menu configuration:
  - Comm.port: MB2 (MASTER),
  - Device: Addr.5:"Flowmeter",
  - Output channel: Out.1:HR 2h/+1,b.0-31,
- for next parameters:
  - **Source** parameter set as logical channel named the **Set point value**,
  - for **Input levels** block parameters:
    - in **Lower level** parameter write **0**,
    - in **Upper level** parameter write **65535**,
  - for **Output levels** block parameters:
    - in **Lower level** parameter write **0**,
    - in **Upper level** parameter write **65535**,
    - in **Alarm level** parameter write **0**,
- using the arrows in the top navigation bar, select **Ext.out.:2**, (it is a relay output),
- there are three constant parameters which are consistent with the **Modbus** menu configuration:
  - Comm.port: MB2 (MASTER),
  - Device: Addr.5:"Flowmeter",
  - Output channel: Out.2:HR 4h, b.0,
- for next parameters:
  - **Mode** parameter set as **above level**; because we want the valve to switch on when the value is greater than **'0'**,
  - **Source** parameter set as logical channel named the **Valve position**,
  - **Alarm state** parameter set as **immed. OFF**, which means that when logical channel 2 will be returning **Err**, **Hi** or **Lo** state, the output will be in low state (valve is off),
  - for **Levels** block parameters:
    - **Level** mode parameter set as **value**,
    - in **Level** parameter write **0**,
    - in **Hysteresis** parameter write **0**,
  - for **Timing** block parameters:
    - in **ON delay** parameter write **0 sec.**,
    - in **OFF delay** parameter write **10 sec.**,
    - in **Min.ON time** parameter write **10 sec.**,

    - in **Min.OFF time** parameter write **10 sec.**; this will be a protection from fast

   switching on and off the valve,
- using the arrows in the top navigation bar, select **Ext.out.:3**, (it is a relay output),
- there are three constant parameters which are consistent with the **Modbus** menu configuration:
  • Comm.port: MB2 (MASTER),
  • Device: Addr.5:"Flowmeter",
  • Output channel: Out.3:HR 8h, b.0,
- for next parameters:
  • **Mode** parameter set as **above level**,
  • **Source** parameter set as logical channel named the **Alarm state**,
  • **Alarm state** parameter set as **immed. ON**,
  • for **Levels** block parameters:
    ◦ **Level mode** parameter set as **value**,
    ◦ in **Level** parameter write **0**,
    ◦ in **Hysteresis** parameter write **0**,
- all the other parameters should be on default settings.

In the next step you need to define a Group of configured logical channels to display them in one window on the device screen. To do this:
- touch the screen and press the **Menu** button,
- press the **Device configuration** button,
- enter the **Groups** menu,
- in the **Name** parameter write **Flowmeter**,
- for **Channels** block parameters:
  • **Slot 1** parameter set as **logical channel 1** named **Set point value**,
  • **Slot 2** parameter set as **logical channel 2** named **Valve position**,
  • **Slot 3** parameter set as **logical channel 3** named **Alarm state**,
  • all the other parameters – **Slot 4÷6** set as disabled, because we want to display only three logical channels on the screen,
- all the other parameters should be on default settings.

The last step is to define the Initial view displayed on the LCD screen when the device starts. The Initial view has to display a defined Group 1 in the Values mode. To do this:
- enter the **General settings** menu,
- for **Initial view** block parameters:
  • **Display mode** parameter set as **values**,
  • **Displayed group** parameter set as **Group 1: "Flowmeter"**.

Finally, exit the menu by pressing the Save changes button. The result of the changes will be visible after a call to the first measurement.

7.13. PROFILES/TIMERS
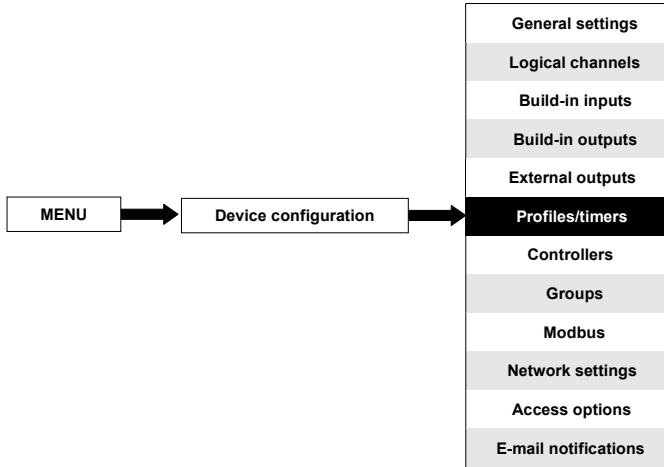
The Profiles/timers menu allows the user to define any profile/timer which can be used to control any process.

7.13.1. Profiles/timers - General settings

In the device there are **8 independent settings of Profiles/timers** available. A configured **Profile/timer** can be used by any **Logical channel** switched to the **Profile/timer** mode - **Fig. 7.93** (see also **Chapter 7.9.8. Logical Channels - Profile/timer mode**).
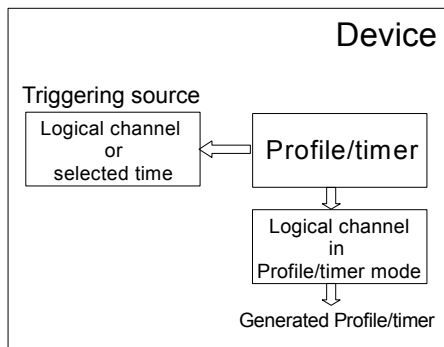
Fig. 7.112. Block diagram of the device configuration for generating Profiles/timers

The window with basic parameters of the **Profile/timer** shown in **7.113**.
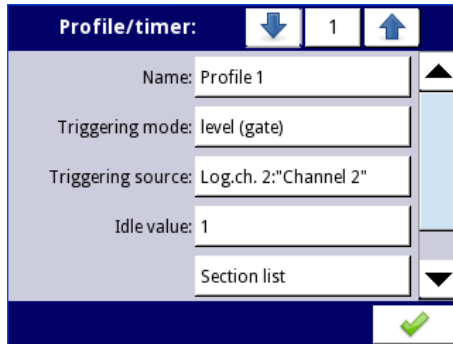
Fig. 7.113. View of the configuration Profiles/timers window

 The arrows placed in the upper right corner of the screen allow you to switch between Profile/timer. The middle button allows you to select a specific Profile/timer from the list directly.

The common parameters for **Profile/timer** are:
- **Name** – it gives the name to the Profile/timer,
- **Triggering mode** – using this parameter the user selects a way to generate a Profile/timer, there are five modes of triggering:
  - **disabled**,
  - **level (gate)** – this means that the **Profile** which was configured by the user will be generated when the source signal will have a value > 0, otherwise (if source value ≤ 0) a defined Profile will not be generated. For more details see **Chapter 7.13.2. Profiles/timers - Triggering mode: level (gate), edge (once), edge (retrig.)**,
  - **edge (once)** – this means that the profile which was configured by the user **Profile** will be triggered by the rising edge signal (from values ≤ 0 to the value > 0) coming from the source signal. After the rising edge the **Profile** will be generated in whole (once), regardless of further changes to the signal source. For more details see **Chapter 7.13.2. Profiles/timers - Triggering mode: level (gate), edge (once), edge (retrig.)**,
  - **edge (retrig.)** - this means that the profile configured by the user **Profile** will be triggered by the rising edge signal (from values ≤ 0 to the value > 0) that comes from the source signal. However, in this mode, unlike the **edge (once)** mode a defined **Profile** will be generated from the beginning every time the Triggering source signal will generate an edge, whether that Profile had been completed or not. For more details see **Chapter 7.13.2. Profiles/timers - Triggering mode: level (gate), edge (once), edge (retrig.)**,
  - **on time** – in this mode the **Profile/timer** is generated in selected time (using parameter **Triggering times**). For more details see **Chapter 7.13.3. Profiles/timers - Triggering mode: on time**,
- **Pause mode** – way to pause a generated Profile/timer,
  - **disabled** – Profile/timer can not be paused,
  - **high level** – Profile/timer will be paused when the channel's value in the **Pause source** parameter is greater than zero (>0) and will be resumed when this value is lower or equals zero (≤0). If a Profile/timer is paused and a new trigger

is detected, the Profile/timer will remain in the pause state. Additionally, when in the **Section list** parameter, **Shape** of **Section num. 1** is set as **const.value**, then the Profile/timer will set itself on **Final value** of this section, and when in the **Section list** parameter, **Shape** of **Section num. 1** is set as **slope**, then the Profile/timer will set itself on its **Idle value**.

- **low level** – Profile/timer will be paused, when the channel's value in the **Pause source** parameter is lower or equal zero (≤0) and will be resumed when this value is greater than zero (>0). If Profile/timer is paused and a new trigger is detected, the Profile/timer will remain in the pause state. Additionally, when in the **Section list** parameter, **Shape** of **Section num. 1** is set as **const.value**, then the Profile/timer will set itself on **Final value** of this section, and when in the **Section list** parameter, **Shape** of **Section num. 1** is set as **slope**, then the Profile/timer will set itself on its **Idle value**.
- **rising edge** – Profile/timer will be paused when the device detects the channel's value change from lower or equal zero (≤0) to greater than zero (>0) in the Pause source parameter, and will be resumed when the device detects another change like that in this channel. If Profile/timer is paused and a new trigger is detected, the Profile/timer will react to this trigger as if it wan not paused.
- **falling edge** – Profile/timer will be paused when the device detects the channel's value change from greater than zero (>0) to lower or equal zero (≤0) in the **Pause source** parameter, and will be resumed when the device detects another change like that in this channel. If Profile/timer is paused and a new trigger is detected, the Profile/timer will react to this trigger as if it was not paused.

– **Pause source** – this parameter is not visible when the **Pause mode** parameter is set as **disabled**. It contains the list of Logical channels, where the selected one will be a pause source for the Profile/timer,

– **Idle value** – the value before and after generating the defined Profile,

– **Section list** – submenu where the user can define Profile/timer consisting of up to 100 sections. For more details see below in this **Chapter**.

– **Looping** – each Profile has a possibility to repeat its generated signal. This parameter has the following options:
- **disabled** – the Profile is generated only once,
- **counted** – allows to generate Profile specified number of times defined in the **Loop count** parameter,
- **from logical channel** – this option allows the user to run a **Profile** a specified number of times set in the selected logical channel defined by the **Looping source** parameter,
- **infinite** – this option allows the user to repeat the generated **Profile** infinitely,

– **Loop count** – this parameter is visible only when the **Looping** parameter is set on **counted**. It allows the user to enter the number of generated Profile repetitions,

– **Looping source** – this parameter is visible only when the **Looping** parameter is set on **from logical channe**l. It allows the user to select the logical channel from the list, whose value will be compared to the number of Profile cycles. If the cycle value is equal or greater than the value in the logical channel, next repetition of the Profile will not start,

– **Return to position** – this parameter is invisible only when the **Looping** parameter is set as **disabled**. It allows the user to select a section from the list, from which any next generated Profile should start. In case when the **Section list** is empty, this parameter is also empty,

Section list sub-menu

It allows the user to configure Profile/timer consisting of up to 100 sections and for each of them the user can set: Duration, Unit, Shape and Final value.

This button allows the addition of a new section to a list.

This button allows the removal of a section from a list.
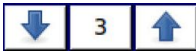
The arrows placed in the upper right corner of the screen allow switching between sections. The middle button allows a direct selection of a specific section.

The **Section list** parameters are:
– **Duration** – defines the duration of the section, it depends on the **Unit** parameter,
– **Unit** – the user can select available options: **second**, **minute**, **hour** which sets a unit of the duration,
– **Shape** – allows to choose the shape of the defined section (**const.value**, **slope**),
– **Final value** – allows to set the level for **const.value**, or the end value for **slope** for the selected Profile/timer.
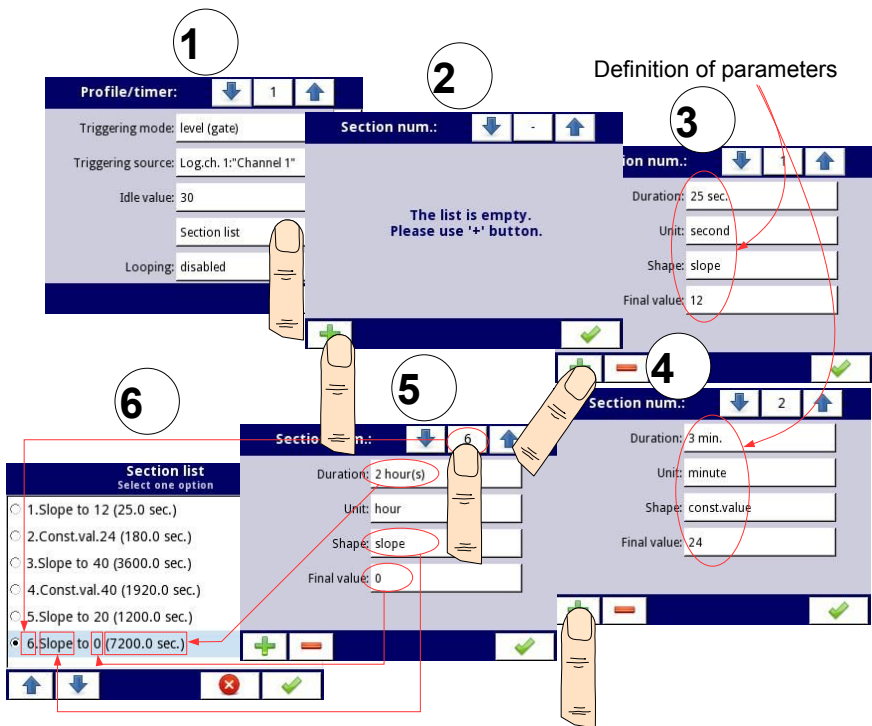
Fig. 7.114. Sample of Profile/timer configuration

Comments to the **Looping** parameter

If the user selects: **counted** or **infinite** repeats of the Profile/timer, the user has two alternatives:

– if the section from which begins the next repeat Profile/timer is a **ramp**, then in the whole duration of this section is linear generating the output signal from the **final value** of the previous section to **final value** of this section. This is shown in **7.115** (dashed line-run profile),

– if the section from which the next repeat of the Profile/timer begins is a constant value, then Profile signal quickly transient (0.1 seconds)from the **final value** of the previous section to a **constant value** in this section. This is shown in **7.115** (profile guided fine line).
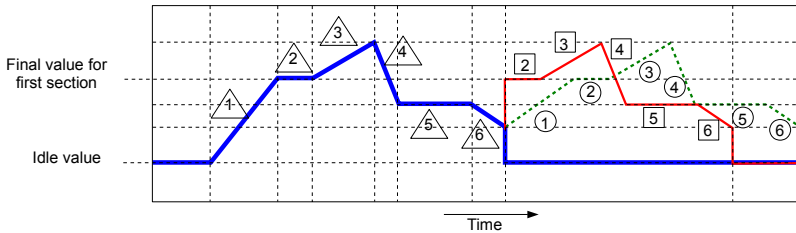
Legend:

△1 Section for Profile without looping

① Section for Profile with looping -> Return to position 1 (slope)

2 Section for Profile with looping -> Return to position 2 (const. value)

Fig. 7.115. Sample of Profile/timer with description of parameters

7.13.2. Profiles/timers - Triggering mode: level (gate), edge (once), edge (retrig.)

The **Profiles/timers** parameters for triggering mode: level (gate), edge (once), edge (retrig.) are:

– **Name** – it gives the name to the Profile,

– **Triggering mode** – selects a way to generate a Profile/timer, there are five modes of triggering:

   • **disabled**,

   • **level (gate)** – this means that the Profile configured by the user when the source signal has the value **> 0**, otherwise (if source value **≤ 0**) the defined Profile will not be generated. See **7.116** and example **a)** in **7.117**,

   • **edge (once)** – this means that the **Profile** configured by the user will be triggered by the rising edge (from values **≤ 0** to the value **> 0**) of the source signal. After the rising edge the **Profile** will be generated in whole (once), regardless of further changes to the signal source, see **7.116** and example **b)** in **7.117**,

   • **edge (re-triggering)** – this means that the **Profile** configured by the user will be triggered by the rising edge (from values **≤ 0** to the value **> 0**) of the source signal. However, in this mode, unlike the **edge (once)** mode the defined **Profile** will be generated from the beginning every time the **Triggering source** signal generates an edge, whether that Profile had been completed or not. See **7.116** and example **c)** in **7.117**,

- **Triggering source** – allows to choose a triggering source of the Profile/timer from the logical channel list,
- **Idle value** – the value which occurs at the idle state which means before and after generating the Profile/timer,
- **Section list** – the submenu in which the user can define up to 100 sections generated by the Profile/timer. For more information see **Chapter 7.13.1. Profiles/timers - General settings**,
- **Looping** – each Profile/timer has an ability to repeat its generated signal, this parameter has the following options:
  - **disabled** – the profile is not repeated,
  - **counted** – allows to generate the profile a specified number of times defined in the **Loop count** parameter,
  - **from logical channel** - allows to run the profile a number of times that equals the value in the selected logical channel determined by the **Looping source** parameter,
  - **infinite** – allows to repeat the generated **Profile** infinitely,
- **Loop count** – this parameter is visible only if the **Looping** parameter is set as **counted**, defines the number of repetitions of the generated Profile,
- **Looping source** – this parameter is visible only if the **Looping** parameter is set as **from logical channel**. It allows to choose a logical channel from the list, whose value will be compared with the number of already made cycles of the Profile,
- **Return to position** – this parameter is invisible if the **Looping** parameter is set as **disabled**, allows to select a fixed position from which every next generated Profile has to start.

Fig. 7.116. Sample of Profile/timer
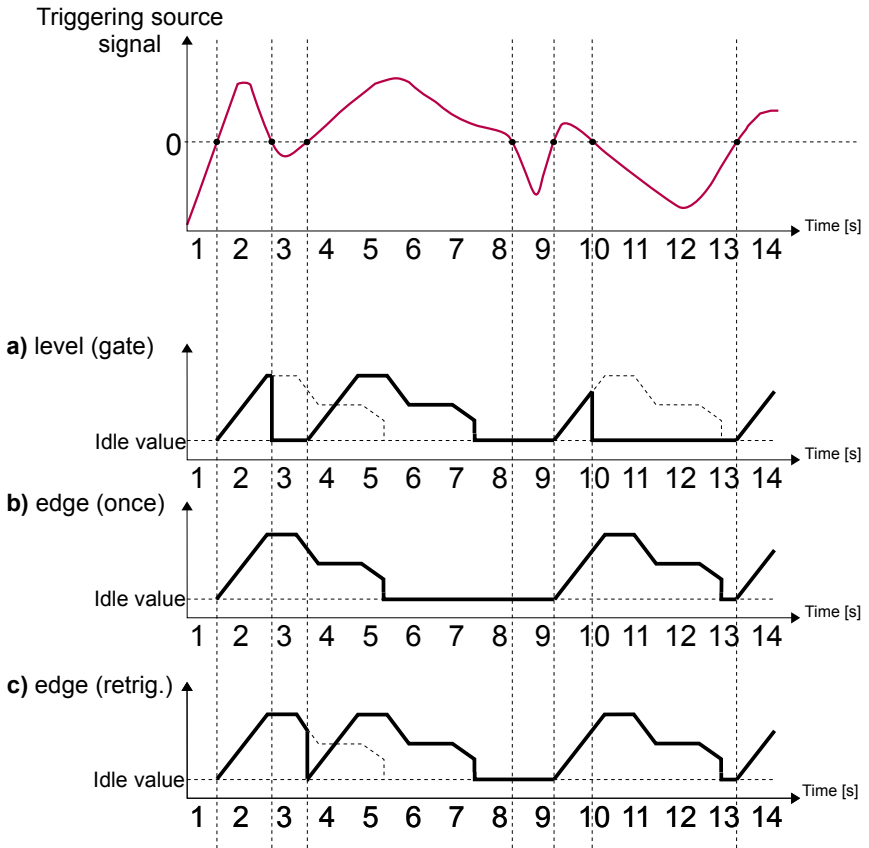
*Fig. 7.117. Samples of **Profile** output waveforms defined in **7.116**, triggered by signal selected in **Triggering source** parameter for **Triggering mode**: **level (gate), edge (once), edge (retrig.)***

7.13.3. Profiles/timers - Triggering mode: on time

The parameters of **Profiles/timers** for **triggering mode**: **on time** are:
– **Name** – it gives the name to the Profile/timer,
– **Triggering mode=on time** – allows to trigger the Profile at a demanded time defined in the **Triggering times** parameter,
– **Triggering times** – the submenu with the parameters which define the times of Profile triggering. These parameters are:
  • **Months**,
  • **Days**,
  • **Week days**,
  • **Hours**,
  • **Minutes**,
  • **Seconds**,

- **Idle value** – the value which occurs at the idle state, which means before and after generating the Profile/timer,
- **Section list** – the submenu in which the user can define up to 100 sections generated by the Profile/timer. For more information see **Chapter 7.13.1. Profiles/timers - General settings**,
- **Looping** – each Profile/timer has an ability to repeat its generated signal, this parameter has the following options:
  - **disabled** – the profile is not repeated,
  - **counted** – allows to generate the profile a specified number of times defined in the **Loop count** parameter,
  - **from logical channel** – allows to run the profile a number of times that equals the value in the selected logical channel determined by the **Looping source** parameter,
  - **infinite** – allows to repeat the generated **Profile** infinitely,
- **Loop count** – this parameter is visible only if the **Looping** parameter is set as **counted**, defines the number of repetitions of the generated Profile,
- **Looping source** – this parameter is visible only if the **Looping** parameter is set as **from logical channel**. It allows to choose a logical channel from the list, whose value will be compared with the number of already made cycles of the Profile,
- **Return to position** – this parameter is invisible if the **Looping** parameter is set as **disabled**, allows to select a fixed position from which every next generated Profile has to start.

ⓘ The parameters in the **Triggering times** submenu allow to define the moment of generating the Profile with an accuracy of seconds. In every parameter the user can select any number of options. If none of the options is selected in any parameter, after accepting, the "**Press to select**" description appears next to the parameter. Additionally, in that case, the Profile will not be generated.

In **7.118** and **7.119** we can see examples of Profile/timer: triggering times configuration and output waveform. Operation of the Profile in **'on time'** mode is similar to **edge (once)** mode - see **Chapter 7.13.2. Profiles/timers - Triggering mode: level (gate), edge (once), edge (retrig.)** because after the rising edge of the triggering source the **Profile** will be generated in whole, regardless of further changes of the signal source at the time of generating the Profile.
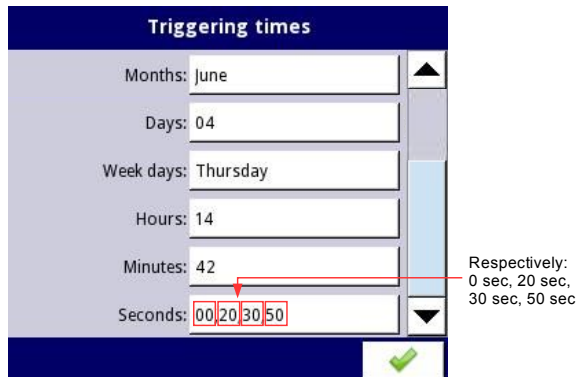
*Fig. 7.118. Samples of time settings for **Profile/timer***
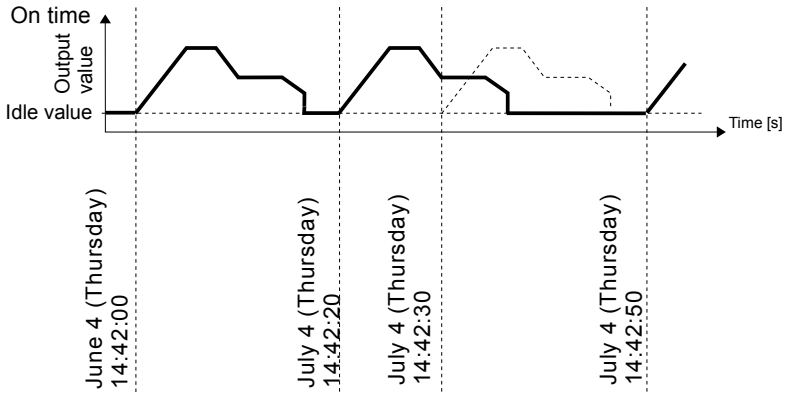
*Fig. 7.119. Waveform for **'on time'** triggering mode and time parameters in accordance with **7.118***

7.13.4. Examples of Profile/timer configurations

7.13.4.1. Application of the Profiles/timers

See also: **Chapter 7.13.2. Profiles/timers - Triggering mode: level (gate), edge (once), edge (retrig.)**.

Task:
        The task is to create a Profile timer in logical channel 1 in the **edge (once)** mode, whose source is logical channel 2 which is connected to the current input of the **FUN2** hardware configuration. The profile consists of four segments:
1. ramp from 0 to 10 within 5 seconds,
2. const.value 8 within 2 seconds,
3. ramp from 8 to 4 within 3 seconds,
4. const.value 4 within 1 second,
Looping is disabled and the idle value is 0.

Solution:
        First configure the device as described below, then connect the module to the measuring system. Example of this connection is shown in **Fig. 7.73** (see also **Appendices 8.2. FUN2, FUN4   - UNIVERSAL HARDWARE INPUT CONFIGURATIONS WITH FLOWMETER MODE**).
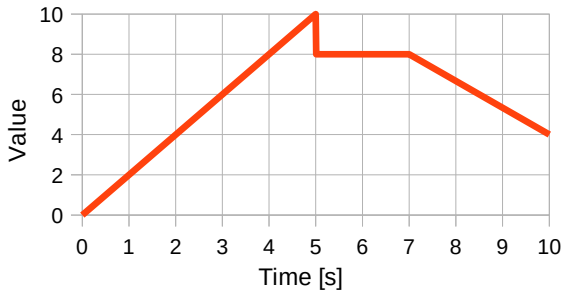
Fig. 7.120. An example of the waveform

In the first step you need to configure the Profile/timer. To do this:
–   touch the screen and press the **Menu** button,
–   press the **Device configuration** button,
–   enter the **Profiles/timers** menu,
–   using the arrows in the top navigation bar, select any **Profile/timer** such as **1**,
–   in the **Name** parameter write **Profile 1**,
–   **Triggering mode** parameter set as **edge (once)**,
–   **Triggering source** parameter set as **Logical channel 2**,
–   in the **Idle value** parameter write **0**, because we want the Profile to have the 0 value on the output in the beginning,
–   enter the **Section list** submenu to define the sections:
    •   press the green plus at the bottom left side of the screen to configure **Section num.1**:
        ◦   in **Duration** parameter write **5 sec.**,
        ◦   **Unit** parameter set as **second**,
        ◦   **Shape** parameter set as **ramp**,
        ◦   in **Final value** parameter write **10**,
    •   press the green plus at the bottom left side of the screen to configure **Section num.2**:
        ◦   in **Duration** parameter write **2 sec.**,
        ◦   **Unit** parameter set as **second**,
        ◦   **Shape** parameter set as **conts.value**,
        ◦   in **Final value** parameter write **8**,
    •   press the green plus at the bottom left side of the screen to configure **Section num.3**:
        ◦   in **Duration** parameter write **3 sec.**,
        ◦   **Unit** parameter set as **second**,
        ◦   **Shape** parameter set as **ramp**,
        ◦   in **Final** value parameter write **4**,
    •   press the green plus at the bottom left side of the screen to configure **Section num.4**:
        ◦   in **Duration** parameter write **1 sec.**,
        ◦   **Unit** parameter set as **second**,
        ◦   **Shape** parameter set as **const.value**,
        ◦   in **Final** value parameter write **4**,
–   **Looping** parameter set as **disabled**, because we want the Profile to be executed only once after triggering from logical channel 2,

  – in the end leave the **Profiles/timers** menu.

In the next step you need to define a **Logical channel**:
  – touch the screen and press the **Menu** button,
  – press the **Device configuration** button,
  – enter the **Logical channels** menu,
  – using the arrows in the top navigation bar, select any **Logical channel** such as **1**,
  – in the **Name** parameter write **Profile**,
  – **Mode** parameter set as **P/T1:"Profile 1"**,
  – for **Displaying** block parameters:
    • **Format** parameter set as **numeric**,
    • **Precision** parameter set as **0.0**,
    • in **Graph low** parameter write **0**,
    • in **Graph high** parameter write **10**.

        The way of configuring a logical channel (**Triggering source**) in the **Hardware input** mode for the **current input** is shown in **7.9.11.1. Application of the Logical channel in the Hardware input mode for the FUN2 hardware configuration** (this is the way to configure the logical channel 2).

        The another method is described in **Chapter 7.9.11.7. Application of Logical channel in the Profile/timer mode**.

7.13.4.2. Application of the Profiles/timers triggered on time

See also: **Chapter 7.13.3. Profiles/timers - Triggering mode: on time**.

Task:
        The task is to create a Profile/timer in logical channel 3, which will generate its profile from the moment of turning on the device till turning it off. The profile consists of two sections:
1. const.value 1 within 0.2 second,
2. const.value 0 within 1.8 second,
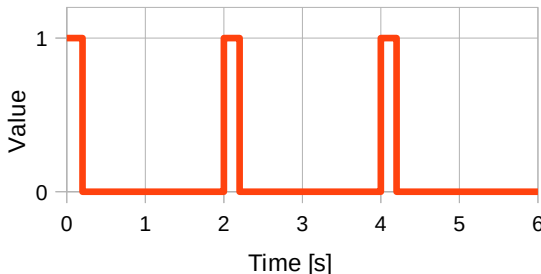
Fig. 7.121. An example of the waveform

Solution:
To complete the task a logical channel in the Profile/timer Mode will be needed:
  – touch the screen and press the **Menu** button,
  – press the **Device configuration** button,
  – enter the **Profiles/timers** menu,

- using the arrows in the top navigation bar, select any **Profile/timer** such as **1**,
- in the **Name** parameter write **Waveform generator**,
- **Triggering mode** parameter set as **on time**,
- enter the **Triggering times** submenu,
    - enter the **Months** menu and press the ⌨ icon at the bottom left side of the screen,
    - enter the **Days** menu and press the ⌨ icon at the bottom left side of the screen,
    - enter the **Week days** menu and press the ⌨ icon at the bottom left side of the screen,
    - enter the **Hours** menu and press the ⌨ icon at the bottom left side of the screen,
    - enter the **Minutes** menu and press the ⌨ icon at the bottom left side of the screen,
    - enter the **Seconds** menu and press the ⌨ icon at the bottom left side of the screen,
- enter the **Section list** submenu,
    - press the green plus at the bottom left side of the screen to configure **Section num.1**:
        ◦ in **Duration** parameter write **0.2** sec.,
        ◦ **Unit** parameters set as **second**,
        ◦ **Shape** parameter set as **const.value**,
        ◦ in **Final value** parameter write **1**,
    - press the green plus at the bottom left side of the screen to configure **Section num.2**:
        ◦ in **Duration** parameter write **1.8** sec.,
        ◦ **Unit** parameters set as **second**,
        ◦ **Shape** parameter set as **const.value**,
        ◦ in **Final** value parameter write **0**,
- **Looping** parameter set as **infinite**,
- **Return to position** parameter set as **1.Const.val.1(0.2 sec.)**,
- leave the **Profiles/timers** menu and enter the **Logical channels** menu,
- using the arrows in the top navigation bar, select the **Logical channel 3**,
- in **Name** parameter write **Waveform generator**,
- **Mode** parameter set as **Profile/timer**,
- **Source** parameter set as **P/T1:"Waveform generator"**,
- for **Displaying** block parameters:
    - in **Graph low** parameter write **-1**,
    - in **Graph high** parameter write **2**.

7.14. CONTROLLERS

General settings
Logical channels
Build-in inputs
Build-in outputs
External outputs
Profiles/timers
Controllers
Groups
Modbus
Network settings
Access options
E-mail notifications

MENU → **Device configuration** →
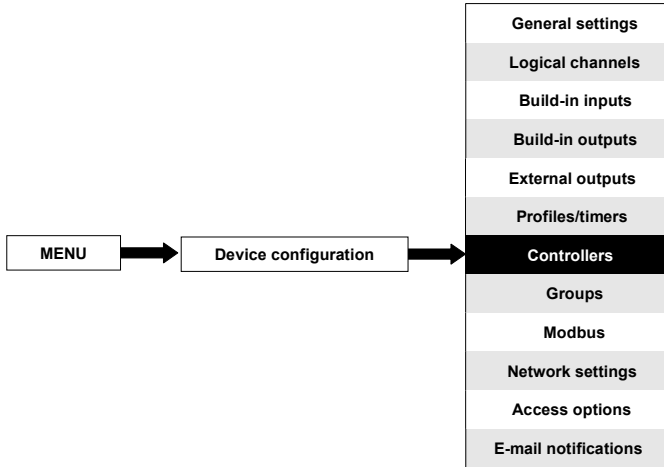
Most controlling processes can be realised using a simple ON - OFF mode, however it is sometimes necessity to use more advanced methods to control the actuators. The **MultiCon CMC-N16** has implemented **proportional–integral–derivative controllers** (**PID controllers**) which is a generic control loop feedback mechanism (controller) by calculating an "error" value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimize the error by adjusting the process control outputs. In the device there are 8 independent settings of PID type controllers available.

7.14.1. Controllers - General settings

In the system there are **8 independent settings of PID type Controllers** available, which can by used by any **Logical channel** switched in the **Controller** mode - see **Chapter 7.9.7. Logical Channels - Controller mode**.

The window with basic parameters of the **Controller** is shown in **7.122**.

Fig. 7.122. Main configuration of Controller profile

The arrows placed in the upper right corner of the screen allow switching between controllers to configure settings of controller parameters. The middle button allows a direct selection of a specific controller from the list.

The parameters of **Controllers** are:
– **Controller name** – it gives the name to the Controller,
– **Autotuning parameters** parameter block - this block contains the following parameters:
 • **Start autotuning –** the button calling the autotuning procedure,
 • **Mode** – the button for enabling / disabling the autotuning mode,
 • **Set point mode** – this parameter sets a source for set point value, only for autotuning process, takes options:
   ◦ **value** – defines constant value as a source for set point value,
   ◦ **from log. Channel –** defines logical channel as a source for set point value,
 • **Set point value** – sets set point value, only for autotuning process,
 • **Set point source** – defines logical channel that will be a source for set point, only for autotuning process,
 • **Feedback source** – defines logical channel that will be a source of feedback, only for autotuning process,
 • **Start mode** – parameter takes three options:
   ◦ **manual** – starts with the button,
   ◦ **from log. channel** – starts if set specified logical channel,
   ◦ **start-up** – starts if device is during start-up,
 • **Start source** – occurs when **Start mode: from log. channel,** defines logical channel that will be a source of start autotuning,
 • **Start triggering** – occurs when **Start mode: from log. channel,** defines the start edge of autotuning, the first edge activates and the second deactivates autotuning, takes options:
   ◦ **rising edge** – reacts to rising edge,
   ◦ **falling edge** – reacts to falling edge,
 • **Timeout** – determines the maximum duration of autotuning in hours (max 24h),
– **Controller parameters** parameter block - this block has the following parameters:
 • **Mode** – defines one of the three available modes: **PD**, **PI** or **PID**,
 • **P coefficient** – this parameter is always visible, allows to set the value of the proportional coefficient,
 • **I coefficient** – this parameter is available for the **Mode**: **PI** and **PID** and allows to set the integral coefficient,
 • **D coefficient** - this parameter is available for the **Mode PD** and **PID** and allows to set the derivative coefficient,
 • **Differentiated signal** - this parameter is available for the **PD** and **PID** mode and allows the selection of the option (see **7.127** and **7.128**):
   ◦ **feedback (measured)** - this option should be used when the value of the **Feedback source** which is the source of data coming from the controlled object is directly transmitted (no error recalculation at the controller input and no dead zone control) on the derivative signal, which enables a faster response of the device to fast changes of the controlled object,
   ◦ **error (deviation)** - this option should be used, when the value of the **Feedback source**, which is a source of data coming from the controlled object, is sent to the derivative signal after an earlier calculation of the error

on the regulator input and checking the condition of the regulator's activity due to exceeding the **Dead zone** range,

- **D (Inertia)** – an additional inertial element, is visible for **Mode: PD** and **PID**, this parameter applies to the differentiator,

$$Iner = \frac{e^{-\frac{t}{T}}}{T}$$

where:

Iner – response of the inertial block,

- **t** – successive moments in time,
- **T** – the inertia factor specified by the parameters:
  - **disable** – inertia is disabled,
  - **hard** – factor T=0,07,
  - **medium** – factor T=0,14,
  - **soft** – factor T=0,21,

The explanations of the parameters are shown in the **7.123** and in the **7.124**.

- **Dead zone** - this parameter determines the difference between the **Set point channel** value and the **Feedback channel** value, which will be included in further signal controlling the output of the controller.
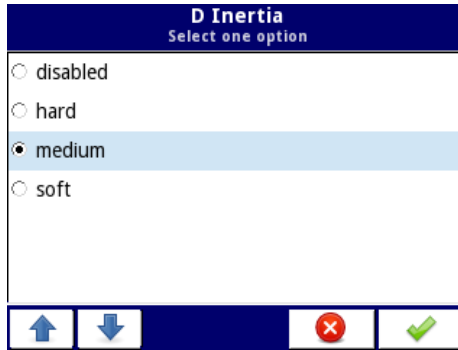
Fig. 7.123. Parameter D (Inertia) selection window
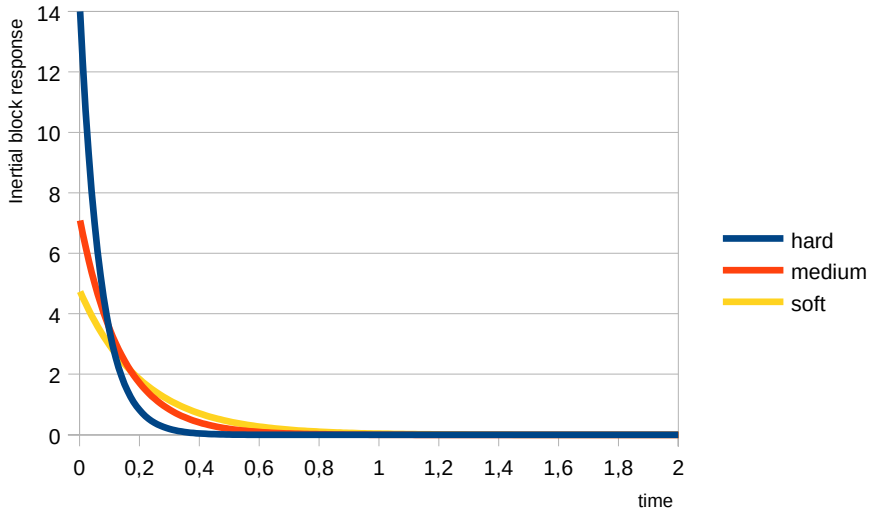
*Fig. 7.124. Responses of the inertial block depending on factor **T***

– **Controller running modes** parameter block – this block contains the following parameters:
  • **Mode** – defines when the controller will turn on:
    ◦ **ON** – always turned on,
    ◦ **OFF**– always turned off,
    ◦ **from log. channel** – turned on depending on the logical channel,
  • **Source –** defines logical channel, that will be a source of run, only visible for **Mode: from log. channel**,
  • **Triggering** – defines the event triggering the controller work, there are the following options:
    ◦ **high level** – triggering of start occurs when triggering channel is in high level,
    ◦ **low level** – triggering of start occurs when triggering channel is in low level,
    ◦ **rising edge** – triggering of start occurs when triggering channel is rising edge,
    ◦ **falling edge** – triggering of start occurs when triggering channel is falling edge,
– **Fuzzy Logic parameters** parameter block - this block has the following parameters:
  • **Mode** – turn on the additional fuzzy logic block to the controller, available options are:
    ◦ **ON** – always turned on,
    ◦ **OFF** – always turned off,
  • **Slew rate** – defines the slew rate, (see **7.125**),
  • **Unit** – defines the unit, available options are:
    ◦ **sec.** – defines the second,
    ◦ **min.** – defines the minute,

- ◦ **hour** – defines the hour,
- • **Hysteresis** – defines the value of hysteresis (see **7.126**),
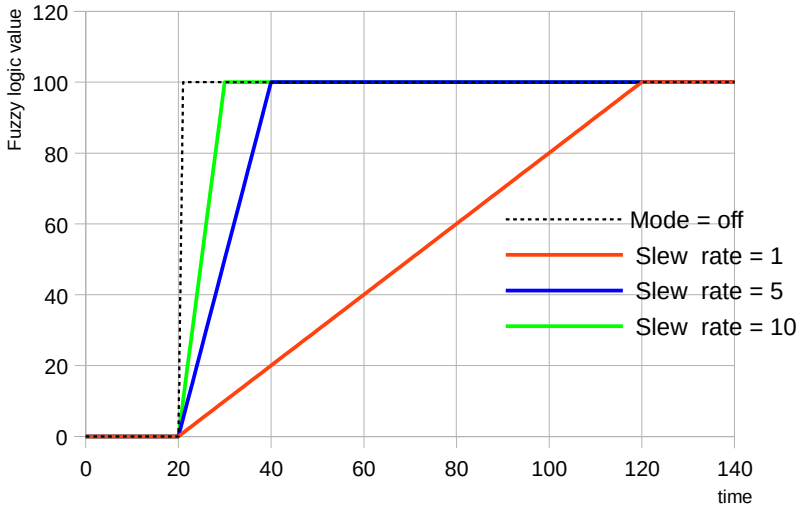
*Fig. 7.125. Change of fuzzy logic value according to **Slew rate** parameter*
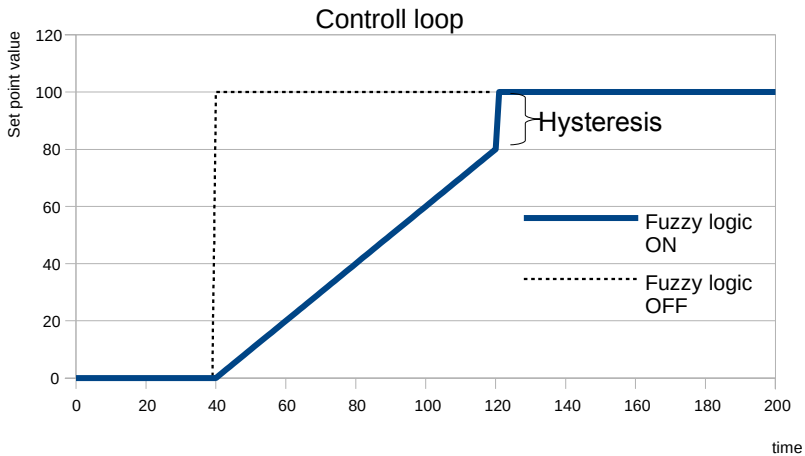
Fig. 7.126. Comparison of response of controller on change set point value with and without fuzzy logic block use

- – **Controller output** parameter block – this block contains the following parameters:

- • **Offset** - value of this parameter causes offset of the controller output value,
- • **Low output limit** – defines the low limit of the controller output signal value,
- • **High output limit** – defines the high limit of the controller output signal value,
– **Initial conditions** parameter block contains the following options:
    - • **Init type** – contains following options:
        - ◦ **disabled** – during the start of the controller its output is in 0 state,
        - ◦ **value** – allows to set the controllers output state during its start depending on **Low output limit** and **High output limit** parameters,
        - ◦ **from log. channel** – allows set up specified logical channel as a controllers output state at the start-up moment,
    - • **Init source** – defines logical channel, which is a source of initial output,
    - • **Initial output** – parameter is visible for **Init type: Value**; defines the controllers output state, where **0%** corresponds to **Low output** limit and **100%** corresponds to **High output limit**,

Note! After offsetting the output signal the output value is limited to the range set in **Low output limit** and **High output limit** parameters (see **7.128**),

7.127 shows the block diagram of a control process of an object with the **Controller** implemented in the device. Set the setting of the selected Controller to be connected to the **Logical channel** operating in the **Controller** mode. In this **Logical channel**, select the **Set point channel** and the **Feedback channel**, which store the data required to control the object. Respectively, the **Set point channel** contains a destination value of the process, while the **Feedback channel** includes the value of feedback coming from the object controlled. The **MultiCon CMC-N16** uses data collected from these channels and the corresponding Controller controls the object.
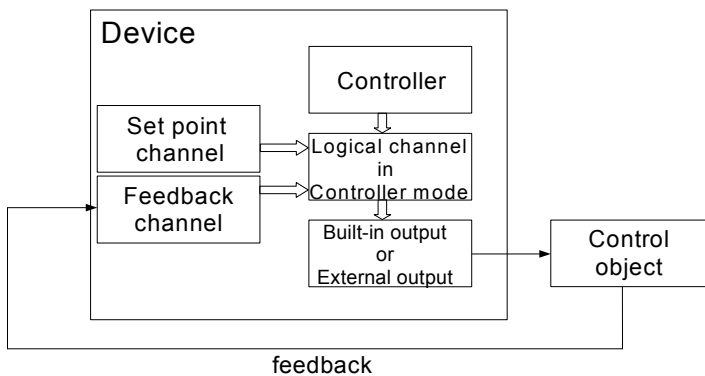
Fig. 7.127. Block diagram of the control loop of the object by **MultiCon CMC-N16**
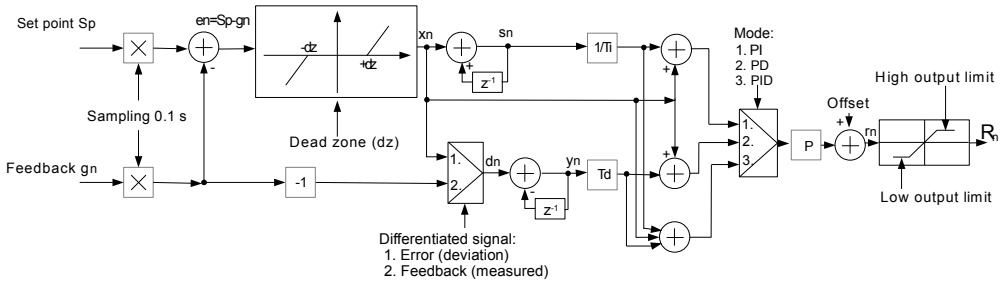
*Fig. 7.128. Block diagram of the **Controller** implemented in the device*

Formula for **Controller** output:

$$r_n = P \cdot \left[x_n + \underbrace{\frac{1}{I} \cdot \left(x_n + s_{n-1} \right)}_{integral\ component} + \underbrace{D \cdot \left(x_n - y_{n-1} \right)}_{differential\ component} \right]$$

The use of 8 independent Controllers in conjunction with the Set point value and feedback provides full possibilities limited only by available logical channels. It means, that the same Controller (e.g. controller No. 3) assigned in several logical channels, with different input data: Set point and Feedback channels it can be used to control several objects at the same time.

PID controller configuration
        The PID controller configuration can be done in two ways: manual and automatic. In manual mode, the user defines all factors and additional elements of the controlers (fuzzy logic, **Inertia**, output limitations, **Dead zone**) at its own discretion. The settings chosen by the device in automatic mode can also be improved and fine-tuned. It is possible to disable each block of the PID Controller and additional blocks.
In automatic mode the device is in the process of autotuning, using the Ziegler-Nichols method, calculates and tunes parameters P, I and D of the controller loop. Additional parameters (fuzzy logic, **Inertia**, output limits, **Dead zone**) are not in this mode configurable and the user must select them for his own applications. The Autotuning process is described later in this **Chapter**.

General configuration diagram is presented in the **7.129**.

*Fig. 7.129. General configuration diagram of the **Controller** using autotuning*

The Controller 1 has been assigned to various logical channels creating independent control loops. It should be noted, that a situation in which both: the controller in autotuning parameters and logical channel can be assigned the same logical channels for **Set point** value and **Feedback** (in the **7.129**, logical channels SP1 and FB1). At start-up autotuning of Controller 1 all logic channels in Controller mode with assigned controller 1 (logic channel 1, 2, n) will use the same autotuning parameters (SP1 and FB1) so they will generate the same value at the output. After the autotuning end logic channels in controller mode with assigned Controller 1 will go to normal operation of the regulator creating independent control loops.

There are 2 methods of configuring the controllers:
  – through the menu Controllers in the Device Configuration menu (see **7.130**),
  – directly in the Logical channel in the Regulator mode by pressing the button Source configuration (see **7.131**).

The appearance of the Regulator configuration window in both cases looks the same.

*Fig. 7.129. General configuration diagram of the **Controller** using autotuning*

The Controller 1 has been assigned to various logical channels creating independent control loops. It should be noted, that a situation in which both: the controller in autotuning parameters and logical channel can be assigned the same logical channels for **Set point** value and **Feedback** (in the **7.129**, logical channels SP1 and FB1). At start-up autotuning of Controller 1 all logic channels in Controller mode with assigned controller 1 (logic channel 1, 2, n) will use the same autotuning parameters (SP1 and FB1) so they will generate the same value at the output. After the autotuning end logic channels in controller mode with assigned Controller 1 will go to normal operation of the regulator creating independent control loops.

There are 2 methods of configuring the controllers:
  – through the menu Controllers in the Device Configuration menu (see **7.130**),
  – directly in the Logical channel in the Regulator mode by pressing the button Source configuration (see **7.131**).

The appearance of the Regulator configuration window in both cases looks the same.

CONTROLLERS MENU   LOGICAL CHANNEL MENU

```
┌─────────────────────────────┐          ┌─────────────────────────────┐
│       CONTROLLER 2          │          │         CHANNEL 1           │
│                             │          │                             │
│  Autotuning coeff.          │          │  Mode: Controller           │
│        SP4                  │          │        SP1                  │
│        FB4                  │          │        FB1                  │
└─────────────────────────────┘          │  Controller num. 2          │
                                          │                             │
step 1                          step 2    │   Config.                   │
                                          └─────────────────────────────┘
┌─────────────────────────────┐
│       CONTROLLER 1          │
│                             │          ┌─────────────────────────────┐
│  Autotuning coeff.          │          │         CHANNEL 2           │
│        SP3                  │          │                             │
│        FB3                  │          │  Mode: Controller           │
└─────────────────────────────┘          │        SP2                  │
                                          │        FB2                  │
                                          │  Controller num. 2          │
                                          │                             │
                                          │   Config.                   │
                                          └─────────────────────────────┘

                                          ┌─────────────────────────────┐
                                          │         CHANNEL 3           │
                                          │                             │
                                          │  Mode: Controller           │
                                          │        SP3                  │
                                          │        FB3                  │
                                          │  Controller num. 1          │
                                          │                             │
                                          │   Config.                   │
                                          └─────────────────────────────┘
```

*Fig. 7.130. Configuration diagram of the **Controller** using Controller Menu*

LOGICAL CHANNELS MENU
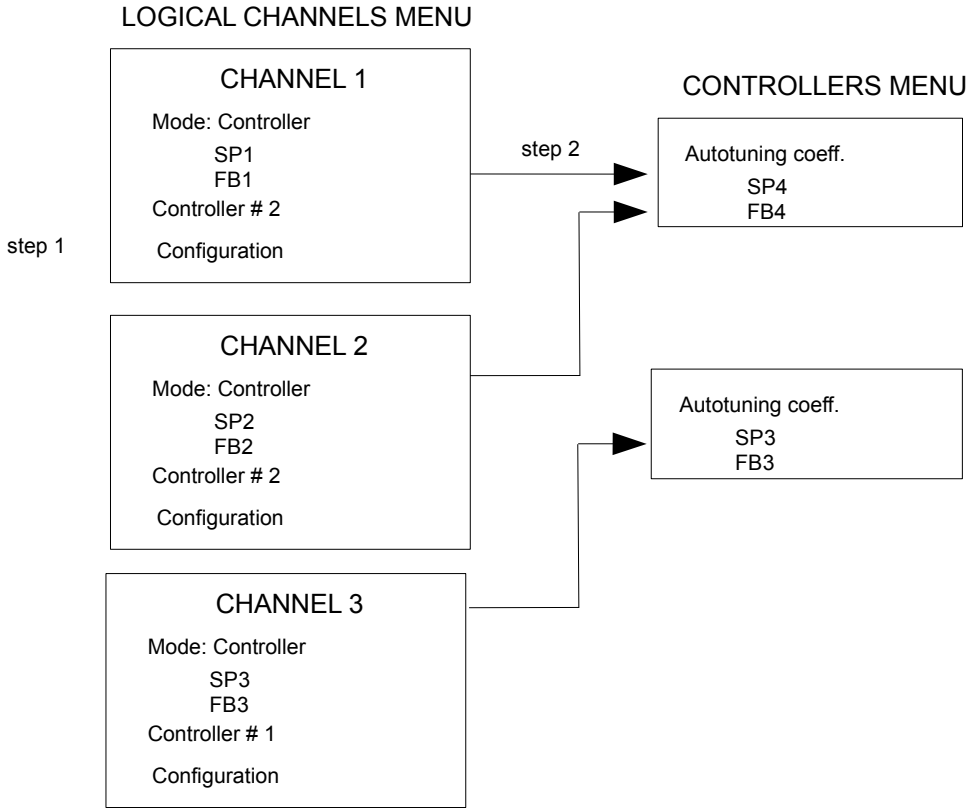
*Fig. 7.131. Configuration diagram of the **Controller** using Logical channels Menu*

The controller output signal

In the **7.132** an exemplary diagram of the output values of the control loop is presented. The diagram shows the double limitation of the control loop supplied to the object. The first limitation with a black line results from the output parameters of the Controller (logic channel output value in the controller mode, see **7.127**), and the second limit marked with a red line results from the parameters built-in/external outputs (value transferred to controlled object, see **7.127**).

The output range of the controller should be within the limit of range of the built-in output.
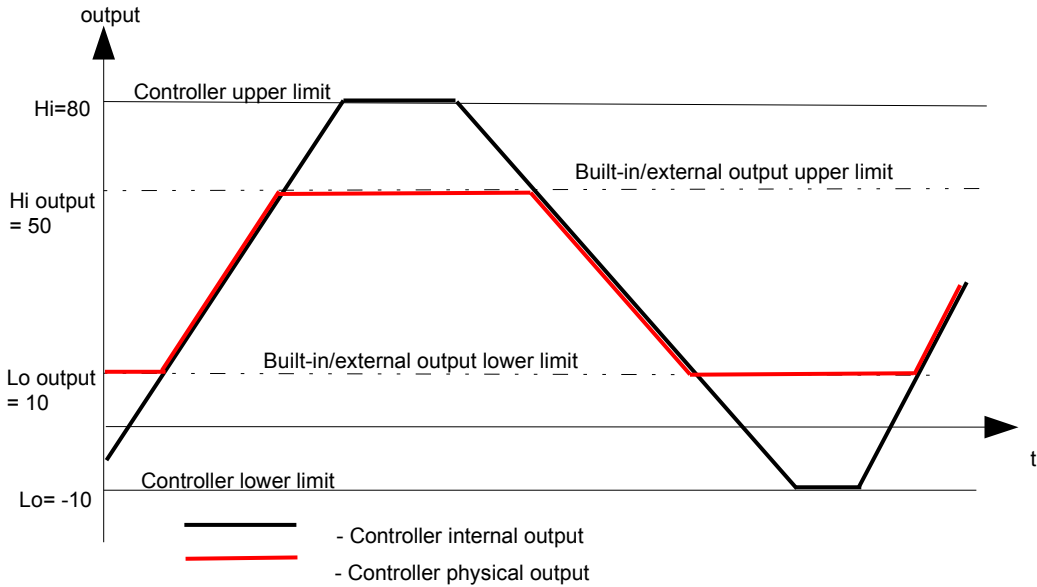
Fig. 7.132. Control loop output limitation

Fuzzy logic block

Step changes in the Set point value can be mitigated by the use of the fuzzy logic block, which inserts speed control of the Set point value change. Thanks to its application the user may dictate how quickly a fuzzy value approximates the target value in the process (**Set point channel**).

This feature is particularly useful in systems where too rapid change in the Set point value and, as it is possible in such a situation, the control signal, could lead to unwanted behaviour on the part of an object, for instance: destruction of the product being manufactured. It is also possible to set the minimum difference between the Set point value, and the fuzzy logic setpoint, beyond which the controller step changes the fuzzy setpoint to the Set point value. This is defined by the **Hysteresis** parameter.

The inertial block

This is used to smooth the progress of the derivative block response, in other words to eliminate overly dynamic changes in the output signal's value caused by the presence of measurement process noise. The use of the inertial block lengthens response time, but provides protection against over-regulation that could lead to failures in the controlled system.

Dead zone

The insertion of this parameter provides the possibility of "anesthetising" the control system to small deviations in the measured value from the setpoint. The **Dead zone** parameter specifies the minimum deviations, to which the controller should react. The use of the **Dead zone** is justified in systems, for which oscillation around the setpoint is a natural

behaviour.

Autotuning

The autotuning process is divided into three phases and after their completion the device automatically switches to operation depending on **Mode** in Controller running modes parameter block (see **Chapter 7.9.7 Logical Channels - Controller mode**). The duration of the auto-tuning process depends on the properties of the controlled system. When the device is during the autotuning process, a visible icon appears on the information bar 🔺.

During normal operation of the controller, when the user wants to carry out the autotuning process and the **Mode** is **from log. channel**, before starting (after triggering autotuning) a message will appear about the ongoing adjustment process and confirmation this message will take user to the autotuning process. If the controller is turned off, this message will not appear.

The Start Autotuning button on a gray background means, that this controller has not been connected to the logical channel, therefore it is inactive. When the user wants to activate it, must define a logical channel in the controller mode with the appropriate controller number.

> ⓘ **Attention!** During autotuning, is not advisable to regulate the actual technological processes, because autotuning can damage the products produced in this process.

The autotuning process may begin when the distance condition is met. Message about this error is shown in **Fig. 7.133**.
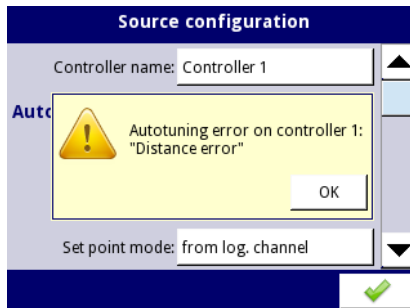
Fig. 7.133. View of Distance error warning

In determining the condition of the distance error, the internal calculation of the set value is considered based on the user set point value and the measured value:

Sp$_{iner}$ = Fb+(Sp – Fb) * 0,5

where:

Sp$_{iner}$ – internal set point,
Fb –feedback (measured value),
Sp – set point value.

Autotuning, which is already under way, is a very sensitive process. Therefore, pay attention to events that may interrupt the tuning of settings. These events include:
1) Device power failure,
2) Exceeding the time of autotuning (max 24h),
3) Measurement error (e.g. sensor failure).

Autotuning is triggered every 30 seconds. During the autotuning process the possibility to save the configuration is blocked (see **7.134**). In the case of several autotuning processes taking place simultaneously, the save of new PID coefficients will be possible after finishing the last autotuning process.
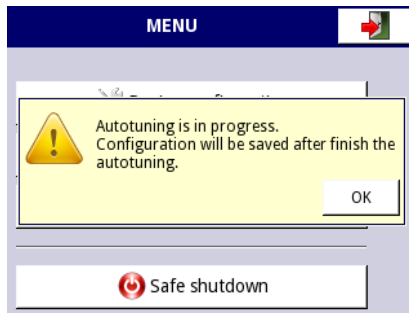
Fig. 7.134. Window with information about blocking save configuration during autotuning process

After the successful completion of the autotuning process, in the **Save mode: ask before save** (see **Chapter 7.8. GENERAL SETTINGS**) parameter of **PID autotuning parameters** a message will appear (**7.135**) with information about new PID coefficients, after which the device switches to the standard configuration save window. With new PID coefficients, the device goes to work with new coefficients. When automatically saving the configuration, there is no window with information about the completed autotuning with new PID coefficients and the question about saving the configuration.
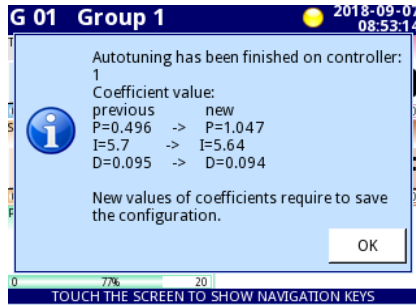
Fig. 7.135. Example window with new PID coefficients

> ℹ️ If the device will turned off during the autotuning, restarting will show the message about the interrupted autotuning process.

> ℹ️ If the device will be turned off before the configuration is saved with the new PID coefficients, after the device restarted, a request to save the configuration with new PID coefficients will occur.

Autotuning starting during device start-up
The device automatically saves new PID coefficients, if the parameter **Save mode** in the parameter block **Autotuning PID coefficients** is set to **automatic save** in the **General Settings**. This will automatically save the whole configuration of autotuning before the device goes into normal operation.
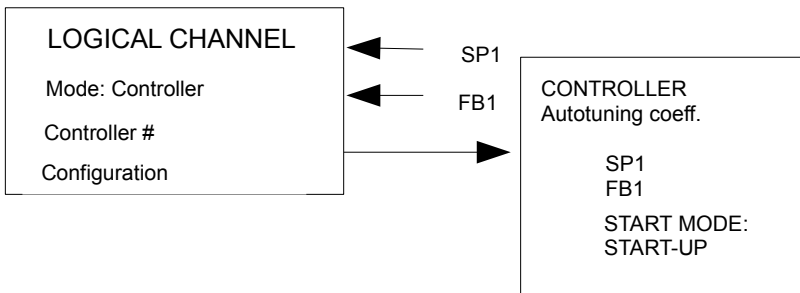
Fig. 7.136. Settings of autotuning automatic start

In order to automatically switch the device to normal operation, after the autotuning process, it is necessary in the **Controller running modes** parameter block set:
  1)   Mode: ON,
       or
  2)   Mode: from log. channel and **Triggering = high/low level**.

7.14.2. Examples of Controller configurations

7.14.2.1. Application of the Controllers

See also: **Chapter 7.14. CONTROLLERS**.

Task:
The task is to configure the controller which will control (via RS-485) the heater in order to obtain a constant object temperature. Logical channels with a set point value and temperature from the PT100 sensor are connected to the controller. After pre-analysis of the sensor type and the transmittance of the object, we set the controller coefficients to the following values: P=0.3, I=0.5, D=0.02. We do not want the controller to respond to a deviation lower than 2ºC. On controller's output set 0-20mA range because that is the range of the converter we use to control the heater in this task.

Solution:
First configure the controller, then connect the Pt100 sensor and Current converter to the device. An example of this connections in presented in **7.137**.

Fig. 7.137. The connection scheme for FUN4 hardware configuration in RTD mode and Modbus MB1 port

In the first step we define **Controller** settings:
- – touch the screen and press **Menu** button,
- – press the **Device configuration** button,
- – enter the **Controllers** menu,
- – using the arrows in the top navigation bar, select any **Controller** such as **1**,
- – in the **Controller name** parameter write **Controller 1**,
- – **Mode** parameter set as **PID**,
- – in the **Dead zone** parameter write **2**,
- – for **Controller parameters** parameter block:
  - • in **P coefficient** parameter write **0.3**,
  - • in **I coefficient** parameter write **0.5**,
  - • in **D coefficient** parameter write **0.02**,
  - • **Differentiated signal** parameter set as **error (deviation)**,
- – for **Controller output** parameter block:
  - • in **Offset** parameter write **0**,
  - • in **Low output limit** write **0 mA**,

- in **High output limit** write **20 mA**,

After leaving the configuration save the settings. In next steps we should set Logical channels (examples of logical channel configuration are presented in **Chapter 7.9.11. Examples of Logical Channels configuration**, especially **7.9.11.6. Application of Logical channel in the Controller mode**), set Modbus for a Slave device (examples of Modbus configuration are presented in **Chapter 7.16.4. Modbus - Example of Modbus protocol configuration in the device**, examples of external outputs configuration which will communicate with a Slave device via RS-485 are presented in **Chapter 7.12.4. Examples of external output configurations**).

7.14.2.2. *Application of the Controllers in cooperation with heater which is controlled by OC-type output*

See also: **Chapter 7.14. CONTROLLERS** and **Chapter 7.11.3. Built-in outputs - OC-type relays**.

Task:
　　　　The task is to configure the controller which will control the heater using the OC-type output in order to obtain a constant object temperature. Logical channels with a set point value and temperature from the thermocouple type K sensor are connected to the controller. Due to the fact that choosing a PID controller is not a matter of the task, we can use the following demonstration values: P=5, I=10, D=1. We do not want the controller to respond to a deviation lower than 2ºC. On the controller's output set 4-20mA range because that is the range of the converter we use to control the heater in this task.

Solution:
　　　　First configure the controller, then connect the type K thermocouple and Current converter to the device. Examples of these connections are presented in **7.138**.
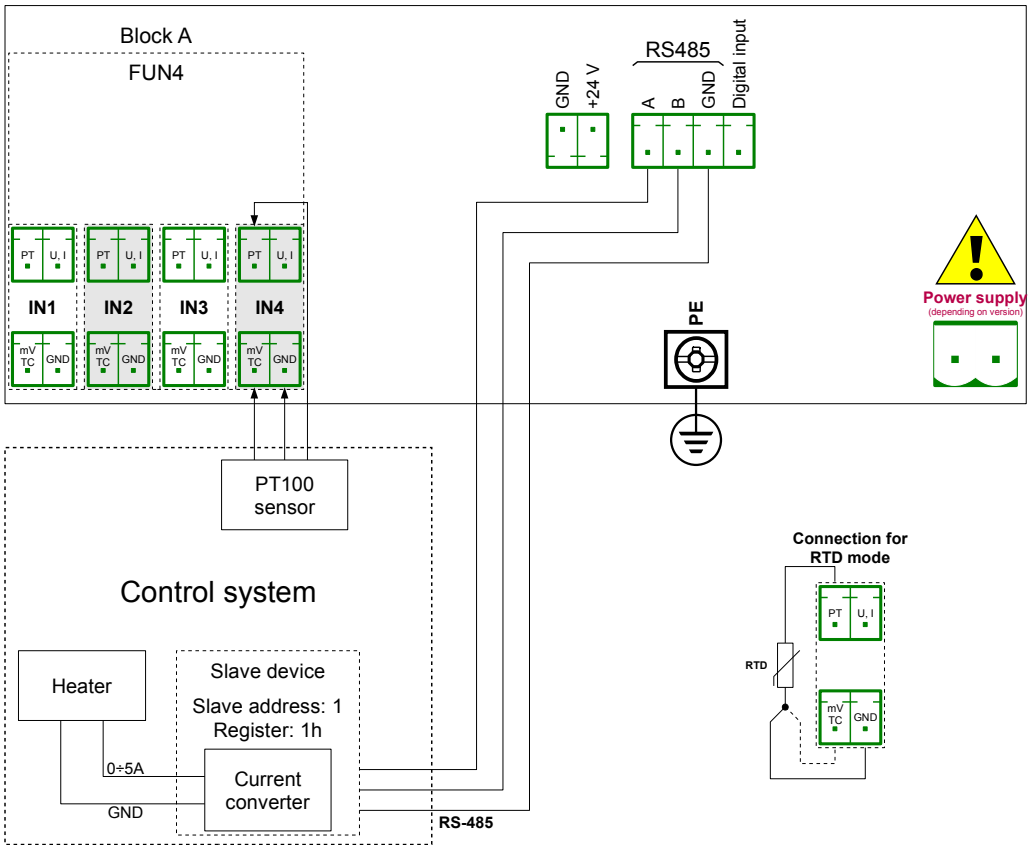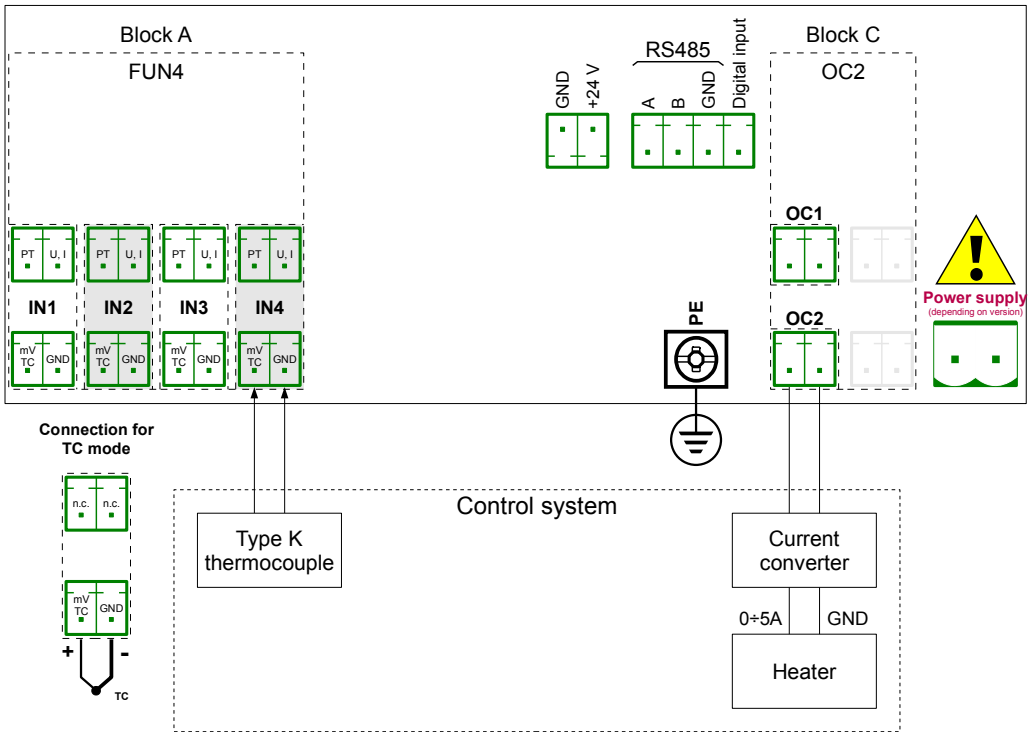
Fig. 7.138. The connection scheme for FUN4 hardware configuration in TC mode and OC2 hardware configuration

A logical channel operating as a PID controller generates an "analog" output signal. This signal can be limited to a range suitable for an output used. On the other hand all built-in outputs have settings allowing to fit them to a source signal (see **7.11.3. Built-in outputs - OC-type relays** or **7.11.4. Built-in outputs - Current output**). It is the user's responsibility to choose a correct limitation of a PID output (as well as other PID settings), but it is important to set a limitation range wide enough to obtain smooth controlling of the actuator, and narrow enough to make PID as fast as possible.

We can assume that the PID output range is set as 0-100. To set this, the user should enter the **Controller** menu, select a requested controller and in the **Controller output** parameters block set:

> **Offset**: 0,
> **Low output limit**: 0,
> **High output limit**: 100,

Other PID controller settings depend only on the application.

Let's assume that the user has to use the TC signal (K) for temperature measurement (FUN4 hardware configuration installed in block A) and the OC-type output driver (OC2 hardware configuration installed in block C) to control a heater keeping a set

temperature of the object. Let's assume that the temperature setpoint can be set between 120 and 200 °C.

To realise this task the user needs at least 3 logical channels and 1 set of Controller settings.
Group 1 – Heater,
Logical channel 1 – Measurement of temperature,
Logical channel 2 – Setpoint of temperature,
Logical channel 3 – PID controller,
Controller 1 – settings of PID controller which runs in channel 3,

There are a few basic, well known methods described in the literature to choose P, I and D parametets optimal for the controlled process. If theuser is not familiar with these methods, also some exemplary values can be used, suitable for demonstration of the PID work process controlling slow processes (like heating 1liter of water using a 1000 Watt heater).

> **Mode**: PID,
> **Dead zone**: 0,

for **Controller parameters** block parameters:
> **P Coefficient**: 5,
> **I coefficient**: 10,
> **D coefficient**: 1,
> **Differentiated signal**: feedback (measured),

for **Controller output** block parameters:
> **Offset**: 0,
> **Low output limit**: 0,
> **High output limit**: 100,

for **Initial conditions** block parameters:
> **Init type**: No init (internal state 0),

The logical channels used for this example should be set as follows:

for **Logical channel 1** :
> **Name**: Temperature,
> **Mode**: Hardware input,
> **Source**: Inp. A4 (we assume that the built-in input is connected and set properly for TC type K),
> **Precision**: 0,
> **Graph Lo**: 0,
> **Graph Hi**: 300,

all the other parameters should be on default settings,

for **Logical channel 2**:
> **Name**: Desired temp.,
> **Mode**: Set point value,
> **Unit**: °C,
> **Edit Button**: enabled,
> **Edit range low**: 120,

Edit range high: 200,
Precision: 0,
Graph Lo: 0,
Graph Hi: 300,

all the other parameters should be on default settings,

for **Logical channel 3**:
Name: PID controller,
Mode: Controller,
Controller num: 1 PID:"Controller 1",
Set point channel: Log.ch. 2:"Desired temp",
Feedback channel: Log.ch. 1:"Temperature",
Precision: 0,
Graph Lo: 0,
Graph Hi: 300,

all the other parameters should be on default settings,

Above channels can be assigned to one free group:
Group 1:
Group: enabled
Name: Heater
Defined mode: controllers
Slot 1: Log.ch.1:"Measurement of temperature"
Slot 2: Log.ch.2:"Setpoint of temperature"
Slot 3: Log.ch.3:"PID controller"
Slot 4: disabled
Slot 5: disabled
Slot 6: disabled
Controllers view -> Mode: enabled

Above parameters set in this way allow to present the process in the form of view appropriate for controllers.

The last thing to do is set of the OC-type output.
Enter the **Device configuration** menu, then **Built-in outputs** and select Output:
Out.C2 : OC. In this output set the parameters as follows:
Mode: PULSE
Source: Log.ch. 3:"PID controller",
Level mode: value,
Lower level: 0,
Upper level: 100,
Alarm level: 0   / heater off if any error occurs/,
Period: 10sec,
Minimum ON-Time: 0,
Minimum OFF-Time: 0.

7.15. GROUPS

General settings
Logical channels
Build-in inputs
Build-in outputs
External outputs
Profiles/timers
Controllers
Groups
Modbus
Network settings
Access options
E-mail notifications

MENU ➡ Device configuration ➡
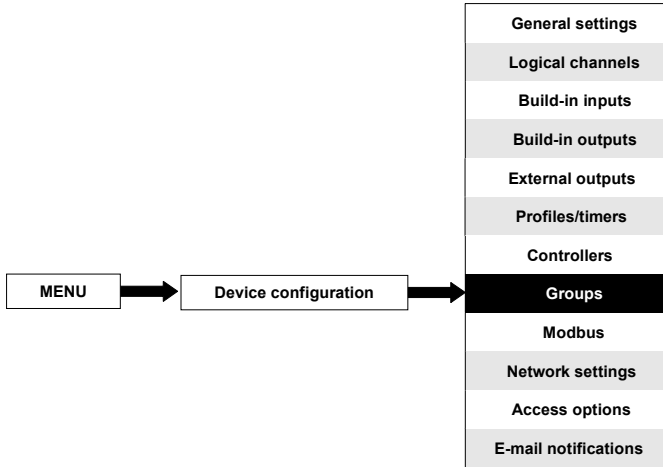
A **Group** is a set of 1 to 6 **Logical channels**, gathered together in one measurement screen to improve the readability of the displayed values. For a definition of **Groups**, see **Chapter 5. INTRODUCTION TO MultiCon CMC-N16**. If the device has a license to record data, it is possible to record data of selected **logical channels** in each **Group**.

7.15.1. Groups - General settings
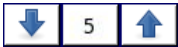
The arrows in the upper right-hand corner of the screen allow you to switch between available **Groups**. The middle button allows you to directly select a specific **Group** from the list.

The main parameter of the **Group** is:
– **Group** - any group can be:
  • **disabled** - after selecting this option, the remaining parameters of the group are invisible and the group itself is not visible on the measurement screen,
  • **enabled** - activates the group.

7.15.2. Groups - Logging

If the device has a license to record data, it is possible to record data of selected **logical channels** in each **Group**.

Approximate time for which the device's internal memory is sufficient when saving the specified number of channels and the time interval between samples, is given in the **Tab. 7.7** for a device with 2 GB internal memory and in the **Tab. 7.8** for a device with 4 GB internal memory.

memory 2 GB	every 1 s	every 10 s	every 60 s
60 channels	20 days	6 months	3 years
48 channels	30 days	8 months	4 years
24 channels	50 days	15 months	7 years

Tab. 7.7 Approximate time for which the internal 2 GB memory will be enough

memory 4 GB	every 1 s	every 10 s	every 60 s
60 channels	46 days	14 months	7 years
48 channels	70 days	18 months	9 years
24 channels	115 days	35 months	16 years

Tab. 7.8 Approximate time for which the internal 4 GB memory will be enough

Each **Group** has its own data logging options and the device can log **10 independent Groups of Logical channels** at the same time with the sam,ple period from 0.1 sec. to 24 hours.

(i) Logging of the data in the device is hardware limited, so the producer recommends to limit the logging to less than 200 samples per second (e.g. at the maximum sampling frequency of 0.1 sec. The user should not log more than 20 Logical channels at one time). Failure to comply with these restrictions may cause the device to slow down.

Data logging (base) can be carried out continuously or be triggered **conditionally** based on the value in the indicated logical channel, e.g. when some process of heating the element in the furnace is running, or when the engine is running.

In addition, during the operation of the base **data logging** with the specified interval of the sampling for a given **Group**, you can start **alternative registration** triggered from selected logical channel with a different set sampling period. Such a solution allows you to record data in a special way in the event of an event on the site, e.g. an alarm condition (e.g. setting an alternative logging with a shorter one sampling period), or idling (e.g. setting an alternative registration with a larger one sampling period).

7.15.2.1. Groups – Logging parameters

Logging options parameter block is always visible, however, it is active only in a device that has a recording license (for more information about the recording license, see **Chapter 7.5. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION**). If no license is entered, the message "Available after entering licence:" is displayed.

With **Logging options** parameters, the user can set the triggering mode of data recording, sampling time or description of recording.

The logging parameters include:
The parameter block **New Recording File - Quick Access** is visible when the **Mode** in the parameter block **Logging Options** takes values **always** or from the **logical channel** and contains:

- − **Create a New File** - a button to create a new recording file. When you press it, a window will appear on the screen in which you must confirm or reject the command to create a new recording file,
- − **Description** - field where you can add a description to the created recording files. The description is visible when browsing the recording files, which makes it much easier to find them; the dynamic variables shown in **7.4** can be used for description.

The include:
- • **#n** – the counter value,
- • **#D** – the current day,
- • **#M** – the current month,
- • **#Y** – the current year,
- • **#h** – the current hour,
- • **#m** – the current minute,
- • **#s** – the current second.

Each of the mentioned dynamic variables in the description can be used only once. The next occurrence of this character combination will be interpreted as text.

Example of **Description**:

When entering the text in the
Description field: **Batch No. #n, #D.#M.#Yr. #h:#m:#s**

we get for a specific batch and
time of file creation: **Batch No. 23, 8.10.2019 10:15:35**

For a description of the **Auto Log Creation** parameters, see **Chapter 7.15.2.2. Groups – On-demand Logging**.

Logging options parameter block contains:
- – **Mode** – defines how to trigger group recording:
  - • **inactive** - registration of a given **Group** is inactive,
  - • **always** - continuous registration of a given **Group**,
  - • **from logical channel - activates an additional parameter Trigger Source**, allowing for conditional activation of the recording - when the logical channel value **> 0** the recording is active, otherwise (logical channel value **≤ 0**) the registration is **inactive**,
- – **Trigger Source** - only for the **Mode: from the logical channel** - in this parameter the user selects the appropriate logical channel that is the source of triggering the **Group** recording - when the logical channel value **> 0** the recording is active, otherwise (logical channel value **≤ 0**) the registration is inactive,
- – **Base period** and **Base unit** - determine at what time interval the next data sample is to be recorded in a given **Group**; the following options are available:
  - • **unit: second ->** duration form **0.1** to **3600 sec.**,
  - • **unit: minute ->** duration form **0.1** to **1440 minute**,

- • **unit: hour ->** duration form **0.1** to **24 hour**,
- – **Counter - specifies the number of recording files created.** You can enter any value here at any time. Creating another recording file will increase the counter value by 1
- – **Alternative mode** - this mode allows the user to log data in a special situation where a deeper analysis is required (for example in the critical state of an object), this parameter has the following options:
  - • **disabled** - alternative logging of selection Group is disabled,
  - • **from logical channel** - this option activates a new parameter - **Triggering source** which enabled logging the data when the value of **Triggering source > 0**,
- – **Alternative source** - this parameter is visible for **Mode=from log.channel**, when the value of this source **> 0** then the **data logging** for an alternative mode is **enabled**,
- – **Alternative period and Alternative unit** - these parameters set the duration of the sample of data logging for an alternative mode, these parameters have the following options:
  - • **unit: second ->** duration form **0.1** to **3600 sec.**,
  - • **unit: minute ->** duration form **0.1** to **1440 minute**,
  - • **unit: hour ->** duration form **0.1** to **24 hour**.

The parameters block **Log file auto-creation** include:
- – **Mode** - specifies the source of the signal to create a new recording file, available options are:
  - ⋅ **inactive** - creating new recording files is disabled,
  - ⋅ **from the logical channel**[*] - new recording files will be created depending on the value in the indicated logical channel; the maximum number of recordings the device can set is 300. Once this limit is reached, new recordings will not be created and the last created will be continued.
- – **Source**[*] - parameter that specifies the trigger source for creating a new recording file.
- – **Trigger**[*] - defines the trigger slope of a new recording file, available options are:
  - ⋅ **rising slope** - the trigger signal will occur when the logical channel value changes from 0 to 1,
  - ⋅ **descending slope** - the trigger signal will occur when the logical channel value changes from 1 to 0.

ⓘ When changing the parameters related to the recorded **Group** (e.g. changing the value of the recording parameters, changing the parameters of the display options, changing the parameters of logical channels attached to the recorded Group, etc.), a new **recording file** is automatically created. If you turn off the device or change other parameters not related to the recorded **Group**, no new recording file is created.

ⓘ When saving or restoring the last saved configuration and loading the configuration from the flash memory, the recording is stopped, which will be manifested by the lack of samples in the recording files.

ⓘ A detailed description of the structure of the data saved during the recording can be found in **Chapter 8.6DATA FORMAT**.

---

[*] parameter available for devices with flash version 1.08 or higher

Configure the device to create a new recording file:
- join the selected **Logical channel** to the **Group**, see **Chapter 7.15.4. Groups - Channels**,
- activate recording by setting the **Recording Options** for the **Group** to which the **Logical channel** has been added (setting these parameters will allow you to create a new recording file for a given **Group** once during the configuration saving),
- for on-demand recording, set the parameters of **Recording file creation options**. Setting these parameters will allow you to create further recording files for a given Group, triggered from the value of the designated logical channel, see **Chapter 7.15.2.2. Groups – On-demand Logging**,
- after leaving the menu, accept the changes by saving the configuration. While saving the configuration, a new recording file is created - the received data recording files can be saved to a removable flash drive, see **Chapter 7.4. FILES MANAGEMENT** or upload using HTTP protocol, see **Chapter 8.7. DIRECT ACCESS TO LOG FILES USING HTTP PROTOCOL**.

7.15.2.2. Groups – On-demand Logging

On-demand Logging allows to generate a new recording file by means of a **button** directly placed in the settings of the given **Group** (see parameter group **New recording file - Quick Access**) or depending on the value of the designated **Logical Channel** (see parameter block **Recording file creation options**). The designated **logical channel** can assume various functions: it can be a button directly on the measurement screen, which makes it very easy to generate new recording files at the touch of a button, it can be a Profile/Timer and e.g. generate new recording files with fixed days and hours making the device a practical tool for generating daily or monthly reports. The designated logical channel can also be an alarm condition that generates new files when the alarm is triggered.

Limitations of On-demand Logging:
- For a device with a **3.5"** display, the functionality of creating a new file from the logical channel is only possible from flash version 1.09 onwards. There are no restrictions for a **5.7"** display device.
- New recording files can be created **no more often than every 3 minutes**; a logical channel triggering new recording files can trigger all recorded groups in one moment, but another successful trigger of a new recording file from this logical channel or another in other recording groups can occur after 3 minutes.
- The device can generate up to **300** recording files, after exceeding this number, the recordings are continued in the current file. In order to create more files, existing files in the device must be deleted.

7.15.2.3. Groups - On-demand Logging Example

Task:

In the task, temperature recording from four PT100 sensors (4-wire) located in the warehouse must be started. The recording is to take place every 1 minute, but if the temperature of 28°C is exceeded on at least one sensor, the alarm state is to be activated in the form of an acoustic signal. The alarm state is to force an alternative recording with data sampling every 1 sec. Every day at 8 o'clock a new recording file is to be generated with a description containing the number of the next recording file, date and time of generation. The

display is to take place on the device screen in one measurement window in value mode. The task is shown in **7.139**.

<u>*Solution*</u>:

In the task, a device with a built-in **RT4** module should be used to measure temperature from 4 PT100 sensors simultaneously. Before starting the temperature measurement, configure the device and then connect the PT100 sensors to the device accordingly. An example of the connection method is shown in **Fig. 7.81**.

In the first step, we move on to configuring **Logical Channels**. For sensor 1 temperature reading:
- touch the screen and press the **Menu** button,
- press the **Device Configuration** button,
- enter the **Logical Channels** menu,
- using the arrows in the top navigation bar, select any **Logical Channel**, e.g. **30**,
  - enter **sensor 1** in the parameter **Name**,
  - set the **Mode** parameter as **Hardware Input**,
  - set the **Source** parameter as **A1 Input: RTD** (see **Fig. 7.74**),
  - go to the **Source Configuration** submenu:
    - set the Mode parameter as **Pt100 4-wire**,
    - in the parameter **Bottom Limit**, enter **-100°C**,
    - in the parameter **Upper Limit**, enter **600°C**,
  - for the **Display** parameter block :
    - set the parameter **Format** as **Numeric**,
    - set the **Precision** parameter to **0.0** because the accuracy of the PT100 sensor allows a value with 1 decimal place to be displayed,
    - in the parameter **Lower Value** of the graph, enter **15°C**,
    - in the parameter **Upper Value** of the graph, enter **30°C**.

The other channels are set in the same way:
Logical Channel 31: **sensor 2**,
Logical Channel 32: **sensor 3**,
Logical Channel 33: **sensor 4**.

To detect an alarm condition use the mathematical function **Con.One X[i]>Y,** e.g. we will set the logical channel **34**:
- enter the **Alarm State** in the **Name** parameter,
  - set the **Mode** parameter as **Mathematical Function**,
  - select **Function**: **Con.One X[i]>Y**,
    - in the **Source X** parameter, select channels **30, 31, 32, 33**,
    - in the **Error Handling of X Source** parameter set: **the errors passed on to the result**,
    - in the **Y Source Type** parameter: select the value and set to **28°C**,
  - for the **Display** parameter block:
    - set the **Format** parameter as **Binary**,
    - set the activation status text as **ALARM**,
    - set the deactivation status text as **OK**,

- ◦ in the parameter **Lower Value** of the graph, enter **0**,
- ◦ in the parameter **Upper Value** of the graph, enter **1**.

To generate a new recording file on a daily basis we will use **Profile/Timer**; for this purpose we need to select a logical channel e.g. **35**:
- − enter the **Profile** in the **Name** parameter,
- − set the **Mode** parameter as **Profile/Timer**,
- − set the **Source** parameter as **P/T 1 "Profile 1"**,
- − press the **Source Configuration** button,
  - • set the **Mode** parameter **at a preset time**,
  - • set the **Trigger Times** parameter as:
    - ◦ set **Months, Days of the month, Days of the week** as **All**,
    - ◦ Set the **Hours** to **8**,
    - ◦ **Minutes** to **0**,
    - ◦ **Seconds** to **0**,
    - ◦ set **Trigger Mode** to inactive,
  - • enter **0** in the parameter **Resting Value**,
  - • press the **Interval List** button:
    - ◦ press green plus in the bottom left-hand corner,
    - ◦ the **Interval** parameter should be set to **1**,
    - ◦ set the **Duration** parameter to **1**,
    - ◦ set the **Unit** parameter to **second**,
    - ◦ set the **Interval Type** parameter to **fixed value**,
    - ◦ set the **Target Value** parameter to **1**,
  - • set the **loop** parameter to: **inactive**
- − for the parameter block **Display**:
- − set the **Format** parameter as Numeric,
  - • in the parameter **Lower Value** of the graph, enter **0,**
  - • in the parameter **Upper Value** of the graph, enter **1**.

In the next step, define the **Group** of the created logical channels to display them in one window on the device screen.
For that purpose:
- − enter the **Group** menu, activate e.g. **Group 1** if it is switched off,
  - • set the parameter block **New Recording File - Quick Access**,
  - • in the **Recording Description** parameter, enter: **No. #n, #D.#M.#Yr. #h:#m:#s**,
  - • for the parameter block **Recording Options** (this parameter block is only available for devices with a data registration license):
    - ◦ set the **Mode** parameter to **always** - we want it to keep recording until the configuration is changed to turn off recording,
    - ◦ In the **Basic Period** parameter, enter **1**,
    - ◦ set the **Base Unit** parameter as **minute** (sampling a group of channels every 1 minute),
    - ◦ set the parameter **Alternative Mode** to **from the logical channel**,
    - ◦ set the parameter **Alternative Source** to **the logical channel 34**,

- ◦   set the **Alternative Period** and **Alternative Unit** to **1 second,**
- •   for the parameter block **Recording File Creation Options**, set:
  - ◦   set the **Mode** parameter to **from the logical channel**,
  - ◦   set the **Source** parameter to logical channel **35**,
  - ◦   set the **Triggering** parameter to **rising slope**,
  - ◦   set the **Counter** parameter to **0**,
- •   for the parameter block **Display Options**:
  - ◦   enter **Warehouse** into the parameter **Name**,
- •   in the parameter block **Channels**:
  - ◦   set **Slot 1** parameter as logical channel **30** named **sensor 1**,
  - ◦   set **Slot 2** parameter as logical channel **31** named **sensor 2**,
  - ◦   set **Slot 3** parameter as logical channel **32** named **sensor 3**,
  - ◦   set **Slot 4** parameter as logical channel **33** named **sensor 4**,
  - ◦   set **Slot 5 and Slot 6** parameters as empty.

In the next step, define the **Initial View** displayed on the LCD screen after the device is started. The **Initial View** is to display the defined **Group 1** with logical channels showing temperature reading.
For that purpose:
- −   enter the **General Settings** menu:
  - •   for the parameter block **Initial View**:
    - ◦   set the **Display Mod**e parameter to **value**,
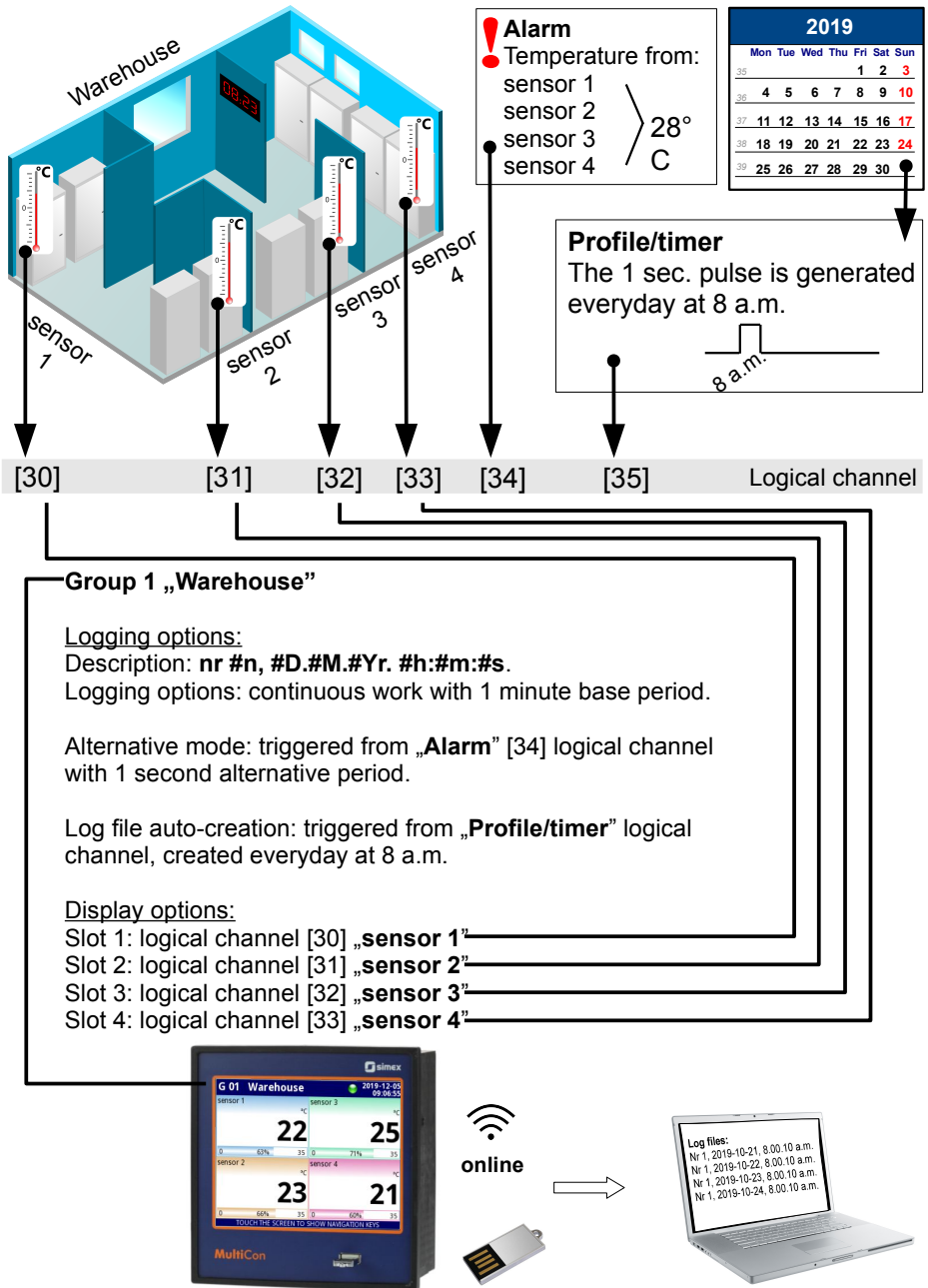   - ◦   set the **Group** parameter to **Group 1: Warehouse**.

Fig. 7.139. Example of On-demand Logging functionality

7.15.3. Groups - Display options

Display options parameter block contains the following options:
- – **Name** – it gives the name to the Group,
- – **Defined mode** – defines the default data presentation mode for the group,
  - • **(not set)** – data are presented in the last active data panel mode.
  - • **numerical values** – data are presented as numerical values,
  - • **bars** - data are presented as vertical or horizontal bars,
  - • **charts** - data are presented as vertical, horizontal or phasor chart,
  - • **needles** - data are presented as needle dials,
  - • **many groups** - data are presented as a value with subsequent groups (then up to 5) on one data panel,
  - • **SCADALite** - data are presented in **SCADALite** mode (see **Chapter 7.15.5. Groups - SCADALite**),
  - • **controllers** - group data are presented in the form of a regulator or regulators (up to 4 on one data panel) if neighboring groups are specified (see **Chapter 7.15.6. Groups – Controllers view**),
- – **Bars** – sets the bars type when a group is displayed in this mode,
  - • **horizontal** - horizontal direction of bars position,
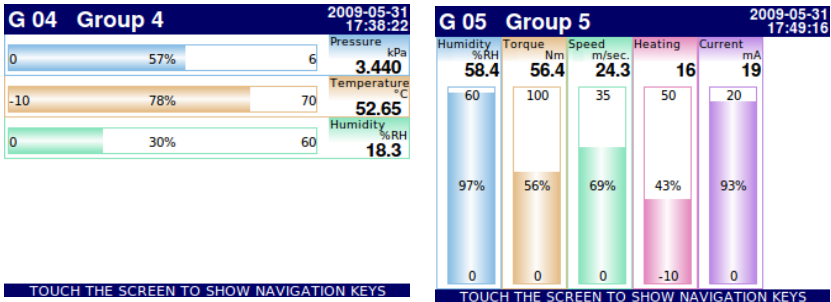 - • **vertical** - vertical direction of bars position.

*Fig. 7.140. Examples of **Horizontal** and **Vertical Bars** Group views*

- – **Charts** – sets the chart type when a group is displayed in this mode,
  - • **horizontal** - time axis is in horizontal position,
  - • **vertical** - time axis is in vertical position,
  - • **phasor** – vector signal representation; each phasor is drawn based on two logical channels, set in one of three **Amplitude and phase** blocks. The logical channel which is set in **Slot 1 (3, 5)** parameter is responsible for the phasor amplitude value and the logical channel which is set in **Slot 2 (4, 6)** is responsible for the phasor phase value. The user should remember that **Graph low** and **Graph high** parameters should be adjusted to the values that may occur in the logical channel, because a minimal and maximal phasor value corresponds to the values in these parameters. A phase of the phasor is module 360 function of the logical channel.
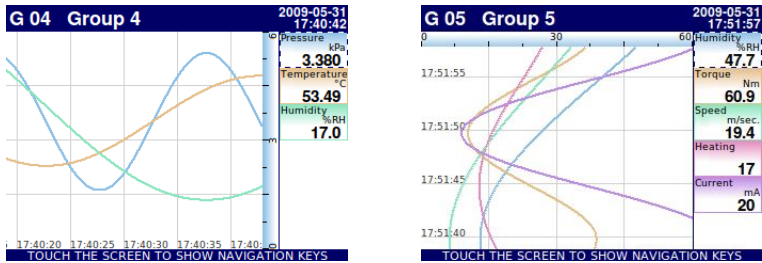
Fig. 7.141. Examples of Horizontal and Vertical Charts Group views
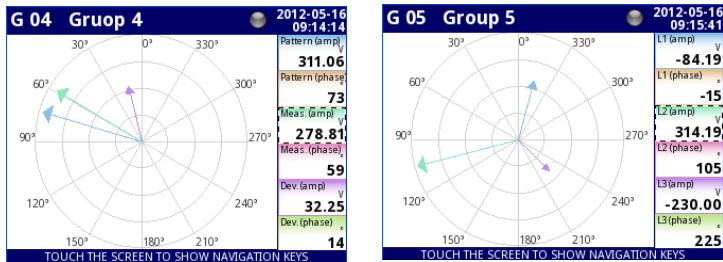
Fig. 7.142 Examples of Phasor Charts Group views

– **Line width** – sets the width of the line:
  • **1 pixel** - the chart line one pixel width,
  • **2 pixels** - the chart line two pixels width,
  • **3 pixels** - the chart line three pixels width,
– **Time scale** – time window which displays a chart which contains samples from the last:
  • **19 sec.**
  • **48 sec.**
  • **95 sec.**
  • **3 min.**
  • **6 min.**
  • **12 min.**
  • **30 min.**
  • **60 min.**
  • **2 h**
  • **4 h**
  • **8 h**
  • **16 h**
  • **24 h**
  • **3d**
  • **7d**

(i) Each group displays samples in the time window (defined in **Time scale** parameter in **Groups** menu) (see **7.143**).

When the **Time scale** is set as **30 min. or more**, the device has a possibility to display all the samples which were measured to present the moment and can fit in this **Scale** (no matter if the device was on or off, see **7.144**). Samples which were not measured will be:

• displayed as "0", when the time when the device was off does not fit between moving vertical markers on the screen,

• repeated on the screen, until new samples after turning on the device occur, when the time the device was off does fit between moving vertical markers on the screen.

When **Time scale** is set as **less then 30 min.** the device has a possibility to display only those samples which were measured from the moment when the device was turned on to the present moment (but they must fit in the **Scale**, see **7.145**).
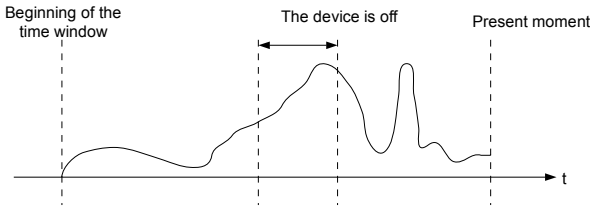
Fig. 7.143. Example of displayed real signal
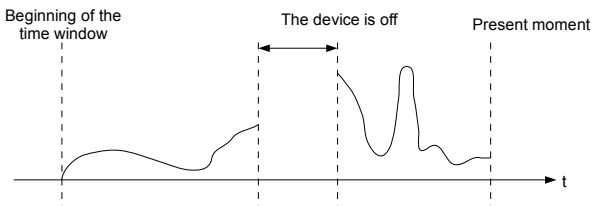
Fig. 7.144. Example of displayed signal when **Time sample** $\geq$ *30min*
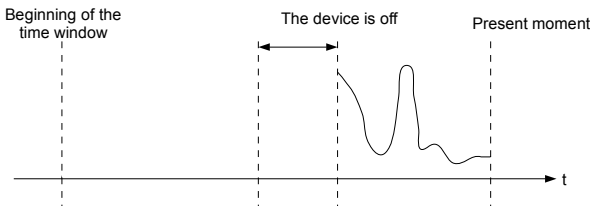
Fig. 7.145. Example of displayed signal when **Time sample** < *30min*

– **Background** - this parameter has the following options:
  • **white** - the background of the window displaying the charts is white,
  • **black** - the background of the window displaying the charts is black.
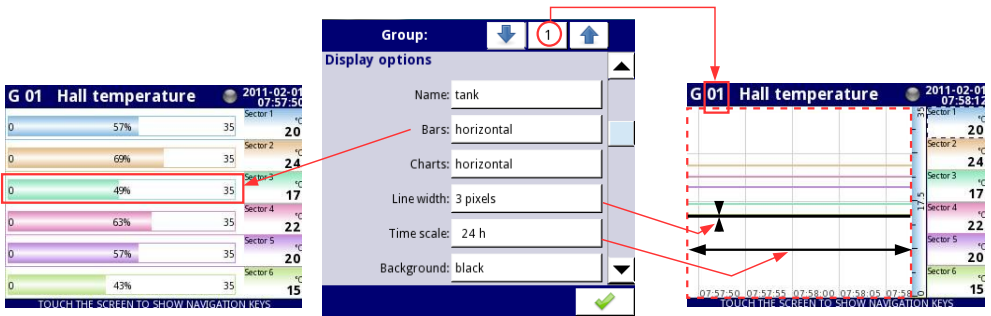
*Fig. 7.146. Sample of **Group** parameters settings - all Slot set to **Logical channel***
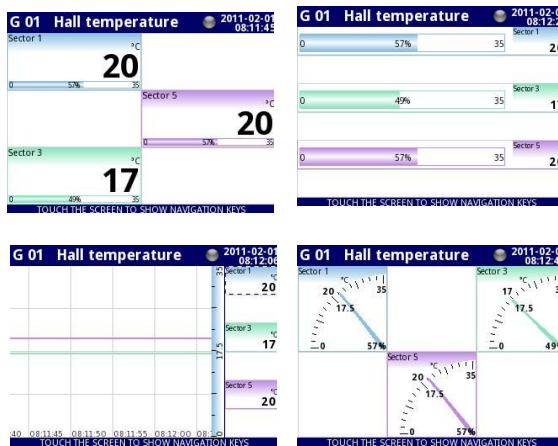
*Fig. 7.147. Sample of **Group** parameters settings - Slots set to **Logical channel** and set to* ***empty***
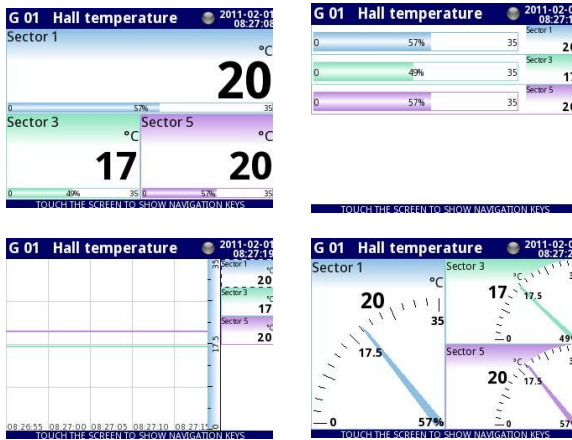
*Fig. 7.148. Sample of **Group** parameters settings - Slots set to **Logical channel** and set to **disabled***
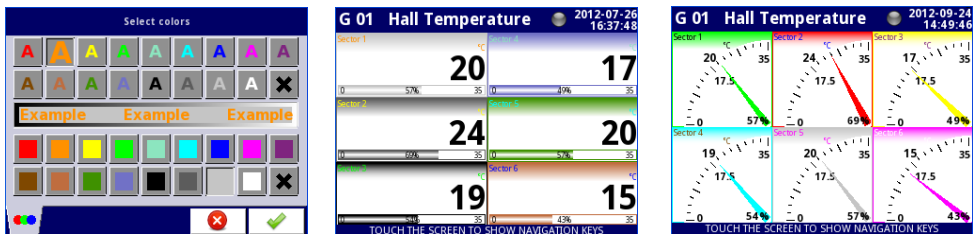
Fig. 7.149. Style color select screen and examples of views with changed styles

7.15.4. Groups - Channels

Channels parameter block defines the number and location of Logical channels that are displayed in the **Group**, includes the parameters:

- **Slot 1÷6** - in each slot the user can select an option (see **7.146÷7.148**):
    - **disabled** – disabled position is skipped which reduces the number of position to deploy in the display window,
    - **empty** – the empty position remains empty so that in contrast to the disabled position it doesn't reduce the number of position to deploy in the display window,
    - **Logical channel** – the user can select one of all available Logical channels which will be displayed in the specific location on the screen,
- **Style** – defines the color which will be used to draw the logical channel selected in this **Slot**. The user can specify the font color in the channel (except value color) and the drawing color of all elements connected with channel visualisation (see **7.149**). By using black **X** buttons the user can return to default settings.

ⓘ The **Slot** constitutes the link between the selected logical channel and the parameters of presenting it.

7.15.5. Groups - SCADALite

SCADALite is a functionality that allows the user to obtain process visualization directly from **MultiCon CMC-N16** display. Information, that is relevant to the operator can be presented in graphical form using values, needles and bars. **SCADALite** allows the user to define a certain number of screens and links between them, presenting different ranges of required information. Examples of visualization in **7.150**.
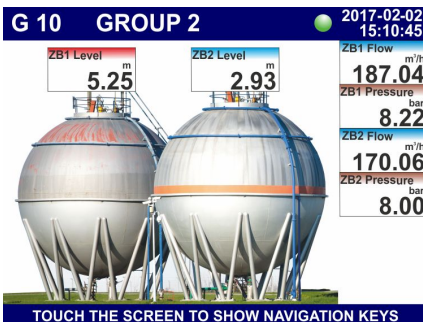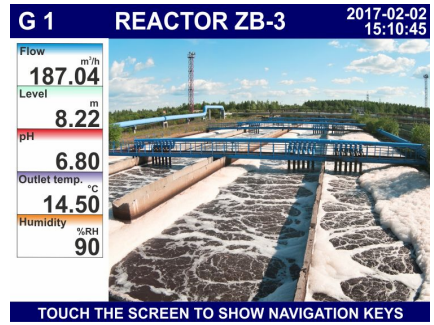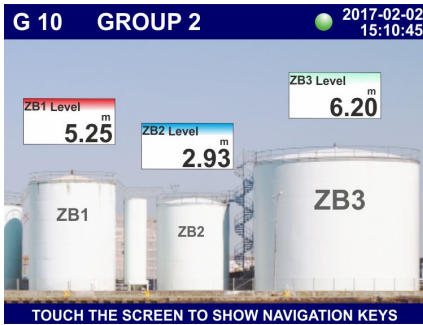
*Fig. 7.150. Example of **SCADALite** visualization*
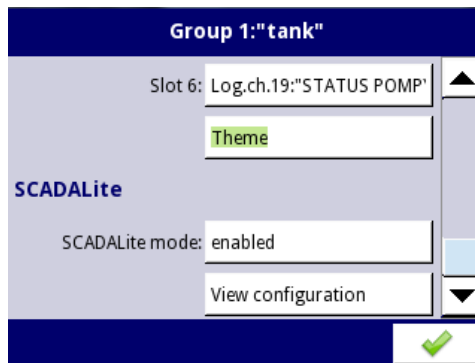
*Fig. 7.151. The view of activation **SCADALite** mode*

SCADALite options parameter block – always visible, but it is active only in devices which have a licence for this functionality (checking the state or entering the licence k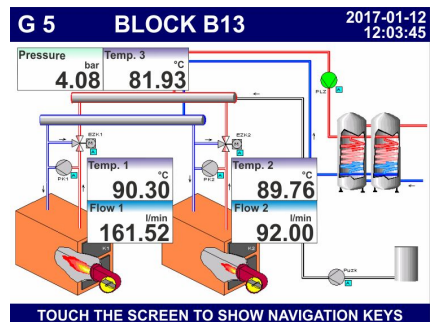ey has been described in **Chapter 7.5. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION**). In all other devices, functionality is inactive and the following message is displayed: "Available after entering licence". By using these parameters the user can set an individual type display mode.

The parameter block of **SCADALite** includes:

– **SCADALite mode** – in case if the user has active **SCADALite** licence this parameter enabled to run this functionality for chosen group. This mode can be:
  • **disabled** – **SCADALite** mode is disabled for this group,
  • **enabled** – **SCADALite** mode is enabled for this group.

– button **View configuration –** options allows the user adjustment **SCADALite** view to the individual needs of the user.
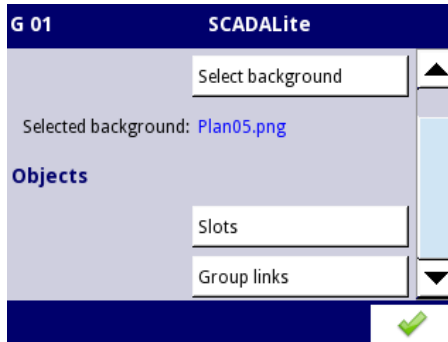
*Fig. 7.152. The view of **SCADALite** parameter*

After choosing the **View configuration** button the user can access the following parameters:

– **Select background –** this button refers to the list of graphic files, that were previously added to the device's memory. From the displayed list the user can choose a background to visualize parameter of a particular group (see **Chapter 7.4. FILES MANAGEMENT**),

– **Selected background –** displays the name of the file that is currently the background for the group,

– **Slots** – this button refers to the configuration of the individual slots. View the submenu presented in **7.153**. It allows the user to access the parameters determining the way of presenting results:

The arrows in the top right corner of the screen allow the user to switch between slots. The middle button allows the user to directly select a particular slot from the list.
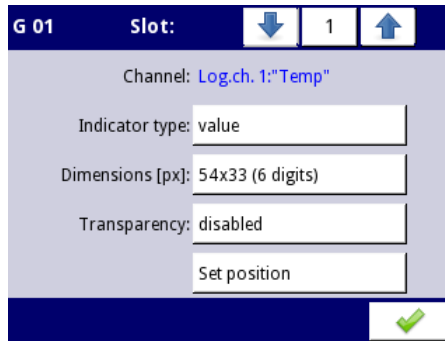
Fig. 7.153. Window of Slot parameter

- **Channel –** indicates the channel for which the display parameters are configured on the slot,
- **Indicator type –** parameter specifying the form of data presentation, the options available include:
  - **value –** value of the logical channel displayed as a number,
  - **needle –** value of the logical channel displayed as a needle,
  - **bar –** value of the logical channel displayed as a bar,
- **Dimensions [px]** – parameter defining indicator dimensions. All the indicators have been presented in **Tab. 7.9**,

  If the **Indicator type** parameter is set to **value**, the dimensions available include:
  - 54x33 (6 digits),
  - 64x34 (6 digits),
  - 80x38 (6 digits),
  - 80x46 (6 digits),
  - 100x50 (6 digits),
  - 160x100 (6 digits),

  If the **Indicator type** parameter is set to **needle**, the dimensions available include:
  - 73x96 (4 digits),
  - 100x87 (5 digits),
  - 106x100 (6 digits),
  - 123x87 (6 digits),
  - 161x101 (6 digits),
  - 160x202 (6 digits),

  If the **Indicator type** parameter is set to **bar**, the dimensions available include:
  - 54x100 (6 digits),
  - 54x150 (6 digits),
  - 75x150 (6 digits),

- ◦ 100x150 (6 digits),
- ◦ 54x202 (6 digits),
- ◦ 75x202 (6 digits),
- ◦ 100x202 (6 digits),
- • **Transparency** – the parameter specifying transparency of a slot in relation to its background:
  - ◦ **enabled –** slot limpidity,
  - ◦ **disabled –** no slot limpidity,
- • **Set position –** this button refers to a simulated view of a group that shows the selected background and positions of all the active slots. The position of the slot can be selected by touching the screen in the right place - the indicated point will be the top left corner of the indicator. The point coordinates can also be entered in the lower left corner of the screen.

ⓘ The default position of a slot is X:0, Y:0, therefore before the user changes position, the individual slots overlap.
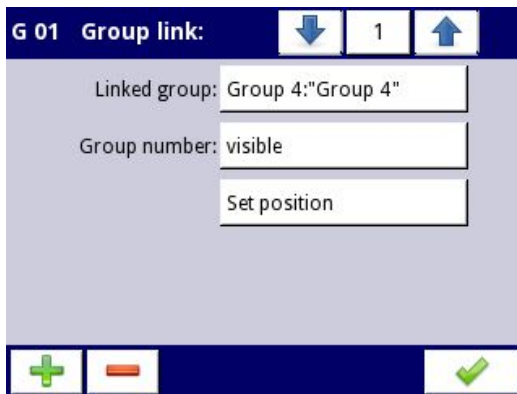
Fig. 7.154. Window of Group link parameter

– **Group links** – this button allows the user to create a link to a given group with a view of another group configured in **SCADALite** display mode; created link as a link placed in the selected place on the measurement panel will transfer the user directly to the view of indicated group,

The arrows in the top right corner of the screen allow the user to switch between links. The middle button allows the user to directly select a particular link from the list.

- • **Linked group –** parameter specifies the group to which the user will be moved when clicking on the link indicator activity area,
- • **Group number –** parameter specifies whether the link number will be visible to the group number referenced by the link:
  - ◦ **invisible –** number of the link group will not be visible,

- ◦ **visible –** number of the link group will be visible.
- **Set position –** this button refers to a simulated view of a group that shows the selected background and positions of all the active slots and links. The position of the link can be selected by touching the screen in the right place - the indicated point will be the top left corner of the link indicator. The point coordinates can also be entered in the lower left corner of the screen. For example, the screen view of **MultiCon CMC-N16** when configuring **SCADALite** display mode is presented in **7.155**.
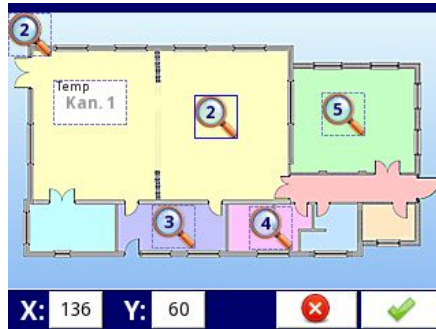
*Fig. 7.155. Example of the screen view during configuring **SCADALite** display mode*

For each group, user can define up to 10 "" group links. In the configuration mode, the link indicator icon always displays the number of the group to which the link is transfering. In configuration mode, the link pointer icon, that is currently in use, is surrounded by a blue frame. Frames represent the active area of the link. An example of the screen view with active links is presented in **7.156**.
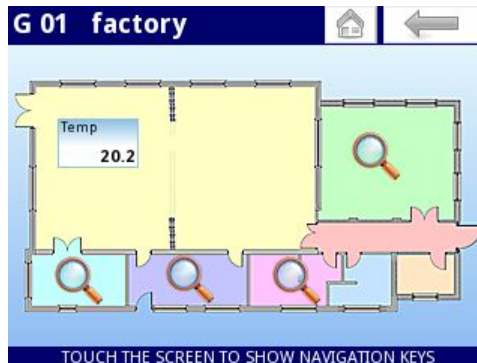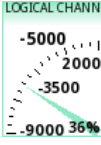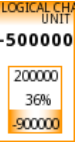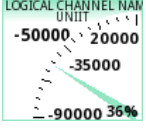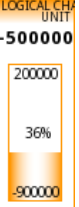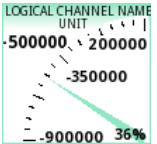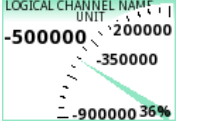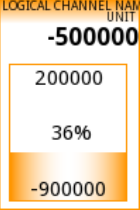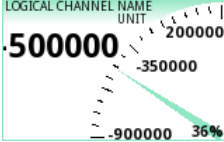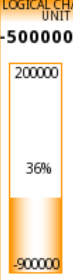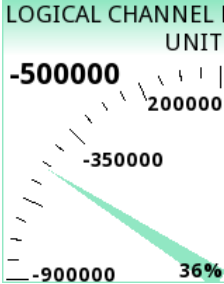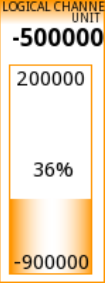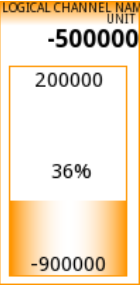
*Fig. 7.156. Example of the screen view in **SCADALite** display mode with active links*

In this mode the top **Home-Back** context menu is available. It appears after double-clicking on the measurement panel area (just after the lower context menu is closed) and after switching to another **SCADALite** view using the link to the group. The top context menu **Home-Back** remembers the last 30 views displayed in **SCADALite** display mode. The view is saved regardless whether it was switched to another link or by using the lower context menu **GROUP**. The history of the last views is deleted after going to another display mode than

SCADALite. If the history of the last views is empty, the **Back** button is inactive and its icon is greyed out. The **Home** button tranferes to the **Home group** view set in **Chapter 7.8. GENERAL SETTINGS**).

The proportions of the individual indicators available in the **SCADALite** display mode have been presented in **Tab. 7.9**.

Slot style		
value	needle	bar
54x33 px	73x96 px	54x100 px
64x34 px	100x87 px	45x150 px
80x38 px	106x100 px	75x150 px

80x46 px	123x87 px	100x150 px
LOGICAL CHANNEL UNIT **-500000**	LOGICAL CHANNEL NAME UNIT **-500000** 200000 -350000 -900000 36%	LOGICAL CHANNEL NAM UNIT **-500000** 200000 36% -900000

100x50 px	161x101 px	54x202 px
LOGICAL CHANNEL NAM UNIT **-500000**	LOGICAL CHANNEL NAME UNIT **-500000** 200000 -350000 -900000 36%	LOGICAL CHA UNIT **-500000** 200000 36% -900000

160x100 px	160x202 px	75x202 px
LOGICAL CHANNEL NAME UNIT **-500000**	LOGICAL CHANNEL N UNIT **-500000** 200000 -350000 -900000 36%	LOGICAL CHANNE UNIT **-500000** 200000 36% -900000

		100x202 px
		LOGICAL CHANNEL NAM UNIT **-500000** 200000 36% -900000

*Tab. 7.9 Dimensions of individual indicator for **SCADALite** display mode*

7.15.6. Groups – Controllers view

The **Controller view** block parameter specifies whether the data presentation mode in the form of a controller is available for this group and whether it will be presented as single or with adjacent groups.

Fig. 7.157. Slots layout in controllers view

In the controllers view, the values of logical channels assigned to slots 5 and 6 are presented in a two-state manner. For a value higher than zero, the slot will be highlighted in red. If a **channel highlight** is set for a given channel, it will replace the above operation (see **Chapter 7.9.1. Logical Channels - General settings**). The displayed name of slots 5 and 6 are limited to 10 first characters (the rest of them will not be visible).

The following parameters belong to this block:

– **Mode**:
  • **disabled** – the data presentation mode in the form of a controller is inactive for this group and the data can not be presented in this form,
  • **enabled** – the data presentation mode in the form of a controller is active for this group and the data can be presented in this form.
– **Adjacent group 1÷3** – these parameters specify which groups will be presented in the data panel together with the main group. If at least one adjacent group is assigned, the data panel will be divided into 4 parts. In each quadrant the data of one group will be presented in the form of a regulator. If the selected adjacent group is inactive – the quarter will be empty. If all adjacent groups are inactive, only the main group in the form of a controller will be displayed on the entire data panel.

7.15.7. Groups - Examples of visualisations of groups

7.15.7.1. Single channel - one big needle

Task:

The task is to create a Group with only one logical channel. The value of that channel should be displayed in the whole screen in the needle mode.

Solution:

Let's assume that we have configured a logical channel in the Hardware input Mode for the FUN4 hardware configuration in RTD mode. The logical channel is named Temperature and it displays heater temperature (example of logical channels configuration is shown in **Chapter 7.9.11. Examples of Logical Channels configuration**. or in **Chapter 7.11.5. Examples of build-in output configurations**.

We need to define a Group with a logical channel in order to display it on the device screen. To do that:
– enter the **Groups** menu and turn on **Group 1** if it is turned off,
– in the **Name** parameter write **Heater**,
– for **Channels** parameter block:
    • **Slot 1** parameter set as **Logical channel 1** named **Temperature**,
    • **Slot 2 ÷ 6** set as **disabled**,
– all the other parameters should be on default settings,

Next we need to define the **Start-up view**. To do this:
– enter **the General settings** menu,
– for **Start-up view** parameter block:
    • **Selection method** parameter set as **specified view**,
    • **Group** parameter set as **Group 1: "Bake"**,
    • **Default mode** parameter set as **needles**,

Finally, leave this menu and press the Save settings button. The result of just made changes will be visible after the first measurement is called. An example of this view in **7.158**.
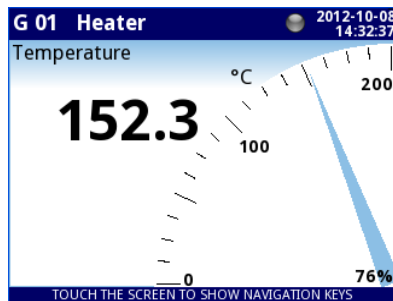
Fig. 7.158. Single big needle example

7.15.7.2. Three channels view - one bigger, two smaller

Task:

The task is to visualize three logical channels which will present the following values: Pressure, Temperature and Humidity.

Solution:
Let's assume that we have configured three logical channels:
– Logical channel 1 named **Pressure**,
– Logical channel 2 named **Temperature**,
– Logical channel 3 named **Humidity**,
(an example of logical channel configuration is shown in **7.9.11. Examples of Logical Channels configuration**)

We need to define a Group with these logical channels to display them in one window on the screen. To do that:
– enter the **Groups** menu and turn on **Group 4** if it is turned off,
– in the **Name** parameter write **Group 4**,
– for **Channels** parameter block:
  • **Slot 1** parameter set as **Logical channel 1** named **Pressure**,
  • **Slot 2** parameter set as **Logical channel 2** named **Temperature**,
  • **Slot 3** parameter set as **Logical channel 3** named **Humidity**,
  • **Slot 4 ÷ 6** set as **disabled**,
– all the other parameter should be on default settings,

Next we need to define the Initial view. To do this:
– enter the **General settings** menu,
– for **Initial view** parameter block:
  • **Display mode** parameter set as **needles**,
  • **Displayed group** parameter set as **Group 4: Group 4**,

Finally, leave this menu and press the Save settings button. The result of just made changes will be visible after the first measurement is called. An example of this view is posted in **7.159**.
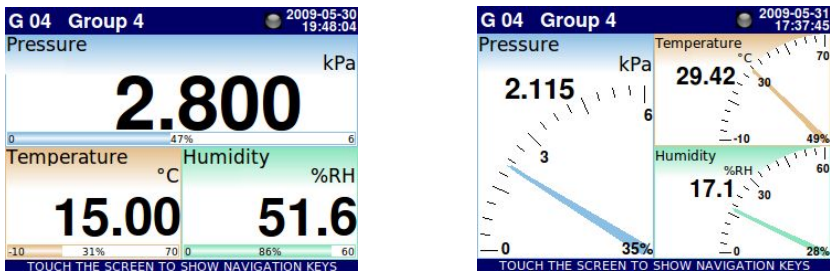
Fig. 7.159. Example of three channels presentation with emphasis on pressure

7.15.7.3. Three channels view - one bigger, two smaller

Task:
        The **MultiCon CMC-N16** controls 4 independent autoclaves (kind of a hermetic furnace) using 4 PID built-in controllers. Each of them controls temperature. They also monitor pressure and heating power. Executive system consist of 2 heaters per each control object, thus the CMC is equipped with 8 relays - 2 for each furnace, and 4 temperature inputs.

The task is to use the controllers view to show crucial parameters for each process on one screen.

Solution:
Let's put the following parameters in 4 separate groups of channels. One parameter should be in one logical channel. Each channel must be put in suitable slot, accordingly to the **7.15.6. Groups – Controllers view**.
   •   temperature (measurement) - slot 1
   •   temperature (set point value) - slot 2
   •   pressure (measurement) - slot 3
   •   power (measurement) - slot 4
   •   relay 1 (hardware output monitor) - slot 5
   •   relay 2 (hardware output monitor) - slot 6

For this purpose:
   •   go to Menu / Device configuration / Groups / Group 1

In "Channels" section of the group settings user must choose proper logical channel for each slot. The result is as shown on **7.160**. The above steps shall be performed for each of the 4 furnaces, so for the 4 groups of channels.
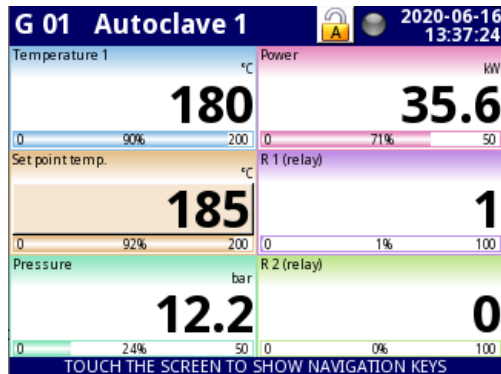
Fig. 7.160. Default view

The next step is to activate the controllers view. For this purpose:
   •   go to Menu / Device configuration / Groups / group 1,
   •   set the mode to "enabled" in the "Controllers view" section,
   •   after exiting the menu, use mode buttons to change it to the **controllers view**.
The above steps shall result in the view shown in **7.161**.
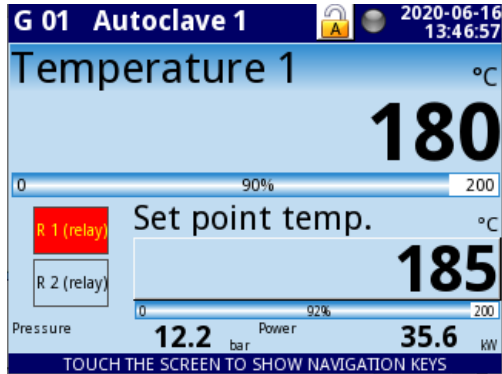
Fig. 7.161. Controllers view of one group

This is how the controllers view looks like for a single group. The **MultiCon CMC-N16** allows to show 4 such views simultaneously. To do so:
- go to Menu / Device configuration / Groups / Group 1,
- set the adjacent groups no. 1, 2 and 3 in the "Controllers view" section.

ⓘ This is important, at this moment, for the adjacent groups no. 1, 2 and 3 to have assigned logical channels to the proper slots. So the physical values they represent will be shown as expected. Analogically to how the group no. 1 has.

The above configuration results as shown in **7.162**.
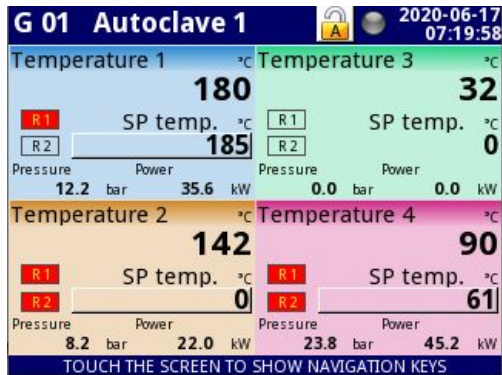
Fig. 7.162. Controllers view of four groups.

7.16. MODBUS

General settings
Logical channels
Build-in inputs
Build-in outputs
External outputs
Profiles/timers
Controllers
Groups
Modbus
Network settings
Access options
E-mail notifications

MENU ➡ Device configuration ➡
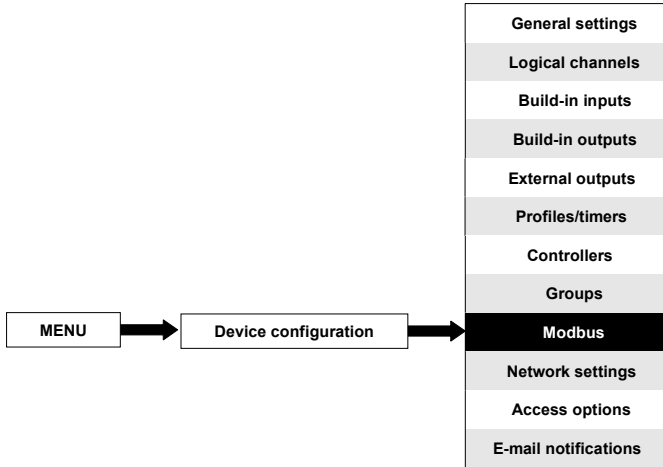
The basic version of **MultiCon CMC-N16** has one RS-485 port built-in. The communication ability can be increased by installing a hardware communication configuration into block D of the device (**7.163**). It offers one additional serial port RS-485 and one ethernet port, which allows the creation of an advanced Multi-Modbus system. In the current software version, a MODBUS RTU and MODBUS TCP/IP protocol are available and every port can be switched to the Slave or Master (except port 4 which can be only SLAVE) mode.

To establish a connection between **MultiCon CMC-N16** and another device using TCP/IP protocol, port number 502 must be unlocked in the software and hardware firewall.
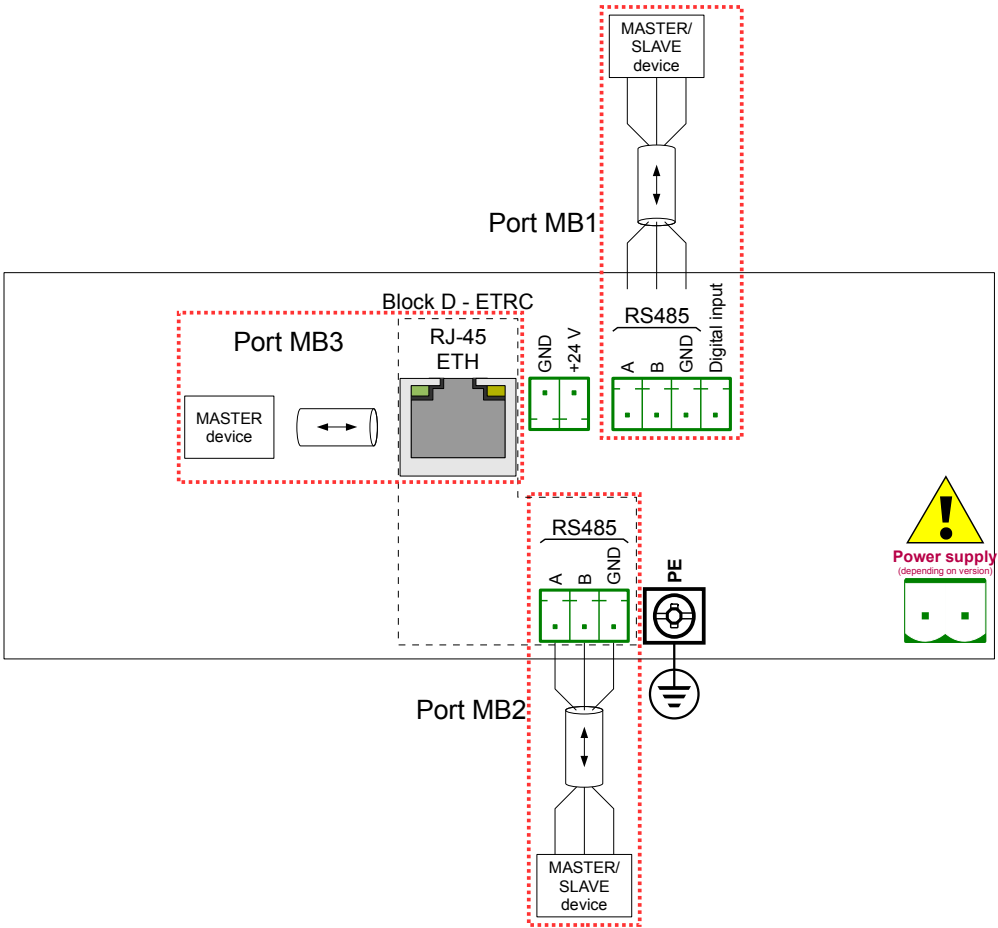
Fig. 7.163. Serial communication ports available in the device

7.16.1. Modbus - General settings

The window with basic parameters of communication interfaces is shown in **7.164**.
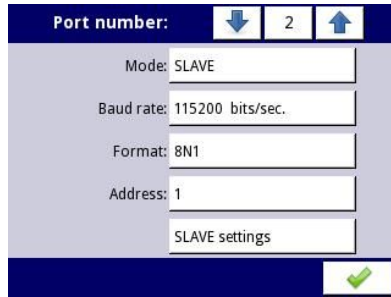
*Fig. 7.164. Configuration parameters for **SLAVE** mode*

The arrows placed in the upper right corner of the screen allow switching between available serial ports. The middle button can be used for direct selection of a specific communication port from the list.

The parameter common for all Modbus protocol modes is:
– **Mode** - which has the following options:
  • **disabled** – the selected Modbus port is inactive,
  • **SLAVE** – this device is a SLAVE device, see **Chapter 7.16.2. Modbus - SLAVE mode**
  • **MASTER** – this device is a MASTER device and manages the Slave devices, see **Chapter 7.16.3. Modbus - MASTER mode**.

7.16.2. Modbus - SLAVE mode

The parameters of the Modbus RTU protocol (serial communication) for the SLAVE mode are:
– **Mode = SLAVE**,
– **Baud rate** – determines the baud rate of the RS-485 interface, available options are: **1200**, **2400**, **4800**, **9600**, **19200**, **38400**, **57600**, **115200** bit./sec.,
– **Format** - data format of the RS-485 interface, available options are shown in the **Tab. 7.10**.

Format	Number of data bits	parity control	Number of stop bits
8N1	8	none	1
8N2	8	none	2
8E1	8	even	1
8E2	8	even	2
8O1	8	odd	1
8O2	8	odd	2

Tab. 7.10 Data format of the RS-485 interface

  – **Address** - SLAVE device address, available address range: 1÷255,
  – **SLAVE settings** – submenu where the user can configure available read and/or write registers. For more information see below,

The parameters of the Modbus TCP/IP protocol (ethernet communication) for the SLAVE mode are:
  – **Mode = SLAVE**,
  – **SLAVE settings** – submenu where the user can configure available read and/or write registers, for more information see below,

   See **Chapter 7.16.2.3. Modbus SLAVE - The Modbus protocol handling** for a detailed description of MODBUS protocol handling in the **MultiCon CMC-N16** device.

Notes on Modbus TCP/IP:
  – allows the device to keep up to 3 connections, more connections disconnect the one that is the longest in the network,
  – not implemented queuing priorities of customers,
  – no access control to device based on client IP address,

SLAVE settings menu
   This menu allows the user to define a type, data format and registers activity which are available to the **MultiCon CMC-N16** device. In this menu the user can also view the assignment of groups of registers to logical channels as well as other important parameters of devices.

The list of parameters of the **SLAVE settings** menu:
  – **Load device template** - this button allows the loading of a template with predefined blocks of input registers, for more information see **Chapter 7.16.2.1. Modbus SLAVE - Modbus Templates for SLAVE mode**,
  – **Save device template** - this button allows saving of templates of predefined blocks of input registers, for more information see **Chapter 7.16.2.1. Modbus SLAVE - Modbus Templates for SLAVE mode**,
  – **Device channels** parameters block - this block of parameters allows the user to define a type, data format and activity registers, the user can also preview the list of registers to read and blocks of registers to transmit into modbus frames,

7.16.2.1. Modbus SLAVE - Modbus Templates for SLAVE mode

In the **SLAVE settings** menu there are 2 buttons for modbus templates:
  – **Load device template** - this button allows to load a template with predefined blocks of output registers. Pressing this button invokes a file selection window. Templates can be imported into the **MultiCon CMC-N16** using a the **File Management** menu,
  – **Save device template** - this button allows the user to save in templates the predefined blocks of input registers. A saved template can be used for:
    • fast copying/moving of the SLAVE device settings to another port (using the **Load device template** button),
    • easy exchange of templates between different MulitCon devices (using the **File management** menu),

   An example of the Modbus template selection window is shown in **7.165**. Additional icons are described below this figure.

Fig. 7.165. Template selection window

This button invokes the software keyboard window allowing to write or search the template name.

This button allows the user to delete a selected template.

Navigation keys allow the user to select an appropriate template.

7.16.2.2. Modbus SLAVE - Device channels for SLAVE mode

Device channels parameter block has 2 buttons:
–   **Output list** – the submenu where the user can configure registers of available logical channels. In this submenu the user decides which registers are read only or read and write.
–   **Avail. output blocks** – the submenu where the user can review the block of registers configured automatically by the device. This submenu is only for review.

In the submenu of the **Device channels** parameter block the visible buttons are:
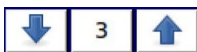
The arrows placed in the upper right corner of the screen allow the user to switch between channels/blocks of input/output registers. The middle button moves directly to specific channel/block of Input/Output registers from the list.

Output list submenu
This menu consists of the following fields:
–   **Value register** – the submenu where the details of the Modbus register can be set,
  •   **Register type**, the user can select two types:
    ◦   **HOLDING** - holding registers of a SLAVE device compatible with the Modbus protocol
    ◦   **INPUT** - input registers of a SLAVE device compatible with the Modbus protocol
  •   **Write mode** - this parameter for HOLDING register type allows the user to

select whether the register is to be read only or read-write, for INPUT type this parameter is read only,

- **Register number** - read-only parameter; the assignment of registers to logical channels and other **MultiCon CMC-N16** parameters see **Chapter 7.16.2.4. Modbus SLAVE - List of registers**,
- **Data format**, we can select one of these options:
  ◦ **32 bits, signed**, integer value, the most significant bit is the sign bit,
  ◦ **32-bits, unsigned** - integer value without information about the sign,
  ◦ **32 bits, float**, floating point IEEE 754 format,
  ◦ **32-bits, BCD**, unsigned BCD value, write two digits in each byte,
- **32 bit reading**, this parameter is only for 32-bit format, the user can select one of these options:
  ◦ **two 16-bit registers**,
  ◦ **one 32-bit register**
- **Ordering** - this parameter is only for 32-bit formats, the letters **ABCD** mean: **A** - most significant byte of high words (word = 2 bytes), **B** - least significant byte of high words, **C** - most significant byte of low words, **D** - least significant byte of low words,
  ◦ **ABCD (standard),**
  ◦ **CDAB**,
  ◦ **DCBA**,
  ◦ **BADC**,
- **Data shift** - read-only parameter, no data shift,
- **Data mask** - read only parameter, no data mask,
– **Status register** - this parameter allows the user to review the number and data format of the status register,
– **Dec. point register** - this parameter allows the user to review the number and data format of the decimal point register,

Avail. output blocks submenu
This submenu is read only and consists of the following fields:
– **Block type** - a read only parameter, the user can review the options that were set in the **Output list** submenu. Available options:
  - **read HOLDING register** - read only register,
  - **write HOLDING register** - read and write register,
  - **read INPUT register** - read only register,
– **Register size** - defines data size, can be set to:
  - **16-bit registers** - data is read/preset as 16 bit registers; this value can also be used for 32-bit registers reading/presetting. In such a case data is composed as two 16-bit registers,
  - **32-bit registers** - for 32-bit registers reading/preseting only, data is read as one 32-bit register.
– **First register** - value indicating the number of the first register of the block,
– **Last register** - value indicating the number of the last register of the block, for a single block with one 16-bit register the parameter **First register** and **Last register** must be the same number register,

HOLDING registers can be set as **read** or **read and write**. That is why when the **Block type** parameter is set as **read HOLDING reg.**, in this block, registers are **read** type. When this block is set as **write HOLDING reg.**, in this block, register are **read and write** type.

7.16.2.3. Modbus SLAVE - The Modbus protocol handling

Parameters for Modbus RTU implemented in the **MultiCon CMC-N16**
Transmission parameters: 1 start bit, 8 data bits, 1/2 stop bit, no/even/odd parity control
Baud rate: selectable from: 1200 to 115200 bits/second
Transmission protocol: MODBUS RTU compatible

7.16.2.4. Modbus SLAVE - List of registers

The device parameters and measurement result are available via the RS-485 interface, as HOLDING and INPUT type registers of Modbus RTU protocol and via Ethernet interface as HOLDING and INPUT type registers of MODBUS TPC/IP. The registers (or groups of the registers) can be read by 04h function when a register is set as INPUT type or by 03h function when a register is set as HOLDING type. HOLDING type registers can be written by 06h (single registers) or 10h (group of the registers) accordingly to the Modbus RTU and TCP/IP specification.

Register	Write	Range	Register description
20h	No	0÷255	Address of device
21h	No	2040h	Device identification code
Measurements results (floating point format) [1]			
200h	Yes	0÷0FFFFh	Measurement result for logical channel 1 (high word)
201h	Yes	0÷0FFFFh	Measurement result for logical channel 1 (low word)
202h	Yes	0÷0FFFFh	Status for logical channel 1: **0h** – data valid, **1h** – data not ready, **20h** – software error, **40h** – bottom border of the software measurement range is exceeded, **80h** – top border of the software measurement range is exceeded, **2000h** – hardware error, **4000h** – bottom border of the hardware measurement range is exceeded, **8000h** – top border of the hardware measurement range is exceeded, **FFFFh** – data nor available (e.g. logical channel not configured)
203h	Yes	0÷4	Decimal point for logical channel 1
Register from 204h to 2EFh			Measurement result, status and decimal point for Logical Channels 2÷60
Measurements results (integer format) [1]			
400h	No	0÷0FFFFh	Measurement result for logical channel 1 (high word, not considering the decimal point)
401h	No	0÷0FFFFh	Measurement result for logical channel 1 (low word, not

Register	Write	Range	Register description
			considering the decimal point)
402h	No	0÷0FFFFh	Status for logical channel 1: **0h** – data valid, **1h** – data not ready, **20h** – software error, **40h** – bottom border of the software measurement range is exceeded, **80h** – top border of the software measurement range is exceeded, **2000h** – hardware error, **4000h** – bottom border of the hardware measurement range is exceeded, **8000h** –top border of the hardware measurement range is exceeded, **FFFFh** – data nor available (e.g. logical channel not configured)
403h	No	0÷4	Decimal point for logical channel 1
Register from 404h to 4EFh			Measurement result, status and decimal point for Logical Channels 2÷60

[1]*IEEE 754 Standard, Float point format represents data as precise as possible. Integer 32 represents value with constant precision, selscted by decimal point position. When decimal is set for example 0.0 then Int32 format represents integer part of the value contained in float registers and multiplied by 10 (e.g.: float is 1.2345, D.P. = 0.0, then Integer = 12). Similarly when decimal point is 0.000 then integer represents integer part of the value contained in float registers and multiplied by 1000 (e.g.: float is 1.234, D.P. = 0.000, then Integer = 1234)*

Tab. 7.11 List of register available in the device

7.16.2.5. Modbus SLAVE - Transmission errors handling

If an error occurs during reading or writing one of the registries, then the unit shall return the frame containing the error code (according to the Modbus protocol).

Error codes should be interpreted as follows:
01h   - illegal function (only functions 03h, 04h, 06h and 10h are available),
02h   - illegal register address
03h   - illegal data value

7.16.2.6. Modbus SLAVE- Example of query/answer frames

The examples concern a unit with address 1. All the values are given in the hexadecimal system.
Designations:
ADDR   Address of the device in the system
FUNC   Function number
REG H,L   Higher and lower part of registry number the command refers to,
COUNT H,L   Higher and lower part of registry counter number the command refers to, starting with the register which is defined by REG (max. 32)
BYTE C   Number of higher bytes in the frame
DATA H,L   Higher and lower part of data word

CRC L,H   Higher and lower part of CRC sum

1. Read of ID code

ADDR	FUNC	REG H,L		COUNT H,L		CRC L,H	
01	03	00	21	00	01	D4	00

The answer:

ADDR	FUNC	BYTE C	DATA H,L		CRC L,H	
01	03	02	20	40	A0	74

   DATA H,L - identification code (2040h)

2. Read of the registers 401h, 402h and 403h in one message (example of reading a number of registries in one frame):

ADDR	FUNC	REG H,L		COUNT H,L		CRC L,H	
01	03	04	01	00	03	55	3B

   COUNT L - the count of registers being read (max. 32)

   The answer:

ADDR	FUNC	BYTE C	DATA H1,L1		DATA H2,L2		DATA H3,L3		CRC L,H	
01	03	06	00	0A	00	02	00	00	18	B4

   DATA H1, L1   - 401h registry (10 – high word of value for channel 1, no decimal point),
   DATA H2, L2   - 402h registry (2 – low word of value for channel 1, no decimal point),
   DATA H3, L3   - 403h registry (0 – status for channel 1).

ⓘ   There is no full implementation of the Modbus Protocol in the device. The functions presented above are the only ones available.

7.16.3. Modbus - MASTER mode

The parameters of a **Modbus** protocol for the **MASTER** mode are:
  – **Mode = MASTER**,
  – **Baud rate** - this parameter allows the user to select the baud rate RS-485 interface, available option: **1200**, **2400**, **4800**, **9600**, **19200**, **38400**, **57600**, **115200** bit./sec.,
  – **Format** - data format of the RS-485 interface, available options are shown in **Tab. 7.12**.

Format	Number of data bits	parity control	Number of stop bits
8N1	8	none	1
8N2	8	none	2
8E1	8	even	1
8E2	8	even	2
8O1	8	odd	1
8O2	8	odd	2

Tab. 7.12 Data format of the RS-485 interface

– **Request timeout** - is the amount of time (any value between 0.01 to 3 sec.) the Master device waits for a response from the Slave device after sending a query,
– **Request retrials** - this is the number of times (integer value between 1 to 5) a Master device tries to send a message,
– **Interval** – minimal amount of time that elapses between read/write cycles. The read/write cycle contains all the operations with Slave devices. If the Master device requests all the Slave devices in shorter time than the interval, then new requests will begin after the Interval elapses.
– **Slave device** - this Button enters the submenu that allows to define the list of Slave devices connected to the current serial port of **MultiCon CMC-N16** and configure registers for read and/or write. See below for more information on this menu.

SLAVE device menu
        MultiCon CMC-N16 allows to address as many as 255 slave devices on the addresses from 1 to 255. To define an external data source, first an address must be chosen, next further parameters of the SLAVE device (having this address) set.

The arrows placed in the upper right corner of the screen allow switching between addresses of SLAVE devices. The middle button allows a direct selection of a specific address from the list.

If the particular address is not used, then a *short menu* is displayed:
– **Device type –** turns on and off a selected address,
   • **not present** - means that this address is not used (in other words - there is no SLAVE device connected with this address),
   • **defined** - after selecting this option an extended list of SLAVE device parameters will appear, see below in this **Chapter** for details
– **Load device template** – this button allows to load a template with predefined blocks of input and / or output registers. Pressing this button invokes a file selection window. After a successful loading of the template an extended list of SLAVE device parameters will appear, see below for details. Templates can be imported into the **MultiCon CMC-N16** using the **File Management** menu and also created by the user using the **Save device template** button after a full configuration of a slave device.

If the **Device type** is set to **defined** or a Template has been loaded, then the *extended SLAVE device menu* is shown. This menu has the following fields:
– **Device type** = **defined**,
– **Device name** - to create or change the name of a SLAVE device, press the button

next to the **Device name** label and enter the name using a displayed editor,
– **Device templates** parameter block – see **Chapter 7.16.3.1** for extended description,
– **Device channels** parameter block – see **Chapter 7.16.3.2** for extended description,
– **Register blocks** parameter block – see **Chapter 7.16.3.4** for extended description.

7.16.3.1. Modbus MASTER - Device templates parameter block

This block is composed of 2 buttons:
– **Load device template** – this button allows to load a template with predefined blocks of input and / or output registers. Pressing this button invokes a file selection window. After a successful loading of the template an extended list of SLAVE device parameters will appear, see below for details. Templates can be imported into the **MultiCon CMC-N16** using the **File Management** menu and also created by the user using the **Save device template** button after a full configuration of a slave device
– **Save device template** - allows the user to save a configured SLAVE device as a template for further usage. A saved template can be used for:
  • fast copying/moving of the SLAVE device to another address (use the **Load device template**)
  • easy creation of similar SLAVE devices by loading the template in another address and modification of parameters.
  • easy moving exchange of templates between different **MultiCon CMC-N16** devices (using the **File Management** menu).

An example of the Modbus template selection window is shown in **7.166**. Below additional icons are described below this figure.

Fig. 7.166. Template selection window

This button invokes the software keyboard window allowing to write or search the template name.

This button allows the user to delete a selected template.

The navigation keys allow the user to select an appropriate template.

7.16.3.2. Modbus MASTER - Device channels parameter block

This block includes the following buttons:
– **Input list** – the submenu in which the user can define logical channels with chosen write registers,
– **Output list** – the submenu in which the user can define logical channels with chosen read registers,

Both these submenus have basic icons presented below. Their functions are as follows:

This button allows the user to add a new Input/Output to the Input list / Output list,

This button allows the user to delete an Input/Output from the Input list / Output list

The arrows placed in the upper right corner of the screen allow the user to switch between Inputs/Outputs. The middle button moves directly to a specific Input/Output channel selected from the list.

Input List submenu.
When at least one Input Channel is added to the **Input list**, then the Input channel menu is displayed. This menu consists of the following fields:
– **Channel value** parameters block composed of buttons:
  • **Value register** – menu where the details of the Modbus register serving data of input can be set - see **Chapter 7.16.3.3. Modbus MASTER - Register settings**.
  • **Decimal point** – list of available decimal point positions, in the last place there is an option: *** exp (-point register).** Thanks to selecting this value a new field (**Decimal point register)** will appear in the **Channel value** block.
  • **Decimal point register** – allows the user to select the SLAVE device's register containing information about the decimal point position. Thanks to using this parameter the **Channel value** is displayed according to the formula:

$$(data\ of\ Value\ register)\cdot 10^{(-Decimal\ point\ register)}$$

See Chapter **7.16.3.3. Modbus MASTER - Register settings** for more info.

– **Channel status '-HI-'** parameter block – allows to define when a status **-HI-** should be displayed (returned) in place of a numerical value of Input Channel,
  • **-HI- state** – defines condition for **-HI-** state appearance:
    ◦ **never** – does not display status **'-HI-'**; for this option, other parameters of **Channel status '-HI-'** block are invisible,
    ◦ **if register = value** - state **'-HI-'** is returned if data read from **'-HI- register'** equals **'-HI- value'** parameter,

        ◦   **if register ≠ value** - state **'-HI-'** is returned if data read from **'-HI- register'** differs from **'-HI- value'** parameter,
- **-HI- register** - allows the user to select a status register to be read (see above)
- **-HI- value** - allows the user to define the value being returned corresponding to status **-HI-** (see above),

– For other blocks (**Channel status '-LO-'**, **Channel status '-WAIT-'**, **Channel status '-ERR-'**) parameters and their configuration are analogous,

(i) When a **Logical channel** is configured to the **Modbus** mode, then while reading the registers from a SLAVE device if connection to the SLAVE device is lost, the device returns an error and displays the state **-ERR-**.

Output list submenu

This submenu allows the user to edit the output channels registers to be written. When at least one Output Channel is added to the list, then the Output channel menu is displayed.

The parameters of the **Output channels** are:
  – **Output active:**
- **no** – the output channel is defined but invisible in the **External output** menu (see **Chapter 7.12. EXTERNAL OUTPUTS**),
- **yes** – the output channel is defined and visible in the **External output** menu (see **Chapter 7.12. EXTERNAL OUTPUTS**).

  – **Control type:**
- **as a relay** - the output has two states, low state: value '0' and high state: maximal value (for 16-bit format is the value 65535),
- **as a linear output** - can take any value depending on the settings in **Output register** and settings the parameters in the **External outputs** menu (see **Chapter 7.12. EXTERNAL OUTPUTS**),
  – **Output register** submenu - menu where the user can set details of the Modbus register to be written - see **Chapter 7.16.3.3. Modbus MASTER - Register settings**.

7.16.3.3. Modbus MASTER - Register settings

The submenu of **registry** settings in the Modbus protocol has the following fields:
  – **Register type** – this parameter is only for the register settings in the **Input list** menu (for the **Output list** menu the registers are HOLDING type), the user can select two types:
- **HOLDING** - holding registers of SLAVE device compatible with Modbus protocol
- **INPUT** - input registers of SLAVE device compatible with Modbus protocol
  – **Write mode** – read only parameter, informs about possibility to read or write register,
  – **Register number** - any value from 0 to 65535
  – **Data format**, we can select one of these options:
- **16 bits, signed** - integer value, the most significant bit is the sign bit,
- **16-bits, unsigned** - integer value without information about the sign,
- **32 bits, signed**, integer value, the most significant bit is the sign bit,
- **32-bits, unsigned** - integer value without information about the sign,

- • **32 bits, float**, floating point IEEE 754 format,
- • **16-bits, BCD**, unsigned BCD value, write two digits in each byte,
- • **32-bits, BCD**, unsigned BCD value, write two digits in each byte,
- – **32 bit reading**, this parameter is only for 32-bit format, the user can select one of these options:
  - • **two 16-bit registers**,
  - • **one 32-bit register**
- – **Ordering** - this parameter is only for 32-bit formats, the letters ABCD mean: A - most significant byte of high words (word = 2 bytes), B - least significant byte of high words, C - most significant byte of low words, D - least significant byte of low words
  - • **ABCD (standard)**
  - • **CDAB**,
  - • **DCBA**,
  - • **BADC**,
- – **Data shift** - values can be a bit moved to the right of any integer value in the range (parameter does not exist for **float** format):
  - • for 16-bit format from 0 to 15,
  - • for 32-bit format from 0 to 31,
- – **Data mask**, the device allows the user to use the masking of data on individual bits, mask 0xFFFF for 16-bit format means that the entire value of register is visible, while the 0x0 mask (no mask) means that the value is zero.

7.16.3.4. Modbus MASTER - Register blocks parameter block

The device has the ability to read data from the SLAVE devices using multi register queries. By default this feature is configured automatically, but can be switched to the manual mode.

Register blocks group has the following fields:
- – **Blocks config. mode**
  - • **automatic** - the device automatically creates a block of registers to be read using the list defined in the **Device channels** sub-menu. Then the **Block list** is informal only and cannot be edited.
  - • **manual** – the user must create a list of registers blocks using the **Block list** parameter
- – **Maximum block size** - occurs only for the **Blocks config.mode** = **automatic.** This parameter allows the user to limit the number of data registers to be read at once. It can be very useful when SLAVE devices have a limitation of max. number of registers read in a single frame.
- – **Block list** – occurs only for the **manual** mode, parameter description below in this **Chapter**,
- – **Avail. Input blocks** – read only submenu, allows to preview the created input blocks automatically or manually,
- – **Avail. Output blocks** – read only submenu, allows to preview the created input blocks automatically or manually,

Block list submenu

| | This button appears only for the manual configuration mode and allows the user to add a new block of registers to the list of register blocks. |

| | This button appears only for the manual configuration mode and allows the user to remove a block of registers from the list of register blocks. |

| | The arrows placed in the upper right corner of the screen allow switching between register blocks. The middle button allows a direct selection of a specific register block. |

The **Block list** parameters are:
– **Block type:**
  • **read Holding reg.** - HOLDING register **Group** for reading purpose,
  • **read INPUT reg.** - INPUT register **Group** for reading purpose,
  • **write HOLDING reg.** - HOLDING register **Group** for writing purpose,
– **Register size** - defines data size, can be set to:
  • **16-bit registers** - data is read/preset as 16 bit registers; this value can also be used for 32-bit registers reading/preseting. In such a case data is composed as two 16-bit registers and params: **First register** & **Last register** must select a minimum of 2 registers (e. g. **First register: 3h**, **Last register: 4h)**. The important parameter is then also **Ordering** - see **Chapter 7.16.3.3. Modbus MASTER - Register settings**),
  • **32-bit registers** - for 32-bit registers reading/preseting only, data is read as one 32-bit register,
– **First register** - value indicating the number of the first register of the block,
– **Last register** - value indicating the number of the last register of the block, for a single block with one 16-bit register the parameter **First register** and **Last register** must be the same number register.

The manual mode configuration of registers blocks introduces a freedom when setting **Block list** parameters. Take care to set **Block list** parameters according to **Input list** and **Output list** in the **Device channels** parameter block of the device (see above in this **Chapter**). If the user creates a block of registers to read / preset in which there were no registers appearing on the Input list and Output list in the **Device channels** parameter, this device in case of:
> - **read** - the read whole register block and registers undefined in the Input list menu will not be visible in the Logical channel in the Modbus mode which cannot be read either data from these registers and to use these registers to control and regulate the process,
> - **preset** - will send the frame to preset the entire block of registers and registers not defined in the **Output list** will not appear on the **External output** menu
> Warning! In this case a random value of these registers will be sent to Slave device, uncontrolled by the user.

7.16.4. Modbus - Example of Modbus protocol configuration in the device

The Modbus protocol sends data in the floating point format. In this connection the user should keep in mind that transmitted and received values using the Modbus protocol can be slightly different. This situation can happen when a decimal value can not be represented as floating point numbers, or when the least significant digit is eight orders of magnitude smaller than the most significant digit.

7.16.4.1. Input configuration of Modbus protocol in MASTER mode

See also: **Chapter 7.16.3. Modbus - MASTER mode**.

Task:
        The task is to configure a logical channel to read Slave registers (e.g. a temperature converter). We read register 1 which contains temperature, register 2 is a status register, register 3 is a decimal point register. The device has addresses 1 and 8. To set the Modbus configuration for the same devices, the Modbus template can be helpful, which will be created in this task.

Solution:
        First configure the device, then connect the SLAVE device to the **MultiCon CMC-N16** (in accordance with **7.167**).
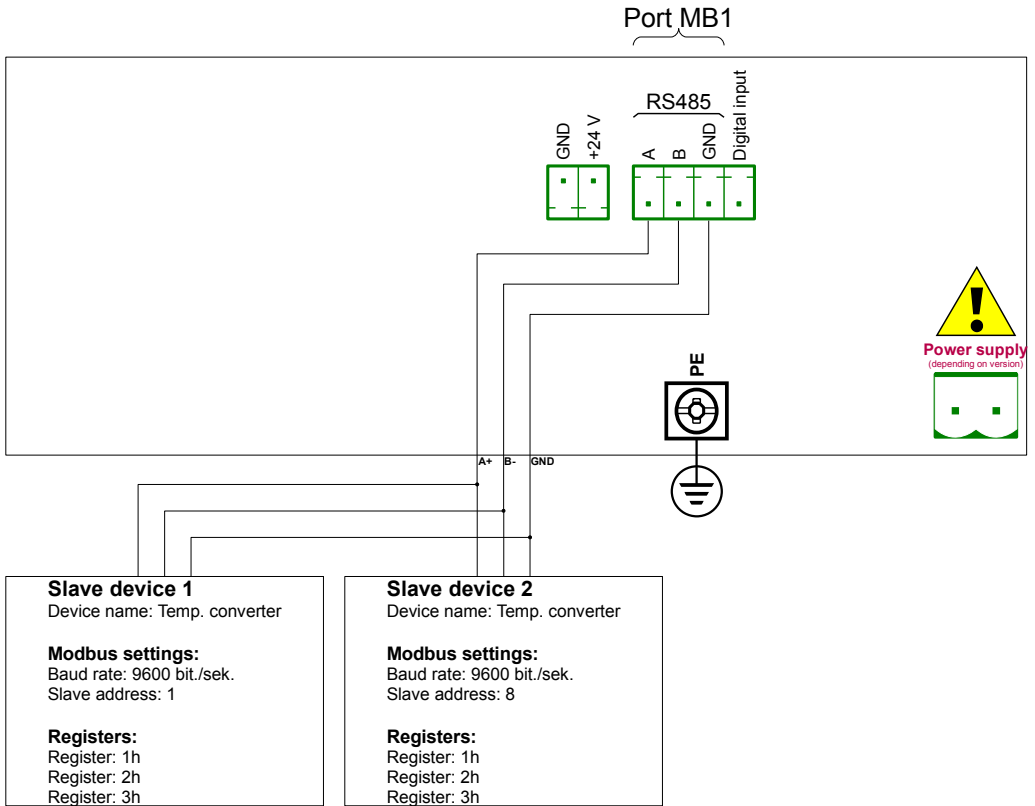
Fig. 7.167. Connection diagram for the Modbus port MB1

Description of the SLAVE device register shown in **Tab. 7.13**.

Register	Range	Register description
01h	-4000 ÷ +8500	The temperature measured with a resolution given to 2 decimal places (in the U2 code, excluding decimal point, for example, the value of 3523 means the temperature of 35.23°C)
02h	00h, 10h, 20h,40h, 80h	Status of the temperature contains the error code which must be interpreted as follows: **00h** - measuring correctly **10h** - flooding the interior of the sensor **20h** - damage temperature sensor **40h** - measured temperature is lower -40°C **80h** - measured temperature is upper 85°C
03h	0÷2	Specify the decimal point as a result, the value of 2 means that a point precedes the last 2 digits, the value of 0 means no decimal point

Tab. 7.13 Description of the SLAVE device registers

In the first step:
- touch the screen and press the **Menu** button,
- press the **Device configuration** button,
- enter the **Modbus** menu,
- using the arrows in the upper right corner choose **Port number**: **1** (**MB1** in accordance with **7.167**),
- **Mode** parameter set as **MASTER**,
- **Baud rate** parameter set as **9600 bits/sec.** (it should be the same as in the SLAVE device),
- **Format** parameter set as **8N1**,
- in **Request timeout** parameter write **0.2 sec.**,
- in **Request retrials** parameter write **3**,
- press the **Slave devices** button:
  - using the arrows in the upper right corner select the Slave device **Modbus address**, in this case it is **1**,
  - **Device type** parameter set as **defined**,
  - in the **Device name** parameter write **Temperature converter**,
  - in the **Device channels** parameter block press the **Input list** button to define the register to read,
    - if the **Input list** is empty, press the sign **'+'** to add an input channel (if the list has defined the input channels, use the arrows in the upper right corner to select the input channels),
    - in the **Value register** parameter set:
      - **Register type: HOLDING** - using **03h** function to read the holding registers as specified by the Modbus RTU,
      - **Register number: 1h** - the number of the read register
      - **Data format: 16 bits, signed** - the choice of a format depends on the SLAVE device register format,
      - **Data shift: 0**, because we do not want to move data read from the Slave device,
      - **Data mask: 0xFFFF**, which reads the whole value of the register, exit the **Value register** parameter menu,
    - in the **Decimal point** parameter select option: * exp (-point register) causes that there will be a new parameter to appear - **Decimal point register** that causes an automatic change of the decimal point, depending on the value of **Decimal point register**,
    - in the **Decimal point register** parameter set:
      - **Register type: HOLDING**,
      - **Register number: 3h**,
      - **Data format: 16 bits, unsigned**,
      - **Data shift: 0**,
      - **Data mask: 0x0003**, because we are interested only in the first 2 bits changing the register value, the value of which determines the decimal point, exit this menu after the settings,

  - **-HI- state** parameter set as **if register = value**,
    - in the **-HI- register** parameter set:
      - **Register type** parameter set as **HOLDING**,
      - in **Register number** parameter write **2h**,
      - **Data format** parameter set as **16 bits, unsigned**,
      - in **Data shift** parameter write **0**,

- in **Data mask** parameter write **0x0080**,
  ◦ in the **-HI- value** parameter for the signal **-HI-** set the value of **0x0080**,

- **-LO- state** parameter set as **if register = value**,
  ◦ in the **-LO- register** parameter set:
    – **Register type** parameter set as **HOLDING**,
    – in **Register number** parameter write **2h**,
    – **Data format** parameter set as **16 bits, unsigned**,
    – in **Data shift** parameter write **0**,
    – in **Data mask** parameter write **0x0040**
  ◦ in the **-HI- value** parameter for the signal **-HI-** set the value of **0x0040**,
- **-ERR- state** parameter set as **if register ≠ value**,
  ◦ in the **-ERR- register** parameter set:
    – **Register type** parameter set as **HOLDING**,
    – in **Register number** parameter write **2h**,
    – **Data format** parameter set as **16 bits, unsigned**,
    – in **Data shift** parameter write **0**,
    – in **Data mask** parameter write **0x00C0**,
  ◦ in the **-HI- value** parameter for the signal **-HI-** set the value of **0x0000**,

- **Blocks configuration mode** parameter set as **automatic**,
- **Max.block size** parameter set as **5**, because the slave device selected in this example allows reading the frame (using the Modbus function 03h) with up to 5 registers,

- move to the **Device templates** parameter block and press the **Save device template** button,
  ◦ in the **Save device template** menu, enter the template name, e.g. **Temperature converter** (stored template can be used to configure the device to a different address),
- using the arrows in the upper right corner of the screen, go to the **Modbus address**: **8**,
- press the **Load device template** button and in the template list select the template named **Temperature converter**, in such a fast way we can set all the parameters for the **SLAVE device 2** identical to **SLAVE device 1**,
- exit from the **SLAVE devices** menu,

An example of a Logical channel in the Modbus mode is shown in **7.9.11.4. Application of Logical channel in the Modbus mode**.

7.16.4.2. Configuration of the Modbus Input in the MASTER mode

See also: **Appendices 8.5. ETE, ETR, ETEC, ETRC – HARDWARE COMMUNICATION CONFIGURATIONS**.

Task:
The task is to configure the Slave device (e.g. a Flowmeter) output channel. The Slave device address is 5. we write data to register 1, 2, 4 and 8. For communication in this task we use the ETRC hardware communication configuration, in which the MB2 port is used.

Solution:
First we need to configure the device according to the following recommendations,

then connect the device in according with **7.168**.
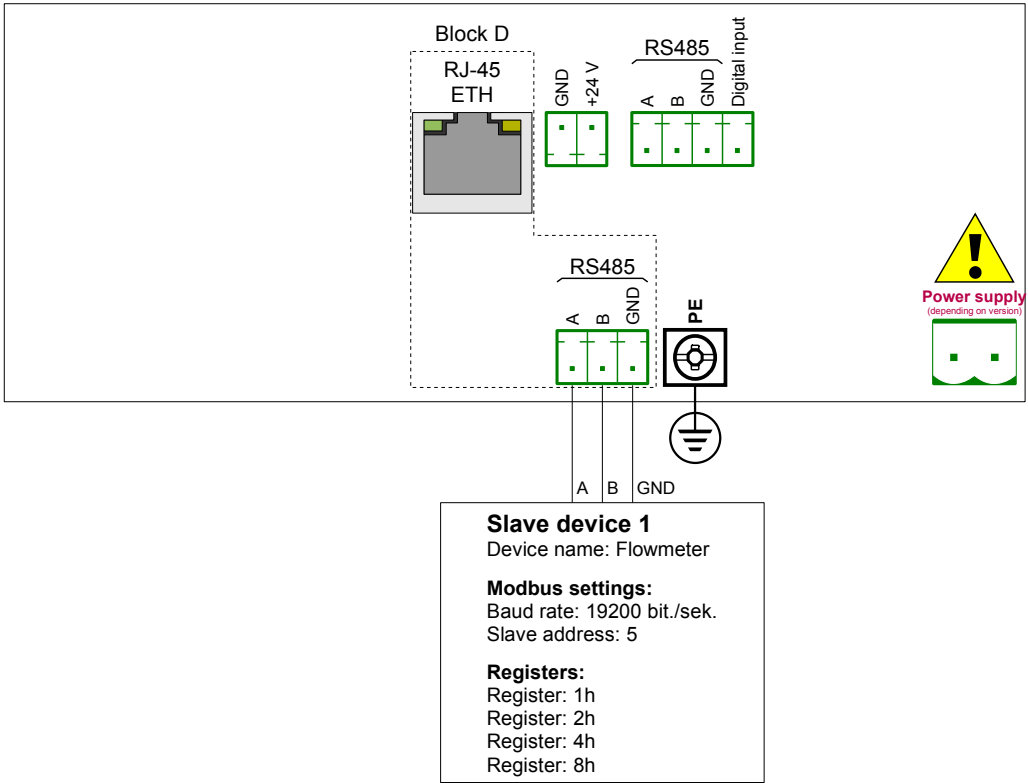
Fig. 7.168. Connection diagram for the Modbus MB2 port

16-bits Slave device registers description is shown in **Tab. 7.14**.

Register	Range	Register description
01h	0÷0x0FFFFh	Set point value (younger word of 32-bits float value)
02h	0÷0x0FFFFh	Set point value (older word of 32-bits float value)
04h	0÷1	Valve position: 0-closed, 1-open,
08h	0÷1	Alarm state: 0-off, 1-on,

Tab. 7.14 Description of 16-bits Slave device registers

In the first step:
- touch the screen and press the **Menu** button,
- press the **Device configuration** button,
- enter the **Modbus** menu,
- using the arrows in the upper right corner choose **Port number**: **1** (**MB1** in

accordance with **7.168**),
– **Mode** parameter set as **MASTER**,
– **Baud rate** parameter set as **19200 bits/sec.** (it should be the same as in the SLAVE device),
– **Format** parameter set as **8N1**,
– in **Request timeout** parameter write **0.2 sec.**,
– in **Request retrials** parameter write **3**,
– press the **Slave devices** button:
  • using the arrows in the upper right corner select **Modbus address** of the Slave device, in this case: **5**,
  • **Device type** parameter set as **defined**,
  • in the **Device name** parameter write **Flowmeter**,
  • in the **Device channels** parameter block press the **Output list** button to define the register to write,
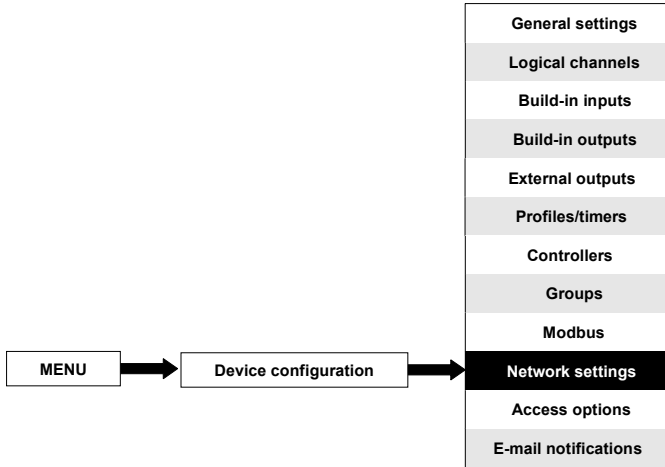   ◦ if the **Output list** is empty, press the sign **'+'** to add the output channel (if the list has defined the output channels, use the arrows in the upper right corner to select the input channels),
    ◦ **Output active** parameter set as **yes**,
    ◦ **Control type** parameter set as **as a linear output**,
    ◦ in **Output register** submenu:
      – in **Register number** parameter write **1h**,
      – **Data format** parameter set as **32 bits, float** (this depends on the register format in the Slave device),
      – **32-bit reading** parameter set as **two 16-bit registers**,
      – **Ordering** parameter set as **CDAB**, because first the younger word part is sent (Register 1h), then the older part (Register 2h),
    ◦ press the sign **'+'** and then:
    ◦ **Output active** parameter set as **yes**,
    ◦ **Control type** parameter set as **as a relay**,
    ◦ in **Output register** submenu:
      – in **Register number** parameter write **4h**,
      – **Data format** parameter set as **16 bits, unsigned** (this depends on the register format in the Slave device),
      – in **Data mask** parameter write **0x0001**,
      – in **Data shift** parameter write **0**,
    ◦ press the sign **'+'** and then:
    ◦ **Output active** parameter set as **yes**,
    ◦ **Control type** parameter set as **as a relay**,
    ◦ in **Output register** submenu:
      – in **Register number** parameter write **8h**,
      – **Data format** parameter set as **16 bits, unsigned** (this depends on the register format in the Slave device),
      – in **Data mask** parameter write **0x0001**,
      – in **Data shift** parameter write **0**,
  • **Blocks config.mode** parameter set as **manual**,
  • in **Block list** submenu:
    ◦ **Block type** parameters set as **write HOLDING reg.**,
    ◦ **Register size** parameter set as **16-bit registers** (according to register format in the Slave device),
    ◦ in **First register** parameter write **1h**,
    ◦ in **Last register** parameter write **2h**,

- ◦ create next block by pressing **'+'** sign,
- ◦ **Block type** parameter set as **write HOLDING reg.**,
- ◦ **Register size** parameter set as **16-bit registers** (according to register format in the Slave device),
- ◦ in **First register** parameter write **4h**,
- ◦ in **Last register** parameter write **4h**,
- ◦ create next block by pressing **'+'** sign,
- ◦ **Block type** parameters set as **write HOLDING reg.**,
- ◦ **Register size** parameter set as **16-bit registers** (according to register format in the Slave device),
- ◦ in **First register** parameter write **8h**,
- ◦ in **Last register** parameter write **8h**,
- • move to the **Device templates** parameter block and press the **Save device template** button,
  - ◦ in the **Save device template** menu, enter the template name, e.g. **Write do device** (the stored template can be used to configure the device to a different address).

7.17. NETWORK AND REMOTE DISPLAY SETTINGS

General settings
Logical channels
Build-in inputs
Build-in outputs
External outputs
Profiles/timers
Controllers
Groups
Modbus
Network settings
Access options
E-mail notifications

MENU ➡ Device configuration ➡

ⓘ This button is visible only for a device equipped with the any hardware communication configuration in block D.

Network settings allows the user to configure the network settings by downloading and visualizing the data from the device through the Ethernet connection. **Remote display** settings allow the user to configure the **MultiCon CMC-N16** and to display any data with an external PC.

The parameters of the **Network settings** menu are:
- **DHCP** (Dynamic Host Configuration Protocol) - allows a device to be configured automatically, eliminating the need for intervention by a network administrator,
  - **disabled** - **DHCP** is disabled, the user needs to manually enter an **IP address** and **Subnet mask** in the following fields, and a **Default gateway** address if required.
  - **enabled** - the network settings are automatically generated by the DHCP server, after setting the DHCP, it takes several seconds before the IP address is obtained from the DHCP server, if the user sets this option, other parameters in this menu are invisible,
- **IP address** - this parameter is visible only if the **DHCP** parameter is set as **disabled**, the user may enter a static IP address,
- **Subnet mask** - this parameter is visible only if the **DHCP** parameter is set as **disabled**, sets a range of IP addresses that can be accessed,
- **Default gateway** - this parameter is visible only if the **DHCP** parameter is set as **disabled**, and allows the user to enter a gateway IP address for use when the device is to communicate outside the local network,
- **DNS server 1** – this parameter is visible only if the **DHCP** parameter is set as **disabled**, and allows the user to enter a DNS server 1 address that manages, maintains and processes Internet domain names and their associated records,
- **DNS server 2** – this parameter is visible only if the **DHCP** parameter is set as **disabled**, and allows the user to enter a DNS server 2 address that manages,

maintains and processes Internet domain names and their associated records,
– **Remote display** parameter block - this parameters lets the user configure the **MultiCon CMC-N16** with an external PC to display the screen of **MultiCon CMC-N16** on the PC monitor, the parameters of this block are:
  • **IP address** - IP address of external PC,
  • **Screen number** – can bet set from 0-9 in the **MultiCon CMC-N16** allowing the PC to display up to 10 device screens,
– **Server port** – the parameter allowing to set the web server port number where internet service works. Allows to connect with the device using a web browser on a selected port. It is also needed while configuring a remote connection with the device which is outside the local network. It is not recommended to set number ports which are reserved to standard services.
– **NTP server address** – the parameter allowing to set the NTP server address, download the current time from it and, if necessary, correct system clock display of the device. The device will connect to the NTP server every 15 minutes and correct its time according to the downloaded server time. This parameter contains one button:
  • **Synchronize with NTP** – this button allows manual synchronization of device clock with time downloaded from NTP server. It also allows to check validation specified NTP server and its quality.

(**i**) If the **DHCP** parameter is set as **disabled,** the parameters of **IP address** and **Subnet mask** must be configured correctly, depending on the local network settings which will work with the device. The user should contact the network administrator in case of errors in communication.

The current network settings are visible in the **Device Information** menu (see **Chapter 7.5**).

(**i**) To establish a connection to remote display of the device, port numbers 6000-6009 (depending on screen number) must be unlocked in the software and hardware firewall.
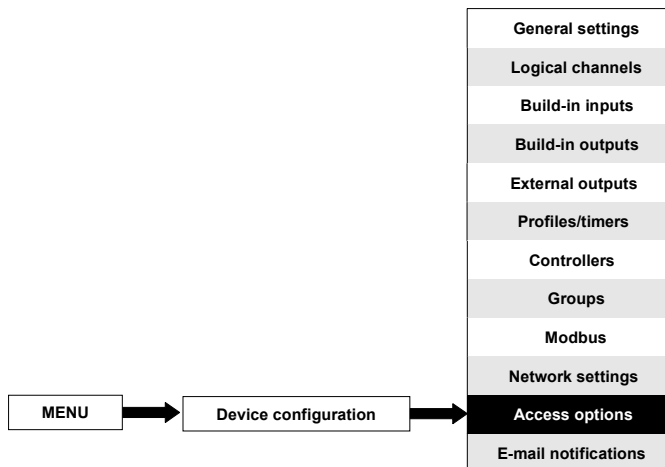
(**i**) To establish a connection between **MultiCon CMC-N16** and another device using the Modbus TCP protocol, port number 502 must be unlocked in the software and hardware firewall.

(**i**) To connect **MultiCon CMC-N16** with another device using the HTTP protocol, port number 80 must be unlocked in the software and hardware firewall.

7.18. ACCESS OPTIONS – MULTILEVEL ACCESS FUNCTIONALITY

General settings
Logical channels
Build-in inputs
Build-in outputs
External outputs
Profiles/timers
Controllers
Groups
Modbus
Network settings
Access options
E-mail notifications

MENU ➡ Device configuration ➡

To prevent an accidental or unauthorized change of the settings the user can set a single level or **MultiLevel Access** protection. Both of them can be activated in the **Access options** menu.

A single level access protection requires to enter a password (see **7.169**) in order to login as the Admin and obtain access to all the settings of the device.

A **MultiLevel Access** protection requires the user to go through the Authentication procedure (see **7.170**) in order to log in and get access to the appropriate permission. It is done by choosing the user (see **7.171**) and passing his password. Only the Administrator can access the **Access options** menu (see **7.172**).

A password can be activated in two different ways depending on the level access protection. In the **Access options** menu by pressing the button next to the **Access password** label (for single-level mode) or in the **Users list** menu (for multi level mode) by typing the user's password in the text box (see **7.1**, **7.2**, **7.3**). After closing the text box, in place of the entered password, 8 asterisks will be displayed independently from the password length.

Removing the password is even easier, it is done by leaving the text box empty. After that, the field next to the **Access password** label will also be empty.

When the user passes through the login process successfully, on the an opened padlock with the letter "A" appears on the information bar (if the single level mode is set or the user is logged as Administrator in the multi level mode) appears (see **7.173a**). Otherwise a number of user will be displayed (from 2 to 16) instead of opened padlock (see **7.173b**).

Only the administrator can freely configure the unit without restrictions.

Furthermore, the administrator configures other users accounts for **MultiLevel Access**. Those users can configure the device according to Administrator's restrictions. Entering an incorrect password or attempt to enter a restricted function will cause denial (see **7.174**).

Only one user can be logged in at a time. The user can be logged out automatically after one minute for the single level mode or after the time set by the Administrator (in both cases the time is measured from the last user's interaction with the device). The user also can be logged out manually by touching the padlock icon on the information bar and confirming a

logout (see **7.175**).

For the **MultiLevel Access** mode there is a possibility to log in by connecting your own USB flash drive to the device, later called the Hardware key. The Hardware key allows the user to log in without using a password but only by plugging the key into the device. Removing the hardware key causes an immediate logout. One key can be assigned to one user only, and also the user can only have one key.

ⓘ Hardware key options are always available but they are active only in devices which have the Access Dongle licence activated (for more information about the licence key see **Chapter 7.5. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION**). In those devices which don't have the Hardware key licence activated the following message will be displayed in the **User list** menu:

<div align="center">"Enter licence to use Dongle"</div>

Access options parameters are:

- **Passwords masking** – by default, editing of all device settings that require confidentiality (passwords and service options) is done using edit dialogues (see Figures 7.1, 7.2, 7.3 and 7.4) in which the entered letters aren't visible (masked with the "*" sign). This setting allows disable masking and enable preview of just typed or previously entered characters. The setting doesn't affect visibility of passwords on the parameter list of any menu, which always remain masked,
- **Access mode** – allows to select the protection mode against an unauthorized usage,
  - **single level** – protects the whole menu against unauthorized changes with one password. The user which is currently logged in can freely configure the device until a manual or automatic logout,
  - **MultiLevel** – allows to configure settings for Administrator and up to 15 users with different access levels and independent passwords,
- **Access password** – password which is required to log in while using the single-level access mode (unless it's left empty),
- **Access timeout** – time in minutes which will elapse from the last interaction with the device to an automatic logout. The value of 0 means that the user will be logged out immediately after closing the protected menu. Each new attempt to enter the protected menu will require a password once again; Available only for the **MultiLevel Access** mode,
- **Users list** – menu containing Administrator's and user's access settings; Visible only for the **MultiLevel Access** mode,
  - **Name** – used during the authentication process,
  - **Password** – used for authentication of aselected user, can be left empty,
  - **Permissions** – menu containing device's functionalities which can be assigned to the users. Every user will get access to only those functionalities which were earlier assigned to his permission list,
  - **Access dongle** – number of the hardware key assigned to the user,
  - **Assign dongle** – button which assigns the hardware key (currently connected flash disk) to the selected user,
  - **Remove dongle** – button which removes the previously assigned hardware key,
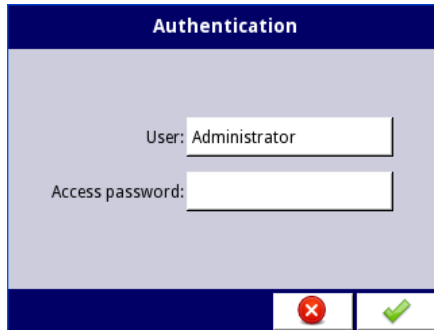
Fig. 7.169. Single-level password entering window

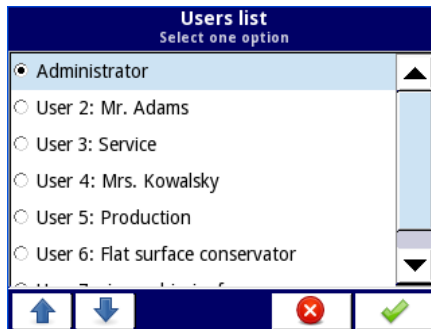*Fig. 7.170. **MultiLevel Access** password entering window*
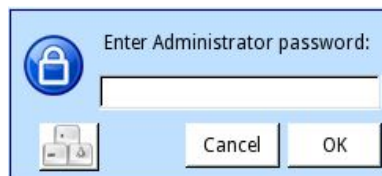

Fig. 7.171. User selection window

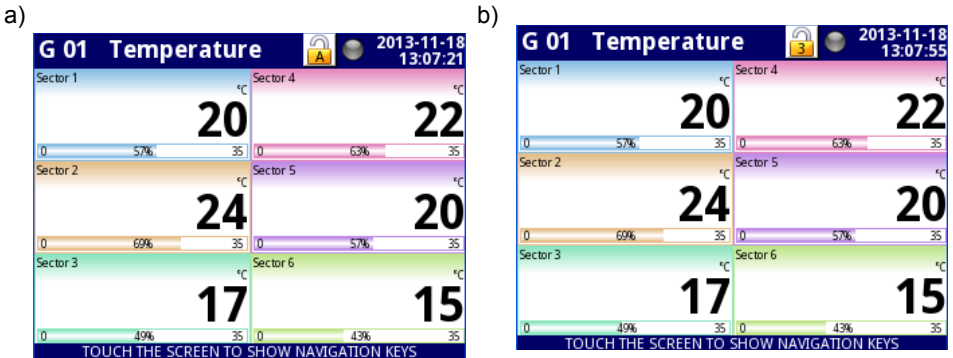*Fig. 7.172. Administrator password window*

a)

b)

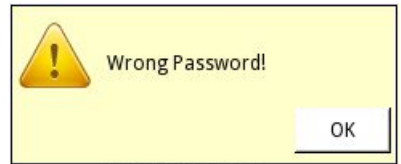*Fig. 7.173. Example of screen view with user logged in as: a) Administrator   b) User 3*

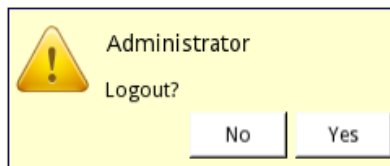*Fig. 7.174. Access denied and Wrong password window*

Fig. 7.175. User logout window

 The button which calls the keyboard allowing to type the user's password.

The settings of the single-level access protection are stored in the main configuration file. They are also copied to other devices together with the main configuration file.

The settings of the **MultiLevel Access** protection have to be saved in the independent file. They are not stored in the main configuration file. They also cannot be removed by restoring the default configuration as it is shown in **Chapter 7.5. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION** The only way to transfer those settings to another device is save them on an external memory as an independent permission file (see **Chapter 7.4. FILES MANAGEMENT**).

7.19. E-MAIL NOTIFICATIONS

General settings
Logical channels
Build-in inputs
Build-in outputs
External outputs
Profiles/timers
Controllers
Groups
Modbus
Network settings
Access options

MENU ➤ **Device configuration** ➤ **E-mail notifications**
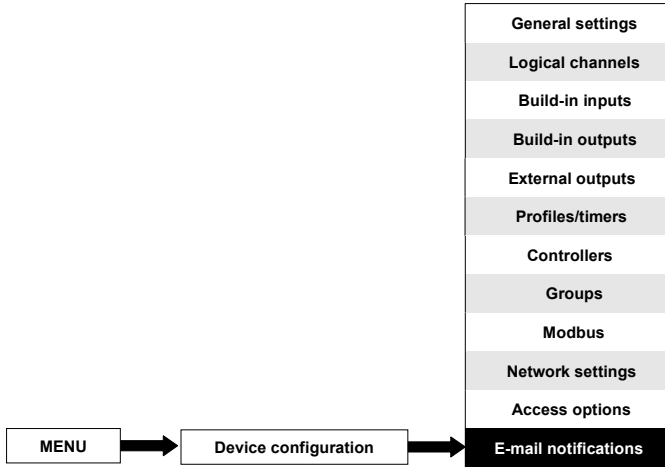
ⓘ This button is visible in all devices but this functionality will work only in the ones with the E-mail notifications licence. Devices without the E-mail notification licence will display the following message in the E-mail notifications menu:
"Available after entering licence"

      The **E-mail notifications** menu allows to send messages using e-mail. Each message can contain any description and attachment with selected **Logical channels** or **Groups**. E-mail notifications can be used as devices state report when an alarm or a specified signal occurs. This report will be sent to all the declared recipients.
      In case of any sending error (e.g. connection with network error) messages can be saved in a queue where they are waiting to be sent, however sending and saving to the queue can not occur more often then once a minute which means that any faster than one minute message trigger will be ignored. Additionally, this queue can not be longer than 10 messages and when it fills up the appearance of a new message will delete the oldest one and the new one will be placed into the queue. When the connection is re-established, the device will send messages in the queue as one e-mail with all the awaiting attachments. All the messages from all the queues will be lost when the device is switched off or when power fails.
      There are 32 available **Messages** which can be configured independently and each of them can be sent to several recipients.

Attachment with **Logical channels** has the following structure:
File name:
serial_number**_Ch_**data**_**hour**_**minute**_**second**_msgID=**Message_number_Suffix**.csv**
File content – first row:
      "Channel Number","Channel Name","Value","Unit"
File content – next rows:
      selected_channel_number**,"**Channel_name**",**Channel_value**,"**unit**"**
      Each Logical channel is stored in a new line of the file.
Attachment with **Groups** has the following structure:

File name:

serial_number_**Gr**_data_hour_minute_second_**msgID=**Message_number_Suffix**.csv**

File content – first row:

"Group Number","Group Name","Channel Number","Channel Name","Value","Unit"

File content – next rows:

Group_number**,"**Group_name**",**Channel_number**,"**Channel_name**",**Channel_value**,"**Unit**"**

Each Logical channel from each **Group** is stored in a new line of the file.

Where: in **black** constant elements are written,

in red are written the elements which are dependent on triggering and the device state but all of these elements must be included.

The content of the attached file can be slightly different depending on the **Decimal separator** parameter (see description of the **Decimal separator** parameter in **Chapter 7.8. GENERAL SETTINGS**).

In case when the channel selected to send I disabled (or a **Group** with a disabled Channel) the name of this Channel will be **"disabled"** and its unit will be **"null"**.

Alarm states (-ERR-, -LO-, -HI- etc.) of Logical channels are not included in the attachment but only the current values.

The parameters on the e-mail notifications menu are:

– **E-mail configuration** – menu which contains the sender account parameters,
  • **Mode** – switches e-mail notifications on and off:
    ◦ **disabled**,
    ◦ **enabled**,
  • **Sender name** – text which will be displayed in the Name field in the user's e-mail account.
  • **Sender address** – sender e-mail address which will be used by the device to send messages to recipients,
  • **Login** – sender e-mail account login,
  • **Password** – sender e-mail account password (every sign is displayed as an asterisk,
  • **SMTP server** – sender e-mail account server address,
  • **SMTP port** – sender e-mail account server port number,
  • **Security** – security mode which is used by the sender's e-mail server,
    ◦ **none** – no security,
    ◦ **STARTLS** – establishes TLS encryption in a network connection when communication with server is realized on unencrypted port,
    ◦ **SSL/TLS** – the server uses SSL or TLS encrypted communication on the selected port,
  • **Retry every [min]** – parameter which describes how much time must elapse before the device will try to send messages again after failure,
– **Messages** – menu which defines up to 32 messages,
  • **Mode** – switches on and off a particular message,
    ◦ **disabled**,
    ◦ **enabled**,
  • **Recipient(s)** – parameter which contains e-mail address (addresses) of the recipient (recipients). All recipients must be separated from each other with a semicolon or semicolon and space signs but there can not be more than 78 signs in total,

- **Subject** – subject of the message,
- **Message text** – additional message text which can be used as a description of triggering occurrence,
- **Attachment** – parameter which defines what should be attached to the message. Every message must have an attachment,
  - ◦ **disabled** – attachment will not be created and message will not be sent. When the Attachment parameter is set as disabled all the parameters described below are not visible,
  - ◦ **channels** – attachment will contain Logical channels selected by the user,
  - ◦ **groups** – attachment will contain **Groups** selected by the user,
- **Selected channels** – parameter which is visible only when the **Attachment** parameter is set as channels. Allows to select logical channels from the list which must be included in the attachment. Until no channel is selected in this parameter, all the parameters described below will not be visible,
- **Selected groups** – parameter which is visible only when the Attachment parameter is set as **Groups**. Allows to select **Groups** from the list which must be included in the attachment. Until no **Group** is selected in this parameter, all the parameters described below will not be visible,
- **Retry when failed** – parameter which defines if the messages the device was unable to send will be stored in a queue or not,
  - ◦ **yes** – messages will be stored in a queue and sent again,
  - ◦ **no** – messages will not be stored and sent again,
- **Expiry period** – this parameter is visible only when the **Retry when failed** parameter is set as yes. Defines for how long (in hours) a message will be stored in a queue since its first send failure,
- **Triggering channel** – parameter which contains a list of channels where the selected one will be a triggering source for the message,
- **Trigger mode** – parameter which allows to select the message triggering method,
  - ◦ **rising edge** – switching value in the channel from less or even to zero (≤ 0) to value greater than zero (> 0) will trigger the message,
  - ◦ **falling edge** – switching value in the channel greater than zero (> 0) to value less or equal zero (≤ 0) will trigger the message.

8. APPENDICES

⚠️ **All the connections must be made while the power supply is disconnected !**

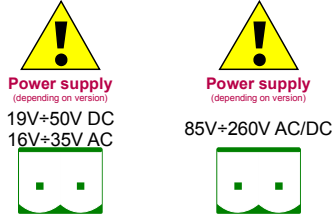## *8.1. PS32, PS42 – HARDWARE POWER SUPPLY CONFIGURATIONS*

Fig. 8.1 Hardware power supply configurations (from left side): PS32, PS42

Most important parameters of PS32, PS42 hardware power supply configurations:

	PS32	**PS42**
Number of inputs/outputs	4	4
Power supply	19V...**24**...50V DC; 16V...**24**...35V AC	85V...**230**...260V AC/DC; 50 ÷ 60 Hz
Sensor power supply output	24V DC ±5% / max. 200mA	
Digital input	0...15...24V DC with galvanic insulated (low state: 0÷1V, high state:8÷24V) power consumption: 7,5 mA / 24V, insulation: 1min @ 500V DC.	
Interface	RS-485, Modbus RTU, 1200bit/s ÷ 115200bit/s	

Tab. 8.1 Technical data for the hardware power supply configurations PS32, PS42

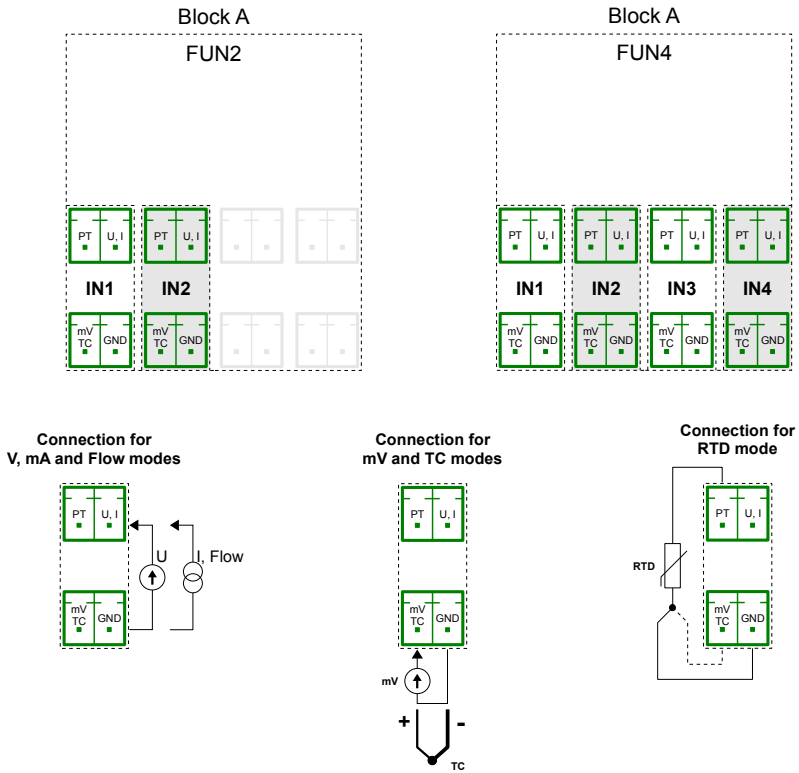## 8.2. FUN2, FUN4 - UNIVERSAL HARDWARE INPUT CONFIGURATIONS WITH FLOWMETER MODE

Fig. 8.2. Hardware input configurations for Block A

The **8.2** shows terminals placement in Block A hardware input configurations. Inputs are gathered into groups to make connections easier. All ground terminals of a particular hardware configuration are common, but separated from power supply and other hardware configurations.

Most important parameters of **FUN2** and **FUN4** hardware input configurations:

	FUN2		FUN4	
Number of inputs	2		4	
Hardware limitations in:				
current mode	-2mA÷30mA		-2mA÷30mA	
voltage mode	-1V÷12V -15÷30mV -15÷120mV -50÷650mV		-1V÷12V -15÷30mV -15÷120mV -50÷650mV	
thermocouple mode	-15mV÷30mV -15mV÷120mV		-15mV÷30mV -15mV÷120mV	
RTD mode	0÷325Ω 0÷3250Ω		0÷325Ω 0÷3250Ω	
Hardware resolution in:				
current mode	1µA		1µA	
voltage mode for range -1÷12 V for range -15÷30 mV for range -15÷120 mV for range -50÷650 mV	1 mV 2 µV 4 µV 40 µV		1 mV 2 µV 4 µV 40 µV	
thermocouple mode for range -15÷30mV for range -15÷120mV	2µV 4µV		2µV 4µV	
RTD mode for range 0÷325Ω for range 0÷3250Ω	0,01Ω 0,2Ω		0,01Ω 0,2Ω	
Permissible Long time overload	20%		20%	
Covered MultiCon CMC-N16 measurement ranges in:				
	type	max	type	max
current mode	0÷20mA 4÷20mA	0÷22mA 4÷22mA	0÷20mA 4÷20mA	0÷22mA 4÷22mA
voltage mode	0÷5V 1÷5V 0÷10V 2÷10V -10÷25mV -10÷100mV	0÷6V 0÷6V 0÷12V 0÷12V -10÷30mV -10÷120mV	0÷5V 1÷5V 0÷10V 2÷10V -10÷25mV -10÷100mV	0÷6V 0÷6V 0÷12V 0÷12V -10÷30mV -10÷120mV
thermocouple mode	0÷600mV	-50÷650mV	0÷600mV	-50÷650mV
RTD mode				

	FUN2	FUN4
resistance mode	K,S,J,T, N, R, E, L(GOST), B, C Pt100, Pt'100, Pt'50, Pt500, Pt'500, Pt1000, Cu50, Cu'50, Cu100, Cu'100, Ni100, Ni500, Ni1000 0÷300Ω 0÷3000Ω	
Max input voltage	5 V	5 V
Totalizer capacity	1 000 000 000 000	1 000 000 000 000
Connection method in **RTD** mode:	2 and 3 wire (switched manually)	2 and 3 wire (switched manually
Input impedance in:		
current mode	< 65Ω (typ. 30Ω)	< 65Ω (typ. 30Ω)
voltage mode	> 100kΩ (while maintaining correct polarization)	> 100kΩ (while maintaining correct polarization)
thermocouple mode	$\geq$ 3MΩ	$\geq$ 3MΩ
RTD mode	$\approx$ 4kΩ	$\approx$ 4 kΩ
Hardware accuracy **current inputs**	0,1% @ 25˚C for all ranges	0,1% @ 25˚C for all ranges
voltage inputs	0,1% @ 25˚C for ranges: 0÷5V, 1÷5V, 0÷10V, -10÷100mV, 0÷600mV 0,15% @ 25˚C for range: -10÷25mV	0,1% @ 25˚C for ranges: 0÷5V, 1÷5V, 0÷10V, -10÷100mV, 0÷600mV 0,15% @ 25˚C for range: -10÷25mV
thermocouple inputs	0,15% @ 25˚C for range 30mV 0,1% @ 25˚C for range 120mV	0,15% @ 25˚C for range 30mV 0,1% @ 25˚C for range 120mV
resistance inputs	0,1% @ 25˚C for ranges: 0÷300Ω, 0÷3000Ω	0,1% @ 25˚C for ranges: 0÷300Ω, 0÷3000Ω
Temperature stability	50 ppm/˚C	50 ppm/˚C
Sapling period in:		
current mode **voltage mode** thermocouple mode **RTD mode** **resistance mode**	535ms 535ms 535ms 1410ms 1410ms	535ms 535ms 535ms 1410ms 1410ms
Isolation strength	1 min @ 500V AC	1 min @ 500V AC

Tab. 8.2 Technical data for the FUN2 and FUN4 hardware input configurations

type	range*	resolution	full range	full range accuracy		limited range	limited range accuracy	
	[mV]	[µV]	[°C]	[°C]	[%]	[°C]	[°C]	[%]
K	-10÷120	4	-200÷1370	±8	±0,51	-100÷1370	±4	±0,27
S	-10÷30	2	-50÷1768	±7,5	±0,41	0÷1768	±6	±0,34
J	-10÷120	4	-210÷1200	±6,3	±0,44	-100÷1200	±3	±0,23
T	-10÷30	2	-200÷400	±1,9	±0,31	-100÷400	±1	±0,20
N	-10÷120	4	-200÷1300	±12	±0,80	-100÷1300	±5,9	±0,42
R	-10÷30	2	-50÷1768	±8,6	±0,47	0÷1768	±5,5	±0,31
E	-10÷120	4	-200÷1000	±4,7	±0,40	-100÷1000	±2,7	±0,25
L	-10÷120	4	-200÷800	±4,4	±0,44	-100÷800	±2,5	±0,28
B	-10÷30	2	250÷1820	±15	±0,96	480÷1820	±6	±0,45
C	-10÷120	4	50÷2290	±13	±0,51	50÷1900	±9,2	±0,5

* taking into account the operation of the device in environment temperature range of -20'C÷50'C.

Tab. 8.3 Thermocouple measurement accuracy for the FUN2 and FUN4 hardware input configurations in TC mode

type	norm
K	PN-EN 60584-1:2014
S	PN-EN 60584-1:2014
J	PN-EN 60584-1:2014
T	PN-EN 60584-1:2014
N	PN-EN 60584-1:2014
R	PN-EN 60584-1:2014
E	PN-EN 60584-1:2014
L	GOST R 8.595:2004
B	PN-EN 60584-1:2014
C	PN-EN 60584-1:2014

Tab. 8.4 Thermocouple norm for the FUN2 and FUN4 hardware input configurations in TC mode

type	range	temperature range	resolution	accuracy	accuracy
	[Ω]	[°C]	[Ω]	[°C]	[%]
Pt100	0÷325	-100÷600	0.01	±1	±0,14
Pt500	0÷3250	-100÷600	0.1	±2	±0,3
Pt1000	0÷3250	-100÷600	0.1	±1	±0,14
Pt'50	0÷325	-200÷600	0.01	±2	±0,25
Pt'100	0÷325	-200÷600	0.01	±1	±0,12
Pt'500	0÷3250	-200÷600	0,1	±2	±0,25
Cu50	0÷325	-50÷200	0.01	±1,8	±0,73
Cu100	0÷325	-50÷200	0.01	±0,9	±0,37
Cu'50	0÷325	-200÷200	0.01	±1,6	±0,4
Cu'100	0÷325	-200÷200	0.01	±0,8	±0,2
Ni100	0÷325	-60÷180	0.01	±0,7	±0,3
Ni500	0÷3250	-60÷180	0.1	±1,4	±0,58
Ni1000	0÷3250	-60÷180	0.1	±0,7	±0,3

Tab. 8.5 RTD measurement accuracy for the FUN2 and FUN4 hardware input configurations in RTD mode

type	norm
Pt100	PN-EN 60751:2009
Pt500	PN-EN 60751:2009
Pt1000	PN-EN 60751:2009
Pt'50	GOST 6651-94 (W_{100}=1,3916)
Pt'100	GOST 6651-94 (W_{100}=1,3916)
Pt'500	GOST 6651-94 (W_{100}=1,3916)
Cu50	GOST 6651-94 (W_{100}=1,3916)
Cu100	GOST 6651-94 (W_{100}=1,3916)
Cu'50	PN-83M-53852 (W_{100} =1,4280)
Cu'100	PN-83M-53852 (W_{100} =1,4280)
Ni100	PN-EN 60751:2009
Ni500	PN-EN 60751:2009
Ni1000	PN-EN 60751:2009

Tab. 8.6 RTD norm for the FUN2 and FUN4 hardware input configurations in RTD mode
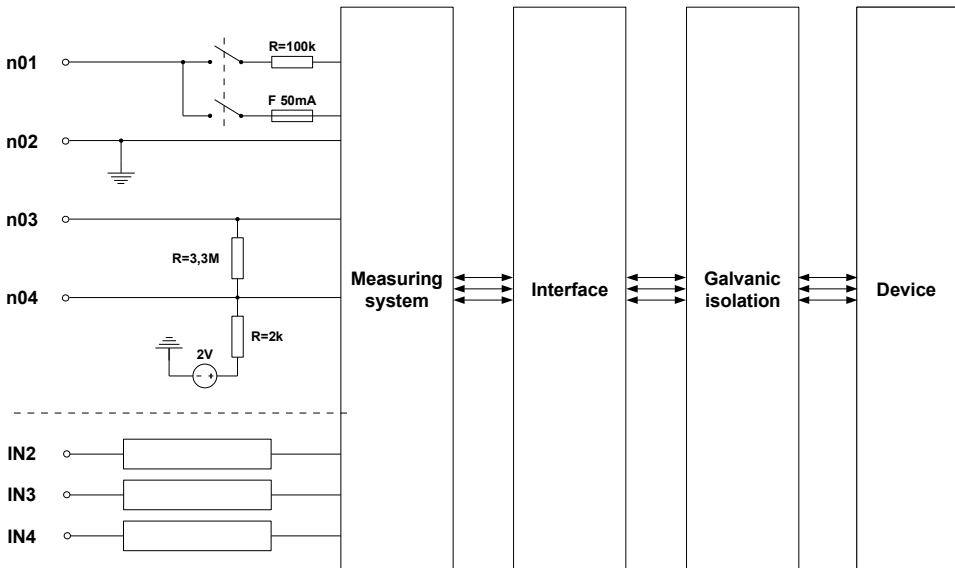
Fig. 8.3. Internal structure of the universal hardware input configuration with flowmeter mode
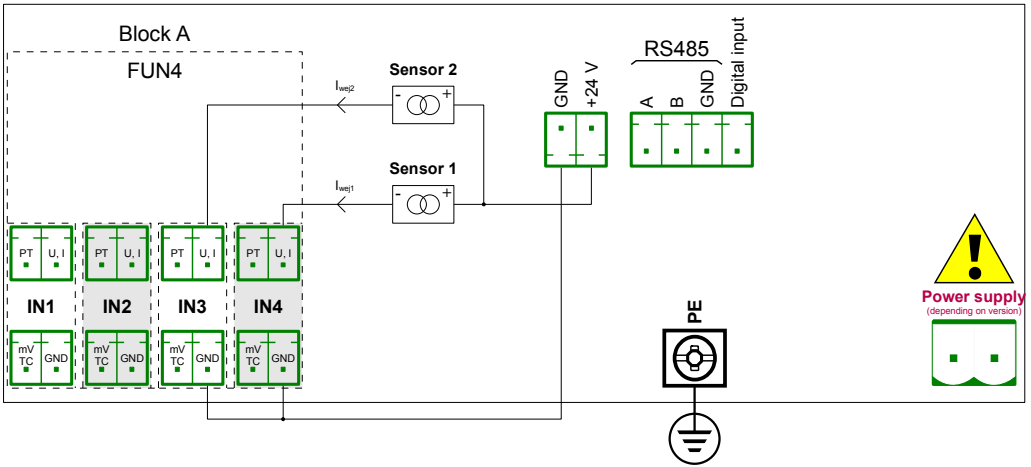
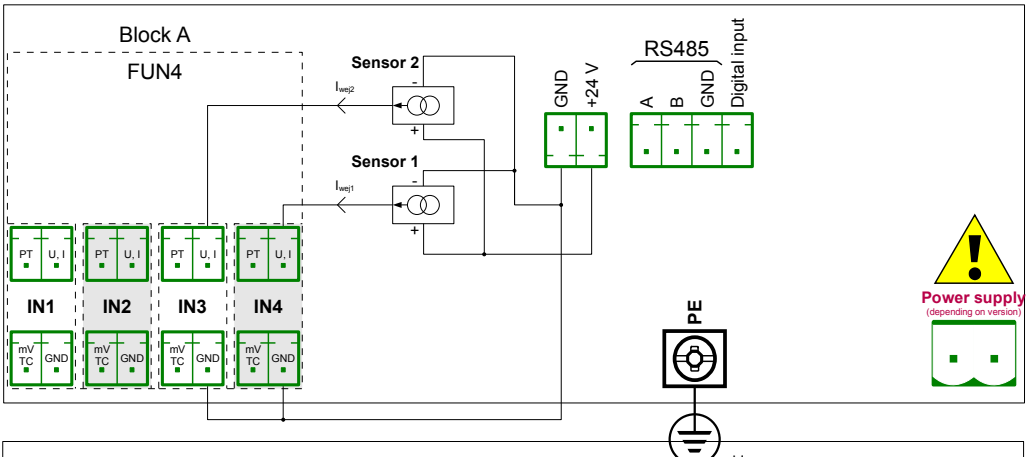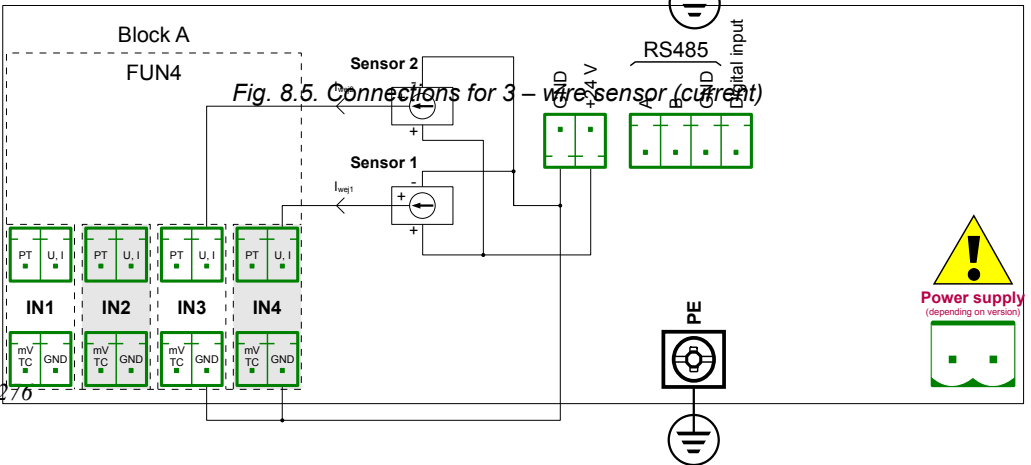*Fig. 8.4. Connections for 2 – wire sensor (current)*

Fig. 8.5. Connections for 3 – wire sensor (current)

Fig. 8.6. Connections for 3 – wire sensor (voltage)

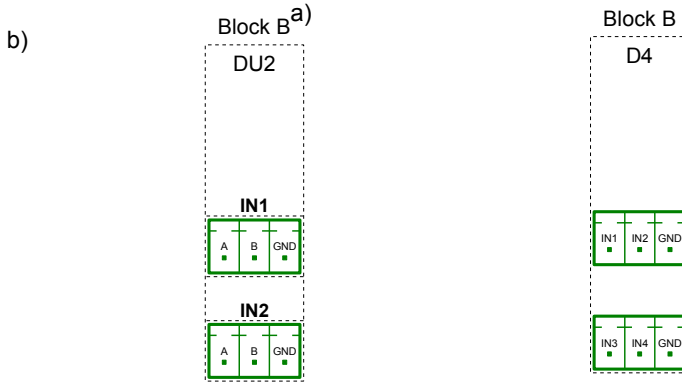## 8.3. DU2, D4 – OPTOISOLATED HARDWARE DIGITAL CONFIGURATIONS

Fig. 8.7. Optoisolated hardware digital configurations for Block B

8.8a shows placement of connectors of optoisolated hardware digital configuration of universal counters **DU2**. Inputs in this configuration can work independently in modes: binary input, universal counter and tachometer.

8.8b shows placement of connectors of optoisolated hardware digital configuration of digital inputs **D4**.

Most important parameters of **DU2** and **D4** hardware configurations:

	DU2	D4
Number of inputs	2 groups of counter inputs (2 groups of 2 inputs optoisolated from other signals)	4 (1 group of 4 inputs, optoisoladted from other signals)
Terminal description	**IN[n] A**, **IN[n] B** – pulse, counting inputs **GND** – common input for each counter **[n] =** 1, 2	**IN1**, **IN2**, **IN3**, **IN4** – digital inputs **GND** – common input
Inputs signal voltage levels: **logical LOW state** **logical HIGH state**	 | Uin | < 1 V | Uin | ≥10 V	 | Uin | < 1 V | Uin | ≥10 V
Max input voltage	30 V	30 V
Max input frequency for quad signals * **	5 kHz	-
Max input frequency for other signals **	8 kHz	5 kHz
Counter capacity	4 503 599 627 970 495 pulses (52 date bits)	-
Input current consumption	about 12 mA @ 24 V about 5 mA @ 10 V	about 12 mA @ 24 V about 5 mA @ 10 V about 2 mA @ 5 V
Insulation strength (between individual measurement inputs)	2 kV	-
Protection	50 mA auto-reset fuse	-
Isolation strength (between internal circuits and measurement inputs)	1 min @ 500 V AC	1 min @ 500 V AC

* For signals with frequency greater than 3 kHz it is recommended to keep duty cycle around 50% and phase shift 90°.
** Shift between input signals edges has to be at least 100 μs.
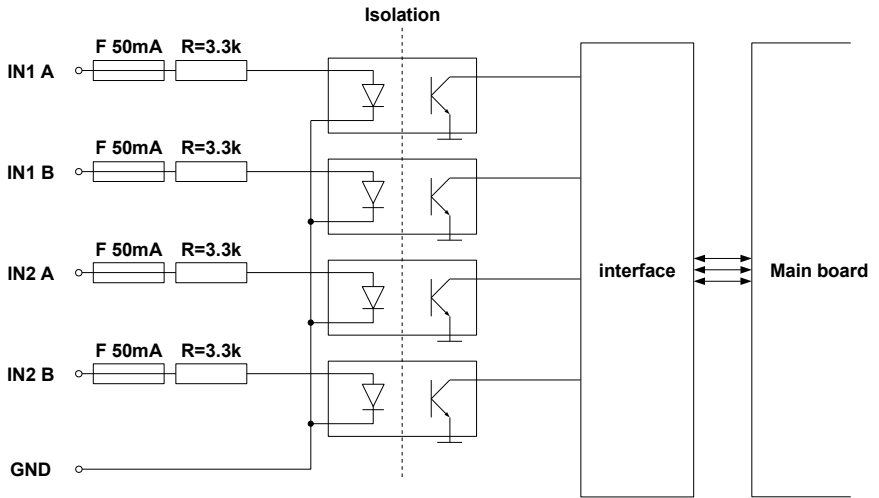
Tab. 8.7 Technical data for digital inputs

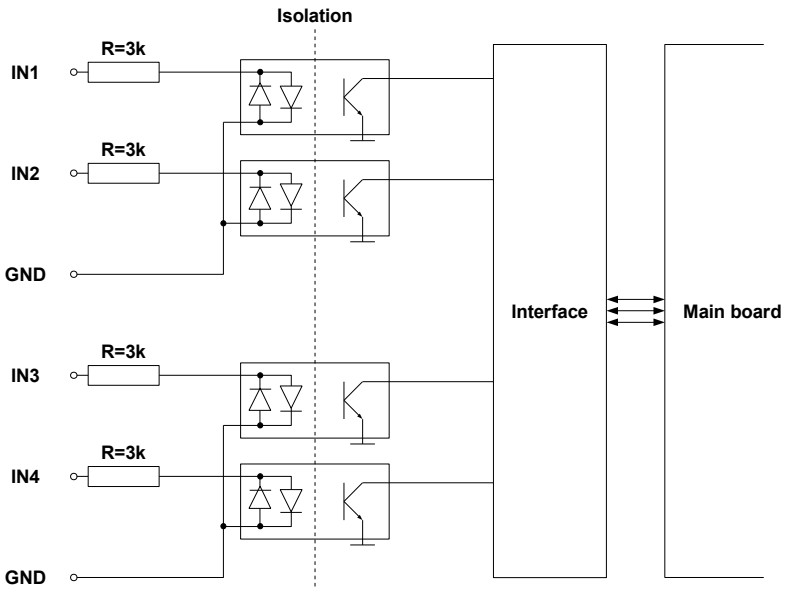*Fig. 8.8. Internal structure for DU2 optoisolated hardware digital input configuration*

Fig. 8.9.Internal structure for D4 optoisolated hardware digital input configuration
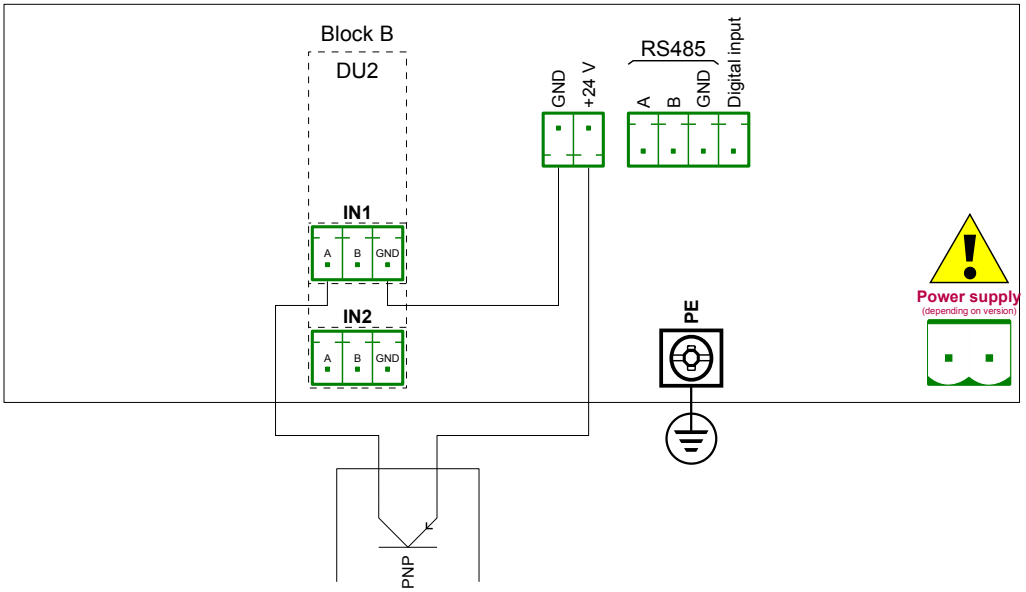
Fig. 8.10.Connection scheme for PNP output type sensor with DU2 hardware configuration
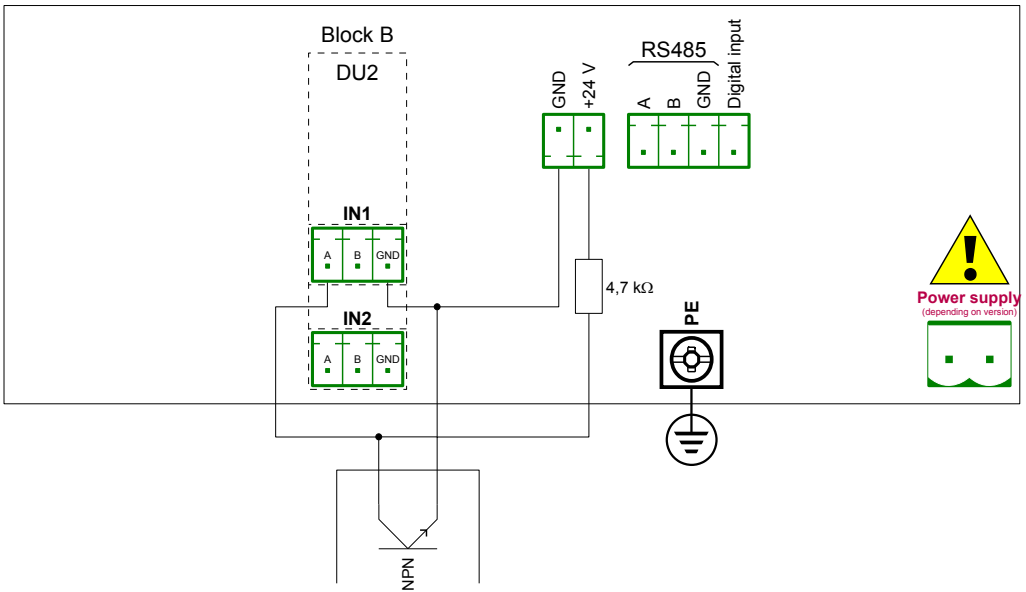

Fig. 8.11.Connection scheme for NPN output type sensor with DU2 hardware configuration

8.4. R21, R41, COP2, COP3, OC2, OC4, R21COP2, COP2OC2, R21OC2 – HARDWARE OUTPUT CONFIGURATIONS
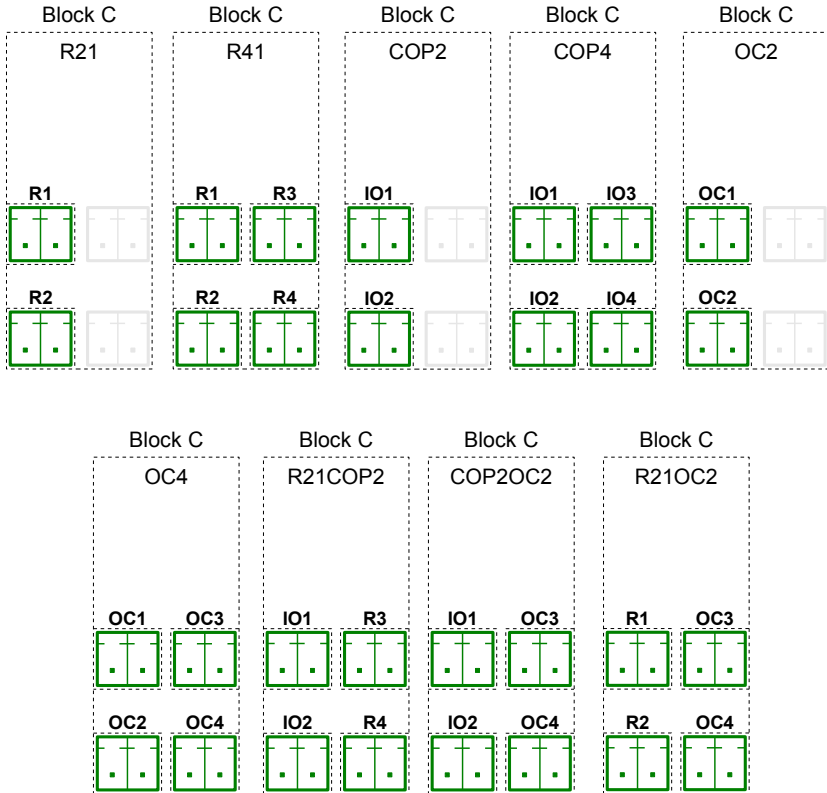

Fig. 8.12. Hardware output configurations for Block C

8.12 hows available hardware output configurations. They are built with different combinations of electrical relay outputs (described as **R21** and **R41**), passive current outputs (described as **COP2** and **COP4**) and electronic relay outputs with PWM function (described as **OC2** and **OC4**).

Most important parameters of electrical relay outputs R21 and R41:

	R21	R41
Number of relays	2 SPDP (N.O.)	4 SPDP (N.O.)
Max. load per relay	1A, cos φ = 1 (resistive load)	1A, cos φ = 1 (resistive load)
Max. voltage switched by relay	250V AC	250V AC
Isolation strength between relays	1 min @ 630 V AC	1 min @ 630 V AC
Isolation strength (relay to MultiCon CMC-N16 supply)	1 min @ 1000V AC	1 min @ 1000V AC

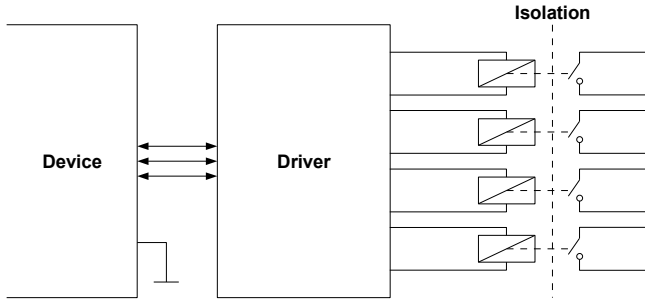*Tab. 8.8 Technical data for electrical relay outputs*

Fig. 8.13. Internal structure of the relay outputs hardware configurations

Most important parameters of passive current outputs COP2 and COP4:

	COP2	COP4
Number of outputs	2	4
Output type	Passive current output	Passive current output
Nominal analogue range	4 - 20 mA	4 - 20 mA
Hardware output limitation	3 - 25 mA	3 - 25 mA
Output voltage dropout	9 V	9 V
Overload protection	Internal resettable fuse 50 mA	Internal resettable fuse 50 mA
Loop supply range	9 - 30 V	9 - 30 V
Output current precision	0.1% @ 25°C, 50 ppm/°C	0.1% @ 25°C, 50 ppm/°C
Resolution	13 bit	13 bit
Isolation strength	1 min @ 500 V AC	1 min @ 500 V AC

Tab. 8.9 Technical data for passive current outputs

Most important parameters of electronic relay outputs with PWM function:

	OC2	OC4
Static parameters		
Number of outputs	2	4
Max current source per ouptut	100 mA	100 mA
Output high level voltage (Iout = 5 mA):	≥ (Vext. - 0.5 V)	≥ (Vext. - 0.5 V)
Output low level voltage (Iout = 5 mA):	< 1 V	< 1 V
Maximum external supply of output	30 V	30 V
Dynamic parameters (set individually for every output)		
PWM period*	0.1 ÷ 1600 s	0.1 ÷ 1600 s
PWM resolution	0.1 s	0.1 s
PWM internal frequency**	5 kHz **	5 kHz **
Pulse – duty factor	0 ÷ 100%	0 ÷ 100%
Pulse – duty factor resolution	15 bits * 0x8000 means 100%	15 bits * 0x8000 means 100%
Lo state minimum time	0 ÷ 800 s	0 ÷ 800 s
High state minimum time	0 ÷ 800 s	0 ÷ 800 s

	OC2	OC4
Isolation strength	1 min @ 500 V AC	1 min @ 500 V AC

* PWM internal frequency and PWM period limit Pulse – duty factor real resolution. For example, if PWM is 0.1 s then real resolution of Pulse - duty factor is about 9 bits (0.1 * 5 kHz = 500 levels). If PWM period is longer than 6.55 seconds then Pulse – duty factor resolution is full 15 bits (6.56 * 5 kHz > 32768 levels).

** PWM output quantization: 20 μs.

Tab. 8.10 Technical date for electronic relay outputs

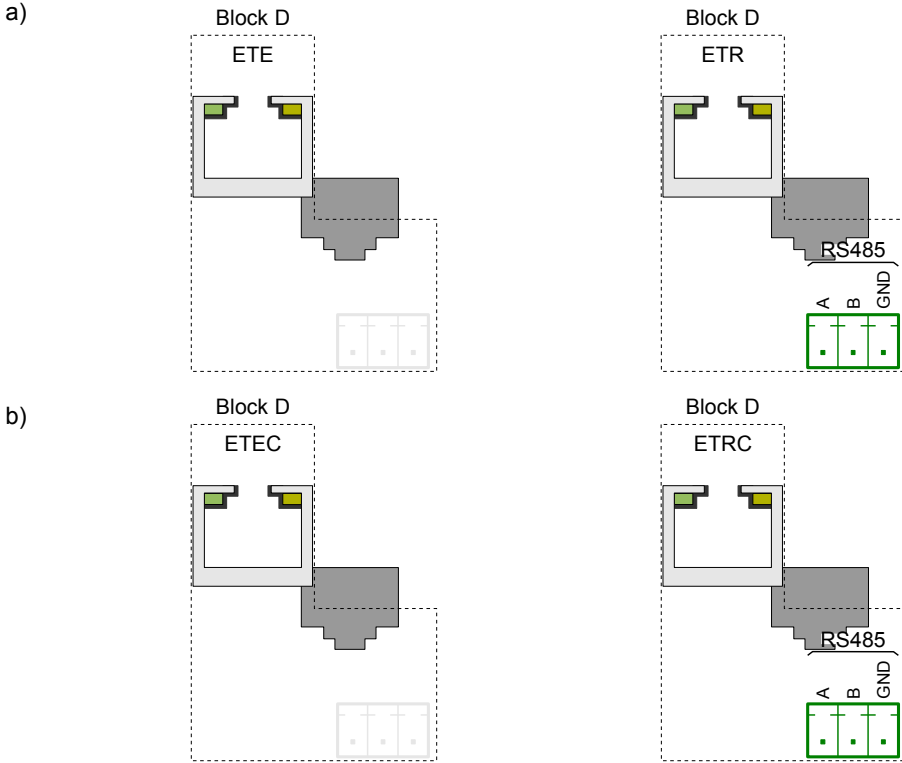## 8.5. ETE, ETR, ETEC, ETRC – HARDWARE COMMUNICATION CONFIGURATIONS

Fig. 8.14. Hardware communication configurations for block D
a) without IP67 connector on housing, b) with IP67 connector on housing

Most important parameters of hardware communication configurations ETE / ETEC i ETR / ETRC*:*

	ETE / ETEC	ETR / ETRC
Number of inputs/outputs	1	2
Inputs/outputs type	RS-485,	RS-485, RJ-45 ETH
Baud rate	RS-485 [bit./sek.]: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, RJ-45 ETH:10Mb/sek.	RJ-45 ETH:10Mb/sek.
Data format	-	RS-232/485: 8N1, 8N2, 8E1, 8E2, 8O1, 8O2

Tab. 8.11 Technical data of ETE / ETEC i ETR / ETRC *hardware communication configurations*
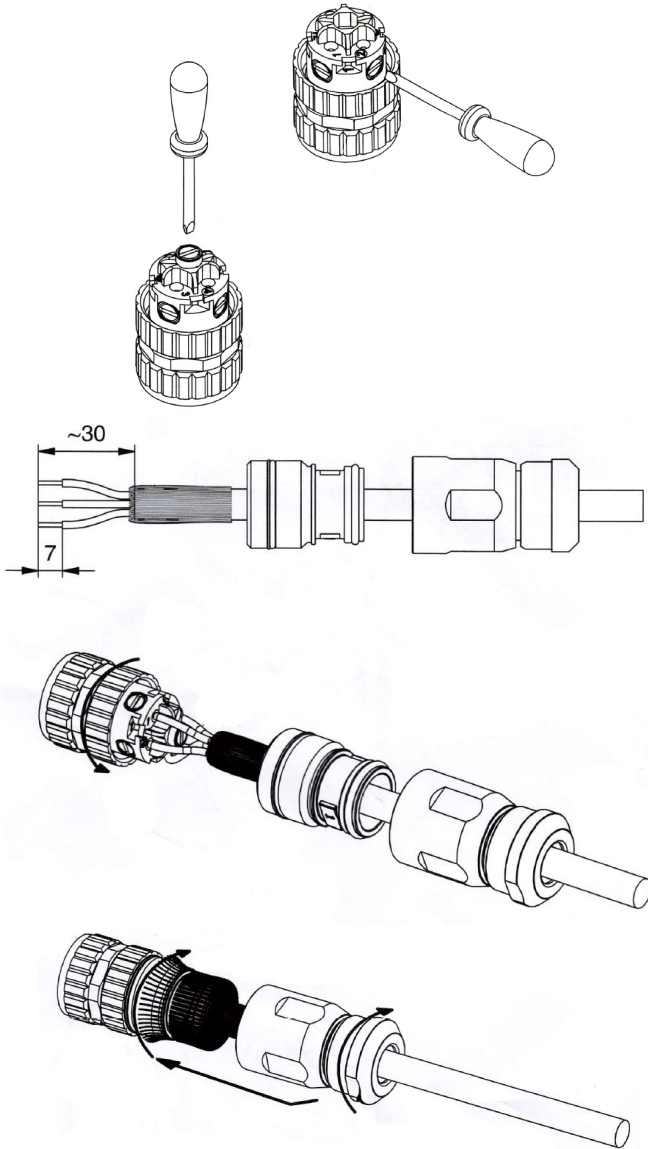
Fig. 8.15. Description of mounting ETH cable into IP67 connector

8.6. DATA FORMAT

Introduction

Every data obtained from the device has a fixed text format. This format can be expanded in the future with backwards compatibility.
Here are some principles that are used in the generated data:
–   To download data manually and load into the delivered PC software, the file structure **must be the same** as described below,
–   If not stated otherwise, all data is a text in which subsequent columns are separated with tabs,
–   If not stated otherwise, all index numbers are counted from zero,
–   All text data have a standard Unix line ending - make sure to convert them correctly under Microsoft Windows® systems,
–   Do not assume that at the end of the file where is an end of a line character, despite its possible presence,
–   If not stated otherwise, all dates are stored in the following format (24-hour clock): YYYY-MM-DD hh:mm:ss:zzz where:
    •   **YYYY** — year,
    •   **MM** — month,
    •   **DD** — day,
    •   **hh** — hour,
    •   **mm** — minute,
    •   **ss** — second,
    •   **zzz** — fraction of a second,

Current measurements data format

Text data have two columns:
–   **channel index** (in decimal notation),
–   **channel value** (decimal point is a dot) – There can be another value than a number:
    •   **-inf** indicates that the channel is in **-LO-** state,
    •   **inf** indicates that the channel is in **-HI-** state,
    •   **nan** indicates that the channel value is unspecified (e.g.: channel is in **ERR** state or is disabled),

Example: channel 1 to 6 measurements.

```
0      1.003
1      2.234
2      -inf
3      inf
4      nan
5      123.223
```

The first column is channel ID (channel number – 1 in decimal notation). In the second column channel 3 (ID = 2) is in -LO- state (-inf), channel 4 (ID = 3) is in -HI- state (inf) and channel 5 (ID = 4) is in ERR state (nan).

Channels settings data format

Text data have the following columns:
– channel ID (channel number – 1)
– name
– selected data source index (see **Mode** parameter in **Chapter 7.9.1. Logical Channels - General settings**)
– unit — channel unit or if there is no unit available – *none* (translated into current device language) label
– format index (see **Format** parameter in **Chapter 7.9.1. Logical Channels - General settings**)
– ON label
– OFF label
– channel **-HI-** level
– channel **-LO-** level
– number precision — text data describing a selected number precision (e.g.: 0.00 means two decimal digits).

Example: channel 1 and 2 settings.

0	Channel C1	1	none	1	ON	OFF	10	0	0.00
1	Channel C2	1	mA	0	ON	OFF	20	0	0.00

Channel one has the name of Channel C1 it is in mode 1 (Hardware input), has no unit, format is set to 1 (binary mode), label **ON** is ON, label **OFF** is OFF, channel **-HI-** state is set to 10 and **-LO-** state to 0, precision is 0.00 (two decimal digits).

When viewing the data in a text editor do not be fooled by the apparent different number of columns. Tabs can be presented differently depending on the user's text editor. It is recommended for the user to enable the visibility of special characters.

General data structure

Data obtained from the device by copying it onto a drive has a fixed structure.
In the flash drive's main directory there is a folder whose name is the same as the device's serial number. All the **main_log.bin_TYPE** files are placed into this directory. The **TYPE** can be:
– **cons_dev_info**,
– **dev_settings**,
– **event_log**,
– **logs**,

Additionally, there are folders with the log data. The name of the folder is the same as the internal log name from the file **main_log.bin_logs**.
Every data folder contains the following files:
– **log.db_TYPE** — log settings files. The **TYPE** can be:
• **chan_conf** — channels settings.
• **conf_xml** — internal device settings.
• **group_settings** — group_settings.
• **log_files** — list of data files.
– **partX.db_TYPE** – log data files. **X** is a number (decimal format without a leading

zero) of the data file the same as the file number stored in **log.db_log_files**. The **TYPE** can be:
- • **raster** — samples interval,
- • **samples** — samples data,
- • **timestamp** — samples synchronization points,

main_log data format description

Contains information about the logging functionality. There can be the following types of data:
- – **const_dev_info** – information specific to this device.
- – **logs** – data describing current and past logs stored on the device. Formatted as follows:
  - • log index,
  - • index of logged group,
  - • internal log name,
  - • date when log was created,
  - • date when log was ended or text **"NULL"** if log is one of the current logs,
- – **event_log** — reserved for future use,
- – **dev_settings** — reserved for future use,

Log data format description

Contains the information about a specific log. There can be the following types of data:
- – **group_settings** — group settings of this log. This is one line in the following format:
  - • group index,
  - • name,
  - • date when this log was created,
  - • date when this log was ended or text "**NULL**" if log is one of the current logs,
  - • log description,
  - • base log period,
  - • alternative log period,
  - • log mode (1 — always, 2 — triggered by channel),
  - • logical channel count in this group,
  - • full group configuration file name,
  - • channel index which is a base trigger or "-1" if there is no channel,
  - • channel index which is an alternative trigger or "-1" if there is no channel,
- – **log_files** — information about all the samples files created in this log in the following format:
  - • ID/file number,
  - • file creation date,
  - • file end date or text "**NULL**" if file is being currently used,
- – **chan_conf** — settings of all the channels that are being logged in the following format:
  - • channel index,
  - • name,
  - • selected data source index (see **Mode** parameter in **Chapter 7.9.1. Logical Channels - General settings**),
  - • unit — channel unit or if there is no unit available *none* (translated into current device language) label,format index (see **Format** parameter in **Chapter 7.9.1. Logical Channels - General settings**),
  - • ON label,

- • OFF label,
- • channel **-HI-** level,
- • channel **-LO-** level,
- **In firmware later than** *2.14.2*:
- • **precision** — text data describing a selected number precision (e.g.: 0.00 mean two decimal digits),

part data format description

Specific log data. There can be the following types of data:
- – **raster** — holds information about samples interval,
    - • Sample ID **in hexadecimal notation** (called: **R$_{id}$**),
    - • raster value (in seconds) (called: **R$_v$**),
- – **timestamp** — samples synchronization markers,
    - • sample ID in hexadecimal notation (called: **T$_{id}$**),
    - • sample time (called: **T$_t$**),
- – **samples** — channels measurements data. Channel order is the same as row order in **log.db_log_files***:*
    - • sample ID in hexadecimal notation (called: **S$_{id}$**),
    - • floating point text representation of sample value for the first logged channel
    - • …
    - • floating point text representation of sample value for the N-th logged channel.

How to calculate the precise sample time

To calculate the time of the sample number **S$_{id}$** use the below procedure:
- – Find **R$_v$** and maximum **R$_{id}$** where **R$_{id}$ ≤ S$_{id}$**,
- – Find **T$_t$** and maximum **T$_{id}$** for **T$_{id}$ ≤ S$_{id}$**,
- – Calculate the sample time (**S$_t$**) using the given equation:

$$S_t = T_t + R_v \cdot (S_{id} - T_{id})$$

Detailed description of files

File **main_log.bin_logs** contains:
- – **ID** – unique decimal number identifying log,
- – **GROUP_NO** – number of a logged group (corresponding to a device's group), starting from 0,
- – **DIR** – name of directory with log files., e.g. "g0_2010_08_04_13_50_39". The name of a directory is created according to the scheme below: "gX_YYYY_MM_DD_hh_mm_ss", where:
    - • **X** – group number,
    - • **YYYY** – year of log start,
    - • **MM** – month of log start,
    - • **DD** – day of log start,
    - • **hh** – hour of log start,
    - • **mm** – minute of log start,
    - • **ss** – second of log start,

- **LOG_START_DATE** – date of log start,
- **LOG_END_DATE** – date of log end. If there is a "**NULL**" in place of date, it means that the particular file is not closed (logging is not finished),
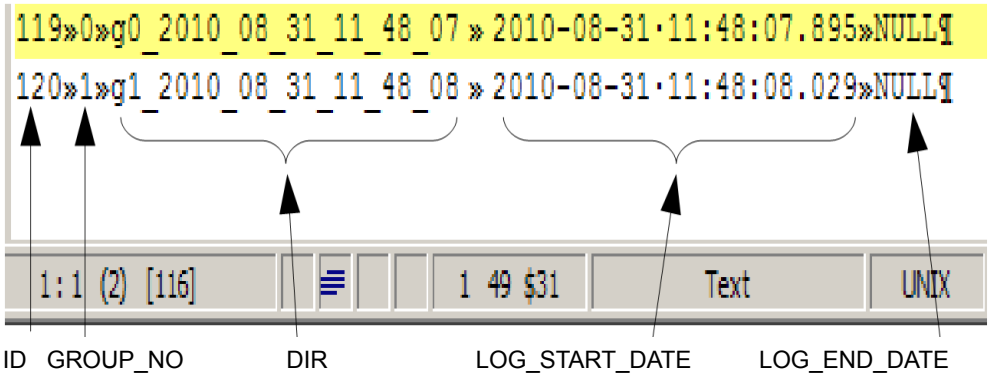

*Fig. 8.16. **main_log_bin_logs** file view example*

File **main_log.bin_event_log** – reserved for future usage,

File **main_log.bin_dev_settings** – reserved for future usage,

File **main_log.bin_cons_dev_info** - -info about the device, contains:
- **dev_type** – type of the device,
- **serial_no** – serial number of the device,
- **s_version** – software version,
- **h_version** – reserved,
- **db_version** – reserved,

File **log.db_group_settings** – settings of a registered group,
1) number of a group,
2) displayed name,
3) logging start date,
4) logging end date,
5) log description,
6) basic logging period,
7) alternative logging period,
8) logging mode (1 – always on, 2 – triggered),
9) number of logged channels (1 to 6),
10) name of a file containing a detailed logging configuration,
11) logical channel number used for basic logging triggering, or -1 if not selected,
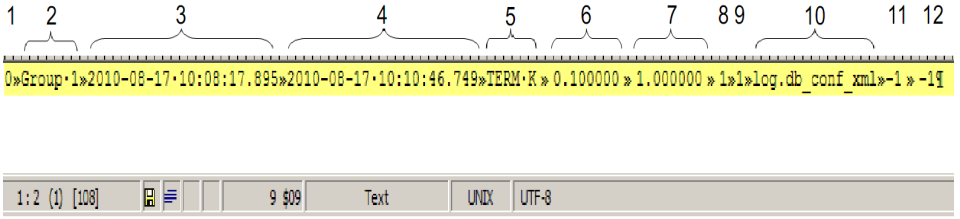12) logical channel number used for alternative logging triggering, or -1 if not selected,

*Fig. 8.17. **log.db_group_settings** file view example*

File **log.db_conf_xml** – file describing settings of the device,

File **log.db_chan_conf** – file describing settings of logged channels,
This file contains as many rows as channels in the logged group.
       1) logical channel number starting from 0,
       2) channel name,
       3) channel source mode,
       4) displayed unit,
       5) display mode (0 – numbers, 1 – binary, 2 – time),
       6) text for binary mode "High",
       7) text for binary mode "Low",
       8) Hi graph limit,
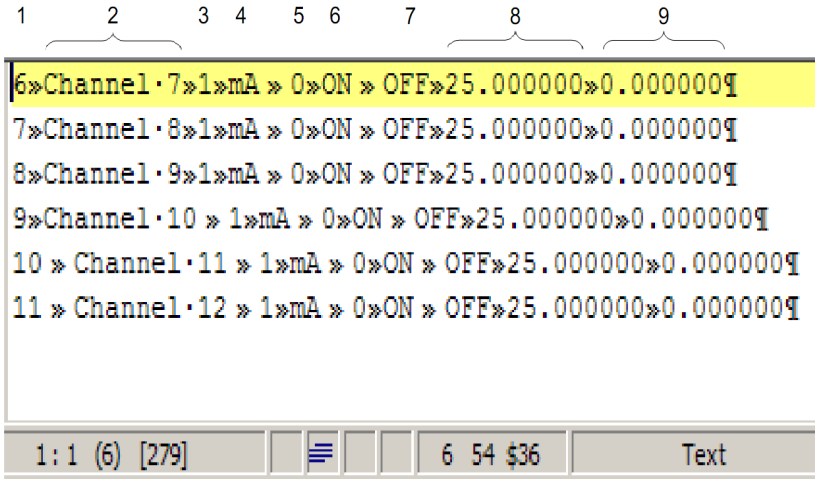   9) Lo graph limit,

*Fig. 8.18. **log.db_chan_conf** file view example*

File log.db_log_files – information about the files with log samples,
      Due to security of data, samples are stored in a separate file which changes from

time to time. File **log.db_log_files** contains information about such files. Every line describes another file.

   1) ID – unique identifier of the file,
   2) file start date,
   3) file end date or NULL if file is not finished,

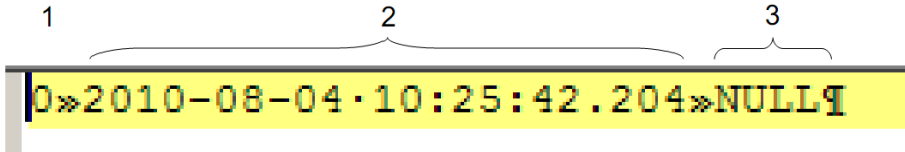*Fig. 8.19. **log.db_log_files** file view example*

Files with samples **partX.db_***
These files contain logged samples. Names of these files are created according to a rule:

  partX.db_SUFFIX, where:

  X – unique identifier of a file (see **log.db_log_files**),
  SUFFIX – can be: **raster**, **samples**, **timestamp**,

If a file has suffix **raster**, then it contains info about the time period between samples.
   1) **IDR** – hexadecimal id of the sample for which raster becomes valid,
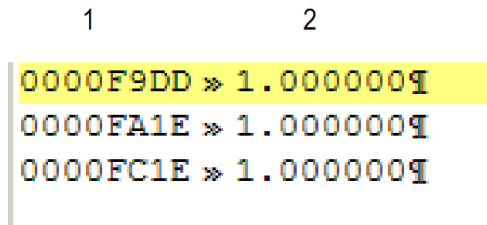   2) **period** – floating point number which expresses the period in seconds,

*Fig. 8.20. **partX.db_raster** file view example*

If a file has suffix **timestamp**, then it contains info about the samples synchronisation.
   1) **IDT** – hexadecimal id of the sample for which synchronisation data is stored,
   2) **T** – date (**timestamp**),

*Fig. 8.21. **partX.db_timestamp** file view example*

If a file has suffix **samples**, then it contains logged samples.
       1) **IDS** – unique hexadecimal id of sample in log,
       2÷7) floating point value of logged channel in a group (see file **log.db_chan_conf**),

Every line in this file contains maximal 7 columns.

```
        1          2          3          4          5          6          7

0000F9DD » 3.0000 » 16.6168»8.4553 » 3.0000 » 16.6168»8.4553¶
0000F9DE » 3.0000 » 14.8750»5.7825 » 3.0000 » 14.8750»5.7825¶
0000F9DF » 3.0000 » 13.4500»4.1751 » 3.0000 » 13.4500»4.1751¶
0000F9E0 » 3.0000 » 11.8665»3.1682 » 3.0000 » 11.8665»3.1682¶
0000F9E1 » 3.0000 » 10.2831»3.0750 » 3.0000 » 10.2831»3.0750¶
0000F9E2 » 3.0000 » 8.7000 » 3.9042 » 3.0000 » 8.7000 » 3.9042¶
0000F9E3 » 3.0000 » 7.1164 » 5.5751 » 3.0000 » 7.1164 » 5.5751¶
0000F9E4 » 3.0000 » 5.5332 » 7.9236 » 3.0000 » 5.5332 » 7.9236¶
0000F9E5 » 3.0000 » 3.9497 » 10.7204»3.0000 » 3.9497 » 10.7204¶
0000F9E6 » 22.0000»3.9500 » 13.6906»22.0000»3.9500 » 13.6906¶
0000F9E7 » 22.0000»6.3248 » 16.5447»22.0000»6.3248 » 16.5447¶
0000F9E8 » 22.0000»8.6998 » 19.0030»22.0000»8.6998 » 19.0030¶
0000F9E9 » 22.0000»11.0748»20.8248»22.0000»11.0748»20.8248¶
0000F9EA » 22.0000»13.4498»21.8317»22.0000»13.4498»21.8317¶
```

*Fig. 8.22. **partX.db_samples** file view example*

Organisation of files stored in a flash drive while copying

Log files are stored on a disk according to the following scheme:
    1) if a directory with the name equal to the device's serial number does not exist, then it is created,
    2) in such a directory, files with information about logs are created (description below),
    3) each log is stored in a separate subdirectory,

Files hierarchy:

```
 DISK:\        Serial_number\   main_log.bin_logs
                               main_log.bin_event_log
                               main_log.bin_dev_settings
                               main_log.bin_cons_dev_info
                               log1_directory\              log.db_chan_conf
                                                            log.db_group_settings
                                                            log.db_log_files
                                                            part0.db_raster
                                                            part0.db_samples
                                                            part0.db_timestamp
                                                            log.db_conf_xml

               log2_directory\
```

8.7. DIRECT ACCESS TO LOG FILES USING HTTP PROTOCOL

ⓘ To connect the **MultiCon CMC-N16** with another device using the HTTP protocol, port number 80 must be unlocked in the software and hardware firewall. The device allows up to 10 concurrent connections using the HTTP protocol.

Request format:

To request specified data the user must perform a CGI query on the address specified below using a described format.

http://<IP_address>/cgi-bin/data.cgi?cmd=<command>[&id_first=x][&id_last=y][&additional parameters]

ⓘ Almost all queries return data where the first column is an ID number. In order to reduce device load, use the parameters of id_first and id_last. These parameters limit the amount of data processed and sent to a given Id range.

Example:
To limit the amount of returned data to the ID range from 123 to 321 execute a query like:

http://<IP_address>/cgi-bin/data.cgi?cmd=<command>&id_first=123&id_last=321[&additional parameters]

Format of the id_first and id_last:

Numbers can be given to each parameter in the decimal or hexadecimal notation (can be mixed – one can be decimal, the other hexadecimal):

      1) **decimal** – id_last=1234,
      2) **hexadecimal** – id_last=0x4D2 (0x4D2=1234),

ⓘ **id_first** and **id_last** are indexed from "0". It means that the first value is at index 0, second at index 1 etc.

The cmd parameter:

This parameter specifies the type of requested data. It can be one of the following values:

      1) **info** – returns devices information text,

                  http://192.168.1.176/cgi-bin/data.cgi?cmd=info

      2) **current** – returns current logical channels measurements text,
      3) **settings** – returns current logical channels settings text,
      4) **main_log** – returns general device logging information text,
      5) **log** – returns specified log information text,
      6) **part** – returns recorded data of specified log,

More about **cmd** parameter see in this **Chapter** below.

Obtaining measurements (cmd=current):

Current measurements can be obtained by using the value **current** of the **cmd** parameter. This feature should not be used as a replacement of recording data functionality because it causes high device load. Use this feature for sporadic device measurement checks.
Example of query viewing settings of channel 2 to 11:

http://192.168.1.176/cgi-bin/data.cgi?cmd=current&id_first=1&id_last=10

Obtaining settings (cmd=settings):

Current channels settings can be obtained by using the value **settings** of the **cmd** parameter. This feature should not be used as a replacement of recording data functionality because it causes high device load. Use this feature for sporadic device settings checks.
Example of query viewing all channels settings:

http://192.168.1.176/cgi-bin/data.cgi?cmd=settings

If a valid data query is performed and no data can be returned (example: requesting list of logs but there was no logs on device or given boundaries are outside available ID range) the blank page will be returned.
If a valid data query is performed but data requested does not exist on the device (example: requesting data of non existent log) the error page *"404: Not found"* will be returned.
If an invalid query is performed the error page *"400: Bad request"* with an inappropriate error message will be returned.

Obtaining general device logging information (cmd=main_log):

To obtain the current device logging information the **main_log** parameter can be used. There must be a specified additional parameter called **file**. Values must be one of the following:

1) **logs** – returns list of past and current logs,
2) **cons_dev_info** – returns device information (similar to the **cmd=info** parameter),
3) **dev_settings** – returns other recorded device information (currently returns blank page – reserved for future use),
4) **event_log** – returns recorded events (currently returns blank page – reserved for future use),

Example of query viewing information about all logs:

http://192.168.1.176/cgi-bin/data.cgi?cmd=main_log&file=logs

Obtaining specified log information (cmd=log):

To obtain specified log information the **log** parameter can be used. There must be specified additional parameters:

1) **log_name** – is the name of log,
2) **file** – is the type of requested data. This parameter can be one of:
- **log_files** – list of log files,
- **chan_conf** – list of channels configuration,
- **group_settings** – group settings,
- **conf_xml** – other internal settings,

Example of viewing list of log files in log gX_YYYY_MM_DD_hh_mm_ss use this query syntax:

http://<IP_address>/cgi-bin/data.cgi?cmd=log&log_name=gX_YYYY_MM_DD_hh_mm_ss&file=log_files

Obtaining recorded data (cmd=part):

To obtain recorded data the **part** parameter can be used. There must be specified additional parameters:

1) **log_name** – is the name of log,
2) **part_nr** – is the number of data file,
3) **file** – is the type of demanded data. They can be:
- **samples** – returns recorded channels data,
- **timestamp** – returns recorded time stamps,
- **raster** – returns samples interval,

(i) If there is no **id_first** or **id_last** specified, the whole data file will be returned. When requesting **samples**, file size can be even **1MB**.

8.8. WWW PAGE

Each device which is equipped with Ethernet port can share its www page which allows to view in a window measurements, device informations and configurations and logs files management.

In order to gain access to www page the user must enter the IP address of the device (see also **Chapter 7.5. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION**) in his web browser. A www page of the device window will appear as shown in **8.23**.

Fig. 8.23. WWW page view

8.8.1. Menu

Contains visualization and logical channels options and also device and logs informations.

Following submenus can be found here:

– **Channels graphical view** – shares very practical and easy in use graphical view of logical channels visualized with widgets made in HTML5 technology.
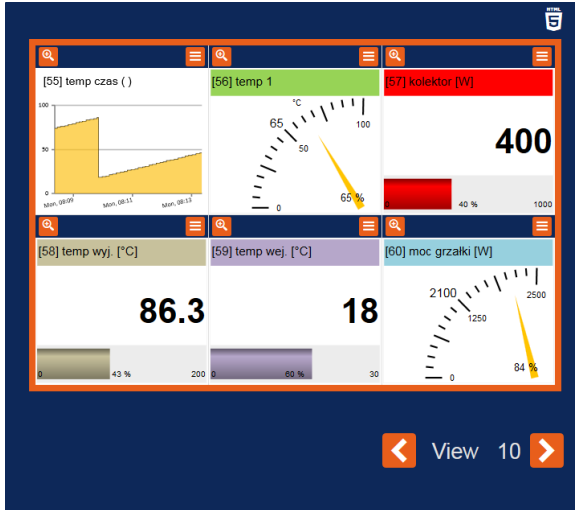

Fig. 8.24. View of logical channels visualization using widgets made in HTML5 technology

Each view marked in **8.24** as VIEW, contains of six logical channels. View 1 shows channels 1-6, view 2 channels 7-12, view 3 channels 13-18 etc..
The ⊕ loupe button, located in the upper left corner of each logical channel is used to enlarge the selected view. Three types of widgets are available:
• **Text –** values are represented with numbers,
• **Chart** – values are represented with chart,
• **Needle** – values are represented with needles.
The number, name, and measuring range of each displayed logical channel is retrieved from the device.

– **Channels tabular view** – a very convenient view that allows to see current values of all logical channels as a single table. Thanks to its own data buffer, this table provides some practical indicators based on value changes of individual channels over the time, such as averages, minimums, maximums and its timestamps. This table is based on HTML5 technology, which makes it possible to view measurements on various types of devices and thanks to the used technology the performance of these devices shouldn't matter. The table automatically formats values according to the channels settings in the device.

Controller 192.168.3.97 Last acquisition at 15:13:10 *	Channel Name	Current Value	Current Value as % of range	Average Value based on last 10h	Minimum Value out of last 10h	Maximum Value out of last 10h
Channel 01	Przepływ pompy	16.9 l/min	84% of range 0..20 l/min	11.9 l/min	5.0 l/min at 15:11:10	20.0 l/min at 15:09:10
Channel 02	PS temp. wej.	21.5 °C	71% of range 0..30 °C	22.6 °C	14.3 °C at 15:05:09	29.3 °C at 15:07:09
Channel 03	PS temp. wyj.	32.8 °C	82% of range 0..40 °C	31.6 °C	27.1 °C at 15:05:09	35.3 °C at 15:07:09
Channel 04	Zbiornik temp. wej.	12.3 °C	49% of range 0..25 °C	12.3 °C	12.3 °C at 15:13:10	12.3 °C at 15:13:10
Channel 05	Zbiornik temp. wyj.	69.4 °C	79% of range 0..85 °C	74.7 °C	66.2 °C at 15:09:10	81.8 °C at 15:11:10
Channel 06	Moc Bojlera	652 W	32% of range 0..2000 W	451 W	200 W at 15:02:09	803 W at 15:09:10
Channel 07	Temp. Bojlera	85.0 °C	94% of range 0..90 °C	85.0 °C		°C at 15:13:10
Channel 08	Przepływ wody	40.2 l/min	67% of range 0..60 l/min	22.9 l/min		l/min at 15:09:10
Channel 09	Moc Pompy	103 W	85% of range 0..120 W	81 W		W at 15:09:10
Channel 10	SŁOŃCE	38.0 °C	95% of range 0..40 °C	36.3 °C		°C at 15:04:09
Channel 11	Zegar bin	0	0% of range 0..2	0		at 15:06:09
Channel 58	US 6	1706	902% of range -100..100	1188		00 at 15:09:10
Channel 59	US 2	-17	41% of range -100..100	-14		0 at 15:11:10
Channel 60	US 3	-5	47% of range -100..100	-5		4 at 15:11:10

Average Values based on other periods

Period	Since	Value	Value%
1m	15:12:10	470 W	23%
2m	15:11:10	390 W	19%
3m	15:10:10	467 W	23%
5m	15:08:10	523 W	26%
10m	15:03:09	474 W	23%
30m	14:43:09	451 W	22%
1h	14:13:09	451 W	22%
2h	13:13:09	451 W	22%
3h	12:13:09	451 W	22%
5h	10:13:09	451 W	22%

Fig. 8.25. Tabular view of configured device channels

ℹ Measurements in the tabular view are refreshed by default every 60 seconds. If it will be necessary to increase the frequency of updates, it can be done by changing the value of **interval** parameter in the URL of the page. The tab with the table should always remain opened (be active) to avoid forced slowdown by the user's web browser.

– **Channels value raw data** – allows to download measurement data and display them in form of a list. The list format is described in **Chapter 8.6. DATA FORMAT** in paragraph **Current measurement data format**.

– **Channel settings raw data** – contains logical channels settings I form of a list. The list format is described in **Chapter 8.6. DATA FORMAT** in paragraph **Channels settings data format**.

– **Device information** – contains basic informations about the device and possible hardware configurations.

– **Information about all logs** – contains information about logs which are stored in the device in form of a list. The list format is described in **Chapter 8.6. DATA FORMAT** in paragraph **Detailed description of files**.

8.8.2. Documentation

Contains manual of the device and instructions about the hardest functionalities of the device which can trouble the user the most.

Following submenus can be found here:
– **User manual** – allows to download the manual which is stored in the device's memory. In order to do this in submenu the user should press "**Proceed**" button, select a language and press the "**Download**" button.
– **Remote display manual** – opens the description of a net hardware configuration with the **2.1** to link with the remote display.
– **Network interface description** – contains the description of a direct access to log files (in the device has active log option) using HTTP protocol.
– Data format – contains log files format description. Full description is also in **Chapter 8.6. DATA FORMAT**.

More information can be found in **Chapter 7.5. DEVICE INFORMATION, LICENCE,**

FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION and **8.7. DIRECT ACCESS TO LOG FILES USING HTTP PROTOCOL**.

8.8.3. Logging and User menu

Most of the options available via www page are available without logging to the device. But access to option which can change the way that the device work are available after logging.

If the device has the single level or multilevel access password set then the user can login to the device via fields in upper right corner of the screen. On the screen the **User menu** will appear which has following submenus:
Following submenus can be found here:
– **Remote display** – allows to switch local screen of the device into remote display without necessity of coming to the device each time. After entering the submenu, an address of the user's computer where the www page is open will appear in the **IP address** field. This address should be entered in **IP Address** in **Access options** menu in the device. Before switching to remote display it is necessary to check that the **Xming** is running and configured properly. Pressing the "**Start**" button will switch from local to remote display. This function is active only for the **Administrator** user and to those users who have access to **Remote display** option in **Permissions** menu.
– **File management** – allows to export and import configuration and templates of the device and also to remotely delete all logs from memory.
   • In **Export** part the user can download main configuration (**Main config**), Modbus master templates (**Modbus template**) and Modbus slave templates (**Modbus slave template**) by choosing particular option and pressing the "**Download**" button.
   • In **Import** part the user can upload main configuration (**Main config**), Modbus master templates (**Modbus template**) and Modbus slave templates (**Modbus slave template**) by selecting particular option and pressing the "**Upload**" button.
   • In **Delete all logs** part the user can delete all stored logs from the internal memory of the device. Pressing the "**DELETE**" button and confirming it will cause deletion of all the logs.
– **Restart device** – allows to remotely restart the device. Pressing this button and confirming it will cause automatic turning of and the turning on of the device. After turning on (unless the user will not prohibit it in the next message) the device will also regain the connection with the www page. Restart of the device is available for the Administrator and any logged user.

More informations are in **Chapter 7.5. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION** and **7.18. ACCESS OPTIONS – MultiLevel Access FUNCTIONALITY**.