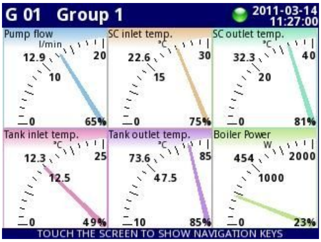
User manual

CONTROLLER/DATA RECORDER

- Firmware: **v.2.27.0** or higher



Read the user's manual carefully before starting to use the unit or software.
Producer reserves the right to implement changes without prior notice.

CONTENTS

Explanation of 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 RESPONSIBLE FOR POSSIBLE DAMAGES.

ℹ *- 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 damages caused by inappropriate installation, not maintaining the proper environmental conditions and using the unit contrary to its assignment.

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

- GND input of device should be connected to PE wire;

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

- **If in the 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 the case of malfunction).**

- Neighbouring and connected equipment must meet the appropriate of 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 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 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. THE USE OF TOUCH-SCREEN

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

2. GENERAL CHARACTERISTICS

The **CONTROLLER/DATA RECORDER** is a sophisticated multichannel unit which allows simultaneous measurement, visualisation and control of numerous channels. This device can operate autonomously or with cooperation with external measurement devices and actuators. Essential features of **CONTROLLER/DATA RECORDER** are listed and briefly described below.

- ***Advanced processing unit and system based on LINUX***
  The powerful **CONTROLLER/DATA RECORDER** 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 simultaneous operation of many processes (like: measurement, communication, visualisation). The use of LINUX also makes software independent of installed hardware.

- ***Color TFT display with Touch-panel***
  The **CONTROLLER/DATA RECORDER** 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 **CONTROLLER/DATA RECORDER** easy and intuitive.

- ***Hardware flexibility and a large variety of possible configurations***
  CONTROLLER/DATA RECORDER is designed as modular device consisting of a ***base*** and optional ***input and output modules***. 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 slots (marked as A, B, C) designed for installation of measurement and/or output modules.
- one slot (marked as D) used for advanced communication module (additional USB Host, RS-485, RS-485/RS-232 and Ethernet).

All measurement and output modules are optional and can be installed inside the device according to the customer's needs.

<u>Input modules that can be installed:</u>

- 4/8/16/24x Voltage/Current input module,
- 16/24x NTC/Voltage/Current/Digital mixed inputs module,
- 6x Isolated current input module,
- 4/6x RTD input module,
- 4/8/12x TC input module,
- 8/16/24x Optoisolated digital input,
- 3/5x universal input module,
- 2/4x universal counter module,
- 2/4x hourmeter module,
- 2/4x flowmeter input + 2/4x current input module.
- 2/4x pulse input + 2/4x current input module

<u>Output modules that can be installed:</u>

- 8/16/24x SSR driver module,
- 4/6x Relay module 5A/250V,
- 8/12x Relay module 1A/250V,
- 2/4/6/8x Passive current output module.

- ***Full freedom of data sources, presentation modes and controlling methods***
  The multi level structure of the **CONTROLLER/DATA RECORDER** firmware allows for selection of presented data sources, presentation modes and controlling methods. The **CONTROLLER/DATA RECORDER** 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 **CONTROLLER/DATA RECORDER** by RS-485 interface),
  - output states and quantities (looped back results of controlling processes),
  - generate profiles/timers or also the mathematical combination of one or more ***logical channels***.

  All of these can be freely named and described by the user, and presented in many forms:
  - as numerical values,
  - vertical and horizontal charts,
  - vertical and horizontal bars,
  - as needle graphs.

  Every ***logical channel*** (visualised or not) can be used as input data for one or more controlling process. The **CONTROLLER/DATA RECORDER** implements many different controlling methods:
  - above defined level,
  - below defined level,

– inside defined range,
– outside of defined range,
– PID control.

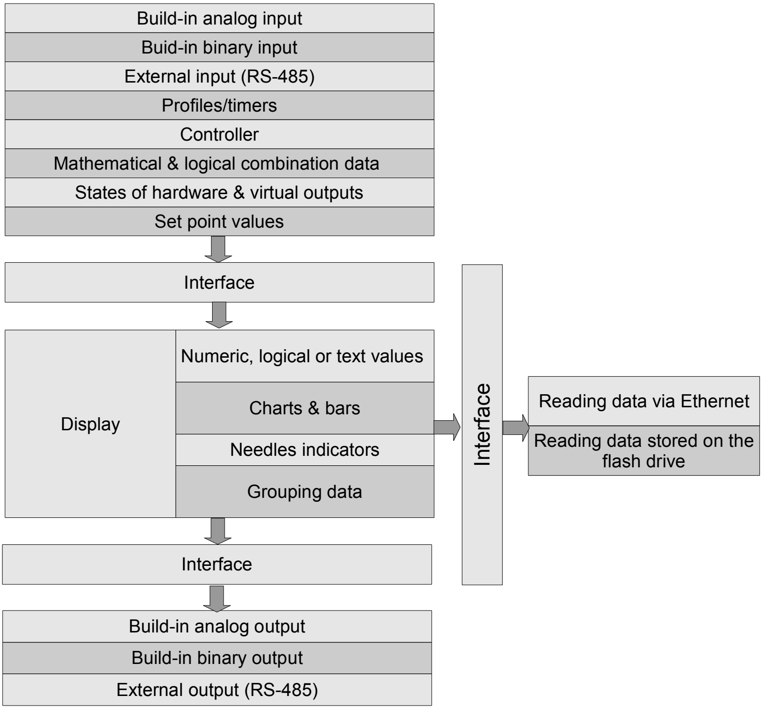


Fig. 2.1. Basic structure of the multichannel device

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***.

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” or 5.7”, 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, USB Device port
Digital input	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 module*	Second USB Host port Serial RS-485 and RS-485/RS-232 Ethernet 10M RJ-45
Optional input modules*	4/8/16/24x Voltage (0÷10V) / Current (0÷20mA)** 16/24x NTC (0÷100k)/ Voltage (0÷10V)/ Current (0÷20mA)/ Digital (TTL,HTL)** 6x Isolated current (4÷20mA), 4/6x RTD (Pt100, Pt500, Pt1000, Cu50, Cu100)** 4/8x/12 TC (J, K, S, T, N, R, B, E, L(GOST)** 8/16/24x Digital input** 3/5x Universal input** 2/4x Universal counter input** 2/4x Hourmeter module** 2/4x Flowmeter + 2/4x Current input** 2/4x Pulse input + 2/4x Current input**
Optional output modules*	4/6x Relay 5A/250V (cos φ = 1)** 8/12x Relay 1A/250V (cos φ = 1)** 8/16/24x SSR driver (10÷15V, up to 100mA per output)** 2/4/6/8x IO Passive current output (4÷20mA)**
Protection level	
USB interface on rear panel	IP 65 (from front, after using waterproof frame) IP 54 (from front, with transparent door) IP 40 (from front, standard) IP 20 (housing and connection clips)
USB interface from front	IP 54 (from front, with transparent door) IP 40 (from front, standard) IP 20 (housing and connection clips)
Housing type	panel
Housing material	NORYL - GFN2S E1

Housing dimensions	96 x 96 x 100 mm (small housing - 3,5” Display) or 144 x 144 x 100 mm (big housing - 5,7” Display)
Mounting hole	90.5 x 90.5 mm (small housing - 3,5” Display) or 137 x 137 mm (big housing - 5,7” Display)
Assembly depth	102 mm
Panel thickness	max. 5 mm
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	5 to 90% no condensation
Altitude	up to 2000 meters above sea level
Screws tightening max. torque	0,5 Nm
Max. connection leads diameter	2,5 mm^2
Safety requirements	according 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
Weight	340g (only base, see **Fig. 4.8**)

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

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 according to the local regulations.

- Read the basic safety requirements on page **5** 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 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 (**Fig. 4.1**) meets the expectations and if necessary use appropriate measures of over voltage protection. Additionally, insure the appropriate air and surface insulation gaps when installing.

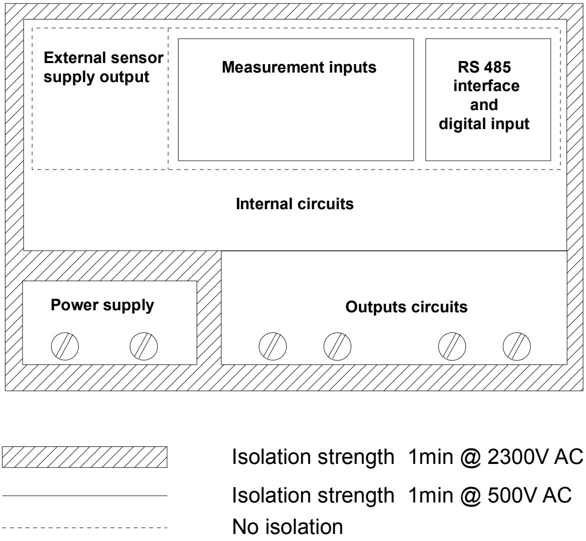

External sensor supply output
Measurement inputs
RS 485 interface and digital input
Internal circuits
Power supply
Outputs circuits
Isolation strength 1min @ 2300V AC
Isolation strength 1min @ 500V AC
No isolation

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:

- assembly brackets - 2 pieces,
- pointer for touch screen,
- user’s manual for **CONTROLLER/DATA RECORDER** unit (device) in **pdf.**,

4.2. ASSEMBLY

- The unit is designed for mounting inside housings (control panel, switchboard) insuring appropriate protection against surges and interference. Metal housings must be connected to ground in a way that complies with the governing regulations.
- 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 according to **Fig. 4.2**. The thickness of the material of which the panel is made must not exceed 5mm. When preparing the mounting hole take the grooves for catches located on both sides of the housing into consideration (**Fig. 4.2**). Place the unit in the mounting hole inserting it from the front side of the panel, and then fix it using the brackets (**Fig. 4.4**). The minimum distances between the centre points of multiple units - due to the thermal and mechanical conditions of operation - are shown on **Fig. 4.3**.

96 x 96 housing:
H, W = 90.5 mm
h = 16 mm

144 x 144 housing:
H, W = 137 mm
h= 38.5 mm

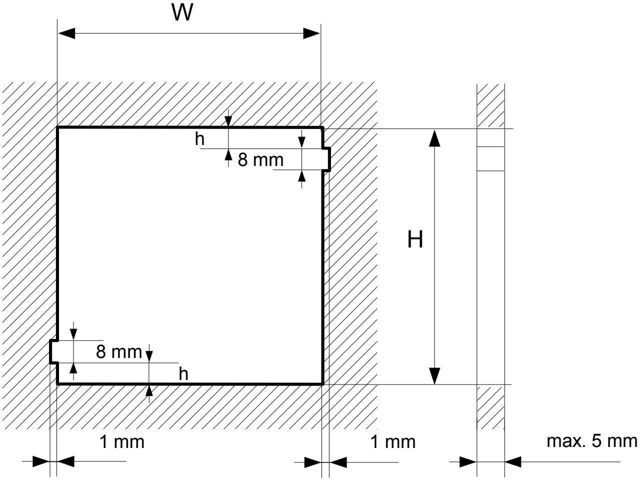


Fig. 4.2. Mounting hole dimensions

<u>96 x 96 housing:</u>
H, W = 115 mm

<u>144 x 144 housing:</u>
H, W = 165 mm

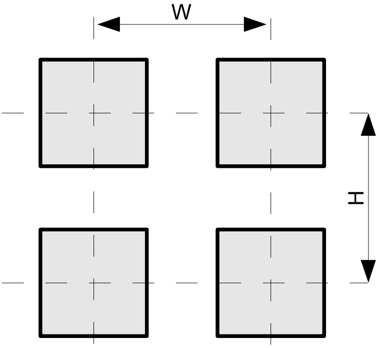


Fig. 4.3. Minimum distances when assembly of a number of units

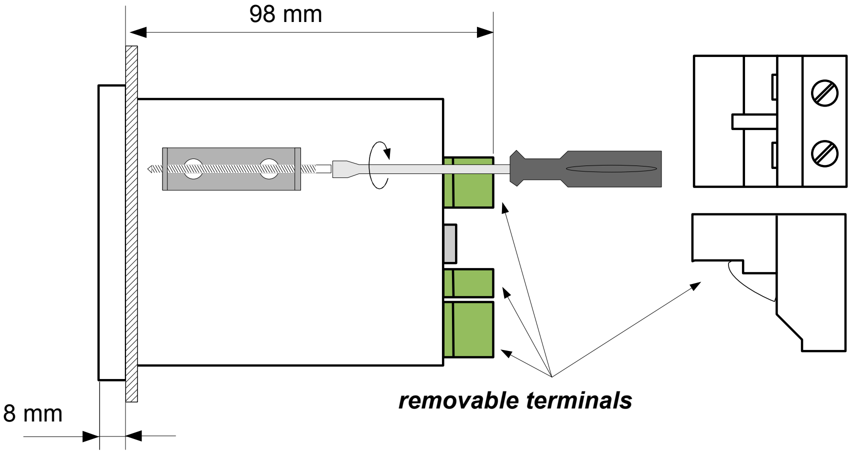


Fig. 4.4. Installing of brackets

To avoid connectors slots destruction use the method shown on **Fig. 4.5**

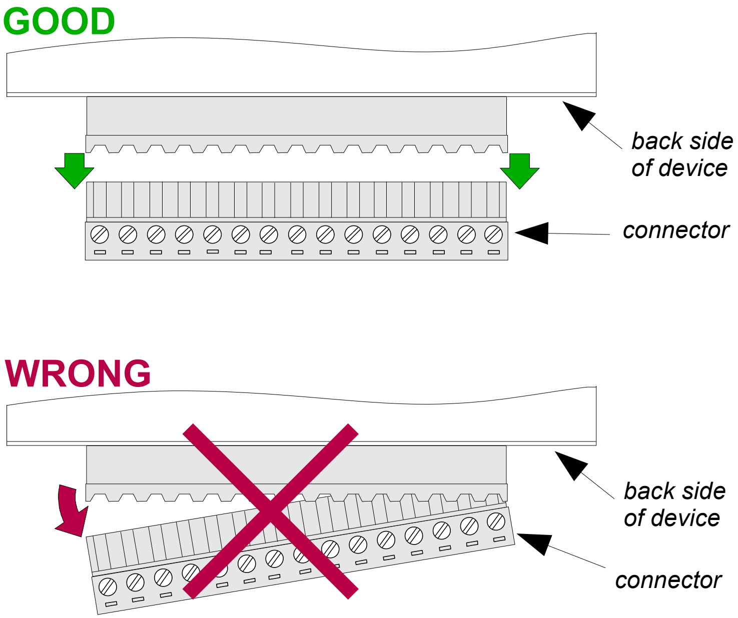


Fig. 4.5. Connectors removing method

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 according to 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 (**Fig. 4.6**). In the 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 that in the case of a short circuit of the cable from the side of the unit the cable shall be protected against destruction with an electrical installation fuse.

- Wiring must meet appropriate standards and local 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 the case of the unit being 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 (marked as n.c.) 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 housing, covers and sealing to protecting against water intrusion, pay special attention to their correct tightening or clamping. In the 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 carries the risk of electrical 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.

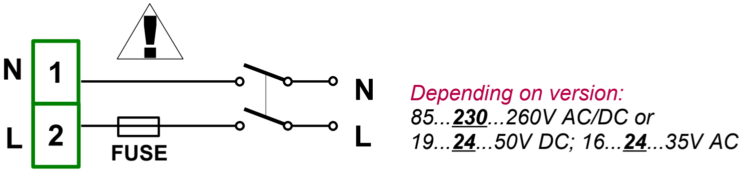


Fig. 4.6. Connection of power supply

- Avoid running signal cables and transmission cables together with power supply cables and cables controlling inductive loads (e.g. contactors). Such cables should cross 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 the 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.
- In the case of measurement or control signals are longer than 30m or go outside of the building then additional safety circuits are required.
- In the 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 with cables with interference (e.g. circuits controlling relays or contactors).

Connections of power supply voltage and measurement signals are executed using the screw connections on the back of the unit's housing.

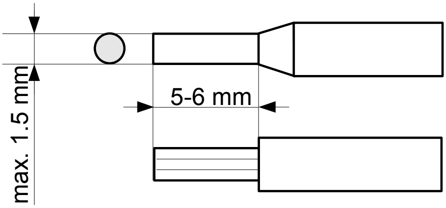


Fig. 4.7. Method of cable isolation replacing and cable terminals dimensions

All connections must be made while power supply is disconnected !

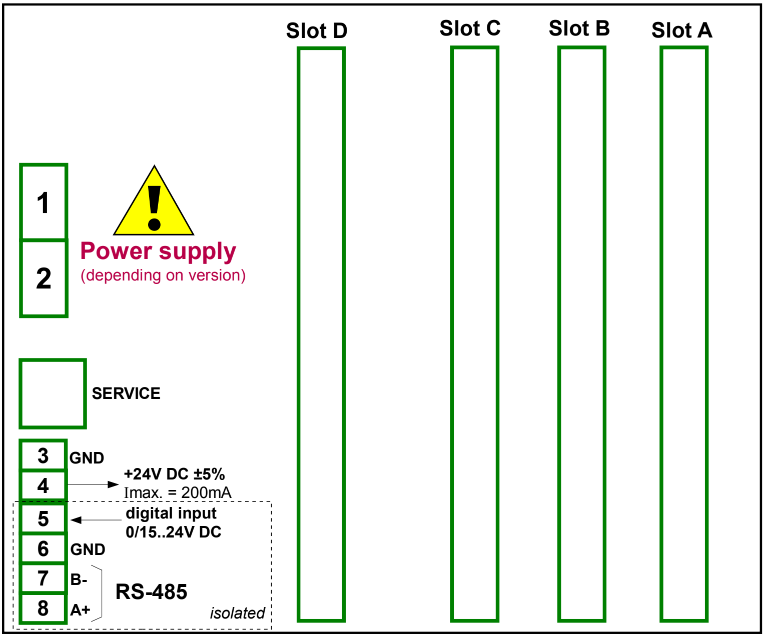


Fig. 4.8. Terminals description

The basic performance of the device (see **Fig. 4.8**) contains only the extreme left terminals:

- Power supply,
- SERVICE,
- Sensor supply output +24V DC Imax=200mA,
- Digital input 0V...15...24V DC (low state: 0÷1V, high state:8÷24V)
- Interface RS-485,

If the **UN3** or **UN5** module is installed, the +24V DC output is available only in devices equipped with **PS42** power supply module. In other cases there is **no +24V DC output** available and these terminals remain disconnected.

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 **Slot A**, **Slot B**, **Slot C**),
- advanced communication module (additional serial, USB and Ethernet interfaces installed in **Slot D**).

According to the order these terminals can look different than shown in **Fig. 4.8** or be not present. Terminals and connections of available modules are shown in **Fig. 4.9÷Fig. 4.21** in **Section 4.3.1.** .

Shown below is an example of a configuration of the installed modules:

- base,
- Slot A - UI8 module (8 current input & 8 voltage input),
- Slot B - RT4 module (4 RTD input),
- Slot C - R81 module (8 relay output 1A/250V),
- Slot D - ACM module (additional serial, USB and Ethernet interfaces).

4.3.1. Available modules

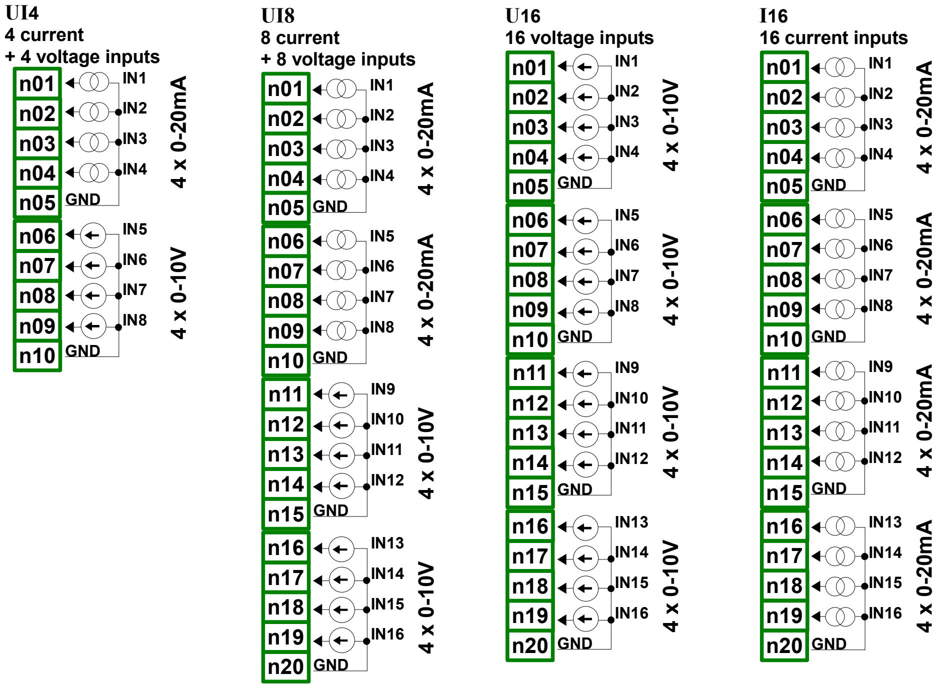


Fig. 4.9. Voltage and current input modules

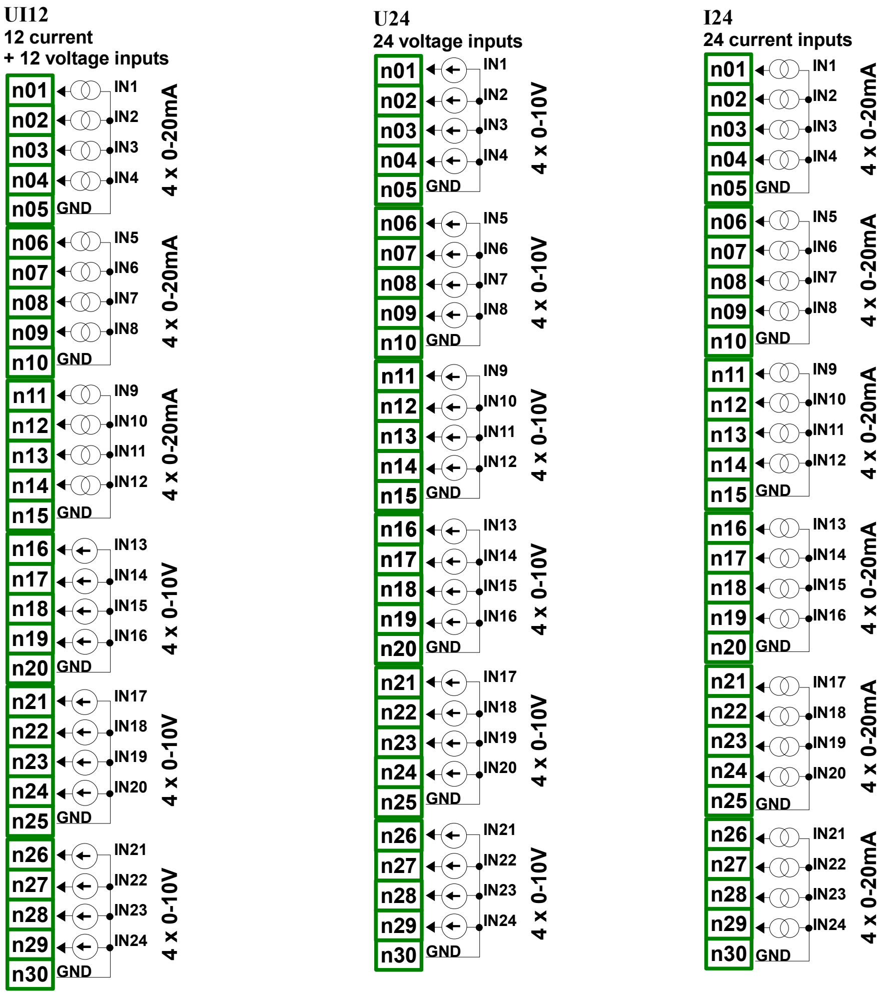


Fig. 4.10. Voltage and current input modules contd.

UI4N8
8 resistance
+ 4 current and 4 voltage inputs

UI4D8
4 current and 4 voltage
+ 8 digital inputs

UI8N8
8 resistance
+ 8 current and voltage inputs

UI8D8
8 current and voltage
+ 8 digital inputs

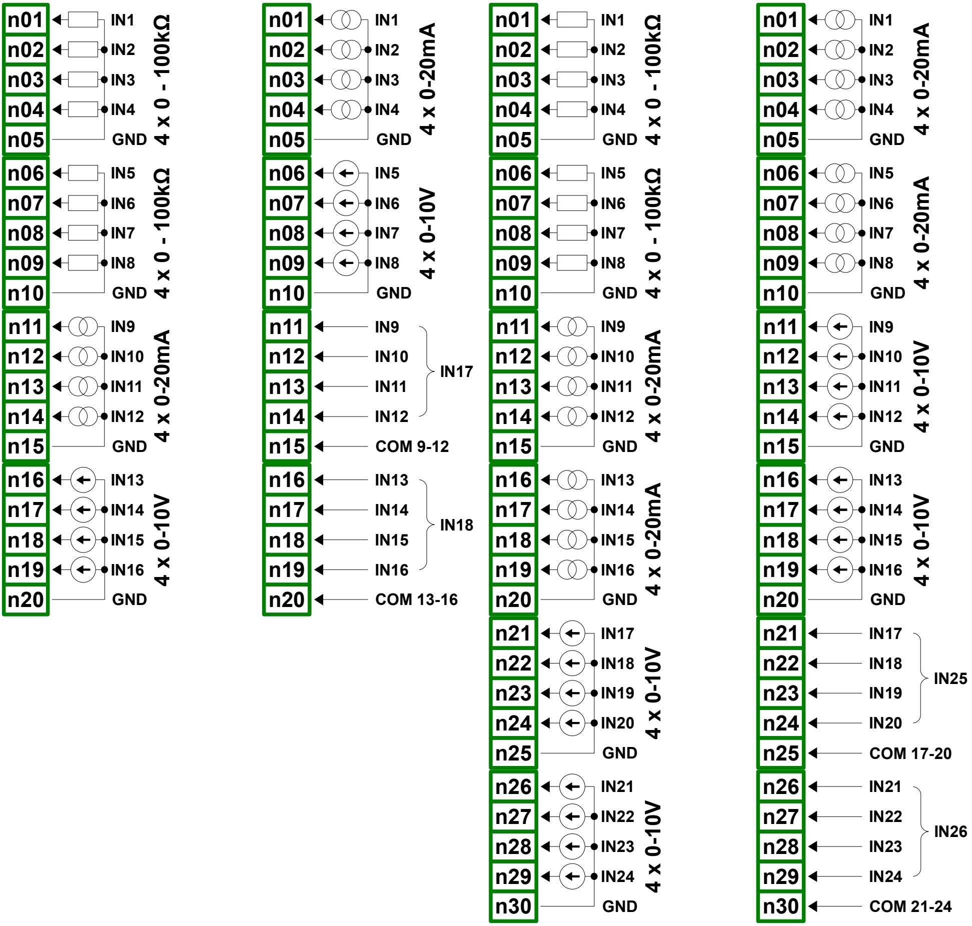


Fig. 4.11. Mixed input modules

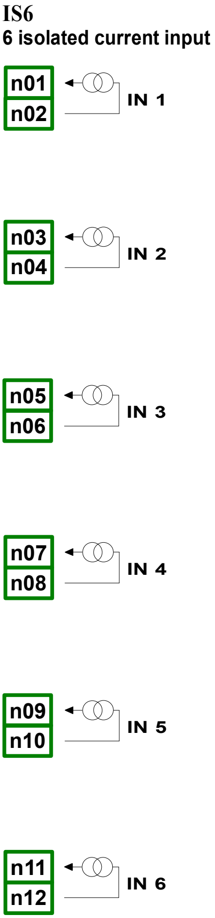


Fig. 4.12.Isolated current input modules

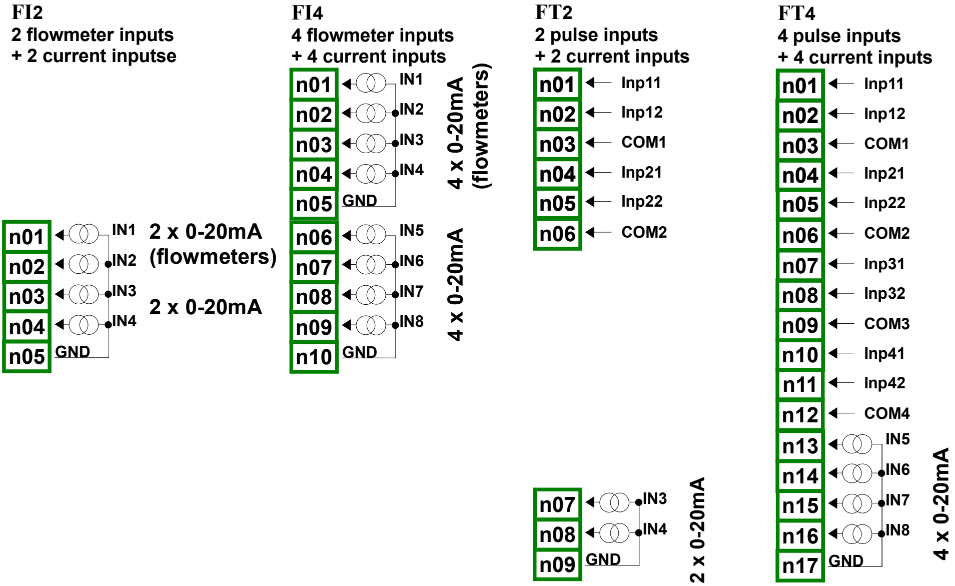
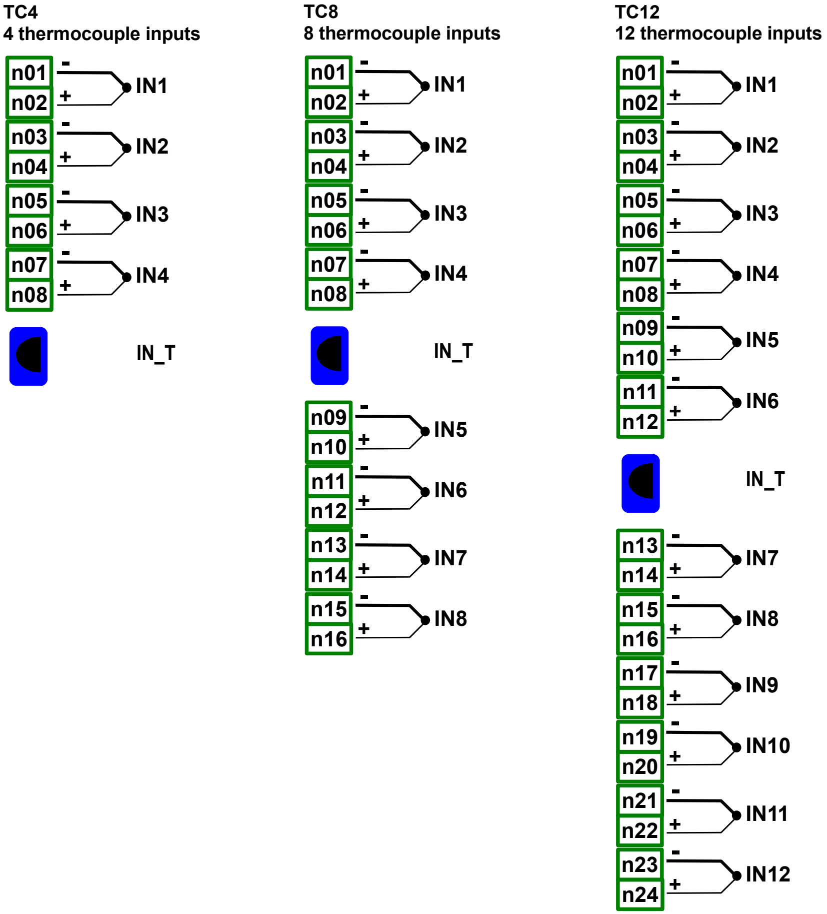


Fig. 4.13. Flowmeter modules



TC4
4 thermocouple inputs

TC8
8 thermocouple inputs

TC12
12 thermocouple inputs

Fig. 4.14. TC input modules

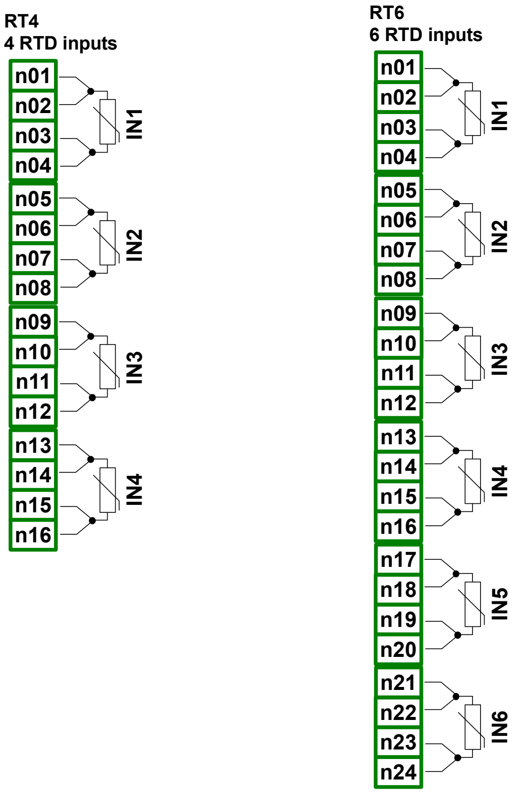


Fig. 4.15. RTD input modules

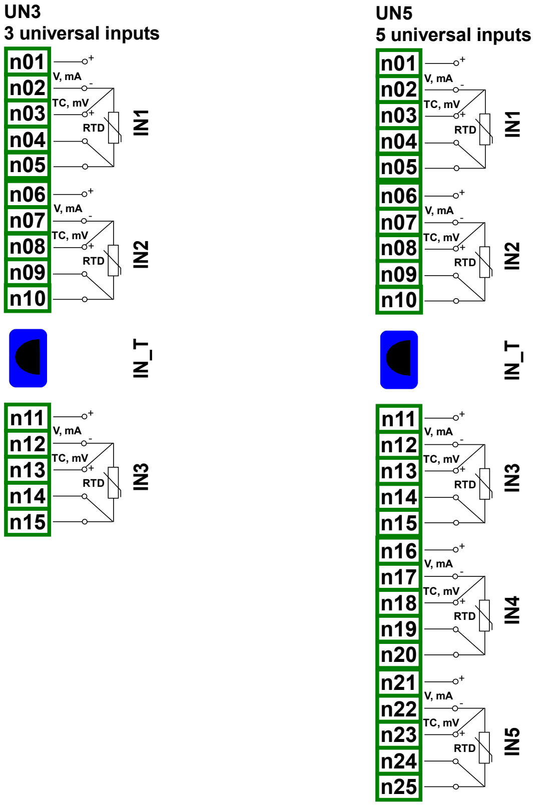


Fig. 4.16. Universal input modules

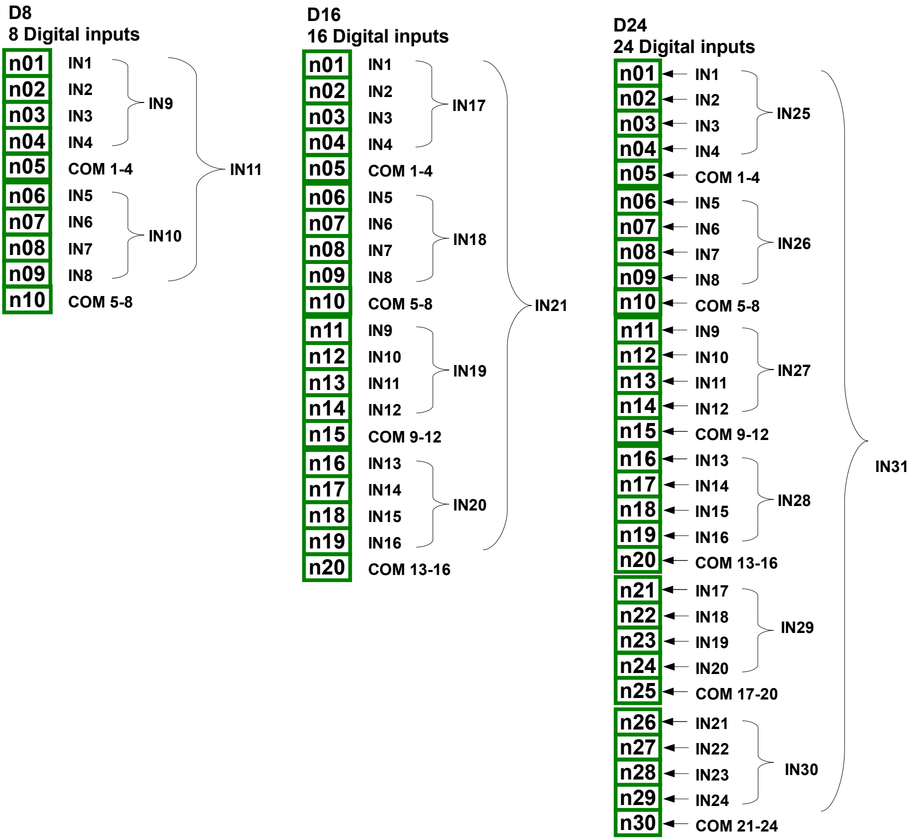


Fig. 4.17. Digital input modules

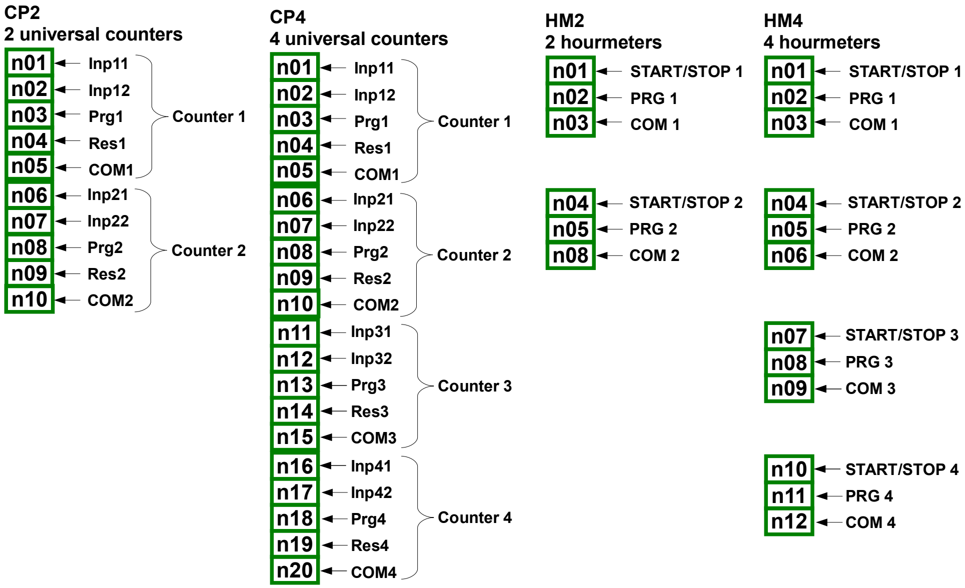
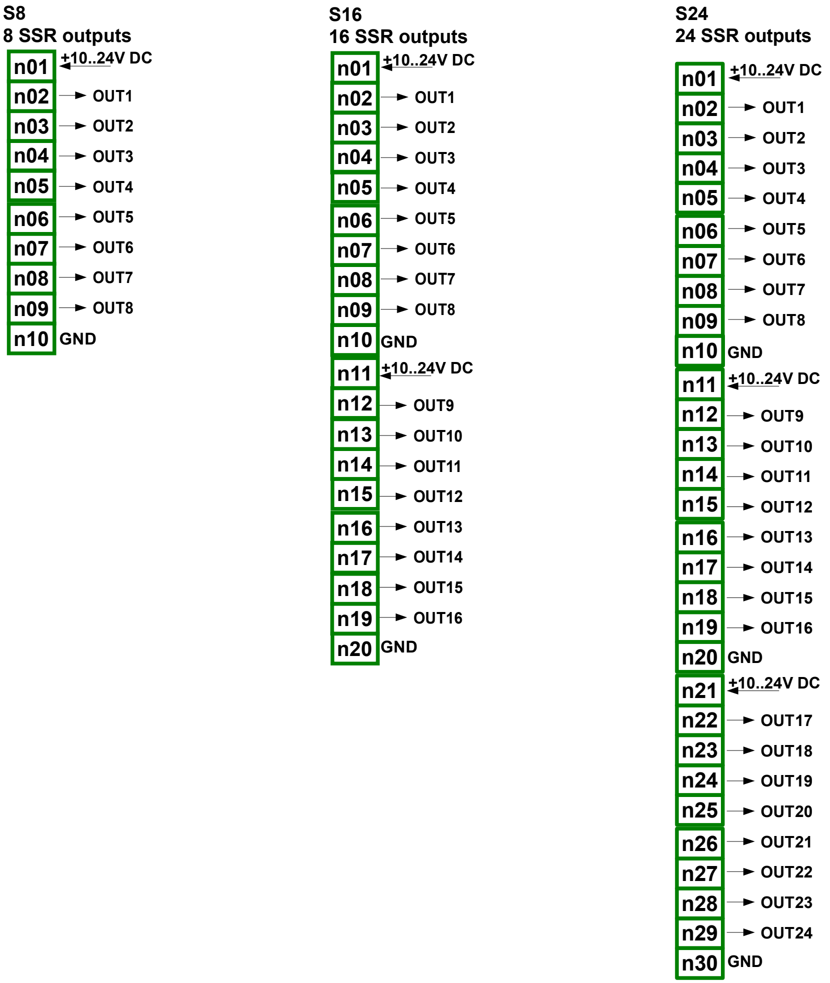


Fig. 4.18. Universal counters and hourmeters modules



S8
8 SSR outputs

S16
16 SSR outputs

S24
24 SSR outputs

Fig. 4.19. SSR output modules

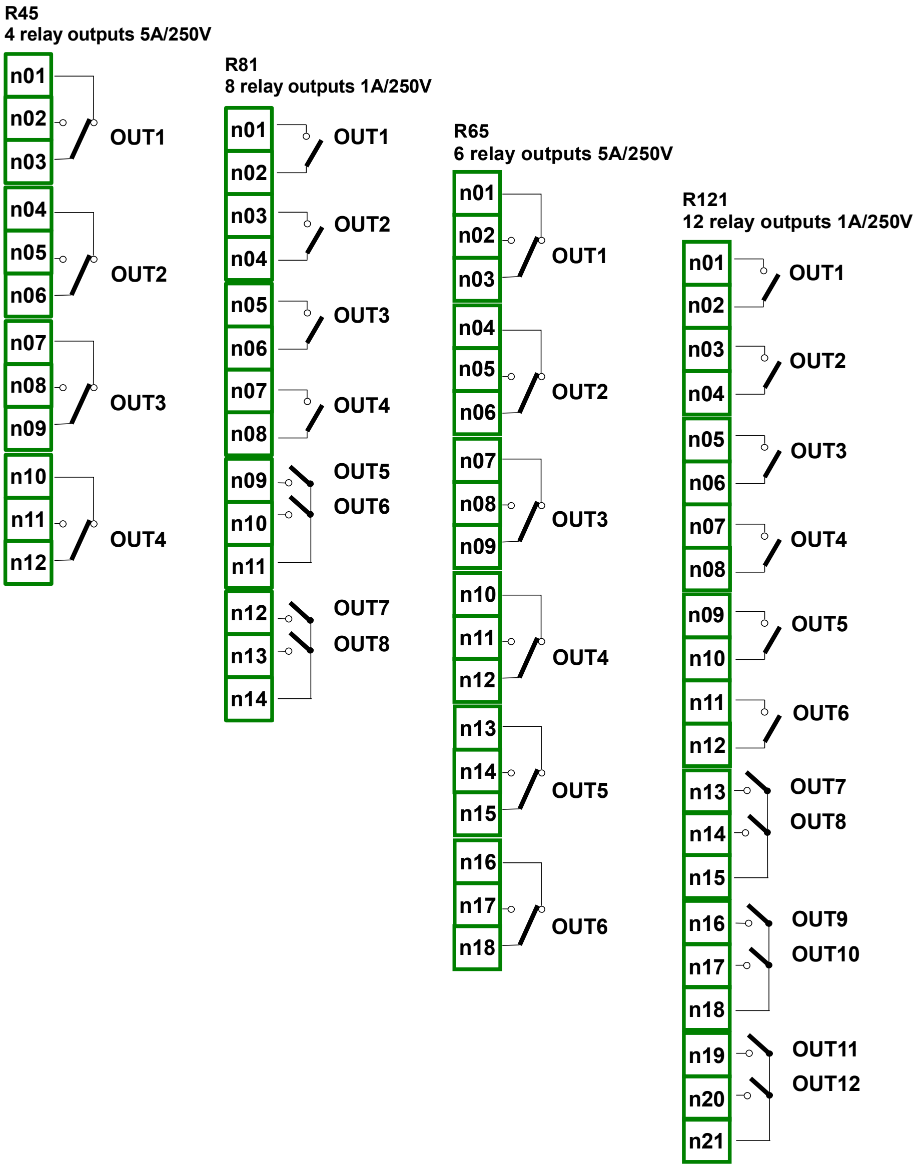


Fig. 4.20. Relay output modules

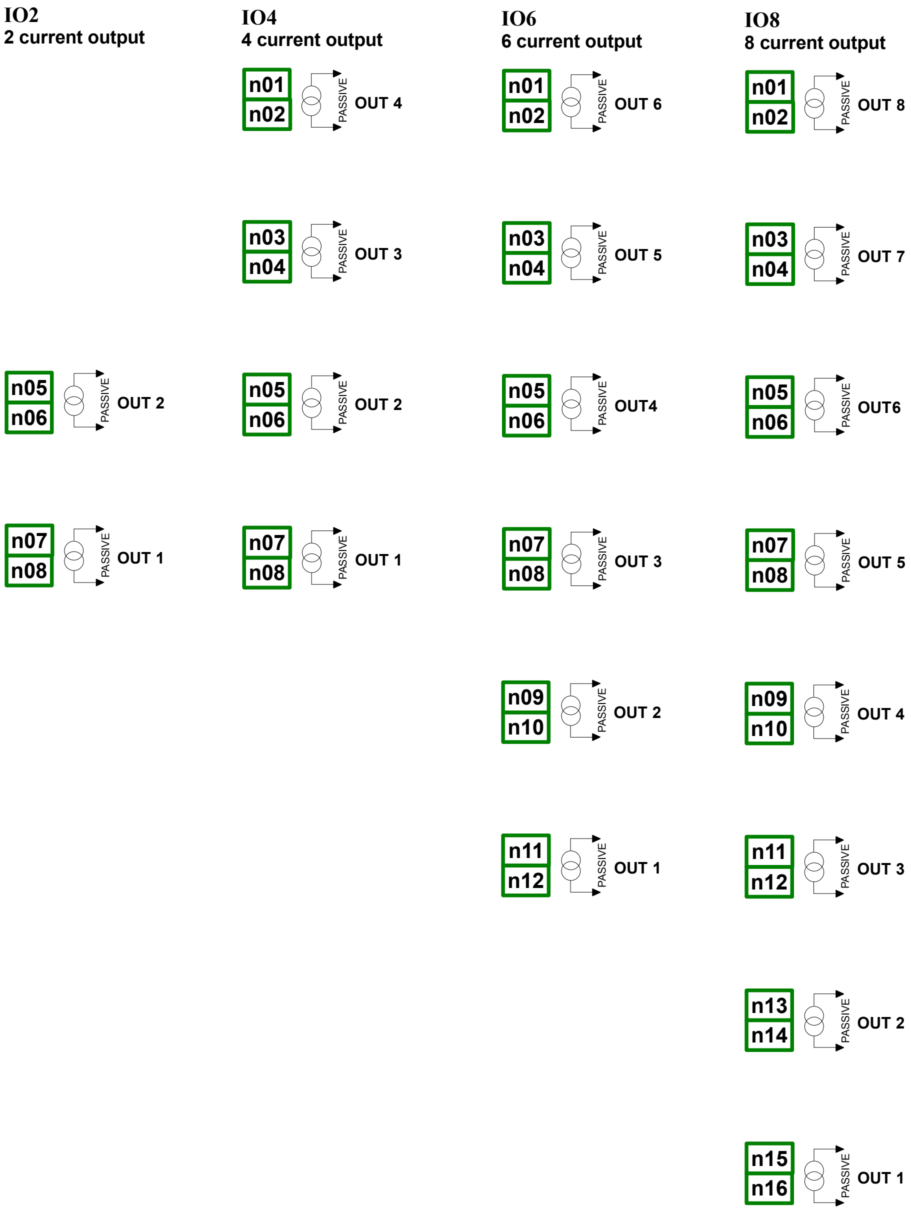


Fig. 4.21. Passive current output modules

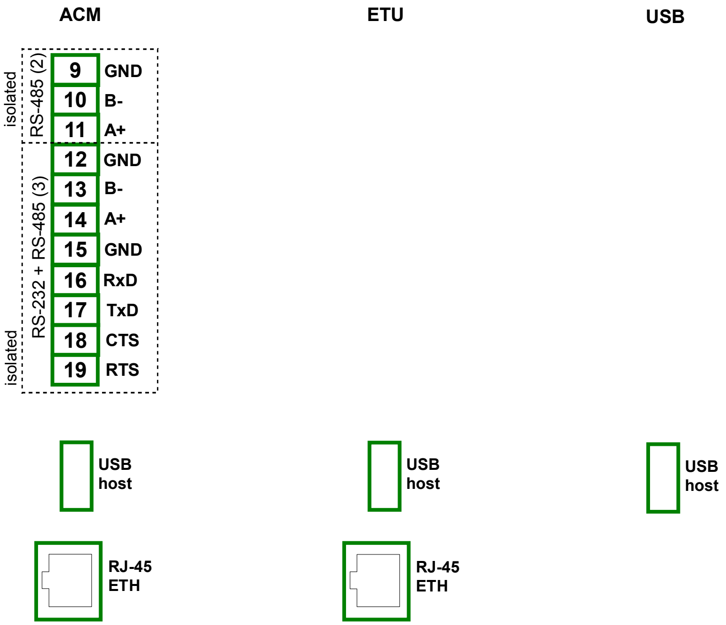


Fig. 4.22. Communication Modules

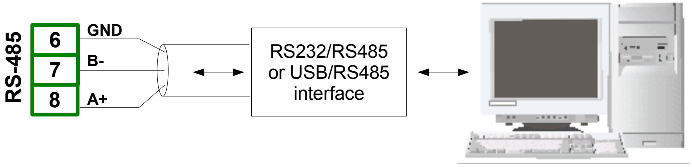


Fig. 4.23. Connection of RS-485 transmission signals

The **CONTROLLER/DATA RECORDER** device supports the following converters:

- USB / RS-485 converter
- RS-232 / RS-485 converter

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 small amount of detergent or in the case of more significant contamination ethyl or isopropyl alcohol.

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

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

5. INTRODUCTION TO CONTROLLER/DATA RECORDER

5.1. UNDERSTANDING CONTROLLER/DATA RECORDER

The **CONTROLLER/DATA RECORDER** device was developed as a universal multichannel controller. To maintain this concept its firmware was written with 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 gives great advantages to high level applications, making it flexible and dynamically configurable. Similarly data structures and streams were 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 **CONTROLLER/DATA RECORDER** decided to use such solution to increase functionality of the device and make software near fully independent on the hardware.

5.1.1. Logical channels

A **Logical Channel** is a data stream existing in the memory of the device, having it's 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.

Depending on type, the device can be equipped with different number of **Logical channels** and other functions. Their amount is shown in table bellow:

type	**96 x 96**	**144 x 144**
LC (logical channels)	60	90 (60 with logging option)
Di (digital inputs)	48	72
Ai (analog inputs)	48	72
Rel/SSR (relays/SSR)	16/48	36/72
Ao (analog outputs)	12	24
MP (Modbus ports)	4	4
F (math functions)	34	34
G (groups)	10	15
P/t (Profile/timers)	8	8
C (controllers)	8	8

Tab.5.1 Maximum amount of functions, included in device

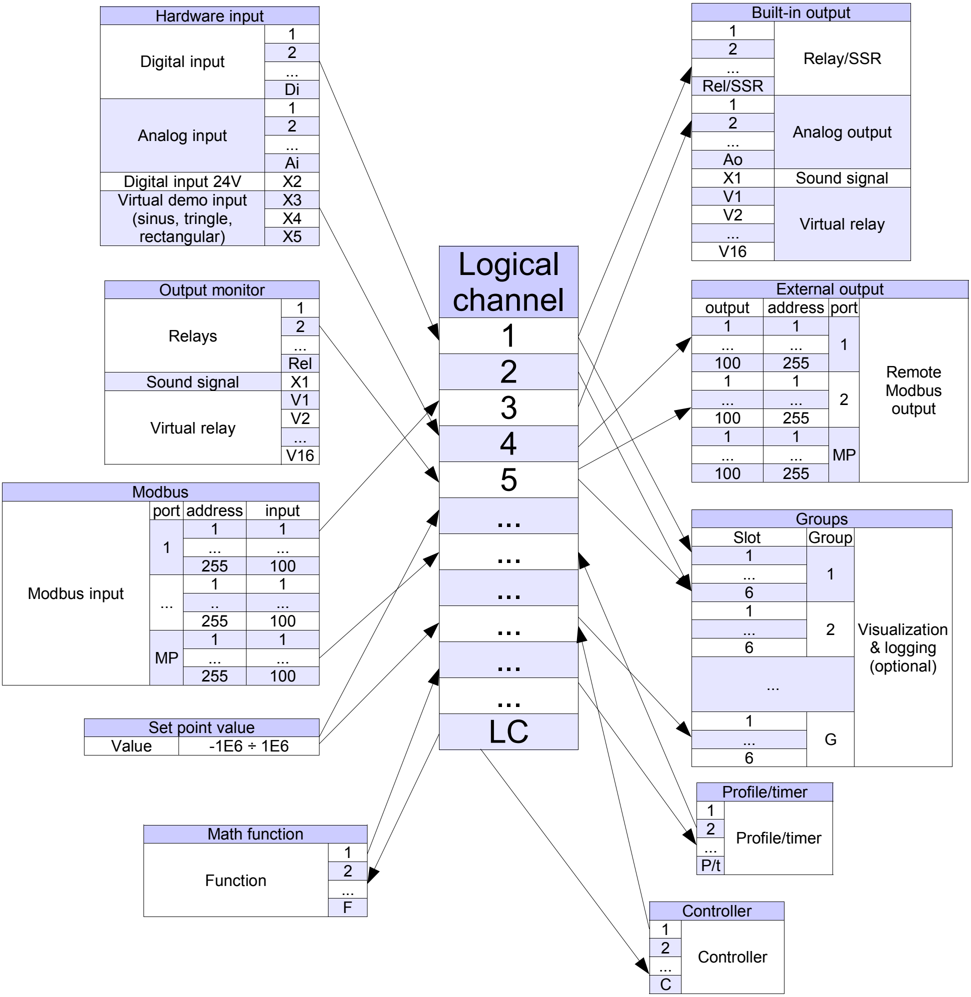


Fig. 5.1. The overall connections structure of the Logical channel with the device I /O

Fig. 5.1 shows general structure of of connections between logical channels, and device inputs/outputs. Each of ***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 **CONTROLLER/DATA RECORDER** via RS-485 interface,
- states and data coming from outputs of controlling processes,
- generated profile/timer
- states of virtual input channels, and timers,
- mathematical combination of other ***Logical Channels***.

More information about Logical channels and samples of configuration Logical channels are presented in **Chapter 7.8. LOGICAL CHANNELS**.

To make visualization clearer ***Logical Channels*** can be gather into ***Groups***.

5.1.2. Groups

A ***Group*** is a set of 1-6 ***Logical Channels***. The **CONTROLLER/DATA RECORDER** 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 single object or representing similar parameters of few separate objects) but it is also possible to create a ***Group*** consisting of completely unrelated channels. Overview of the concept is presented in **Fig. 5.2**.

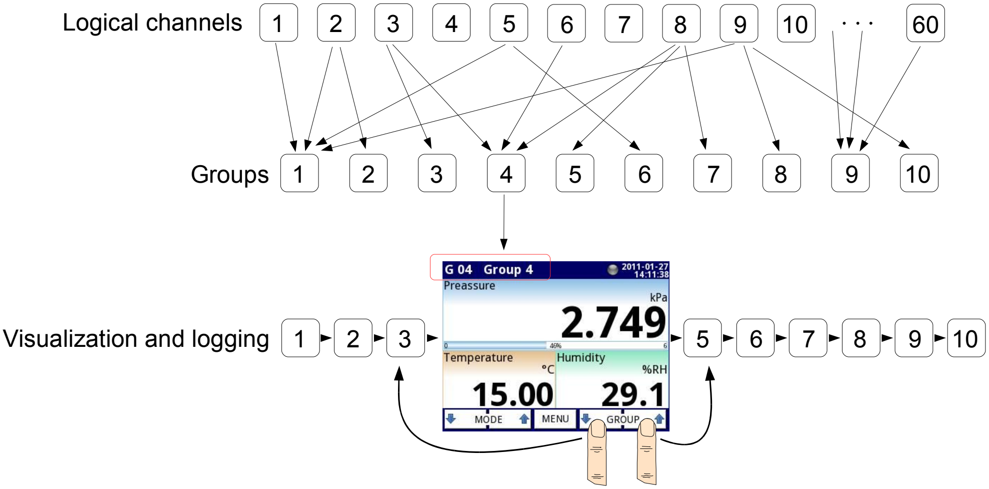


Fig. 5.2. An overview of the concept of Group in the device

Using ***Groups***, ***Logical Channels*** and mathematical combinations of them gives incredible flexibility to the software, allowing for ease in designing advanced control methods and visualisation with a low cost **CONTROLLER/DATA RECORDER**.

More information about **Groups** and samples of Group configurations are presented in **Chapter 7.14. GROUPS**.

5.2. HARDWARE CONFIGURATIONS

The functionality of **CONTROLLER/DATA RECORDER** can fit to the user's needs. The ***base*** of the **CONTROLLER/DATA RECORDER** contains: the main processor, display with 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 **Fig. 4.8** - most far left connectors. All other modules are optional and can be installed inside the device according to customer's needs. Next to the basic connectors is the slot for an advanced communication module. In the simplest version this module can be equipped only with rear USB Host connector (this is standard for the IP-65 version of the **CONTROLLER/DATA RECORDER**). The full version of this module contains also 2 additional serial ports (RS485 and RS485/RS232) and a 10Mb Ethernet RJ-45 connector (see **Fig. 4.22**).

Three slots designed for built-in hardware inputs and outputs are installed on the right side of the case (see **Fig. 4.8**, terminals marked: slot A, slot B and slot C). The number and size of these terminals varies depending on module type. Brief descriptions of available modules are shown in **Fig. 4.9÷Fig. 4.22**. Measurement and actuator modules are constantly being developed, so the current list of available modules varies (visit manufacturers website to check current list of **CONTROLLER/DATA RECORDER** modules).

Basic measurement modules are:

- 4/8/16/24x Voltage/Current input module,
- 16/24x NTC/Voltage/Current/Digital input module,
- 6x Isolated current input module,
- 2/4x flowmeter + 2/4x current input,
- 2/4x pulse input + 2/4x current input,
- 4/6x RTD input module,
- 4/8/12x TC input module.
- 8/16/24x Digital input
- 3/5x universal input
- 2/4x universal counter input
- 2/4x hourmeter input

Output modules are:

- 8/16/24x SSR driver module,
- 8/12x Relay 1A/250V module,
- 4/6x Relay 5A/250V module,
- 2/4/6/8x IO passive current output.

Communication module:

- ACM,
- ETU,
- USB,

6. WORKING WITH THE CONTROLLER/DATA RECORDER

6.1. CONTROLLER/DATA RECORDER POWER UP

After powering up a starting Logo is showed on the **CONTROLLER/DATA RECORDER** 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 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 in the lower left corner displays the software version. After that the main application is started. The view of the main program depends of the **General settings** (see the **Chapter 7.7. GENERAL SETTINGS**) and **Group** settings (see **Chapter 7.14. GROUPS**). An example view of the main program shown in **Fig. 6.2**.

6.2. THE USE OF THE TOUCH-SCREEN

Do not use pointers with sharp edges (like tips of pencils and pens, knifes, scissors, needles, wires, nails, screws, bolts etc.) while working with touch-screen. It is strongly recommended to use a special stylus made of plastic or another soft material with round ends (for example the pionter delivered with the device) or a finger. The display of the **CONTROLLER/DATA RECORDER** 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 LCDs and a soft cloth.

6.3. DISPLAY

The **CONTROLLER/DATA RECORDER** displays all data and dialogue on a 3,5” or 5,7” 320x240 pixel, color TFT screen with embedded touch screen panel. New devices have the display protected with a thin transparent plastic cover. This protective layer should be removed before use to ensure perfect visibility of pictures and sensitivity of the touch-screen.

While normal operation the **CONTROLLER/DATA RECORDER** displays data in a form selected by user, at any time it is possible to switch presentation mode and group or show configuration menu. All details of the user interface are designed to make use of device easy and intuitive. To change display mode, group or to enter the menu, touch the screen of the **CONTROLLER/DATA RECORDER** and press appropriate button in the **Navigation bar**.

ⓘ Further information about menu and presentation modes are described in further chapters.

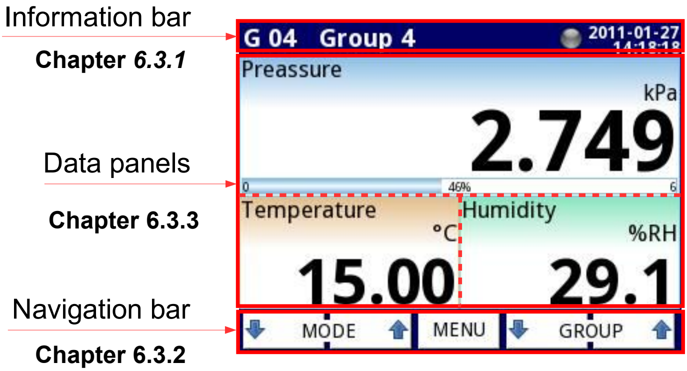


*Fig. 6.1. Typical view of a **CONTROLLER/DATA RECORDER** main page, after touching display*

6.3.1. Information bar

The **Information bar** informs the user about current, display group, logging, actual date and time.

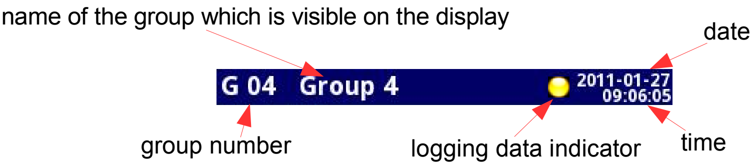


Fig. 6.2. View of information bar

Information bar displays (**Fig. 6.2**):

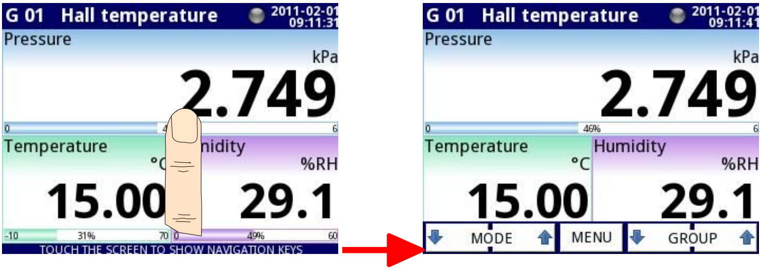
- **name of the Group** visible on the screen, in place of standard name (e.g. **Group 4**). It is possible to enter a more descriptive name for clarity (for more information see **Chapter 7.14. GROUPS**),
- **group number** - number of the currently displayed Group, to change the displayed **Group** press button **[↓GROUP]** or **[GROUP↑]** in the **Navigation bar** (see **Chapter 6.3.2. Navigation bar**),
- **time** and **date** - actual time and date display on the right upper corner on the screen can be set in **General settings** (see **Chapter 7.7. GENERAL SETTINGS**),
- **logging data indicator** - located in the **Information bar** the **logging data indicator** changes color depending of state 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 7.4. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION**), or when the logging option is activated but is 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 7.14.2. Groups - Logging options**),

- **yellow color** – It is possible to log data in the device but there is less than **10MB** of free memory remaining (to clear the memory you need to move onto a removable flash drive any important data logging files and possibly the Modbus templates, after which remove them from the device - more information see **Chapter 7.3. FILES MANAGEMENT**),
- **red color** - warning about the lack of space on memory card, meaning data logging would 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 7.3. 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 logging 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 **Fig. 7.14**).

6.3.2. Navigation bar

The touching the screen at any place causes the **Navigation bar** to display (see **Fig. 6.3**) which allows the user to switch between visualisation modes, groups and to enter the menu.



TOUCH THE SCREEN TO SHOW NAVIGATION KEYS

Fig. 6.3. Main window of device – displaying the ***Navigation bar***

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

⇩ MODE ⇧	switching between visualisation modes of current group (for possible modes see **Chapter 6.3.3. Data panels** and **Chapter 7.14. GROUPS**)
MENU	entering the main menu (see details in **Chapter 7. CONFIGURATION OF THE CONTROLLER/DATA RECORDER**)
⇩ GROUP ⇧	switching between presented groups of logical channels (activation and settings for **Group** view see **Chapter 7.14. GROUPS**)

Tab. 6.1 Buttons of the navigation bar

To enter directly into the configuration menu of particular **Logical channel**, press and hold screen over the channel data panel for 3-4 seconds (see option **(1)** in the **Fig. 6.4** entering configuration of the **logical channel** named **'Temperature'**). Similarly to go directly to configuration of displayed **Group**, touch and hold **the group number** or **group name** in the upper **Information bar** for a few seconds (see option **(2)** in the **Fig. 6.4** entering configuration of the **Group** named **'Group 4'**). In both cases if a password is set (see **Chapter 7.17. ACCESS OPTIONS**) then the user has to enter the password before entering the configuration.

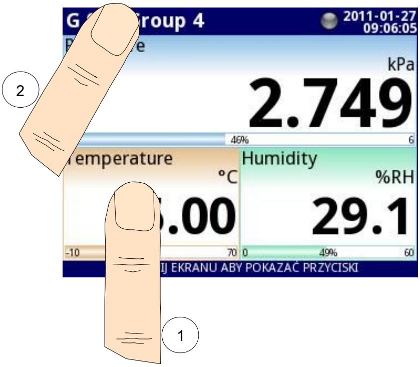


*Fig. 6.4. Methods for direct entry to **Logical channel** configuration **(1)** and **Group** configuration **(2)***

6.3.3. Data panels

The great deal of the screen is being used for channel visualisation. Data can be presented in one of the following modes:

- as numerical values,
- as charts,
- as bars,
- as needle dials.

All 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 **6.6 ÷ 6.11** show examples of different views. The switching between visualisation modes can be done by pressing the buttons **[MODE↑]** or **[MODE↓]** in the **Navigation bar** (see **Chapter 6.3.2. Navigation bar**). Switching between groups can be done by pressing the buttons **[GROUP↑]** or **[GROUP↓]**.

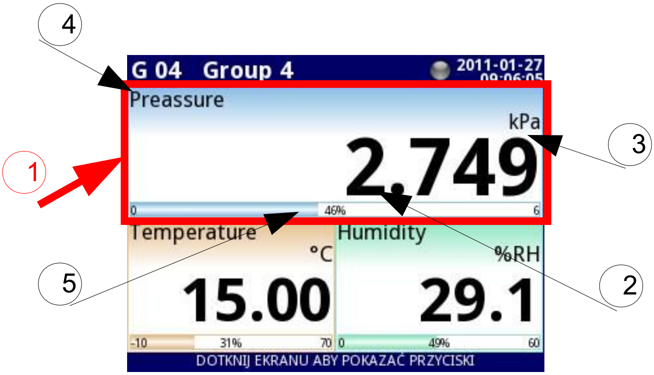


*Fig. 6.5. View of the **Data panel***

In all data panels (a sample of a data panel is shown inside the frame marked **(1)** in **Fig. 6.5**) the following information is available:

- value of the logical channel (denoted by **(2)** in **Fig. 6.5**),
- 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 it's full scale (denoted by **(5)**),

Every ***Group*** of ***Logical Channels*** can be presented in one of 6 modes:

- as numerical values **Fig. 6.6**
- as horizontal bars **Fig. 6.7**
- as vertical bars **Fig. 6.7**
- as horizontal charts **Fig. 6.8**
- as vertical charts **Fig. 6.8**
- as phasor charts **Fig. 6.9**
- as needle dials **Fig. 6.10**

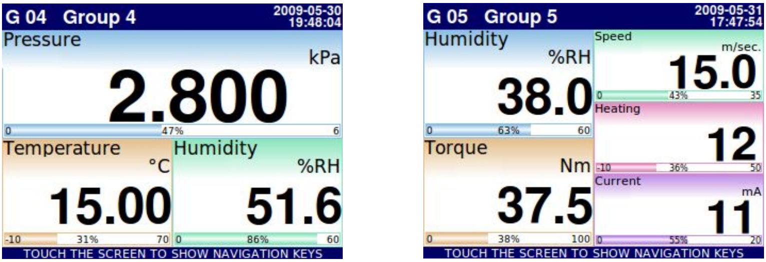


*Fig. 6.6. Examples of **Numerical Values** presentation mode*

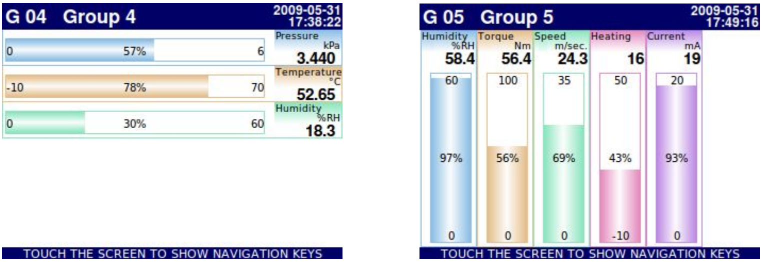


Fig. 6.7. Examples of **Horizontal** (for 3 channels) and **Vertical Bars** (for 5 channels)

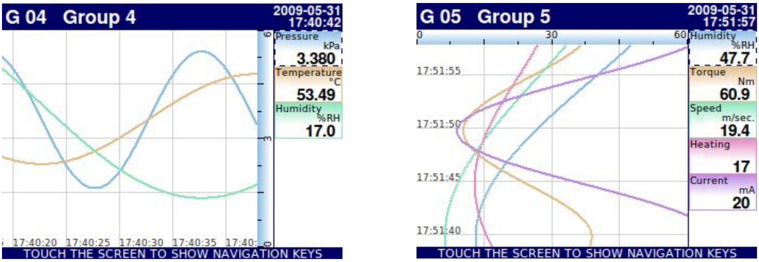


Fig. 6.8. Examples of **Horizontal** (for 3 channels) and **Vertical Charts** (for 5 channels)

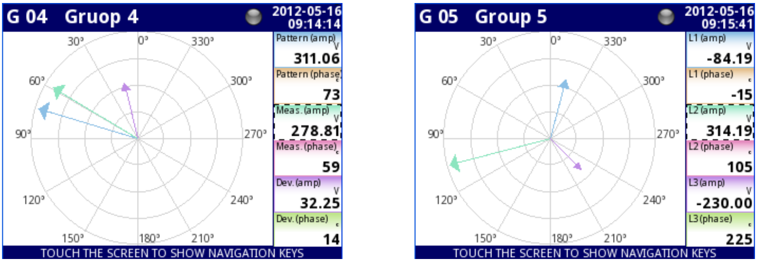


Fig. 6.9 Examples of **Phasor Charts** presentation mode

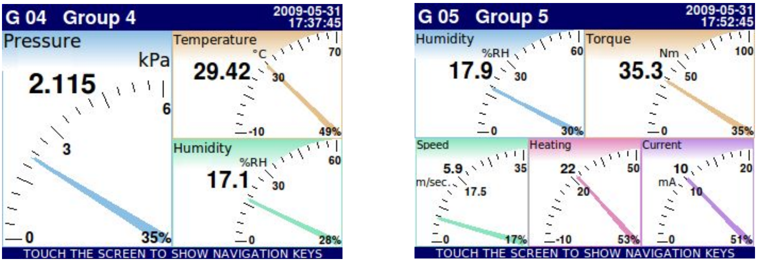
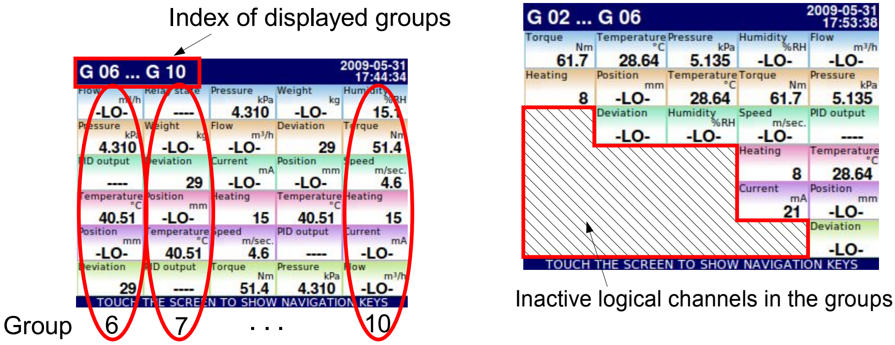

G 04 Group 4
2009-05-31 17:37:45
Pressure kPa 2.115
6 3 0 35%
Temperature °C 29.42
70 30 -10 49%
Humidity %RH 17.1
60 30 0 28%
TOUCH THE SCREEN TO SHOW NAVIGATION KEYS
G 05 Group 5
2009-05-31 17:52:45
Humidity %RH 17.9 60 30 0 30%
Torque Nm 35.3 100 50 0 35%
Speed m/sec. 5.9 35 17.5 0 17%
Heating 22 50 20 -10 53%
Current mA 10 20 10 0 51%
TOUCH THE SCREEN TO SHOW NAVIGATION KEYS

Fig. 6.10. Examples of **Needle Dials** for 3 channels and for 5 channels


Index of displayed groups
Group 6 7 . . . 10
Inactive logical channels in the groups

G 06 ... G 10 2009-05-31 17:44:34

Flow m³/h -LO-	Relay state ----	Pressure kPa 4.310	Weight kg -LO-	Humidity %RH 15.1
Pressure kPa 4.310	Weight kg -LO-	Flow m³/h -LO-	Deviation 29	Torque Nm 51.4
PID output ----	Deviation 29	Current mA -LO-	Position mm -LO-	Speed m/sec. 4.6
Temperature °C 40.51	Position mm -LO-	Heating 15	Temperature °C 40.51	Heating 15
Position mm -LO-	Temperature °C 40.51	Speed m/sec. 4.6	PID output ----	Current mA -LO-
Deviation 29	PID output ----	Torque Nm 51.4	Pressure kPa 4.310	Flow m³/h -LO-

TOUCH THE SCREEN TO SHOW NAVIGATION KEYS

G 02 ... G 06 2009-05-31 17:53:38

Torque Nm 61.7	Temperature °C 28.64	Pressure kPa 5.135	Humidity %RH -LO-	Flow m³/h -LO-
Heating 8	Position mm -LO-	Temperature °C 28.64	Torque Nm 61.7	Pressure kPa 5.135
	Deviation -LO-	Humidity %RH -LO-	Speed m/sec. -LO-	PID output ----
			Heating 8	Temperature °C 28.64
			Current mA 21	Position mm -LO-
				Deviation -LO-

TOUCH THE SCREEN TO SHOW NAVIGATION KEYS

Fig. 6.11. Examples of simultaneous presentation of ***Many Groups***

There is also the possibility to show many groups on a single screen (**Fig. 6.11**). 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 display on the screen, starting from the left side of the page, groups: 8, 9, 10, 1, 2),

See **Chapter 7.14. 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. Figures show below (**Fig. 6.12÷Fig. 6.15**) present examples of message types.

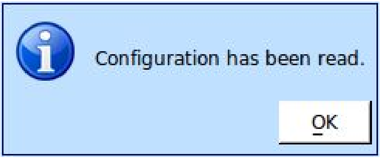

Configuration has been read.

Fig. 6.12. Information message

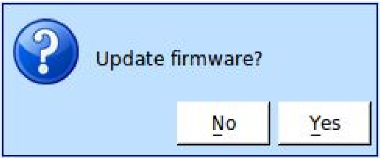

Update firmware?

Fig. 6.13. Question message

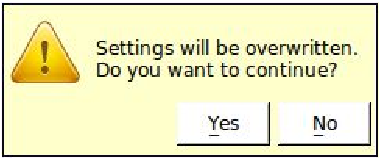

Settings will be overwritten.
Do you want to continue?

Fig. 6.14. Warning message

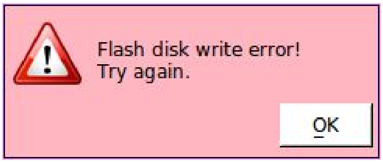

Flash disk write error!
Try again.

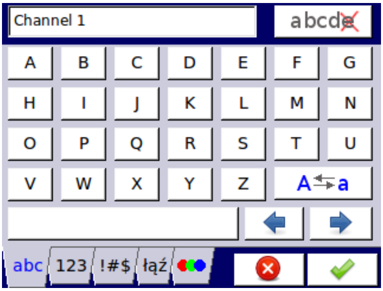
Fig. 6.15. Alert message

7. CONFIGURATION OF THE CONTROLLER/DATA RECORDER

7.1. EDIT DIALOGUES

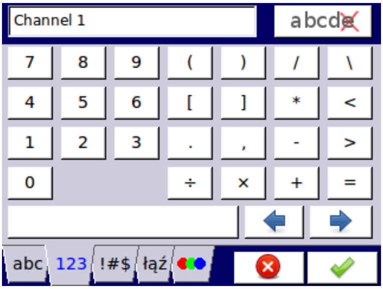
Configuration process are based on edit dialogues. Some of the dialogues are common to different menus, such dialogues are:

- text editor, which is divided into tabs:
  - letters, see **Fig. 7.1**,
  - numbers and arithmetic signs, see **Fig. 7.2**,
  - the special symbols, see **Fig. 7.3**,
  - diacritical letters, **Fig. 7.4**,
  - font and background colors, see **Fig. 7.5**,
- values editor, which is divided into tabs:
  - decimal form, see **Fig. 7.6**,
  - hexadecimal form, see **Fig. 7.7**,
  - binary form, see **Fig. 7.8**,
- switch editor, which is divided:
  - single choice type options, see **Fig. 7.9**,
  - multiple choice type options, see **Fig. 7.10**,
- file editor, which is divided:
  - single file selection, see **Fig. 7.11**,
  - multiple files selection, see **Fig. 7.12**,



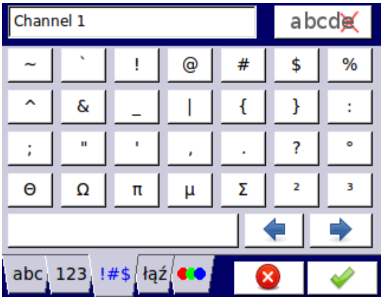
Channel 1

Fig. 7.1. Text editor – letters



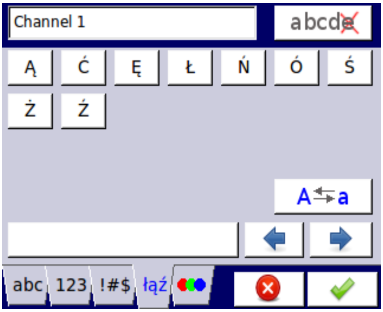
Channel 1

Fig. 7.2. Text editor – numbers



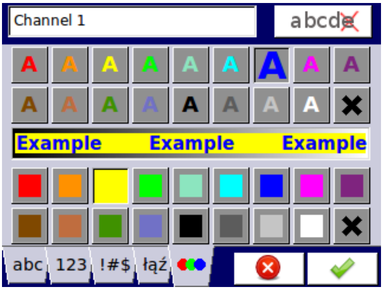
Channel 1

Fig. 7.3. Text editor – special symbols



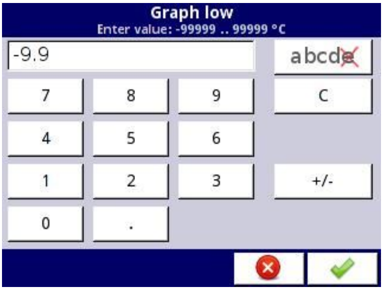
Channel 1

Fig. 7.4. Text editor example – diacritical letters



Channel 1

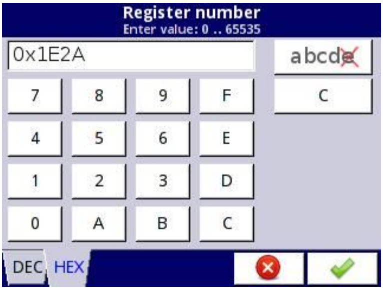
Fig. 7.5. Text editor – font and background color selection



Graph low
Enter value: -99999 .. 99999 °C

-9.9

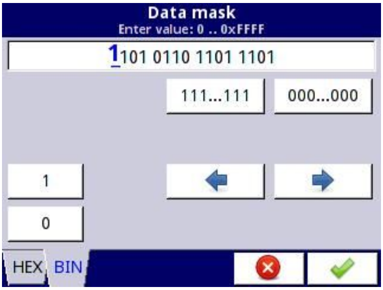
Fig. 7.6. Value editor – decimal form



Register number
Enter value: 0 .. 65535

0x1E2A

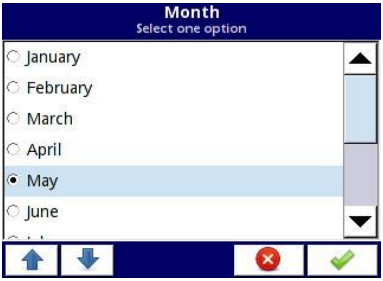
Fig. 7.7. Value editor – hexadecimal form



Data mask
Enter value: 0 .. 0xFFFF

1101 0110 1101 1101

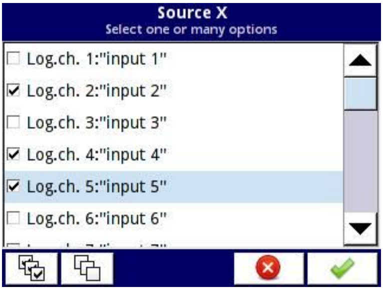
Fig. 7.8. Value editor – binary form



Month
Select one option

○ January
○ February
○ March
○ April
● May
○ June

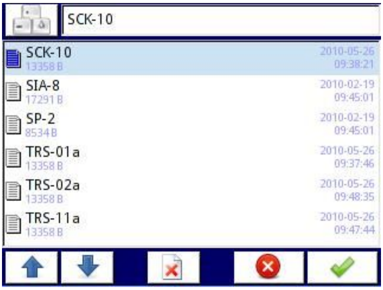
Fig. 7.9. Single choice type editor



Source X
Select one or many options

□ Log.ch. 1:"input 1"
☑ Log.ch. 2:"input 2"
□ Log.ch. 3:"input 3"
☑ Log.ch. 4:"input 4"
☑ Log.ch. 5:"input 5"
□ Log.ch. 6:"input 6"

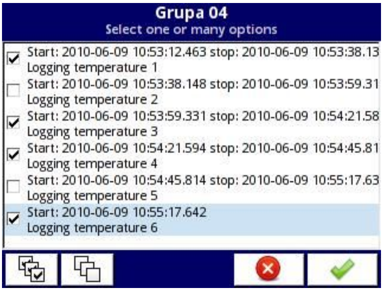
Fig. 7.10. Multiple choice type editor



SCK-10

SCK-10 13358 B 2010-05-26 09:38:21
SIA-8 17291 B 2010-02-19 09:45:01
SP-2 8534 B 2010-02-19 09:45:01
TRS-01a 13358 B 2010-05-26 09:37:46
TRS-02a 13358 B 2010-05-26 09:48:35
TRS-11a 13358 B 2010-05-26 09:47:44

Fig. 7.11. File editor - single file selection



Grupa 04
Select one or many options

☑ Start: 2010-06-09 10:53:12.463 stop: 2010-06-09 10:53:38.13
Logging temperature 1
□ Start: 2010-06-09 10:53:38.148 stop: 2010-06-09 10:53:59.31
Logging temperature 2
☑ Start: 2010-06-09 10:53:59.331 stop: 2010-06-09 10:54:21.58
Logging temperature 3
☑ Start: 2010-06-09 10:54:21.594 stop: 2010-06-09 10:54:45.81
Logging temperature 4
□ Start: 2010-06-09 10:54:45.814 stop: 2010-06-09 10:55:17.63
Logging temperature 5
☑ Start: 2010-06-09 10:55:17.642
Logging temperature 6

Fig. 7.12. File editor - multiple file selection

Functions of common buttons

⇥	"**Exit**" - exits from current menu or sub-menu
✓	"**OK**" - accept choice or changes of edit dialogue (and exit from this dialogue)
✗	"**Cancel**" - reject entered choice or changes of edit dialogue (and exit from this dialogue)
⬇ 9 ⬆	Selection of 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 particular element from the list.
⬆ ⬇	Navigation keys in choice type dialogues.
⬅ ➡	Move arrows. Allow to move cursor along the edited text.
A⇆a	"**Caps lock**" - switches between lower and upper case letters.
abcd✗	"**Backspace**". When editing values, pressing this button deletes last visible number. When editing text, the last edited symbol shown directly before cursor is deleted.
C	"**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 Text editor window.
+	Add a new object
−	Delete a selected object

Tab.7.1 Button functions common for different views

7.2. 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 **Fig. 7.13**). 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 are described in further chapters.

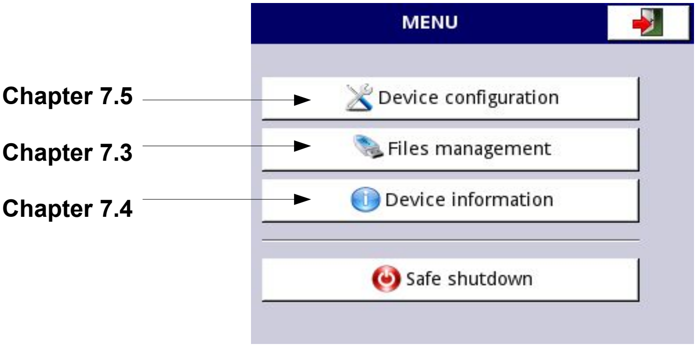


Fig. 7.13. Main menu window

The **Safe shutdown** button allow for a safe power down of the device. After pressing the button and accepting the **warning message** the screen will look like in the **Fig. 7.14**. 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. Not complying with these instructions could cause loss of recently logged data samples.

IT IS NOW SAFE TO POWER OFF DEVICE.

*Fig. 7.14. The view of the screen after pressing the **Safe shutdown** button*

7.3. FILES MANAGEMENT

Fig. 7.15. This button allows to entry to the files management menu

After pressing **MENU -> Files management** (see **Fig. 7.15**) opens the files management menu which is used to exchange data with a flash drive.

Requirements for a flash drive:

- Maximum current consumption is 100mA. Some flash drives with large capacities are not supported by the device (in this case can use an external USB hub with power supply). The manufacturer recommends the use of flash drives of 2GB in size.
- The flash disk must be formatted for Windows as FAT (NOTE!! not FAT32).
- update files, configurations files, and Modbus templates must be located in the main folder (the root of the drive).

A view of the **File management** menu is shown in **Fig. 7.16**. There are three buttons called **Logging files**, **Configuration files** and **Permission files**. The **Configuration files** button is always available. The **Logging files** button is available when the device has the data logging option activated (more information about the licence key for data logging is located in **Chapter 7.4.**) and the **Permission files** button is available when **Access mode** parameter in **Access options** menu is set as **MultiLevel** (see **Chapter 7.17.**).

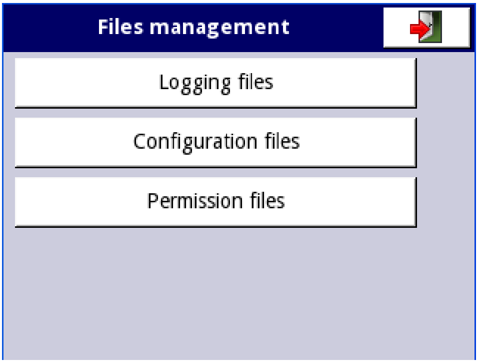
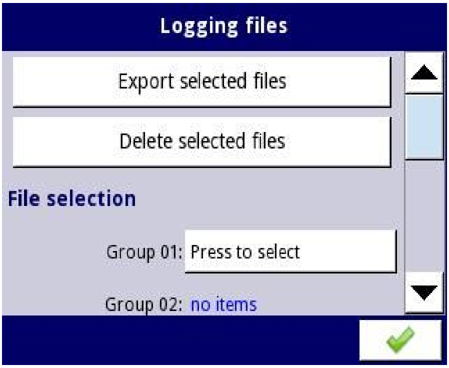


Fig. 7.16. ***Files management*** *menu*

Logging files:

The **Logging files** button (see **Fig. 7.16**) opens the logging files management menu (see **Fig. 7.17**). This button exists only when the user has has input a valid licence for logging data. To export and/or delete logged files follow these steps:

- select a file/s of logged data from a group,
- select the more files in the other groups (if needed),
- export selected files to flash drive and / or delete selected logged data files,



Logging files

Export selected files

Delete selected files

File selection

Group 01: Press to select

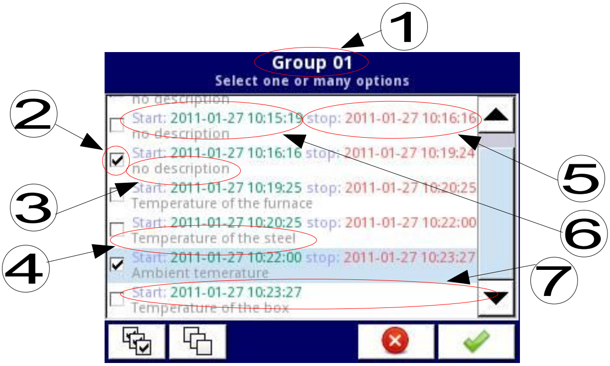
Group 02: no items

✓

*Fig. 7.17. **Logging files** menu*

The **Logging files** menu is presented in **Fig. 7.17**. The menu consists of buttons:

- **Export files** - after pressing this button the selected logged files will be exported to a flash drive,
- **Delete files** - after pressing this button the selected logged files will be removed from the device,
- **Press to select** next to the **Damaged files** label- (this button is visible if at least one damaged file exists) when the user presses this button the window with a list of damaged files is dispayed (the damaged logging files are those that contain errors caused by inappropriate shutdown the device during data logging). There is no guarantee that this logged data will be readable,
- **Press to select** next to **Group number** label - if the user has enabled the logging of particular group of logical channels (see **Chapter 7.14.2. Groups - Logging options**) in the **Logging files** menu next to the label of the group number appears the button **'Press to select'**. Depending on how many groups (the device can define 10 groups) logging is enabled (past or present) as many **'Press to select'** buttons will be active.



Group 01

Select one or many options

no description

□ Start: 2011-01-27 10:15:19 stop: 2011-01-27 10:16:16

no description

☑ Start: 2011-01-27 10:16:16 stop: 2011-01-27 10:19:24

no description

□ Start: 2011-01-27 10:19:25 stop: 2011-01-27 10:20:25

Temperature of the furnace

□ Start: 2011-01-27 10:20:25 stop: 2011-01-27 10:22:00

Temperature of the steel

☑ Start: 2011-01-27 10:22:00 stop: 2011-01-27 10:23:27

Ambient temerature

□ Start: 2011-01-27 10:23:27

Temperature of the box

Fig. 7.18. A sample view of selected logged files from Group 1

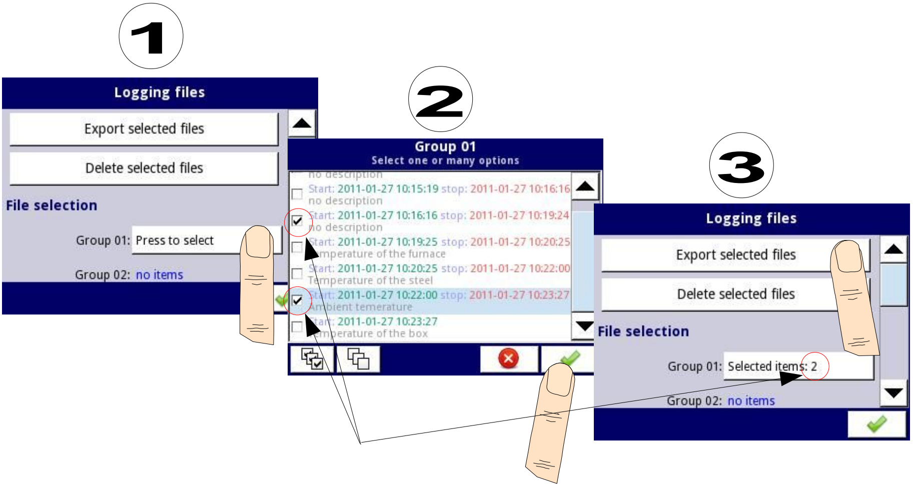
In **Fig. 7.18** presents a sample view of selected logged files from Group 1. The numbers refer to:

- **(1)** - group number,
- **(2)** - the selected logged file,
- **(3)** - no description for the logged file,
- **(4)** - description defined by the user (a description of the logged file is defined in the Groups menu - see **Chapter 7.14.2. Groups - Logging options**
- **(5)** - date and time of the end of the logged data file,
- **(6)** - date and time of the start of the logged data file,
- **(7)** - date and time of the start of the logged data file whose logging has not yet ended.

An example of exporting the logged data to a flash drive

An example of exporting 2 logged files from group 1 is shown in **Fig. 7.19**. First plug the flash drive into the device (when the device is equipped with one of the communication modules).

- In step **(1)**, press the button **Press to select** next to the **Group 01** label,
- In step **(2),** select 2 files by pressing the selected files and then choose the button to accept:
  - 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 step **(3)**, press the **Export selected files** button and wait for a message to end the operations of exporting data to a flash drive,



no description
□ Start: 2011-01-27 10:15:19 stop: 2011-01-27 10:16:16
no description
☑ Start: 2011-01-27 10:16:16 stop: 2011-01-27 10:19:24
no description
□ Start: 2011-01-27 10:19:25 stop: 2011-01-27 10:20:25
Temperature of the furnace
□ Start: 2011-01-27 10:20:25 stop: 2011-01-27 10:22:00
Temperature of the steel
☑ Start: 2011-01-27 10:22:00 stop: 2011-01-27 10:23:27
Ambient temerature
□ Start: 2011-01-27 10:23:27
Temperature of the box

Fig. 7.19. Steps of exporting logged files to flash drive

After exporting logged files a folder is created on the flash drive with the same name as the product identification number, which includes a folder with the selected logged files.

Deleting files from the device is similar export logging files, the difference is that instead of pressing the **Export selected files** button in **(3)** step (see **Fig. 7.19**), press the **Delete** button.

Files management:

The second button on the **File management** menu is the **Configuration files** button. Pressing this button, will open the menu shown in **Fig. 7.20**, which allows the user to load/save the configuration and Modbus templates.

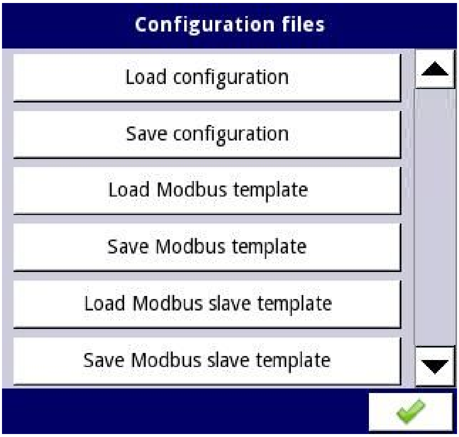


Fig. 7.20. View window when the configuration changes

Load/save configuration will load/save the configuration defined by the user, which includes:

- general settings (see **Chapter 7.7. GENERAL SETTINGS**),
- logical channel settings,
- built-in, external inputs settings,
- built-in, external output settings,
- Modbus protocol settings,
- profile/timer settings,
- control settings,
- network settings,
- group settings,

Load/save Modbus templates allows the user to load/save the configuration of the Modbus MASTER protocol, e.g.:

- name,
- configuration of the device channels (the list of inputs and outputs)
- configuration of register blocks (block list) - see **Chapter 7.15.3. Modbus - MASTER mode**,

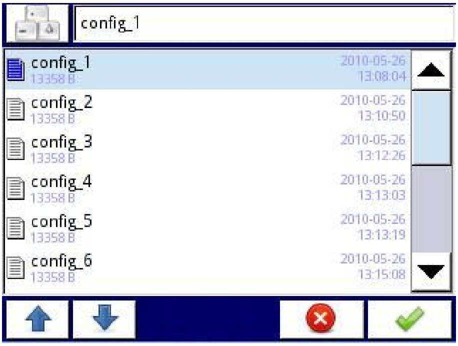
Having saved these Modbus templates means the user can at any time quickly establish a connection between the MultiCon and the SLAVE devices, needing only to choose the appropriate address of the SLAVE devices (more about templates in **Chapter 7.15.3. Modbus - MASTER mode,**).

Load/save Modbus slave templates allows the user to load/save the configuration of the Modbus SLAVE protocol with a defined block of registers, e.g.: register type (Holding/Input), write mode of register, data format (16 bits or 32 bits). For configuration of Modbus SLAVE see **Chapter 7.15.2. Modbus - SLAVE mode**,

The process of exchanging configuration files or Modbus templates between the MultiCon and flash drive starts when you plug the flash drive to the unit. Then enter **MENU -> File management -> Configuration Files**. If you want to **Load configuration/template** the window panel will show a view of the contents of the flash drive with the available files:

- for **configuration file** with extension **.cfg**,
- for **Modbus template file** with extension **.mcfg**,
- for **Modbus slave template file** with extension **.scfg**,

Please note that the file name is defined by the user. If the user wants to **Save configuration/Modbus template**, press the **Save configuration**, **Save Modbus template** or **Save Modbus slave template** button. A window appears with the available files that can be overwritten or a new file can be created by pressing button with keys symbol in the upper left corner (see **Fig. 7.21**). After confirming the write process the data is stored on the flash drive. An example of configuration files in **Fig. 7.21**.



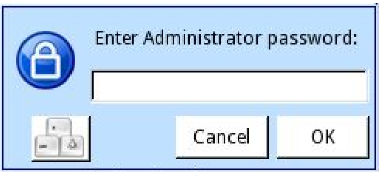
config_1

config_1 13358 B	2010-05-26 13:08:04
config_2 13358 B	2010-05-26 13:10:50
config_3 13358 B	2010-05-26 13:12:26
config_4 13358 B	2010-05-26 13:13:03
config_5 13358 B	2010-05-26 13:13:19
config_6 13358 B	2010-05-26 13:15:08

Fig. 7.21. Example of logging and configuration files

Permissions files:

Third button in the **Files management** menu is **Permission files**. This menu is related with **MultiLevel Access** mode (see **Chapter 7.17.**). Access to that menu has only user logged as Administrator. If no user is logged, the administrator's password must be passed to open that menu (see **Fig. 7.22**).



Enter Administrator password:

Fig. 7.22. Enter password dialogue

In this menu Administrator has access to two buttons: **Load configuration** and **Save configuration** (see **Fig. 7.23**) which allows to store and move saved access settings using external USB flash drive. That streamlines the process of setting up another devices. It is the only way to save **Access options** settings (see **Chapter 7.17.**) , main configuration file doesn't contain those settings.

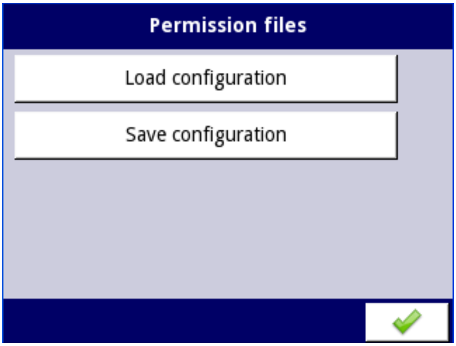
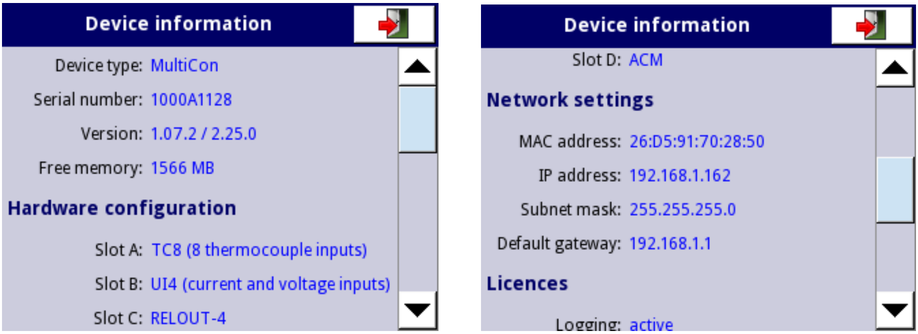


Fig. 7.23. Enter password dialogue

7.4. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION

Fig. 7.24. The button which will show information about the device

The **Device information** menu gives 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.



Device type: MultiCon
Serial number: 1000A1128
Version: 1.07.2 / 2.25.0
Free memory: 1566 MB

Hardware configuration

Slot A: TC8 (8 thermocouple inputs)
Slot B: UI4 (current and voltage inputs)
Slot C: RELOUT-4

Slot D: ACM

Network settings

MAC address: 26:D5:91:70:28:50
IP address: 192.168.1.162
Subnet mask: 255.255.255.0
Default gateway: 192.168.1.1

Licences

Logging: active

Fig. 7.25. Device Information screens

Pressing the **MENU -> Device information** button (see **Fig. 7.24**) will show window (see example window information **Fig. 7.25**) with basic information about software and hardware on the device, such as:

- type of device,
- serial number,
- version of the software
- available free memory,
- hardware configuration - a list of installed modules (**number of slot: module type**)
- network settings,
- active licences.

Enter licence key button (see **Fig. 7.27**) 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 (if the licence key for data logging is entered the text under the **Licences** heading appears as: **Logging**: the period of validity – see **Fig. 7.27**).

Firmware update button (see **Fig. 7.27**) allows the user to update the device software. To perform the update:

- download the latest software version available form the multicon24.eu website ,
- unzip file and copy to a flash drive,
- plug in the flash drive (some high capacity Pendrives are not supported, best choice is pendrive with 2GB capacity) - start the update process by pressing the **Firmware update** button (see **Fig. 7.27**).

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 go 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, depend on the version of the device.

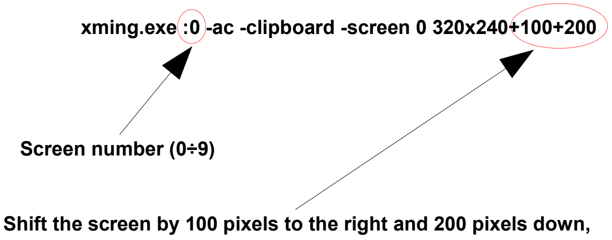
The requirements for removable flash drives are presented in **Chapter 7.3. FILES MANAGEMENT**.

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

This button is available only for devices with installed ACM or ETU module.

Below please find a step by step description of how to configure the **MultiCon** 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 "**Xming**" option and disable "**Launch Xming**" option.
3. After installation of "**Xming**" use the right mouse button to open the properties window for desktop icons of the "**Xming**" and replace the "**-multiwindow"** with "**-ac -screen 0 320x240+100+200"**, then close the window properties. "**Xming**" in cooperation with the **MulitCon** device allows to change the following parameters selected below:



xming.exe :0 -ac -clipboard -screen 0 320x240+100+200

Screen number (0÷9)

Shift the screen by 100 pixels to the right and 200 pixels down,

Fig. 7.26 Xming parameters description

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

4. Next close the window properties and run "**Xming**" using 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 menu and save configuration changes.
6. In the "**Device information**" menu press the "**Remote display**" button and confirm switching to remote display. After a while the device screen will be redirected to your computer screen.

To establish connection between the device and PC, port number 6000 must be unlocked.

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

The **Service options** button (see **Fig. 7.27**) allows the user to delete current configuration and restore default settings (excepting Access options – see **Chapter 7.17.**). It is done by passing the password, which is “**RENEW CONFIGURATION**".

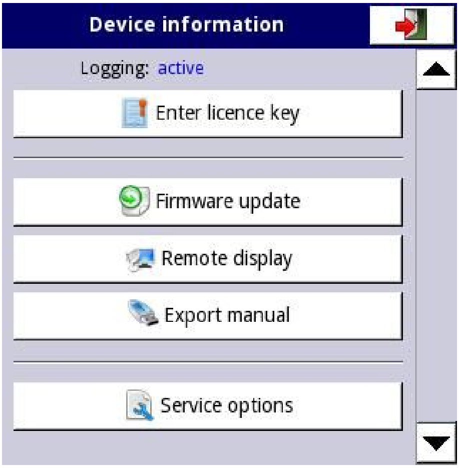


Fig. 7.27. Device information menu

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

7.5. DEVICE CONFIGURATION

Device configuration

Fig. 7.28. The device configuration menu

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

To prevent accidental or unauthorized change the settings in the **Device configuration** menu the user can set the access protection (see **Chapter 7.17. ACCESS OPTIONS**). 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 **Fig. 7.30**).

There is a possibility to press the ➡ button again. In that case warning message will appear (see **Fig. 7.31**), and the user must choose to save new configuration or go to main screen with unsaved but active configuration. If second option is chosen, the icon: 🔧, will be displayed on the information bar, the device will also keep new configuration in memory and user can determine are those changes gave the desired effect (see **Fig. 7.32**). This operation can be repeated any number of times, as long as new configuration is saved or the old one is reverted.

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

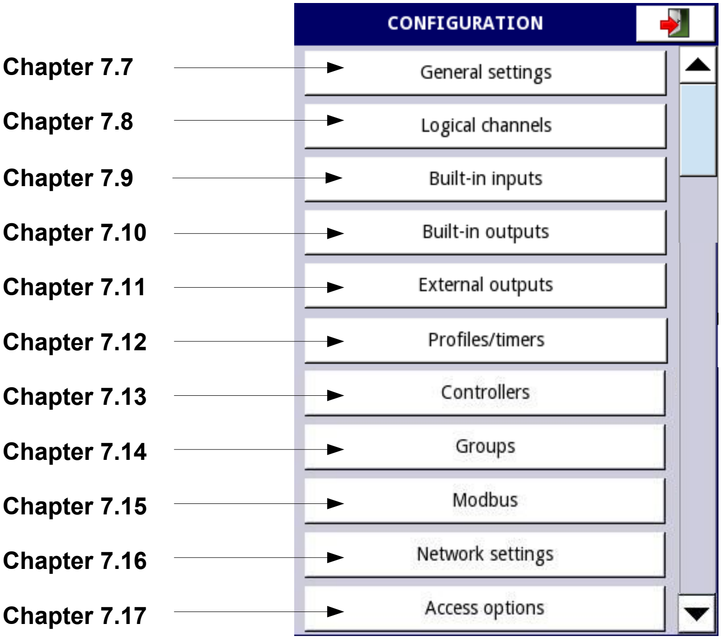


Fig. 7.29. Main menu selection panel

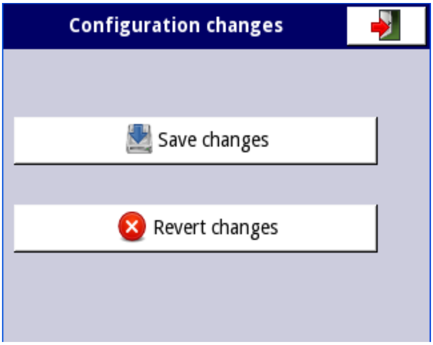
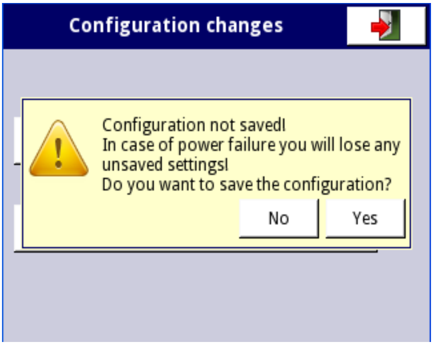
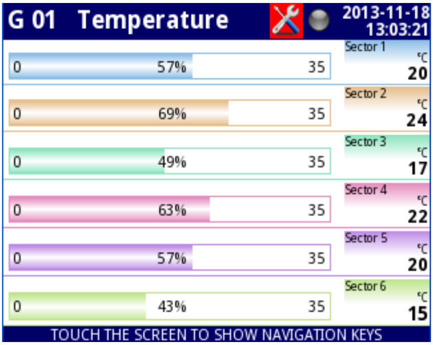


Fig. 7.30. Save / revert changes window



Configuration not saved!
In case of power failure you will lose any unsaved settings!
Do you want to save the configuration?

Fig. 7.31. Warning message about unsaved configuration



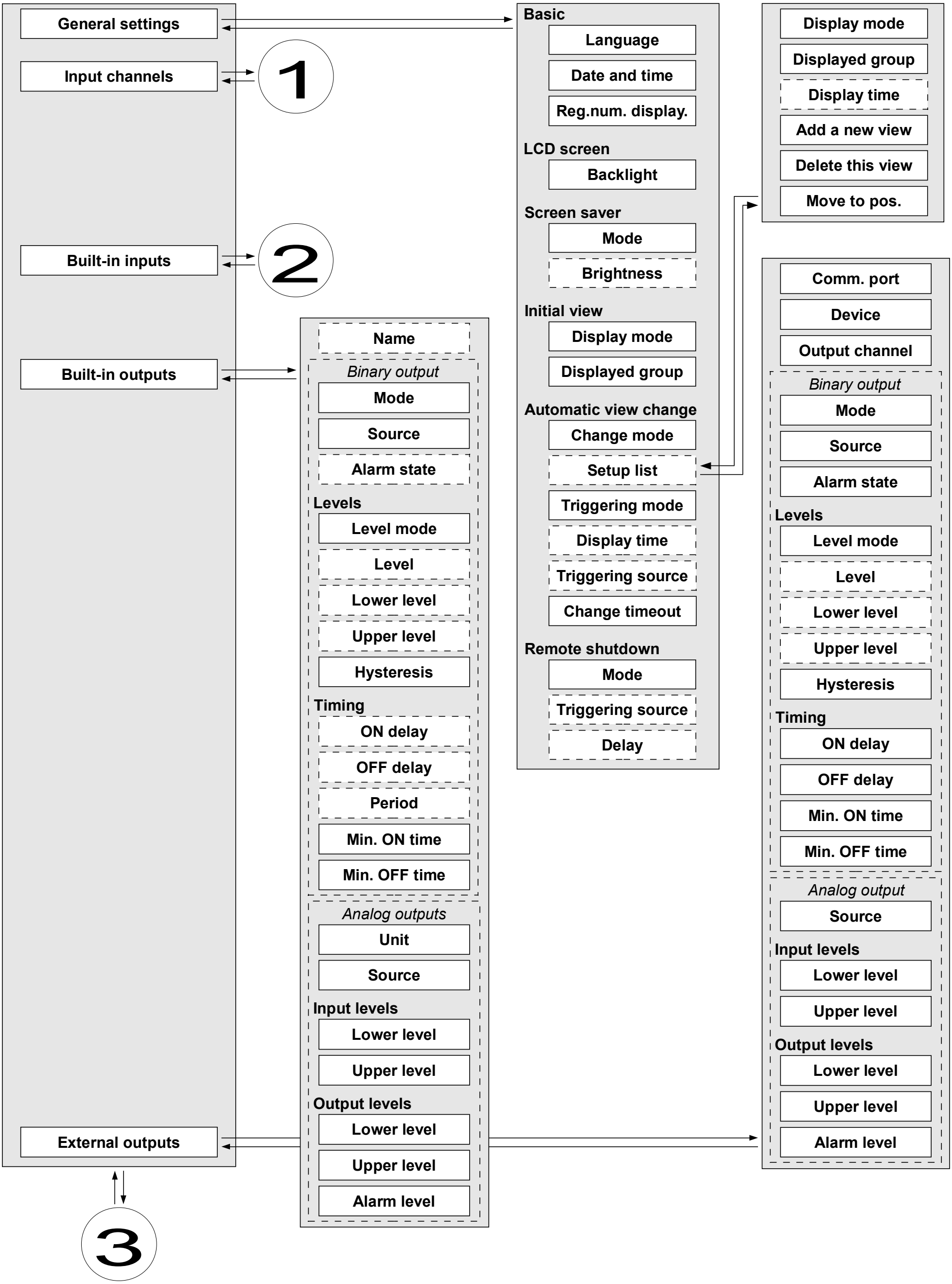
G 01 Temperature 2013-11-18 13:03:21

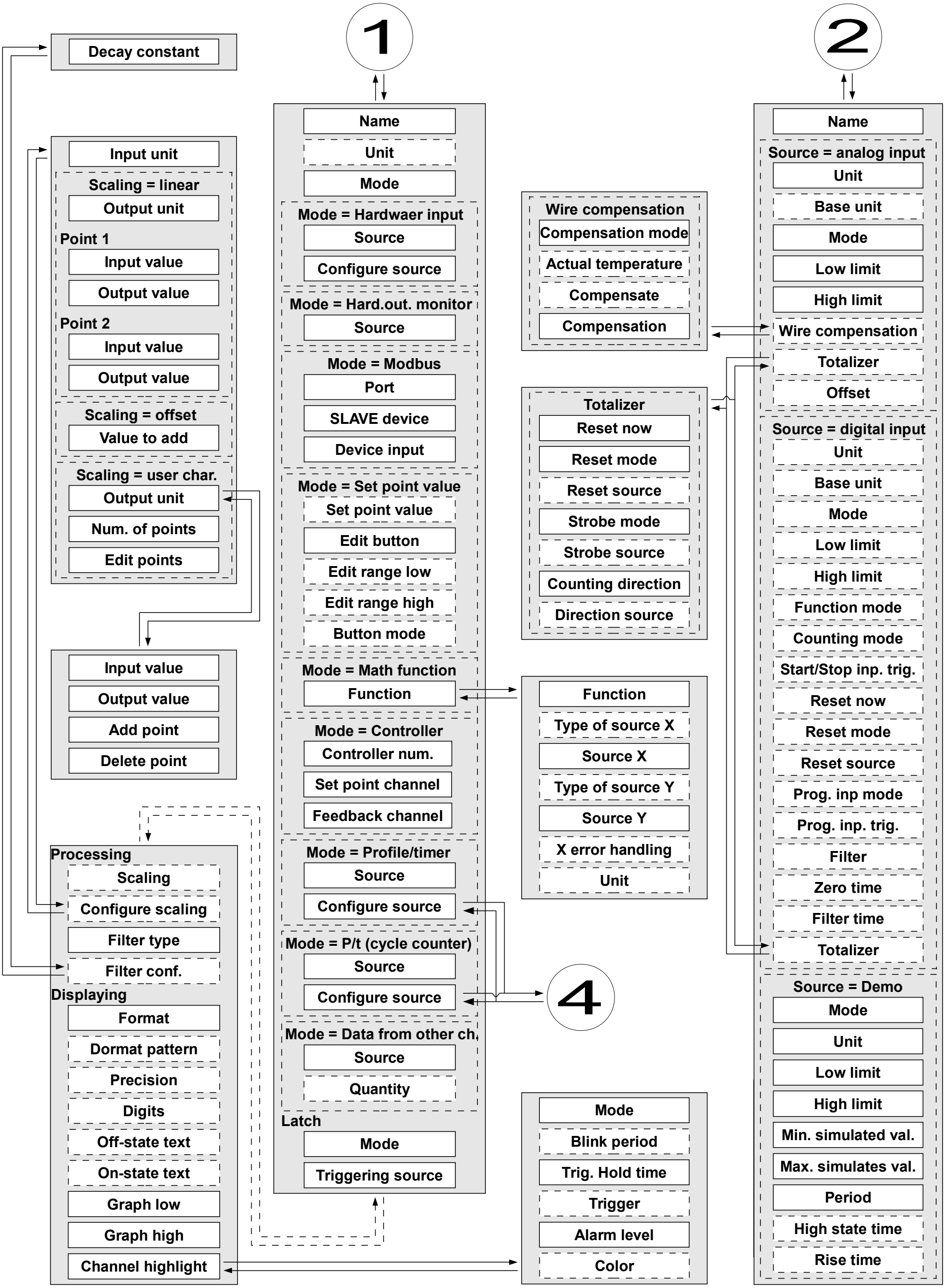
0	57%	35	Sector 1	°C 20
0	69%	35	Sector 2	°C 24
0	49%	35	Sector 3	°C 17
0	63%	35	Sector 4	°C 22
0	57%	35	Sector 5	°C 20
0	43%	35	Sector 6	°C 15

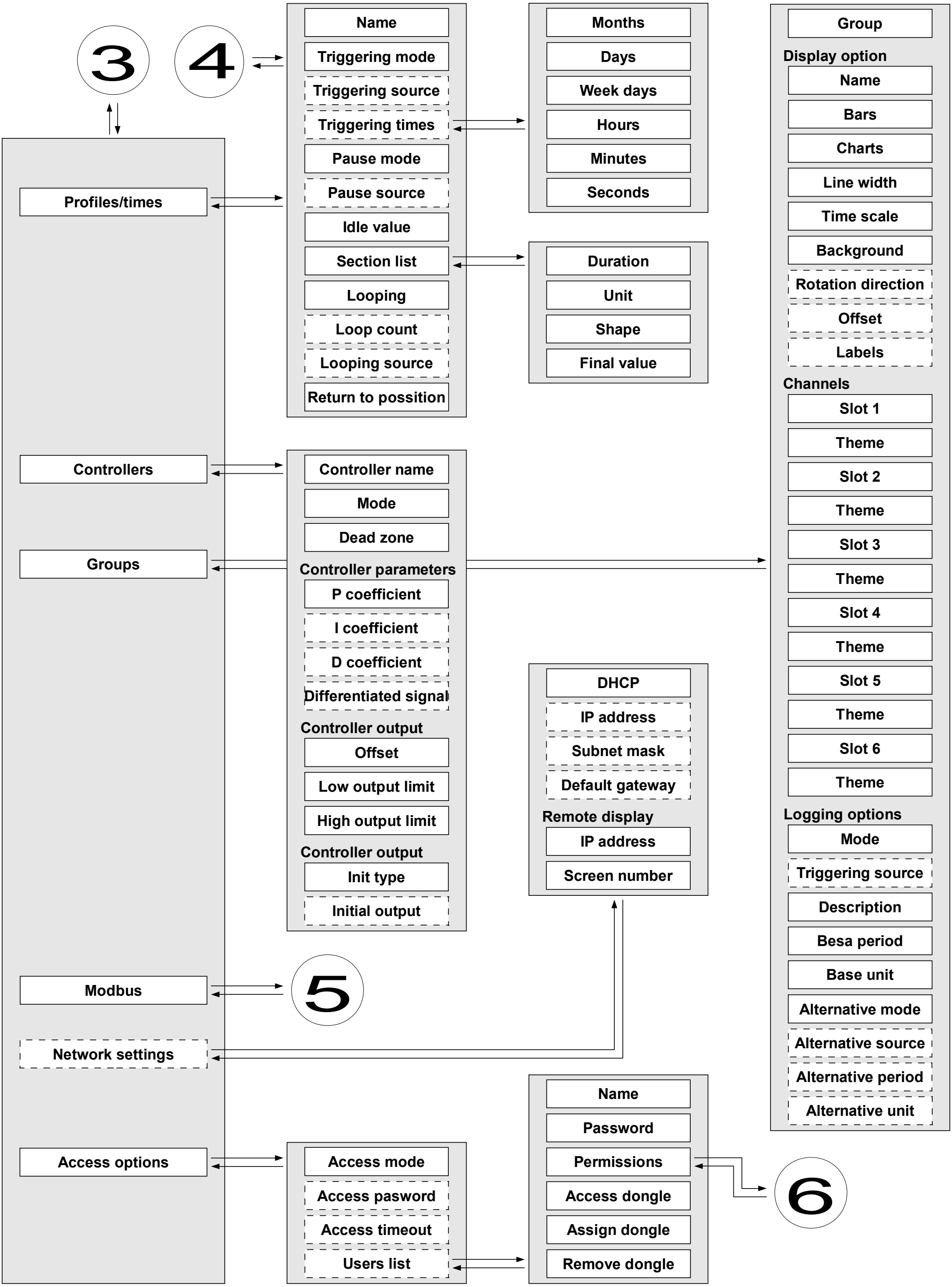
TOUCH THE SCREEN TO SHOW NAVIGATION KEYS

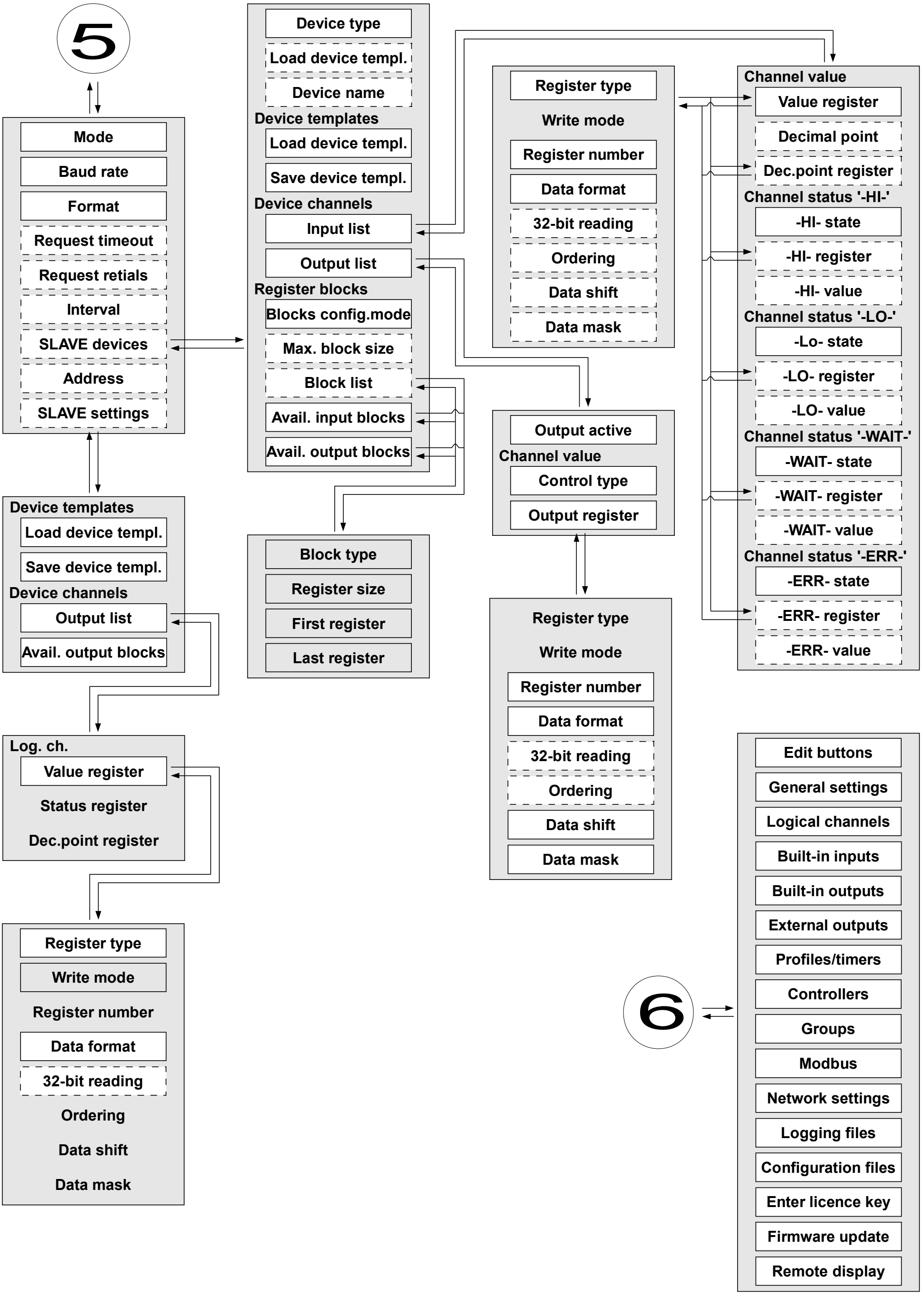
Fig. 7.32. Example of unsaved configuration screen view

7.6. CONFIGURATION MENU STRUCTURE

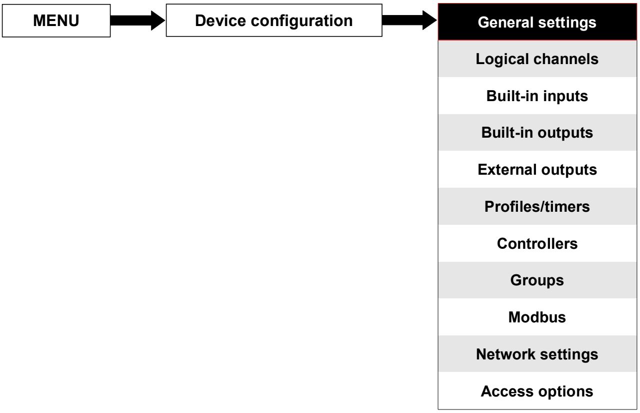



1
2
4
Decay constant
Input unit
Scaling = linear
Output unit
Point 1
Input value
Output value
Point 2
Input value
Output value
Scaling = offset
Value to add
Scaling = user char.
Output unit
Num. of points
Edit points
Input value
Output value
Add point
Delete point
Processing
Scaling
Configure scaling
Filter type
Filter conf.
Displaying
Format
Dormat pattern
Precision
Digits
Off-state text
On-state text
Graph low
Graph high
Channel highlight
Name
Unit
Mode
Mode = Hardwaer input
Source
Configure source
Mode = Hard.out. monitor
Source
Mode = Modbus
Port
SLAVE device
Device input
Mode = Set point value
Set point value
Edit button
Edit range low
Edit range high
Button mode
Mode = Math function
Function
Mode = Controller
Controller num.
Set point channel
Feedback channel
Mode = Profile/timer
Source
Configure source
Mode = P/t (cycle counter)
Source
Configure source
Mode = Data from other ch.
Source
Quantity
Latch
Mode
Triggering source
Wire compensation
Compensation mode
Actual temperature
Compensate
Compensation
Totalizer
Reset now
Reset mode
Reset source
Strobe mode
Strobe source
Counting direction
Direction source
Function
Type of source X
Source X
Type of source Y
Source Y
X error handling
Unit
Mode
Blink period
Trig. Hold time
Trigger
Alarm level
Color
Name
Source = analog input
Unit
Base unit
Mode
Low limit
High limit
Wire compensation
Totalizer
Offset
Source = digital input
Unit
Base unit
Mode
Low limit
High limit
Function mode
Counting mode
Start/Stop inp. trig.
Reset now
Reset mode
Reset source
Prog. inp mode
Prog. inp. trig.
Filter
Zero time
Filter time
Totalizer
Source = Demo
Mode
Unit
Low limit
High limit
Min. simulated val.
Max. simulates val.
Period
High state time
Rise time


3
4
5
6
Profiles/times
Controllers
Groups
Modbus
Network settings
Access options
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
Controller name
Mode
Dead zone
Controller parameters
P coefficient
I coefficient
D coefficient
Differentiated signal
Controller output
Offset
Low output limit
High output limit
Controller output
Init type
Initial output
DHCP
IP address
Subnet mask
Default gateway
Remote display
IP address
Screen number
Group
Display option
Name
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
Logging options
Mode
Triggering source
Description
Besa period
Base unit
Alternative mode
Alternative source
Alternative period
Alternative unit
Access mode
Access pasword
Access timeout
Users list
Name
Password
Permissions
Access dongle
Assign dongle
Remove dongle


5
Mode
Baud rate
Format
Request timeout
Request retials
Interval
SLAVE devices
Address
SLAVE settings
Device type
Load device templ.
Device name
Device templates
Load device templ.
Save device templ.
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
Load device templ.
Save device templ.
Device channels
Output list
Avail. output blocks
Block type
Register size
First register
Last register
Output active
Channel value
Control type
Output register
Register type
Write mode
Register number
Data format
32-bit reading
Ordering
Data shift
Data mask
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
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
Enter licence key
Firmware update
Remote display

7.7. GENERAL SETTINGS



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:

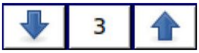
- **Basic** parameter block – this block includes two parameters:
  - **Language** - this parameter allows the user to select the language, available languages are: English, Polish, Spanish, German, Russian, French, Czech, Romanian and Hungarian.
  - **Date and time** - this parameter allows the user to set the current date and time,
  - **Reg. num. displaying** - this parameter allows to change mode of displaying the Register addresses for **Logical channel** and **External output** menu. Two formats are available:
    - **hexadecimal** - the registers are displayed in 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 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%** (least backlight), **40%**, **60%**, **80%**, **100%** (the most backlight),
- **Screen saver** parameter block - these parameters can reduce 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 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%**,

- **Initial view** parameter block - allows the user to set the initial display screen on the LCD screen when the device is turned on, this block includes two parameters:
    - **Display mode** - select the presentation of data in the displayed group (see parameter: **Displayed group**). For possible modes see **Chapter 6.3.3. Data panels**, and **Chapter 7.14. GROUPS**,
    - **Displayed group** - selects a group displayed at startup. If **Display mode** is chosen as **many groups**, parameter **Displayed group** selects 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 **Display group** as **Group 8** then the unit will display: starting from the left side of: Group 8 -> Group 9 -> Group 10 -> Group 1 -> Group 2,
- **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 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 displayed view at fixed time intervals,
        - **from log. channel** – changes displayed view depending on value in selected logical channel. If value in this channel is less than 1, displayed view number is 1, if value in this parameter equal or more than 1 but less than 2 displayed view number is 2, and so on. The last defined view does not has 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 logical channel from the list, which value will sets up the displaying view,
    - **Change timeout** – this parameter determines the time from the device startup or manually change of the view to first automatic view change,
    - **Setup list** – this button is visible for **Change mode** parameter setted as **detailed list**, this parameter is described below in this **Chapter**,
- **Remote shutdown** parameter block – allows to set automatic shutdown of the device,
    - **Mode** – turns on and off shutdown from logical channel,
        - **disable** – remote shutdown is off,
        - **from log. channel** – remote shutdown is on,
    - **Triggering source** – contains the logical channels list, where selected one will be source for the remote shutdown; the shutdown procedure will start each time when signal in selected channel will change from value ≤ **0** to value **> 0**,
    - **Delay** – settable time that will pass before the device will shutdown; this procedure will be stopped if during this time:
        - signal in **Triggering source** parameter will drop to ≤ **0** value,
        - the user will press **cancel** button on the screen,

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

Setup list parameter

After pressing the **Setup list** button the user enters the **View** menu allowing the creation of **1** to **20** views.



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 7.14. GROUPS**,
- **Displayed group** - allows the user to select a group displayed at startup. If the **Display mode** is chosen as **many groups**, parameter **Display group** selects the first of five following groups displayed on the screen (mode many group 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 selected view from the list,
- **Move to position** - this parameter allows the user to move the current view to the appropriate position,

Example:

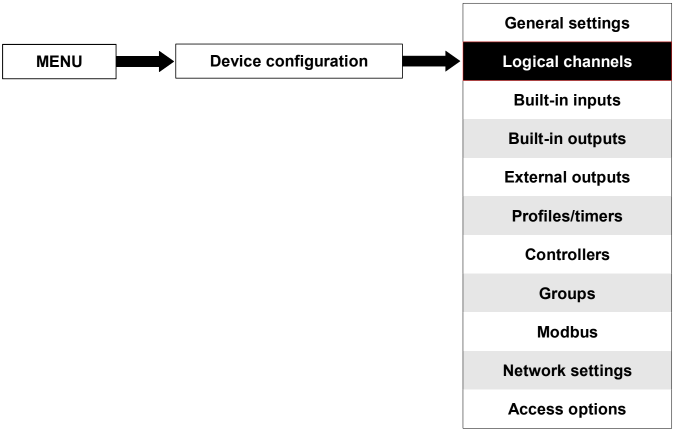
Steps to create four views are as follows:

1. In the Change mode parameter select detailed list,
2. Press the **Setup list** button and enter the **View** menu,
3. Set parameters for a first View,

Arrows placed in the upper right corner of the screen allow you to move to the next view. The middle button allows 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, user can choose two 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 new view on 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,

7.8. LOGICAL CHANNELS



The **Logical channels** menu is used to configure the logical channels. 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 directly into the configuration menu of particular channel, press and hold on the screen over the channel data panel for a few seconds (see option **(1)** in the **Fig. 6.5**). If the password is set (see **Chapter 7.17. ACCESS OPTIONS**) then the user has to enter the password before entering the configuration.

7.8.1. Logical Channels - General settings

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 directly select a specific logical channel from the list.

The parameters of a logical channel depends on the **Mode** of the logical channel. The **Logical channel** has modes:

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

A detailed description of the models is 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.

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 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 logical channel. It is possible to select one of ten modes:
  - **disabled**,
  - **Hardware input** - see Chapter **7.8.2**,
  - **Hardware output monitor** - see Chapter **7.8.3**,
  - **Modbus** - see Chapter **7.8.4**,
  - **Set point value** - see Chapter **7.8.5**,
  - **Math function** - see Chapter **7.8.6**,
  - **Controller** - see Chapter **7.8.7**,
  - **Profile / timer** - see Chapter **7.8.8**,
  - **Profile/timer (cycle counter)** – see Chapter **7.8.9**,
  - **Data from other channel** – see Chapter **7.8.10**,

- **Latch** parameter block - allows 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 of the value of channel selected in the **Triggering source** parameter,
  - **Triggering source** - this parameter is only visible if 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 triggering channel is **≤0** the latch is **active**, for a value **>0** latch is **disabled**),

During device restart, the logical channels, which have the latch function enabled have 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**,

To enter the scaling menu press the button next to the **Scaling** label. The menu has the following options:

a) **disabled** – no scaling of input data,

b) **linear** – after choosing this option, parameter appears:

- **Configure scaling** – submenu which allows the user to change unit in this logical channel and rescale value using two points. This submenu contains following options:
  - **Input unit** – read only parameter, it displays unit of the logical channel source,
  - **Output unit** – unit of rescaled data,
  - **Point 1** block contains following options:
    - **Input value** – value of point 1 before scaling,
    - **Output value** – value of point 1 after scaling,
  - **Point 2** block contains following options:
    - **Input value** – value of point 2 before scaling,
    - **Output value** – value of point 2 after scaling,

In MultiCon, linear scaling can be done in with table placed below, but slope and offset are constant values and they should be determined by user earlier.

	Input value	Output value
Point 1	x_1	y_1=slope*x_1+offset
Point 2	x_2	y_2=slope*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 modules are scaled, to degrees Fahrenheit. The formula, which should be used for this task is shown bellow:

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

where:

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

Lets 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 Configure scaling submenu:
  - in **Output unit** parameter write: **˚F**,
  - in **Input value** parameter for **Point 1** block write: **-50** (which is low range limit),
  - in **Output value** parameter for **Point 1** block write: **-58** (according to the formula),
  - in **Input value** parameter for **Point 2** block write: **85** (which is low range limit),
  - in **Output value** parameter for **Point 2** block write: **185** (according to the formula),

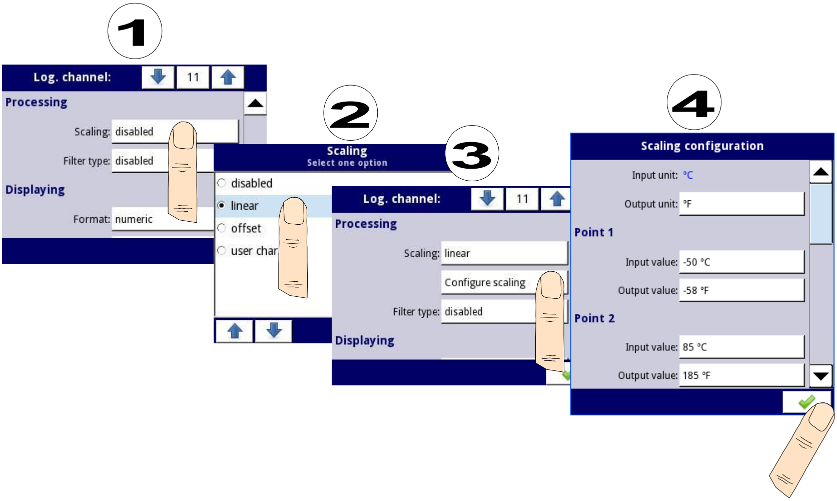


Fig. 7.33. Example of scaling configuration

c) **offset** – after choosing this option, parameter appears:

- **Configure scaling** – function, which allows to add a constant value to input value. It is given by formula:

$$y = x + offset$$

In this function, there are following parameters:

- **Input unit** – read only parameter, it displays unit of the logical channel source,
- **Value to add** – it defines offset value,

d) **user char.** - is defined as set of X-Y points. Number of the points is variable and may be set from 2 to 20 points which make linear segments (see **Fig. 7.34**). After choosing this option, parameter appears:

- **Configure scaling** – submenu which allows to define further points of the user characteristic. There are following parameters:
  - **Input unit** – read only parameter, it displays unit of the logical channel source,
  - **Output unit** – unit of rescaled data,

- **Num. of point** – read only parameter, in displays number of points that currently are defined in characteristic,
- **Edit points** – submenu that defines point in characteristic:
  - **Input value** – point value before scaling,
  - **Output value** – point value after scaling,
  - **Add point** button – adds new point, behind currently displayed,
  - **Delete point** button – deletes currently displayed point, parameter is inactive when number of points in characteristic is **2**,

The arrows placed in the upper right corner of the screen allows to switch between points. The middle button allow 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 input value **'x'** the user obtain an output value **'y'** which is described by the relationship:

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

,where **'slope'** and **'offset'** are coefficients of a segment contained between two points (see **Fig. 7.34**), 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.

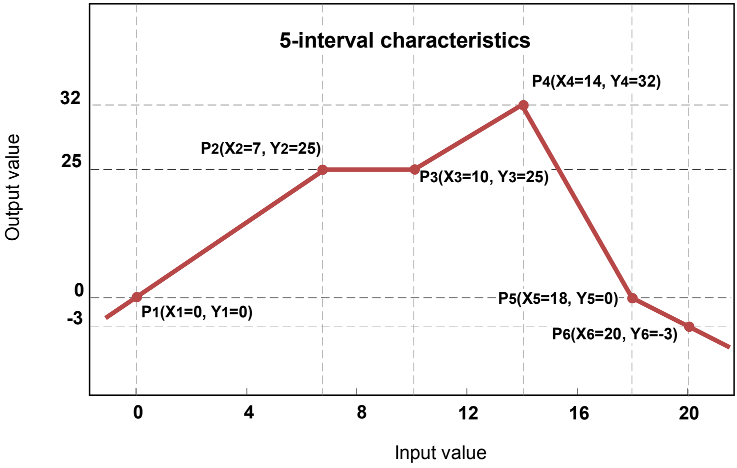
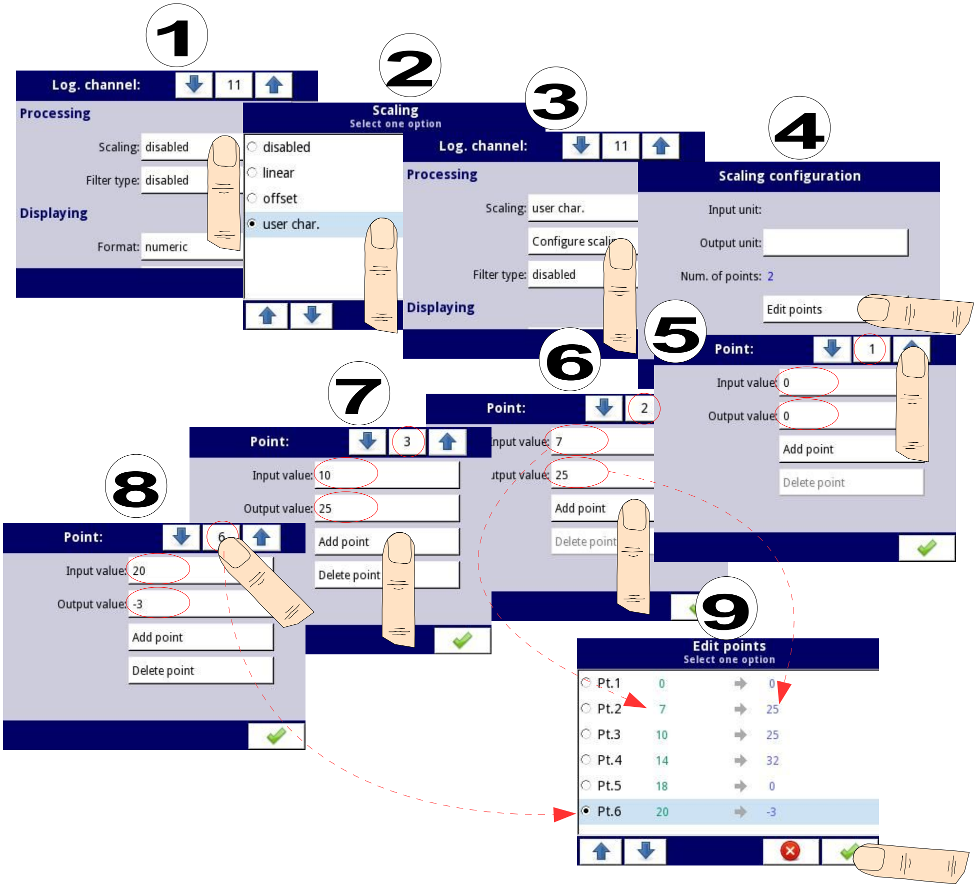


Fig. 7.34. Example of user characteristic

Example:

Steps to create a user characteristic consisting of 6 points (like on **Fig. 7.34**):

1. Press the button next to the **Scaling** label and select **user characteristic** option (point **(1)** and **(2)** in **Fig. 7.35**).
2. Press the **Configure scaling** button and enter the **Scaling configuration** menu (point **(3)** in **Fig. 7.35**).
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 **Output Unit** label.
4. Press the **Edit points** button (point **(4)** in **Fig. 7.35**) and go to **Edit points** menu.



1: Log. channel: 11; Scaling: disabled; Filter type: disabled; Format: numeric

3: Log. channel: 11; Scaling: user char.; Filter type: disabled

4: Input unit: ; Output unit: ; Num. of points: 2

5: Point: 1; Input value: 0; Output value: 0

6: Point: 2; Input value: 7; Output value: 25

7: Point: 3; Input value: 10; Output value: 25

8: Point: 6; Input value: 20; Output value: -3

9:

○ Pt.1	0	➜	0
○ Pt.2	7	➜	25
○ Pt.3	10	➜	25
○ Pt.4	14	➜	32
○ Pt.5	18	➜	0
● Pt.6	20	➜	-3

Fig. 7.35. Configuring the user characteristic

For **Point 1** set input and output value (point **(5)** in **Fig. 7.35**).

5. Switch to **Point 2** by using the arrow keys and there also set the value of input and output (point **(6)** in **Fig. 7.35**).
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 press 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 **Fig. 7.35**).

Filter type

The **Filter type** parameter has options:

– **disabled** - filtering of the input value is turned off,

– **smoothing** - this option enables a filter that is expressed by the formula:

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

,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 **smoothing** option in **Filter type** parameter new button is available - **Filter configuration** which allows the user to enter a time constant with the **Decay constant** parameter (see above filter formula).

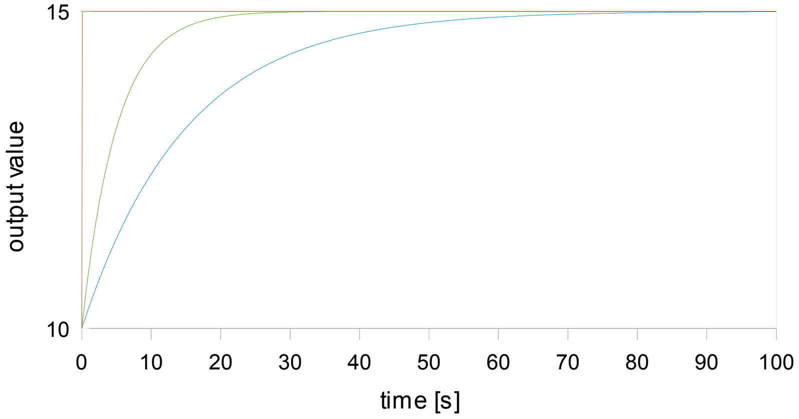


Fig. 7.36. Example of the input signal (red chart), filtered signal for the time constant = 5s (green chart) and time constant = 15s (blue chart)

– **peak detect** – function which allows to detect and visualize peak values in measured signal. Detection of the peak value can be done if the measured signal raises and drops (or the opposite) by value, which is settable by the user. Next, the detected value can be hold for settable by user period of time. If during the time when peak is detected, new peak is detected, then the peak value is updated and the time of hold is reset. When this time will elapse, or no peak is detected, the device will display current value of the signal (see **Fig. 7.37**).

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

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

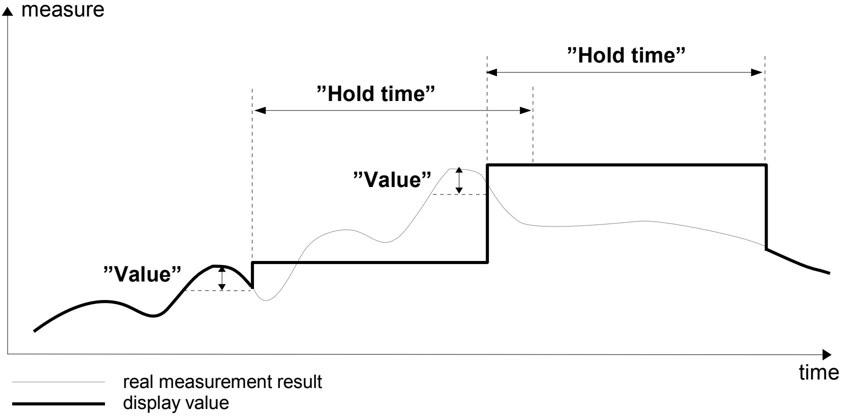


Fig. 7.37. Process of peak detection

Displaying parameter block

The constant parameters of **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 low state and **'1'** for high state,
  - **time** – value in this channel is displayed as a time. Logical channel in time Format will always considers its value as a number of seconds and base on this and settings in **Time pattern** parameter, the logical channel will calculate the value to display,
- **Time pattern** – this parameter occurs only when **Format** parameter is set as **time**; it allows user to define time pattern at his will. Only restrictions are that higher units have to be before lower units and different units have to be separated by colon character. Allowed is to create units which contains any number of digits. When user creates a pattern which is forbidden, the device will detect it and set the pattern as default.
- **Precision** – this parameter occurs when **Format** parameter is set as **numeric** or **time**, which specifies the precision to be displayed on the output value (see **Fig. 7.39**),
- **Off-state text** – this parameter occurs when **Format** parameter is set as **binary**, when the input value is **≤ 0** the value is replaced by the text defined by the user, default settings text is: **OFF**,
- **On-state text** - this parameter occurs when **Format** parameter is set as **binary**, when the input value is **>0** the value is replaced by the text defined by the user, default settings text is: **ON**,

  The text of **Off-state** and **On-state** can be:
  - text with 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 **Format** parameter is set as **numeric**, allowing user to choose the part of the number that is to be the value of the logical channel displayed on the screen,

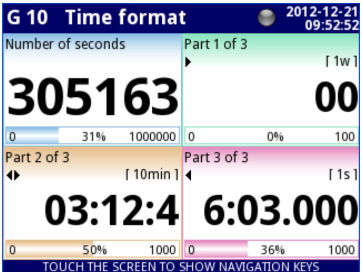
Example of a number representing the value of the logical channel:

<table>
<tr><td>1</td><td>2</td><td>5</td><td>6</td><td>2</td><td>8</td><td>0</td><td>2</td><td>2</td><td>1</td><td>5</td><td>8</td><td>6</td><td>4</td><td>.</td><td>8</td><td>7</td><td>3</td></tr>
<tr><td colspan="6"><b>part 1 of 3</b></td><td colspan="6"><b>part 2 of 3</b></td><td colspan="6"><b>part 3 of 3</b></td></tr>
<tr><td colspan="6"></td><td colspan="6"><b>part 1 of 2</b></td><td colspan="6"><b>part 2 of 2</b></td></tr>
<tr><td colspan="10"></td><td colspan="8"><b>all digits</b> (limited ability to display up to 5 digits plus decimal point on the screen)</td></tr>
</table>

– Available options for **Digits** parameter are:
  - **all digits** – display all parts. With this option the number is limited only to 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,

For time Format in logical channel only 6 digits, three special signs (colons, dot), but not more than eight signs and minus. Using Digits and Precision parameters, the device will display time from youngest unit with maintaining all restrictions, see **Fig. 7.38**.

Minus signs is displayed only before the highest unit of the **Time Pattern**. When the highest unit can not be displayed in logical channel, minus sign will not appear and value in this channel will be considered as a positive value.



G 10 Time format 2012-12-21 09:52:52

Number of seconds
305163
0 31% 1000000

Part 1 of 3
[1w]
00
0 0% 100

Part 2 of 3
[10min]
03:12:4
0 50% 1000

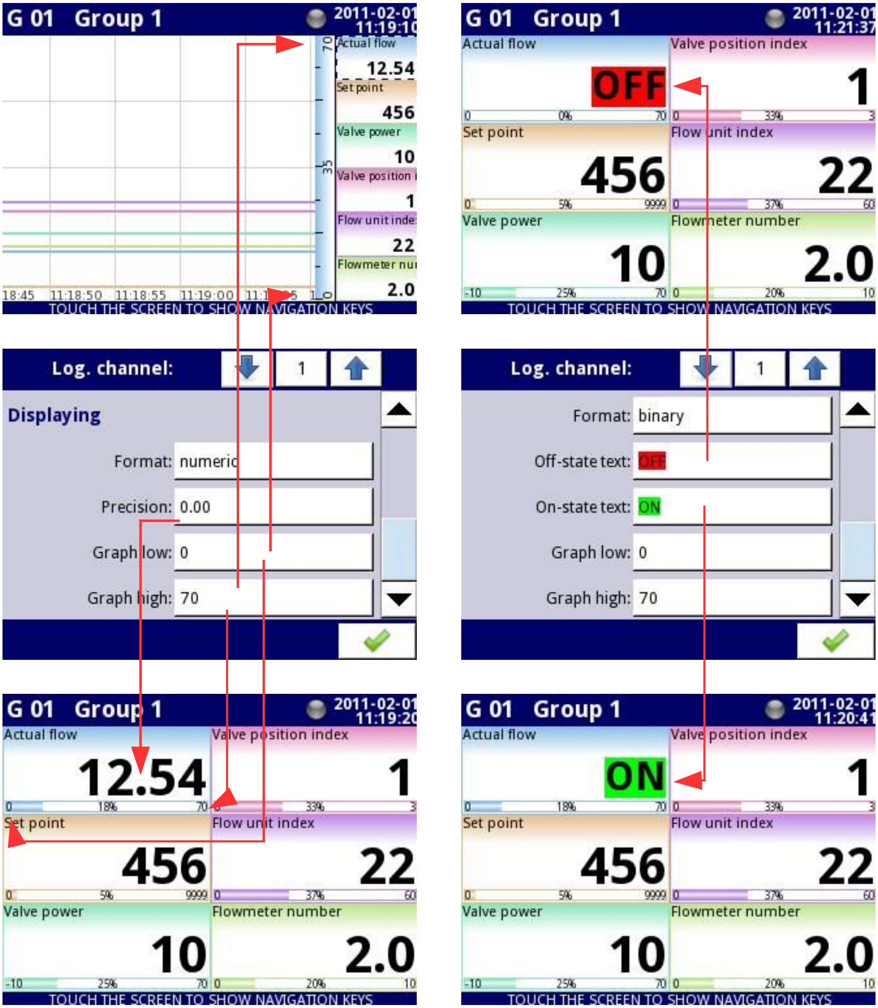
Part 3 of 3
[1s]
6:03.000
0 36% 1000

TOUCH THE SCREEN TO SHOW NAVIGATION KEYS

*Fig. 7.38.Example of **time Format** view with **Precision 0.000** and standard **Time Pattern** [ww:dd:hh:mm:ss]*

– **Graph** low - minimum range value for graphs, bars, needle dials and percentage bars (see **Fig. 7.39**),

– **Graph high** - maximum range value for graphs, bars, needle dials and percentage bars (see **Fig. 7.39**),



Log. channel: 1

Displaying

Format: numeric

Precision: 0.00

Graph low: 0

Graph high: 70

Log. channel: 1

Format: binary

Off-state text: OFF

On-state text: ON

Graph low: 0

Graph high: 70

Fig. 7.39. Input channels menu – 2 different kinds of Displaying parameters

To calculate percentage which will be displayed on the indicator, the device is using 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** – block of parameters, used to change background color and its mode in the logical channel. User can set up to three different variants of highlights, depending to event weight, which should trigger highlight in this channel.

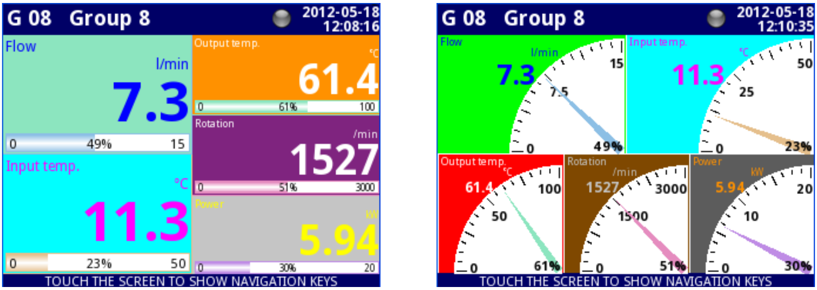


Fig. 7.40 Example of logical channel highlights

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

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

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

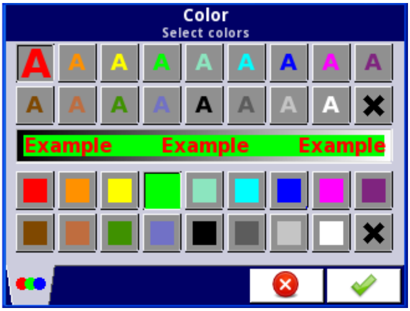


Fig. 7.41 Color edit dialog

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%.
- User should pay attention on the highlight hierarchy. If there are two or three variants of highlight generated at once, 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 7.14. GROUPS**).

Examples of Channel highlight configuration.

Example 1:

Task:

User's task is to set blinking highlight on alarm channel, which will be active when measured current value is too high or too low.

Solution:

After connecting all measuring devices and configuring them in MultiCon, 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, when is 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 on: **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 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 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, when is time to activate the alarm:

- **Mode** parameter set as: **outside range**,
- **Source** parameter set as logical channel named as **Current value**,
- **Alarm state** parameter set as: **immed. ON**,
- **Level mode** parameter set as: **channel**,
- **Lower level** parameter set as logical channel named as **Overhead alarm threshold**,
- **Upper level** parameter set as logical channel named as **Bottom alarm threshold**,
- All other parameters should be set as default,

Example 2:

Task:

User's task is to set blinking highlight on temperature channel, which will be activated on three different levels.

Solution:

After connecting all measuring devices and configuring them in MultiCon, 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 that will tell when will first alarm appear.
e) Channel that will tell when will second alarm appear.
f) Channel that will tell when will third alarm 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,

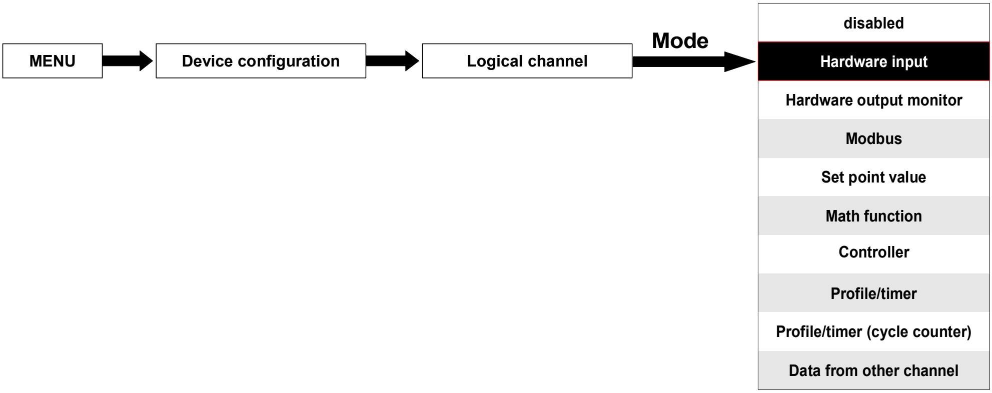
Logical channel 7: Temperature measurement:

- 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 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 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 screen) and background color (at the bottom of the screen),

– All other parameters should be set as default,

7.8.2. Logical channels - Hardware input mode



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

The **Logical channels** parameters in **Hardware inpu**t mode are:

- **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 **Processing** parameter block,
- **Mode=Hardware input** - in this parameter user can select the source type of the data for the logical channel,
- **Source** - in this parameter 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 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 7.9. BUILT-IN INPUTS**),
- **Latch** parameter block - allows user to set the **latch function** which will hold the last value of a channel (discussed in **Chapter 7.8.1. Logical Channels - General settings**),
- **Processing** parameter block - is used for scaling and filtering data (discussed in **Chapter 7.8.1. 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 7.8.1. Logical Channels - General settings**),

<u>**Source** parameter in **Hardware input** mode</u>

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 input module **I16** - 16 current inputs (see **Appendix 8.2. UI4, UI8, UI12, U16, U24, I16, I24 – VOLTAGE and CURRENT MEASUREMENT MODULES**) is shown in **Fig. 7.42**.

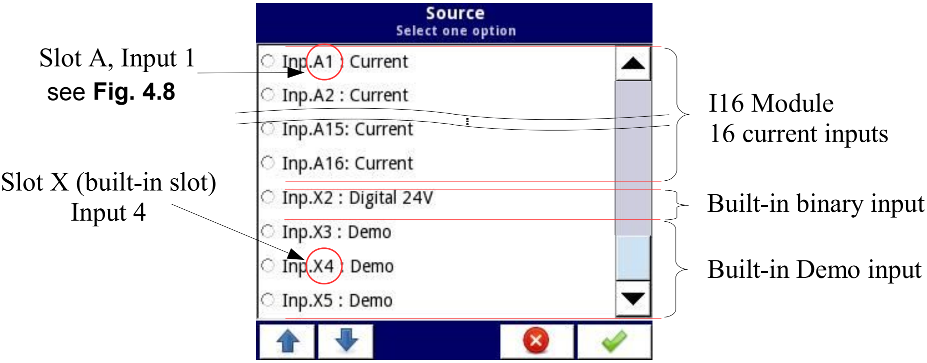


Fig. 7.42. The view of a sample list of available hardware inputs for a device

In the device there are two methods to change the source configuration of the **Built-in inputs**:

- using **Configure source** button in **Logical channel** menu in **Hardware input** mode,
- using the **Built-in inputs** menu,

The **Source** for **Hardware input** mode can be (in the same order as list in the device - see **Fig. 7.42**):

a) installed **input modules** in the appropriate slots A, B or C (see **Fig. 4.8**) - the list of currently available modules is on the website,

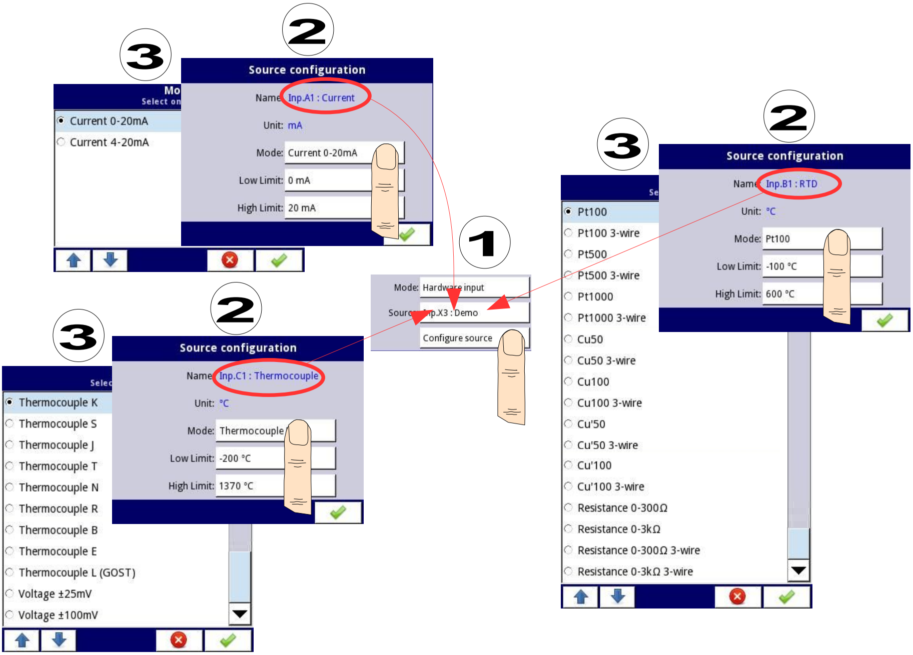
Description of input modules parameters is shown in **Chapter 7.9.2. Built-in inputs - Input modules**.

The following steps change the **Source configuration** for the sample of 3 modules shown in **Fig. 7.43**:

– Step **(1)** - selection of **Source** for channel in **Hardware input** mode, for example: **Inp.A1:Current**, next press the **Configure source** button to enter **Source configuration** menu,

– Step **(2)** - press the **Mode** button to change range of the current input,

– Step **(3)** - choose from the list of available options for signal range - for example: **Current 0-20mA** (for current module),



Source configuration

Name: Inp.A1 : Current

Unit: mA

Mode: Current 0-20mA

Low Limit: 0 mA

High Limit: 20 mA

Mode: Hardware input

Source: Inp.X3 : Demo

Source configuration

Name: Inp.C1 : Thermocouple

Unit: °C

Mode: Thermocouple

Low Limit: -200 °C

High Limit: 1370 °C

Source configuration

Name: Inp.B1 : RTD

Unit: °C

Mode: Pt100

Low Limit: -100 °C

High Limit: 600 °C

Fig. 7.43. Change source configuration for different types of modules

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

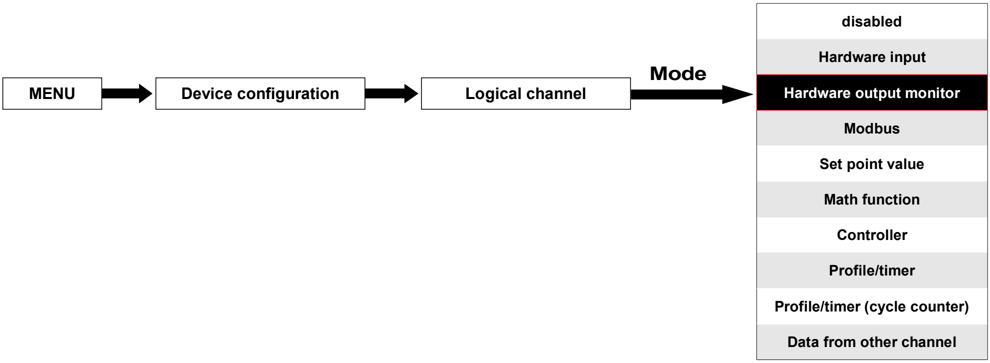
The device has a built-in digital input, which can be used, for example as a switch for a process. Descriptions of parameters is shown in **Chapter 7.9.3. Built-in inputs - Binary input Inp.X2 : Digital 24V**.

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. Descriptions of parameters is presented in **Chapter 7.9.4. Built-in inputs - Demo input numbered X3, X4, X5**.

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

7.8.3. Logical Channels - Hardware output monitor mode



This mode allows the user to display data from built-in output modules, processed in any other logical channels (e.g. by mathematical function or virtual relay) or it can be the data source for controlling another output.

A view of the configuration of a **Logical channel** in **Hardware output monitor** mode is shown in **Fig. 7.44**.

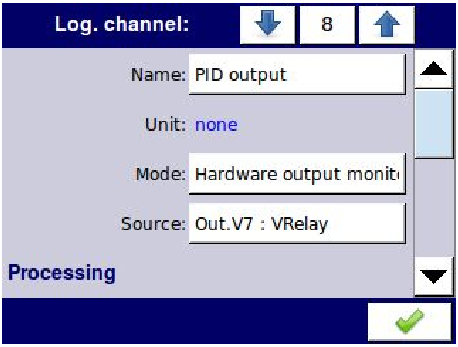
The parameters of a **Logical channel** in **Hardware output monitor** mode are:

- **Name** - it gives the name to the logical channel,
- **Unit** - for **Built-in modules** it will automatically default to the unit of the module. To change the unit use the **Scaling** parameter in **Processing** parameter block,
- **Mode=Hardware output monitor** - in this parameter user selects the source of data for logical channel,
- **Source** - in this parameter 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 7.8.1. Logical Channels - General settings**),

- **Processing** parameter block - is used for scaling and filtering data (discussed in **Chapter 7.8.1. 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 7.8.1. Logical Channels - General settings**),



Log. channel: 8

Name: PID output

Unit: none

Mode: Hardware output monit

Source: Out.V7 : VRelay

Processing

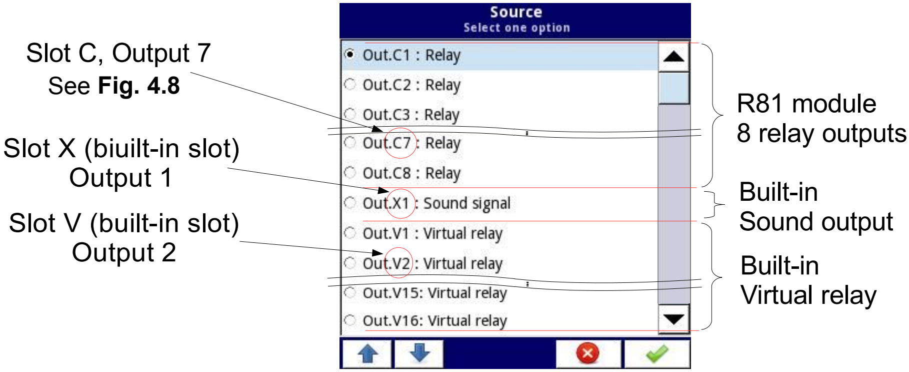
*Fig. 7.44. Input channels menu – parameters specific for **Hardware Output Monitor** mode*

<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 module such as **R81** - 8 relay outputs module (see **Appendix 8.13. R45, R81, R65, R121 - RELAY MODULES**) is shown in **Fig. 7.45**.

Source for the **Hardware output monitor** mode can be (in the same order as in the list in device - see **Fig. 7.45**):

- installed hardware output modules in the respective slots A, B or C (see **Fig. 4.8**) -a list of modules currently available is on the website; more about the output modules in **Chapter 7.10. BUILT-IN OUTPUTS**,
- built-in **Sound signal** output is always marked as **Out.X1: Sound signal** - more about the **Sound** outputs is in **Chapter 7.10. BUILT-IN OUTPUTS**,
- built-in Virtual relays marked as Out. V1 - V16 - more about Virtual relay in Chapter **7.10. BUILT-IN OUTPUTS**.

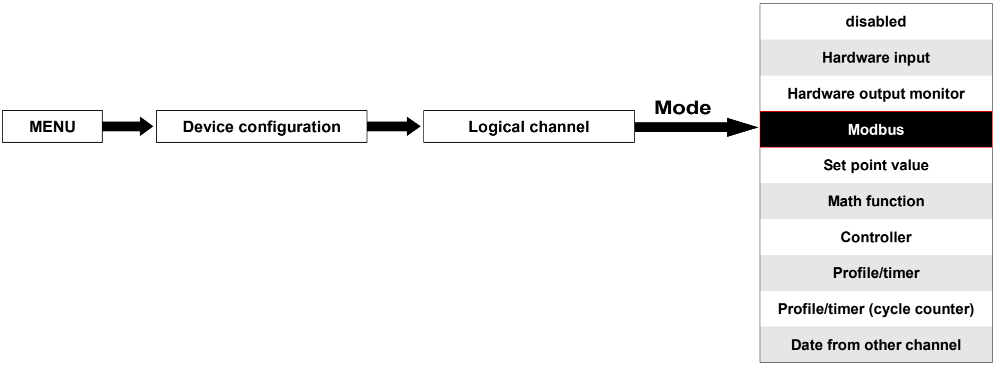


Source
Select one option

● Out.C1 : Relay
○ Out.C2 : Relay
○ Out.C3 : Relay
○ Out.C7 : Relay
○ Out.C8 : Relay
○ Out.X1 : Sound signal
○ Out.V1 : Virtual relay
○ Out.V2 : Virtual relay
○ Out.V15: Virtual relay
○ Out.V16: Virtual relay

Fig. 7.45. Sample list of available hardware outputs

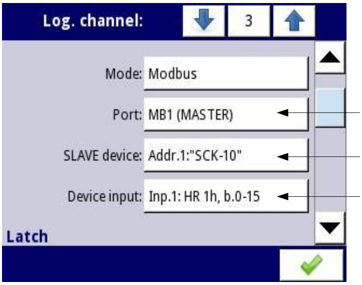
7.8.4. Logical Channels - Modbus mode



Modbus mode allows the user to configure the logical channel to:

- read the data from a SLAVE device transmitted over RS-485 Modbus RTU (MultiCon device in **MASTER** mode),
- read/write the data from/to logical channel of MultiCon device transmitted over RS-485 Modbus RTU (MultiCon device in SLAVE mode) or by Ethernet port (Modbus TCP/IP, MultiCon device in **SLAVE** mode)

This data stored in the logical channel in Modbus mode can be display on screen, processed by other logical channels (e.g. by mathematical function or virtual relay) or it can be the data source for controlling an output.



Log. channel: 3

Mode: Modbus

Port: MB1 (MASTER)

SLAVE device: Addr.1:"SCK-10"

Device input: Inp.1: HR 1h, b.0-15

Latch

Port MB1, Multicon in MASTER mode for this port

Modbus address: '1', Device name: "SCK-10"

Input channel number: '1', HOLDING Register: '1h', size: 'b.0-15':

Fig. 7.46. Sample Logical channel setting in Modbus mode

The parameters of **Logical channel** in **Modbus** mode are (see **Fig. 7.46**):

- **Name** - it gives the name to the logical channel,,
- **Unit** - to create a unit use the **Scaling** parameter in **Processing** parameter block (more information see **Chapter 7.8.1. Logical Channels - General settings**),
- **Mode=Modbus** - in this parameter the user selects source type of data for logical channel,
- **Port** - this parameter allows the user to select a serial/ethernet port from the list. The basic version of MultiCon has one built-in RS-485 port. The number of serial ports can be increased by installing a communication module into slot D of the device. This module offers 2 additional serial ports (one RS-485, and one RS-485/RS-232)

and one ethernet port, which allows for an advanced Multi-Modbus system (see **Fig. 7.47**).

- **Slave device** - this parameter is only visible if the selected option in the **Port** parameter is in **Modbus MASTER** mode (see **Chapter 7.15.3. 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 depend on the mode of the port
  - for **Modbus MASTER** mode (see **Chapter 7.15.3**), using this parameter the user can select the read register of the SLAVE device from the list defined in the **Modbus** menu,
  - for **Modbus SLAVE** mode (see **Chapter 7.15.2**), next to **Device input** label is the logical channel number and register number assigned to this logical channel are displayed,
- **Latch** parameter block - allows user to set the **latch function** which will hold the last value of a channel (discussed in **Chapter 7.8.1. Logical Channels - General settings**),
- **Processing** parameter block - is used for scaling and filtering data (discussed in **Chapter 7.8.1. 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 7.8.1. Logical Channels - General settings**),

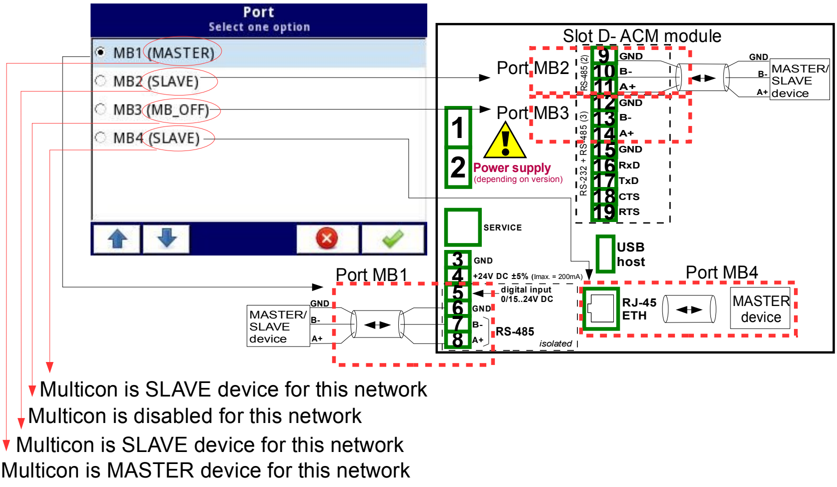


Fig. 7.47. Sample list of available Modbus ports

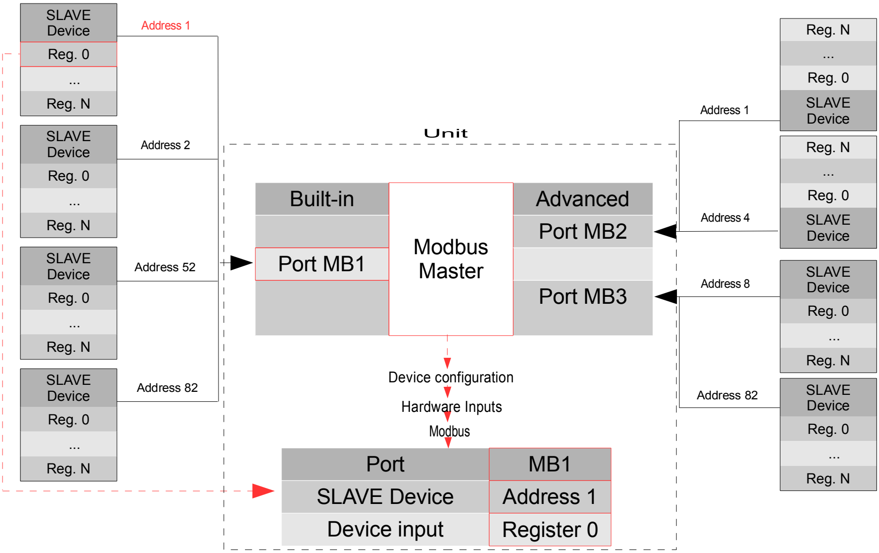
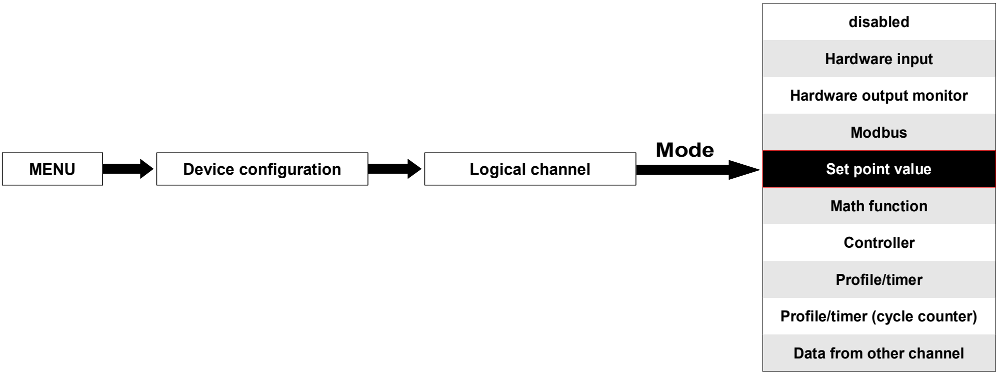


Fig. 7.48. An example of configuration block diagram of the Modbus protocol implemented in the device

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

For more information about the Modbus protocol implemented in the MultiCon see **Chapter 7.15. MODBUS**.

7.8.5. Logical Channels - Set point value mode



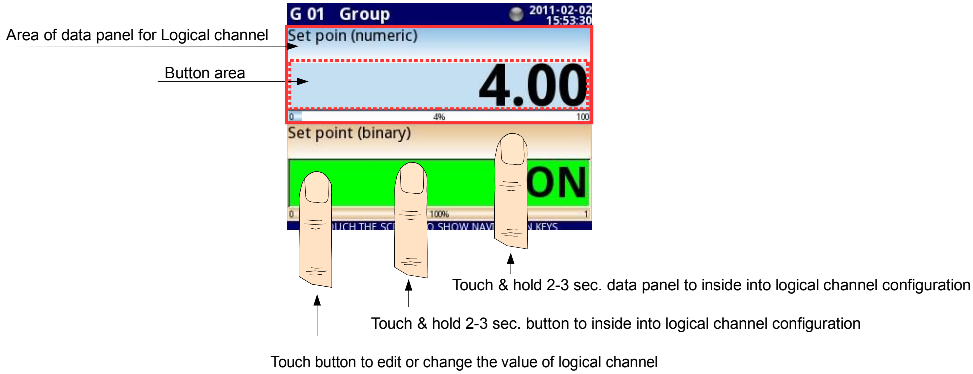
In **Set point value** mode the user defines constant value which can be displayed on the screen, used for calculations in other channels (e.g. in math functions, controllers etc.) or as a data source for built-in or external outputs which will control any object.

The parameters of the **Logical channels** in **Set point value** mode should be:

- **Name** - it gives the name to the logical channel,
- **Mode=Set point value** - in this parameter user selects source type of data for 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 **Set point value** parameter appears in window allowing entry of a value (see **Fig. 7.6**), 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 **Fig. 7.49**), this parameter has two options:
  - **disabled** - the button is disabled on the display, in this case the data source of logical channel will be the value set in the **Set point value** parameter, (see **Fig. 7.50**),
  - **enabled** - this button is active, in this case the data source of logical channel will be the value set after pressing the button, but the initial value is the number written in **Set point value** parameter, before setting the **Edit button** parameter to **enabled**,
- **Button mode** – parameter is visible only in case when **Edit button** parameter is set on **enabled**, and **Format** parameter in **Displaying** parameter block is set on **binary**, it has two options:
  - **bistable** – pressing and releasing the button will cause permanent change of state,
  - **monostable** - pressing and releasing the button will cause that the button on 0,1 second will switch from **OFF** state to **ON** state, and after that time, it will switch again to **OFF** state,

– **Latch** parameter block - allows user to set the **latch function** which will hold the last value of a channel (discussed in **Chapter 7.8.1. Logical Channels - General settings**),

– **Displaying** parameter block - for these parameters the user select format and range of data to be displayed on the screen (discussed in **Chapter 7.8.1. Logical Channels - General settings**),



Area of data panel for Logical channel

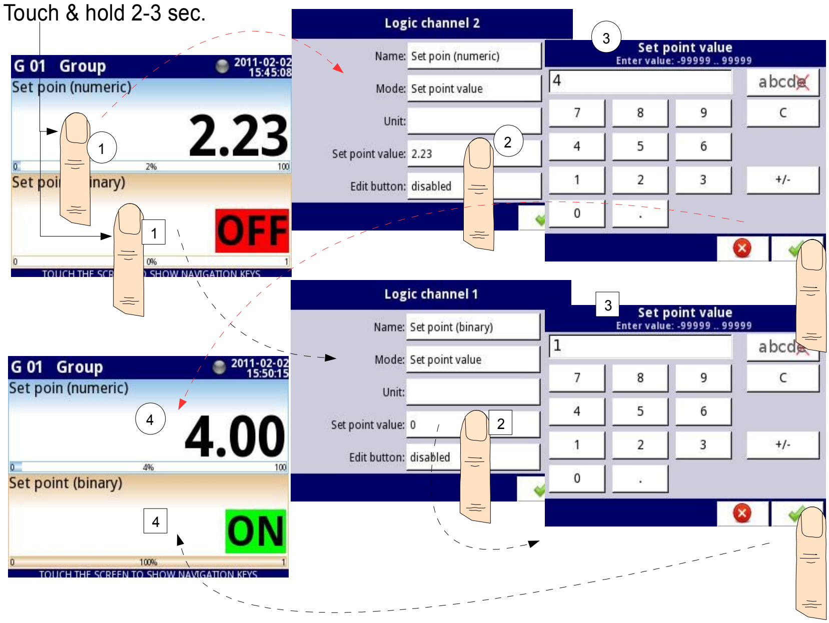
Button area

Touch & hold 2-3 sec. data panel to inside into logical channel configuration

Touch & hold 2-3 sec. button to inside into logical channel configuration

Touch button to edit or change the value of logical channel

*Fig. 7.49. Data panels for Logical channel in **Set point value** mode*


Touch & hold 2-3 sec.

Logic channel 2

Name: Set poin (numeric)
Mode: Set point value
Unit:
Set point value: 2.23
Edit button: disabled

Set point value: 4

Logic channel 1

Name: Set point (binary)
Mode: Set point value
Unit:
Set point value: 0
Edit button: disabled

Set point value: 1

Fig. 7.50. Configuration of the value for Logical channel in Set point value for the disabled button

Manual operation of a button in the data panel

Action button in the data panel depends on the parameters from **Latch** block (description of the **Latch** parameter block in Chapter **7.8.1. Logical Channels - General settings**) and on the **Format** parameter in the **Displaying** parameter block (description of the **Displaying** parameter block in Chapter **7.8.1. Logical Channels - General settings**).

a) **for disabled latch function**, see **Fig. 7.51**

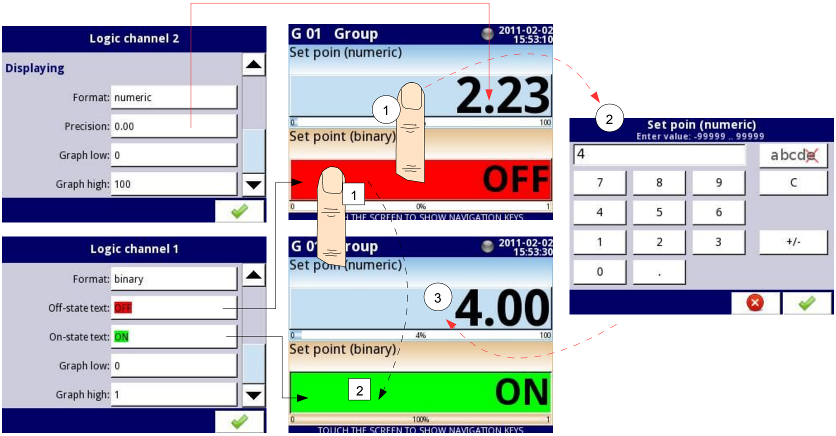
For format:

- **numeric** and **time** – after pressing the button an edit window appears allowing entry of value (see **Fig. 7.6**), 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 entry of value (see **Fig. 7.6**), 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 button is stored in the buffer and set the value of logical channel with the currently stored state when the latch function is disabled again.



Logic channel 2

Displaying

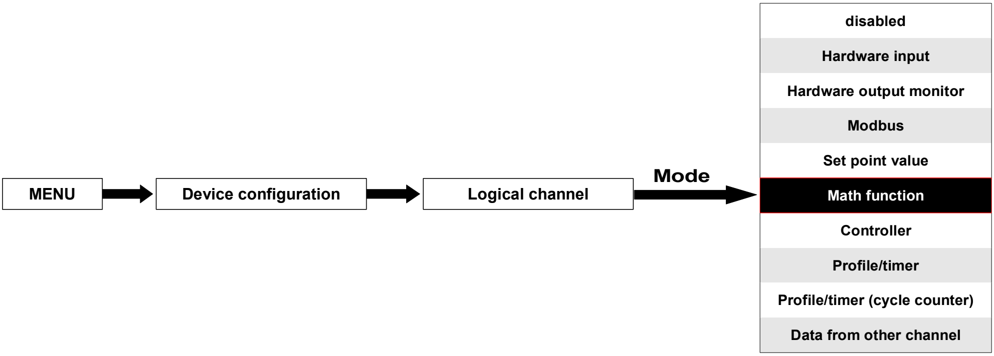
Format: numeric
Precision: 0.00
Graph low: 0
Graph high: 100

Logic channel 1

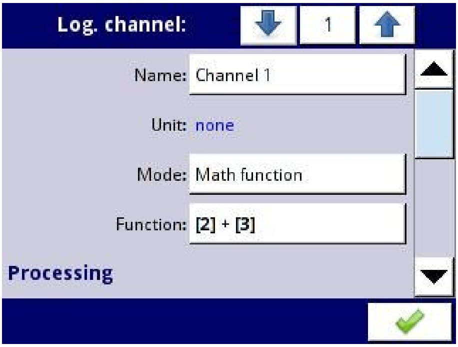
Format: binary
Off-state text: OFF
On-state text: ON
Graph low: 0
Graph high: 1

Fig. 7.51. Manual operation of the button for numeric and binary format

7.8.6. Logical Channels - Math function mode



This mode allows the user to define the Math function for display on the screen, processed this data in any other logical channels or it can be the data source for output to the control and settings of any object.



Log. channel: 1

Name: Channel 1

Unit: none

Mode: Math function

Function: [2] + [3]

Processing

Fig. 7.52. Input channels menu - parameters specific for Math function mode

MultiCon has a wide variety mathematical function which increases functionality and the range of the applications of the device. **Fig. 7.52** presents parameters of **Input channel** into **Math function** mode. This mode allows the user to display channel value, process this data in any other logical channel or it can be the data source for an output to control and set any object.

The parameters of **Logical channel** in **Math function** mode are:

- **Name** - it gives the name to the logical channel,
- **Unit** - to create a unit use the **Scaling** parameter (more information see **Chapter 7.8.1. Logical Channels - General settings**),
- **Mode=Math function** – in this parameter user selects source type of data for logical channel,
- **Function** - this parameter allow user to select math function from the list, for more information see below in this **Chapter**,
- **Latch** parameter block - allows user to set the **latch function** which will hold the last value of a channel (discussed in **Chapter 7.8.1. Logical Channels - General settings**),

- **Processing** parameter block - is used for scaling and filtering of data (discussed in **Chapter 7.8.1. Logical Channels - General settings**),
- **Displaying** parameter block - for these parameters the user select the format and range of data displayed on the screen (discussed in Chapter **7.8.1. Logical Channels - General settings**),

Function parameter in the **Math function** mode

Basic math functions implemented into 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 function that are discussed in the table below. After pressing the button next to the **Function** label go to a 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 the 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 a list in **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 **Type of source X** parameter. This parameter allows user to:
  - select logical channel from list (**Type of source Y=channel**),
  - or enter set point value (**Type of source Y=value**),
- **X error handling** – available for some math functions, depending on this parameter, the user can set 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 selected math function,
- **Init. value mode** – available for some math functions, depending on this parameter, the user can set following options:
  - **channel** – means that initial value will be loaded from selected logical channel,
  - **value** – means that initial value will be a constant value entered in the **Initial value** parameter,
- **Init. value source** – available for certain math functions, selects logical channel from list which will be initial value source for the function,
- **Initial value** – available for certain math functions, constant value which will be 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 following options:
  - **channel** – means that maximal value will be determined by selected logical channel,
  - **value** – means that maximal value will be a constant value entered in the **Maximal value** parameter,
- **Max. value source** – available for certain math functions, selects logical channel from list which will be maximal value source for the function,

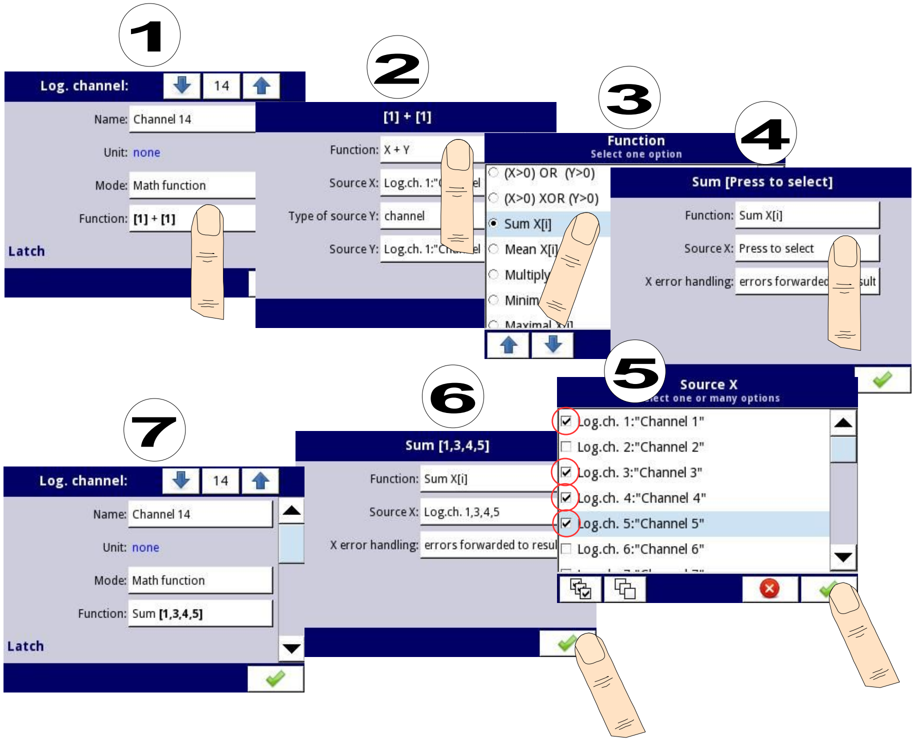
- **Maximal value** – available for certain math functions, constant value which will be maximal value for the function,
- **Averaging time** – available for **Function=avarege of X**, determines logical channel amount of samples which fits in this time and will be averaged; after this time function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

  - **rising edge** – flip-flop triggers when in selected in **Clock source** parameter channel rising edge occurs,
  - **falling edge** – flip-flop triggers when in selected in **Clock source** parameter channel falling edge occurs,
- **R input mode** – available for **Function=flip flop**, the user can set following options:
  - **disabled** – resetting flip-flop R input in inactive,
  - **from log. channel** – means that flip-flop R input source will be logical state of the selected logical channel,
  - **from channel (negation)** – means that flip-flop R input source will be negated logical state of the selected logical channel,
- **R input source** – available for **Function=flip flop**, selects a logical channel which will be flip-flop R input source,
- **S input mode** – available for **Function=flip flop**, the user can set following options:
  - **disabled** – setting flip-flop S input in inactive,
  - **from log. channel** – means that flip-flop S input source will be logical state of the selected logical channel,
  - **from channel (negation)** – means that flip-flop S input source will be negated logical state of the selected logical channel,
- **S input source** – available for **Function=flip flop**, selects a logical channel which will be 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 a list in **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 of **Type of source Y** parameter. This parameter allows user to:
  - select logical channel from list (**Type of source Y=channel**),
  - or enter set point value (**Type of source Y=value**),
- **Unit** - available for certain trigonometric functions. Allow an option to be selected:
  - **degree**
  - **radian**

Example of configuration of logical channel in Math function mode

Steps to configure the function which summarizing values of logical channels: 1, 3, 4, 5 are shown in **Fig. 7.53**. If any logical channel has **Error** state or value exceeds the range of a logical channel (Hi, Lo) then output has the same state. The following steps are:

(1) - Select **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 function from the list, in this case: **Sum X[i]**,
(4) - Press button next to the Source X label to enter to the channel list,
(5) - Choose logical channel from a 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 function next to the **Function** label: **Sum [1, 3, 4, 5]**,



1
Log. channel: 14
Name: Channel 14
Unit: none
Mode: Math function
Function: [1] + [1]
Latch

2
[1] + [1]
Function: X + Y
Source X: Log.ch. 1:"(el
Type of source Y: channel
Source Y: Log.ch. 1:"Ch el

3
Function
Select one option
○ (X>0) OR (Y>0)
○ (X>0) XOR (Y>0)
● Sum X[i]
○ Mean X[i]
○ Multiply
○ Minim
○ Maximal X[i]

4
Sum [Press to select]
Function: Sum X[i]
Source X: Press to select
X error handling: errors forwarded sult

5
Source X
ect one or many options
☑ Log.ch. 1:"Channel 1"
□ Log.ch. 2:"Channel 2"
☑ Log.ch. 3:"Channel 3"
☑ Log.ch. 4:"Channel 4"
☑ Log.ch. 5:"Channel 5"
□ Log.ch. 6:"Channel 6"

6
Sum [1,3,4,5]
Function: Sum X[i]
Source X: Log.ch. 1,3,4,5
X error handling: errors forwarded to resul

7
Log. channel: 14
Name: Channel 14
Unit: none
Mode: Math function
Function: Sum [1,3,4,5]
Latch

Fig. 7.53. 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 the 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 the 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 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 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 the 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 the 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 the 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 the 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 the 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 the ***Example 7.8.11.1***, *Chapter **7.8.11***)
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

Math function	Description	Example
arctan(X)	Arctangent of **Source X** value	**arctan([4])** - arctangent value of channel 4
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 power to value of channel 2
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 flop-flop with clock input and asynchronous **R** and **S** inputs	**filp 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. 7.3 Available math functions

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

User should pay attention for a fact that some mathematical functions have limitations. Those limitations are listed bellow:

Function **X/Y**:

- If **Y** equals **0**, then logical channel will be in **-Err**- state; in other way:
  Y == 0 ⇒ ERROR

Function **arcsin(X)**:

- If **absolute value** of **X** is greater than **1**, then logical channel will be in **-Err-** state; in other way:
  abs(X) > 1.0 ⇒ ERROR

Function **arccos(X)**:

- If **absolute value** of **X** will be greater than **1**, then logical channel will be in **-Err**- state; in other way:
  abs(X) > 1.0 ⇒ 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; in other way:
  abs(X[rad]) - k*π < 1.0e-8, k∈N ⇒ ERROR

Function **arctan(X)**:

- If **absolute value** of **X** (in radians) is lower than **one hundred millionth**, then the function result will be **0**; in other way:
  abs(X[rad]) < 1e-8 ⇒ 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**; in other way:
  abs(X[rad]) > 1e8 ⇒ 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; in other way:
  abs(X[rad]) - k*π < 1.0e-8, k∈N ⇒ 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**; in other way:
  X [rad] < 1e-8 ⇒ 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**; in other way:
  abs(X[rad]) > 1e8 ⇒ arcctg(X) = 0

Function **X^Y**:

- If **X** jest is equal **0**, and **Y** is lower than **0**, then logical channel will be in **-Err-** state; in other 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; in other way:

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

Function **$\log_y(X)$**:

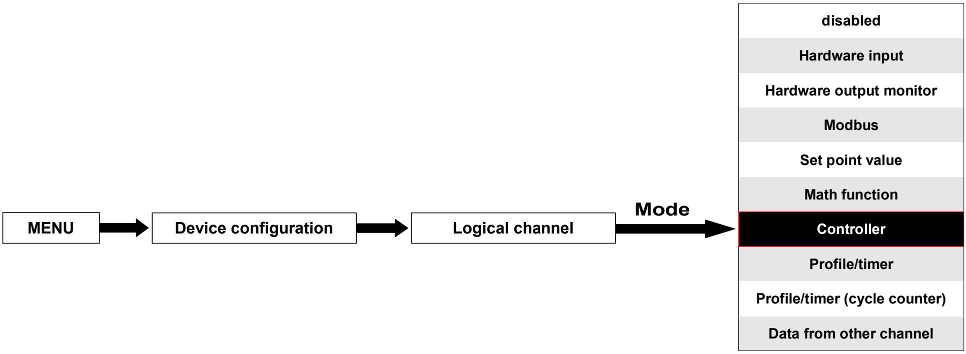
- If **X** or **Y** is lower than **$1.40129846*10^{-45}$**, then logical channel will be in **-Err-** state; in other way:

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

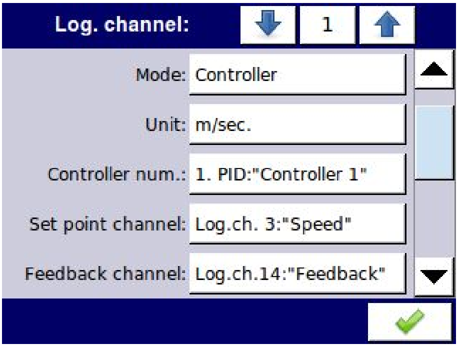
- If **absolute value** of **Y** minus **1** is lower than **$1.19209290*10^{-7}$**, then logical channel will be in **-Err-** state; in other way:

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

7.8.7. 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 **controlling process** the **Logical Channel** should be set in **Controller** mode as in **Fig. 7.54**.



Log. channel: 1

Mode: Controller

Unit: m/sec.

Controller num.: 1. PID:"Controller 1"

Set point channel: Log.ch. 3:"Speed"

Feedback channel: Log.ch.14:"Feedback"

Fig. 7.54. Input channel configuration in Controller mode

To the parameters of **Logical channel** in **Controller** mode should be:

- **Name** - it gives the name to the logical channel,
- **Mode=Controller** - in this parameter user selects source of data for 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). User must configure the selected controller in the **Controllers** menu (overview and configuration parameters that define the controllers can be found in **Chapter 7.13. CONTROLLERS**).
- **Set point channel** - this parameter allows the user to select a logical channel with set point value (see **Fig. 7.55**), **Set point channel** defines input data for process control,
- **Feedback channel** - this parameter allow the user to select a channel with data returned from control system (see **Fig. 7.55**), **Feedback channel** define input data for process control,

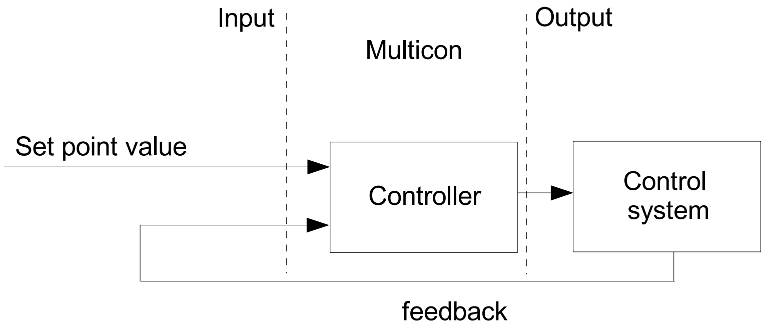


Fig. 7.55. The overall structure of control system implemented in the device

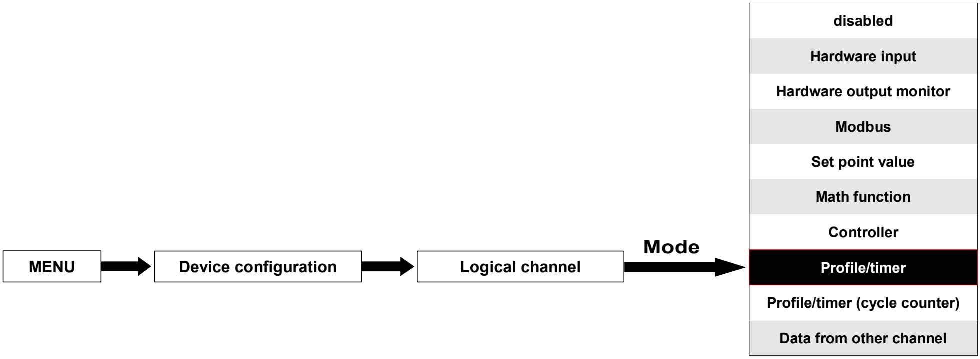
- **Latch** parameter block - allows user to set the **latch function** which will hold the last value of a channel (discussed in **Chapter 7.8.1. Logical Channels - General settings**),

- **Processing** parameter block - is used for scaling and filtering data (discussed in **Chapter 7.8.1. Logical Channels - General settings**),

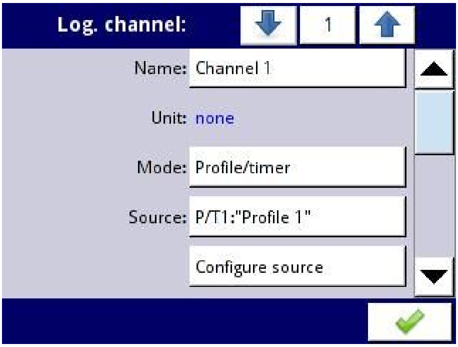
- **Displaying** parameter block - for these parameters the user select format and range of data displayed on the screen (discussed in **Chapter 7.8.1. Logical Channels - General settings**).

ℹ User must configure the controller parameters in the **Controllers** menu before using this controller to control real object (To read more about **Controllers profiles** see **Chapter 7.13. CONTROLLERS**).

7.8.8. Logical Channels - Profile/timer mode



This mode allows the user to set up appropriate Profile/timer defined in **Profiles/timers** menu which may generate 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 **Profile/timer** the **Logical Channel** should be set in **Profile/timer** mode as in **Fig. 7.56**.



Log. channel: 1

Name: Channel 1

Unit: none

Mode: Profile/timer

Source: P/T1:"Profile 1"

Configure source

Fig. 7.56. View of the input channel configuration in Profile/timer mode

The parameters of the **Logical channel** in **Profile/timer** mode should be:

- **Name** - it gives the name to the logical channel,
- **Unit** - to create unit use the **Scaling** parameter in **Processing** parameter block (for more information see **Chapter 7.8.1. Logical Channels - General settings**),
- **Mode=Profile/timer** - in this parameter user selects source of data for logical channel,
- **Source** - this parameter allows the user to select a Profile/timer from the list (**8 Profiles/timers** are implemented in the device), which selected option will be data source for this logical channel. User must configure the selected Profile/timer by pressing the **Configure source** button or enter into the **Profiles/timers** menu (overview and configuration parameters that define the Profiles/timers can be found in **Chapter 7.12. PROFILES/TIMERS**),
- **Configure source** - this button allows the user to configure **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 7.8.1. Logical Channels - General settings**),

- **Processing** parameter block - is used for scaling and filtering data (discussed in **Chapter 7.8.1. Logical Channels - General settings**),

- **Displaying** parameter block - for these parameters the user select format and range of data displayed on the screen (discussed in **Chapter 7.8.1. Logical Channels - General settings**),

User must configure the Profile/timer parameters in the **Profiles/timers** menu before using this Profile/timer to control real object (To read more about **Profiles/timers** profiles see **Chapter 7.12. PROFILES/TIMERS**).

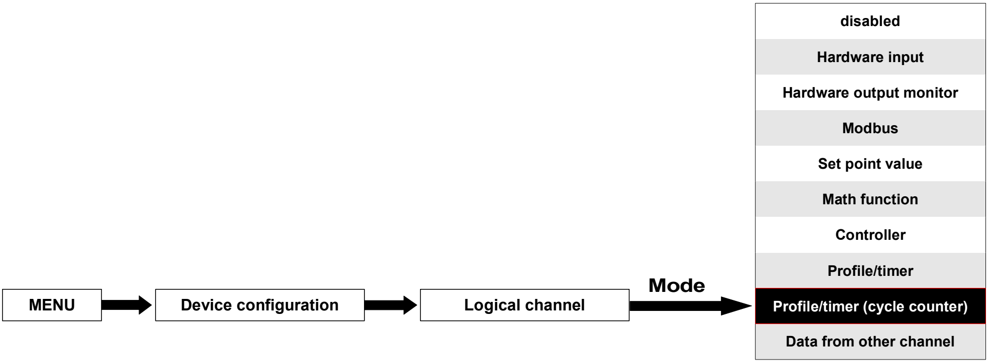
Configure source button in **Profile/timer** menu

There are two methods for configuring Profiles/timers:

- by the **Profiles/timers** menu, following steps: **MAIN->Device configuration-> Profiles/timers** (see **Chapter 7.12. PROFILES/TIMERS**),
- directly in the **Logical channel** in **Profile/timer** mode by pressing the **Configure source** button.

View of window of configuration the profile/timer in both cases is the same. Overview of setting profile/timer is presented in **Chapter 7.12. PROFILES/TIMERS**.

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



This mode is similar to **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.

Cycle counter is generated depending on the **Looping** parameter value, ie when **Looping=disabled** the cycle counter is equal to 0.

The parameters of the **Logical channel** in **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 **Processing** parameter block, to create unit use the **Scaling** parameter (for more information see **Chapter 7.8.1. Logical Channels - General settings**),
- **Mode=Profile/timer** - in this parameter 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 7.8.1. Logical Channels - General settings**),
- **Processing** parameter block - is used for scaling and filtering data (discussed in **Chapter 7.8.1. 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 7.8.1. Logical Channels - General settings**),

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 enter into the **Profiles/timers** menu,

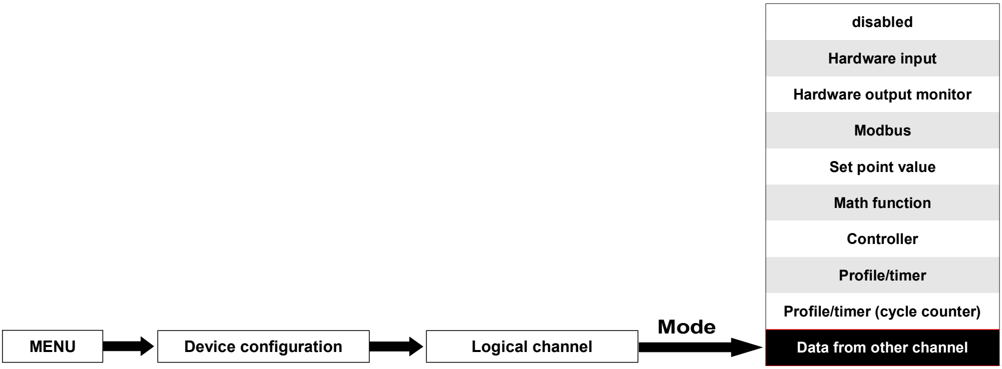
Configure source button in **Profile/timer (cycle counter)** menu

There are two methods for configuring Profiles/timers:

- by the **Profiles/timers** menu, take the following steps: **MAIN->Device configuration->Profiles/timers** (see **Chapter 7.12. PROFILES/TIMERS**),
- directly in the **Logical channel** in **Profile/timer** mode by pressing the **Configure source** button.

The configuration of the profile/timer in both cases is the same. An overview of settings of the profile/timer is presented in **Chapter 7.12. PROFILES/TIMERS**.

7.8.10. Logical Channels - Data from other channel mode



This mode has 2 applications. We can use this mode to:

- copy logical channel,
- read the value from a source which has more than one quantity of output value (e.g. **FI4 input** module has 2 output quantities: flow and balance) to be able to display all possible quantities defined in the logical channel,

The parameters of the **Logical channel** in **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 7.8.1. 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 modules can return more than one measured quantity (e.g. FI4 input module has 2 output elements: flow and balance). 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 7.8.1. Logical Channels - General settings**),
- **Processing** parameter block - is used for scaling and filtering data (discussed in **Chapter 7.8.1. 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 7.8.1. Logical Channels - General settings**),

7.8.11. Examples of Logical Channels configuration

Example 7.8.11.1: *Application of Logical channel in the Hardware input mode for UI4 module (see the **Appendix 8.2 UI4, UI8, UI12, U16, U24, I16, I24 – VOLTAGE and CURRENT MEASUREMENT MODULES**)and **Math function** mode*

Task:

This example shows how to measure voltage and current and calculate a power consumption of the heater. Voltage, current and power of the heater must be display in one window in needle dials mode. Additional in this example user must use external converters to measure voltage and current exceed the range of **UI4** input module.

Solution:

First configure the device as described below, then connect the heater to converters and then to the device (in accordance with **Fig. 7.57**).

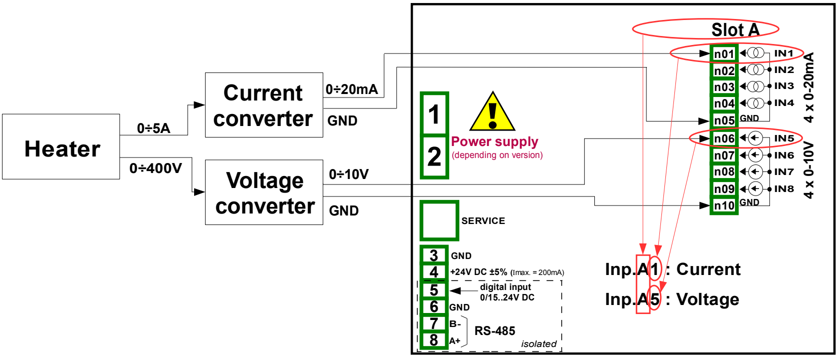


*Fig. 7.57. Schematic diagram for the **UI4** module*

In the next step you need to define three logical channels: for current, voltage and power. To do this:

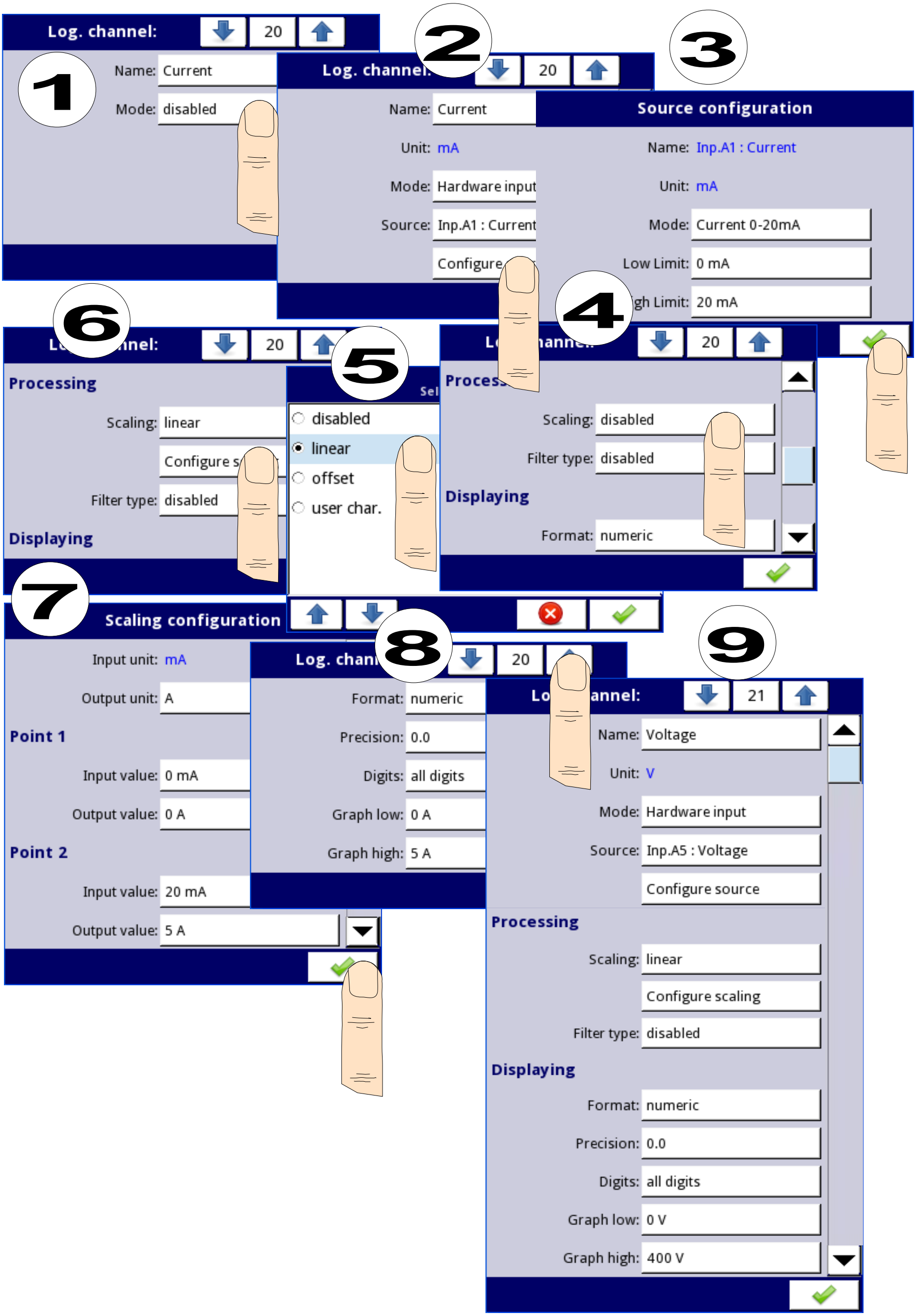
a) For a current,
   - touch screen and press the **Menu** button,
   - press the **Device configuration** button,
   - enter the **Logical channels** menu,
   - using the arrows in the top navigation bar, select any **Logical channel** such as 20,
   - in **Name** parameter write **Current**,
   - **Mode** parameter set as **Hardware input**,
   - **Source** parameter set as **Inp. A1: Current** (see the connection in **Fig. 7.57**),
   - 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 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 other parameters should be on default settings,

Configuration steps are shown on **Fig. 7.58**.

b) For a voltage,

- using the arrows in the top navigation bar, select any **logical channel** such as 21,
- in **Name** parameter write **Voltage"**,
- **Mode** parameter set as **Hardware input**,
- **Source** parameter set as **Inp. A5: Voltage** (see the connection in **Fig. 7.57**),
- press the **Source configuration** button to set the source configuration options:
  - **Mode** parameter set as **0-10V**, because this voltage range it comes from voltage converter,
  - **Low Limit** parameter set as **0V**,
  - **High Limit** parameter set as **10V**,
- **Scale** parameter set as **linear**,
- press **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 other parameters should be on default settings,



1

Log. channel: 20

Name: Current

Mode: disabled

2

Log. channel: 20

Name: Current

Unit: mA

Mode: Hardware input

Source: Inp.A1 : Current

Configure

3

Source configuration

Name: Inp.A1 : Current

Unit: mA

Mode: Current 0-20mA

Low Limit: 0 mA

High Limit: 20 mA

4

Log. channel: 20

Processing

Scaling: disabled

Filter type: disabled

Displaying

Format: numeric

5

○ disabled

● linear

○ offset

○ user char.

6

Log. channel: 20

Processing

Scaling: linear

Configure s

Filter type: disabled

Displaying

7

Scaling configuration

Input unit: mA

Output unit: A

Point 1

Input value: 0 mA

Output value: 0 A

Point 2

Input value: 20 mA

Output value: 5 A

8

Log. chan 20

Format: numeric

Precision: 0.0

Digits: all digits

Graph low: 0 A

Graph high: 5 A

9

Log. channel: 21

Name: Voltage

Unit: V

Mode: Hardware input

Source: Inp.A5 : Voltage

Configure source

Processing

Scaling: linear

Configure scaling

Filter type: disabled

Displaying

Format: numeric

Precision: 0.0

Digits: all digits

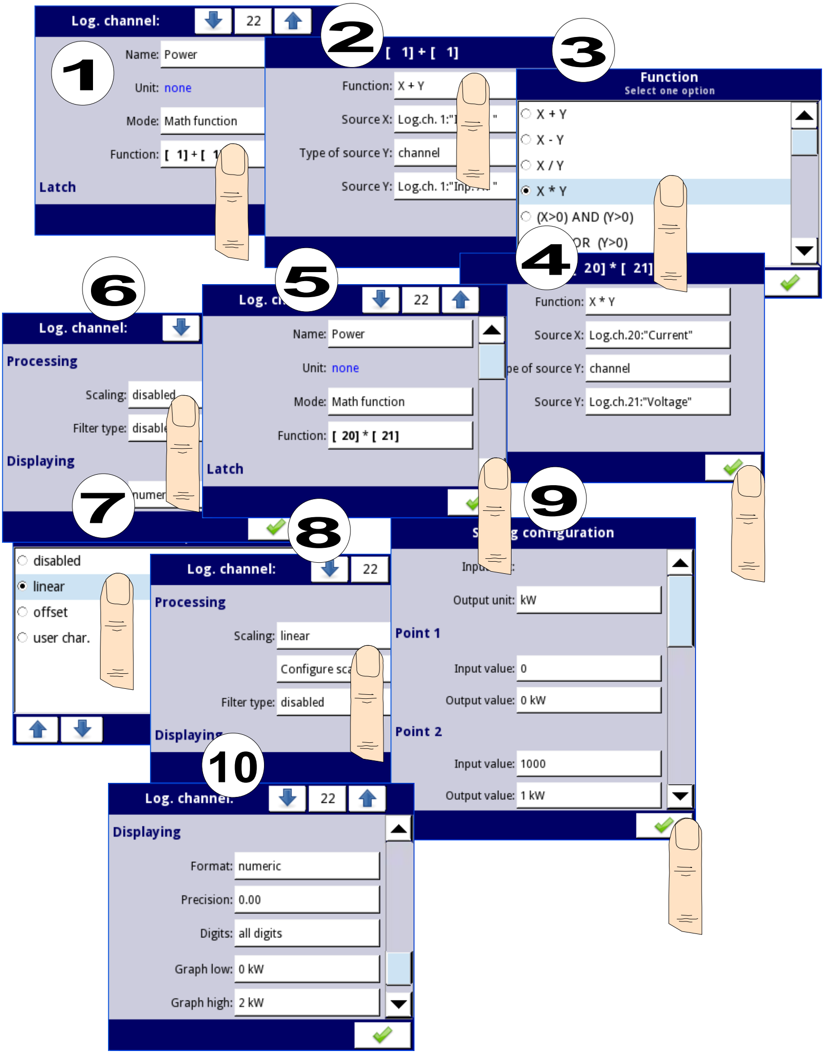
Graph low: 0 V

Graph high: 400 V

Fig. 7.58. Current and voltage source configuration example

c) For a 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 defined as a 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 reading of the heater voltage,
- **Scale** parameter set as **linear**,
- press **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 heater power consumption exceeds 2kW=5A*400V),
- all other parameters should be on default settings,



1

Log. channel: 22

Name: Power

Unit: none

Mode: Math function

Function: [1] + [1

Latch

2

[1] + [1]

Function: X + Y

Source X: Log.ch. 1:"I "

Type of source Y: channel

Source Y: Log.ch. 1:"Inp... "

3

Function

Select one option

○ X + Y

○ X - Y

○ X / Y

● X * Y

○ (X>0) AND (Y>0)

OR (Y>0)

4

20] * [21]

Function: X * Y

Source X: Log.ch.20:"Current"

pe of source Y: channel

Source Y: Log.ch.21:"Voltage"

5

Log. c 22

Name: Power

Unit: none

Mode: Math function

Function: [20] * [21]

Latch

6

Log. channel:

Processing

Scaling: disabled

Filter type: disable

Displaying

numer

7

○ disabled

● linear

○ offset

○ user char.

8

Log. channel: 22

Processing

Scaling: linear

Configure sca

Filter type: disabled

Displaying

9

S g configuration

Inpu :

Output unit: kW

Point 1

Input value: 0

Output value: 0 kW

Point 2

Input value: 1000

Output value: 1 kW

10

Log. channel: 22

Displaying

Format: numeric

Precision: 0.00

Digits: all digits

Graph low: 0 kW

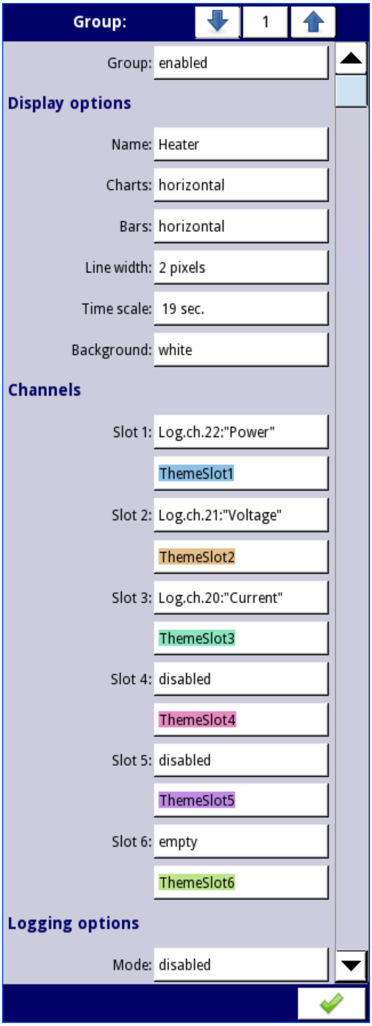
Graph high: 2 kW

Fig. 7.59.Math function configuration example

In the next step you need to define **Group** the 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 its 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 other parameters should be on default settings,



Group: 1

Group: enabled

Display options

Name: Heater

Charts: horizontal

Bars: horizontal

Line width: 2 pixels

Time scale: 19 sec.

Background: white

Channels

Slot 1: Log.ch.22:"Power"

ThemeSlot1

Slot 2: Log.ch.21:"Voltage"

ThemeSlot2

Slot 3: Log.ch.20:"Current"

ThemeSlot3

Slot 4: disabled

ThemeSlot4

Slot 5: disabled

ThemeSlot5

Slot 6: empty

ThemeSlot6

Logging options

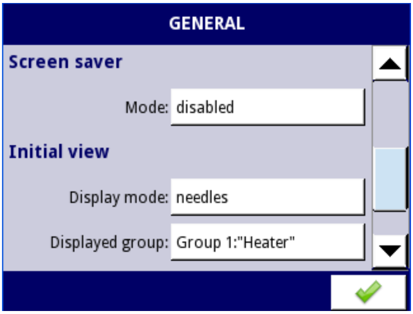
Mode: disabled

Fig. 7.60.Example of Group configuration

The last step is to define the **Initial view** (see **Fig. 7.61**) displayed on the LCD screen when the device starts. Initial View has to display a defined group of logical channels: power, voltage and current in-needle mode. To do this:

- enter in the **General settings** menu,
- **Basic** block parameters left unchanged unless indicated date and time were to 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 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 displayed only **Group 1** in the same needle mode,
- after setting all parameters exit the **General settings** menu,



GENERAL

Screen saver

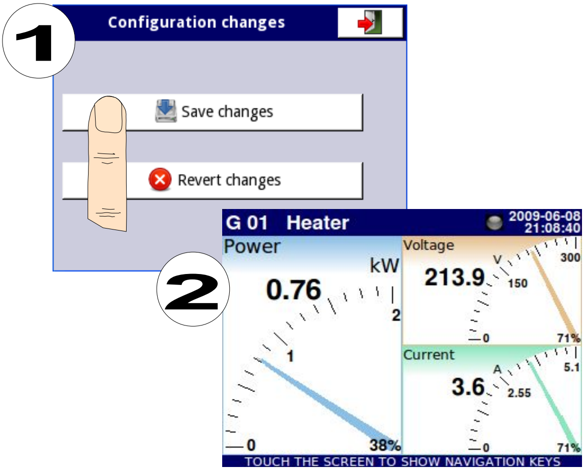
Mode: disabled

Initial view

Display mode: needles

Displayed group: Group 1:"Heater"

Fig. 7.61 An example of Initial view configuration



TOUCH THE SCREEN TO SHOW NAVIGATION KEYS

Fig. 7.62 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 manual switch for displaying Group 1 and needles mode (see Chapter **6.3.2 Navigation bar**). Example of a window view for 3 logical channels is shown in **Fig. 7.62**.

Example 7.8.11.2: *Application of Logical channel in the* **Hardware input** *mode for* **TC4** *modules (see the* **Appendix** ***8.6. TC4, TC8, TC12 – THERMOCOUPLE SENSOR MEASUREMENT MODULES****) and* ***Math function*** *mode.*

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

Solution:
Before measuring temperature the thermocouple sensors should be connected to the device. An example of how the connection is shown in **Fig. 7.63** (see also **Appendix** ***8.6. TC4, TC8, TC12 – THERMOCOUPLE SENSOR MEASUREMENT MODULES***).

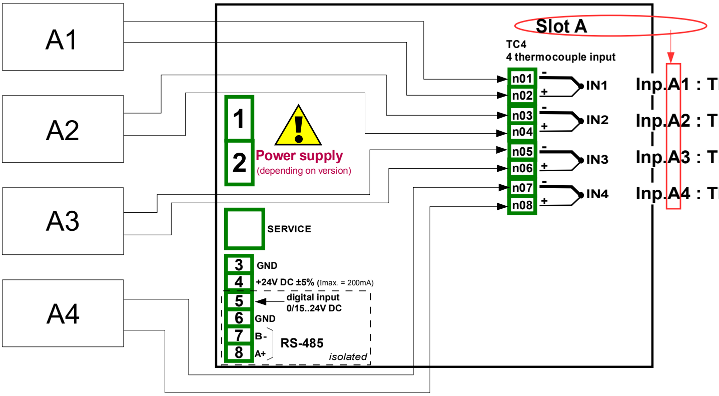


Fig. 7.63. Schematic diagram for the ***TC4*** *module*

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 an temperature alarm condition. To do this:

a) For the temperature reading of sensor 1 with the symbol A1:
   - touch 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 **Name** parameter write **Temperature A1**,
   - **Mode** parameter set as **Hardware input**,
   - Source parameter set as **Inp. A1: Thermocouple** (see the connection in **Fig. 7.63**),
   - press the **Source configuration** button to set the source configuration options,
     - **Mode** parameter et 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 **Name** parameter write **Temperature A2**,
- Mode parameter set as **hardware input**,
- Source parameter set as **Source** - **Inp. A2: Thermocouple** (see **Fig. 7.63**),
- 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 **Name** parameter write **Temperature A3**,
- Mode parameter set as **hardware input**,
- Source parameter set as **Source** - **Inp. A3: Thermocouple** (see **Fig. 7.63**),
- 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 **Name** parameter write **Temperature A4**,
- Mode parameter set as **hardware input**,
- Source parameter set as **Source** - **Inp. A4: Thermocouple** (see **Fig. 7.63**),
- Other parameters are set as the sensor 1.

d) For alarm state,

- using the arrows in the top navigation bar, select any **Logical channel** (except for 1, 2, 3 and 4) eg 5,
- in Name parameter write **Alarm**,
- Mode parameter set as **Math function**,
- enter the **Function** submenu to select the appropriate function that allows the calculation 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 from **Logical channels** menu,

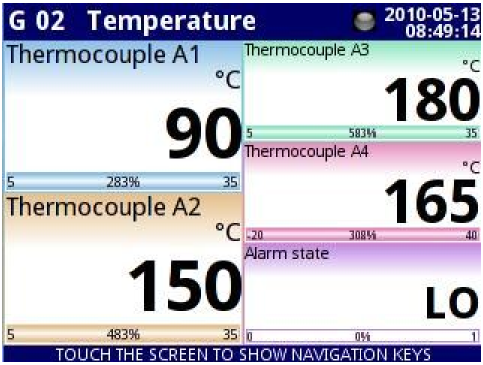
In the next step you need to define **Group** the set up logical channels to view them in one window on the screen. To do this:

- enter the **Group** menu, to enable the Group 1 if it is off,
- in **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 A1**,
  - **Slot 3** parameter set as **logical channel 3** called the **Temperature A1**,
  - **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 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. Initial View has to display a defined group of logical channels: 1, 2, 3, 4, 5 in value mode. To do this:

- enter in the **General settings** menu,
- **Basic** block parameters left unchanged unless indicated date and time were to set incorrectly and should be improved,
- for **LCD screen** block parameters:
  - **Backlight** parameter set to a 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.Measurment**,
- for the **Automatic view change** block parameters:
  - **Change mode** parameter set to disabled, as in this example we want to all the time displayed 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. Example of a window view for 5 logical channels is shown in **Fig. 7.64**.



G 02 Temperature 2010-05-13 08:49:14

Thermocouple A1 °C 90 5 283% 35

Thermocouple A3 °C 180 5 583% 35

Thermocouple A4 °C 165 -20 308% 40

Thermocouple A2 °C 150 5 483% 35

Alarm state LO 0 0% 1

TOUCH THE SCREEN TO SHOW NAVIGATION KEYS

Fig. 7.64. The proposition of the result the Example 2

Example 7.8.11.3*: Application of Logical channel in the Hardware input mode for RT4 modules (see the* Appendix ***8.7. RT4 , RT6 – RTD MEASUREMENT MODULES****).*

Task:
The MultiCon with built-in **RT4** module can measure simultaneously temperature at four points. Suppose you 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 connections 3-wire, and for the rest of PT100 and connections 2 and 4-wire. We want to read the temperature from 4 sensors in one window in vertical bars mode. In addition, we want to log temperature every 1 minute (for more details about logging see Chapter **7.14.2 Groups - Logging options**).

Solution:
Before measuring temperature, user should configure the device, and then connect the sensors as shown in **Fig. 7.65** (see Appendix ***8.7. RT4 , RT6 – RTD MEASUREMENT MODULES***).

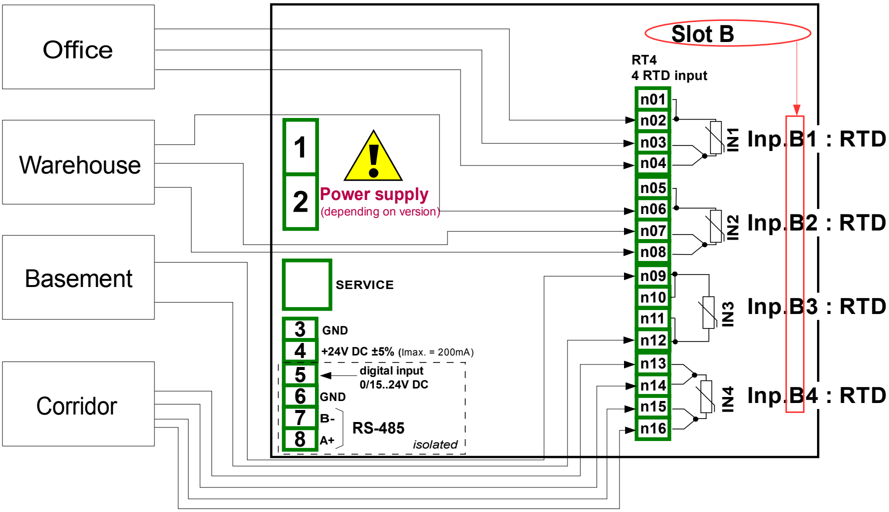


Fig. 7.65. Schematic diagram for the RT4 module

In first step you need to configure **Logical channels**.
To read the temperature from sensor 1 (Bathroom):

- touch 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 **Name** parameter write **Bathroom**,
- **Mode** parameter set as **Hardware input**,
- **Source** parameter set as **Inp. B1: RTD** (see the connection in **Fig. 7.65**),
- enter **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 **Source configuration** set **Mode** parameter as PT100,
Logical channel 33: **Corridor** ,in **Source configuration** set **Mode** parameter as PT100,

In the next step you need to define **Group** the 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 appear if you have license for logging):
  - **Mode** parameter set as **always**, we want to register at all times until the we change the configuration to stop the registration,
  - in **Description** parameter write M**easure 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. Initial view has to display a defined group of logical channels: 30, 31, 32, 33 in bars mode. To do this:

- enter **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 manual switch for displaying Group 2. Example of a window view for 4 logical channels is shown in **Fig. 7.66**.

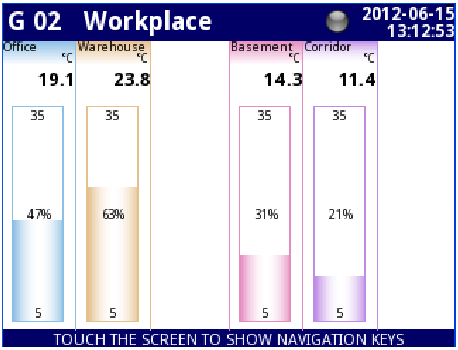


Fig. 7.66. Logical channels configuration - Example 3

Example 7.8.11.4**: Application of Logical channel in the Hardware output monitor mode for **R45, R81** modules (see the* **Appendix 8.13. R45, R81, R65, R121 - RELAY MODULES).*

Task:
This example shows how to read the value from output module **R45**.

Solution:
Before connecting relay to the device, user should configure the device, and then connect them as it is shown in **Fig. 7.67** (see **Appendix 8.7. R45, R81-relay modules**). After setting the configuration of the physical inputs can configure the input channel to read the output value.

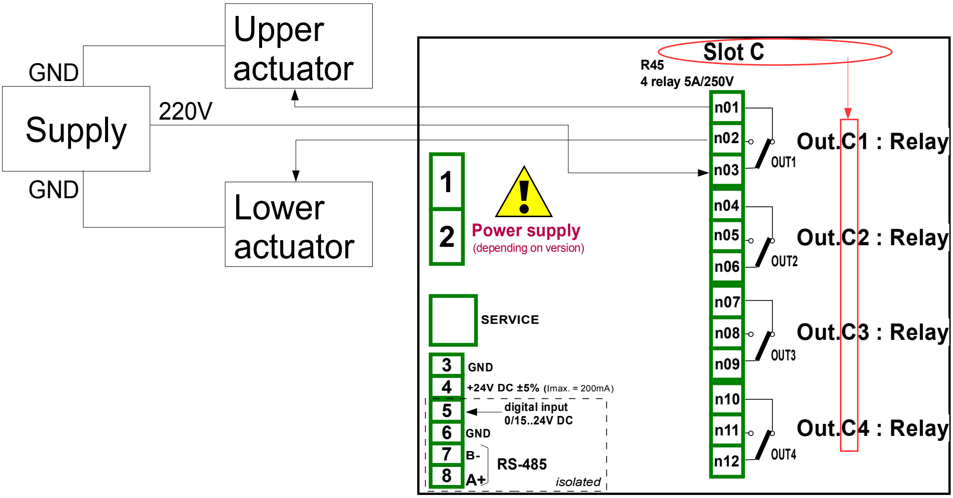


*Fig. 7.67. Schematic diagram for the **R45** module*

To read the output value from module:

- touch screen and press the **Menu** button,
- press **Device configuration** button,
- enter the **Logical channels** menu,
- using the arrows in the top navigation bar, select any **Logical channel** such as 5,
- in **Name** parameter write **Relay 1**,
- **Mode** parameter set as **Hardware output monitor**,
- **Source** parameter set as **Out. C1: Relay** (see the connection in **Fig. 7.67**),
- for **Displaying** block parameters:
  - **Format** parameter set as **binary**,
  - in **Off-state text** write **OFF** with red font color and yellow background,
  - in **On-state text** write **HI** with green font color and violet background,
  - in **Graph low** write **0**,
  - in **Graph high** write **1**,

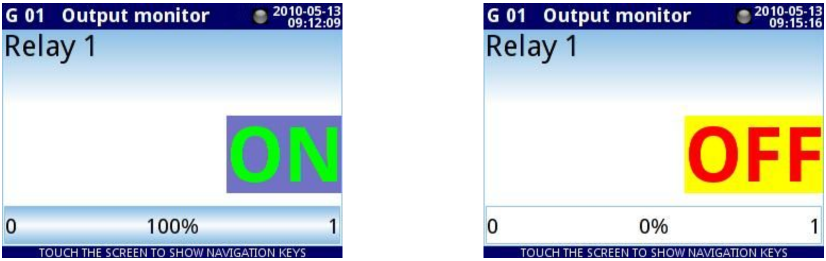
In the next step you need to define **Group** the set up logical channels to view them in one window on the screen. To do this:

- enter the **Group** menu, to enable the **Group 1** if it is off,
- in **Name** parameter write **Monitor output**,
- for **Channels** block parameters,
  - **Slot 1** parameter set as **Logical channel 5** called **Relay 1**,
  - other parameters in this block - **Slot 2÷6**, set as **disabled**, because we want to show only one logical channel,

The last step is to define the **Initial view** displayed on the LCD screen when the device starts. Initial View has to display a defined Group 1 in values mode. To do this:

- enter **General settings** menu,
- for **Initial view** block parameters:
  - **Display mode** parameter set as **value**,
  - **Displayed group** parameter set as **Group 1: Output monitor**,

After whole configuration exit the menu pressing **Save changes**, the result will be visible as soon as first measurement is done. View the window of MultiCon with the above configuration is shown in **Fig. 7.68**.



G 01 Output monitor 2010-05-13 09:12:09

Relay 1

ON

0 100% 1

TOUCH THE SCREEN TO SHOW NAVIGATION KEYS

G 01 Output monitor 2010-05-13 09:15:16

Relay 1

OFF

0 0% 1

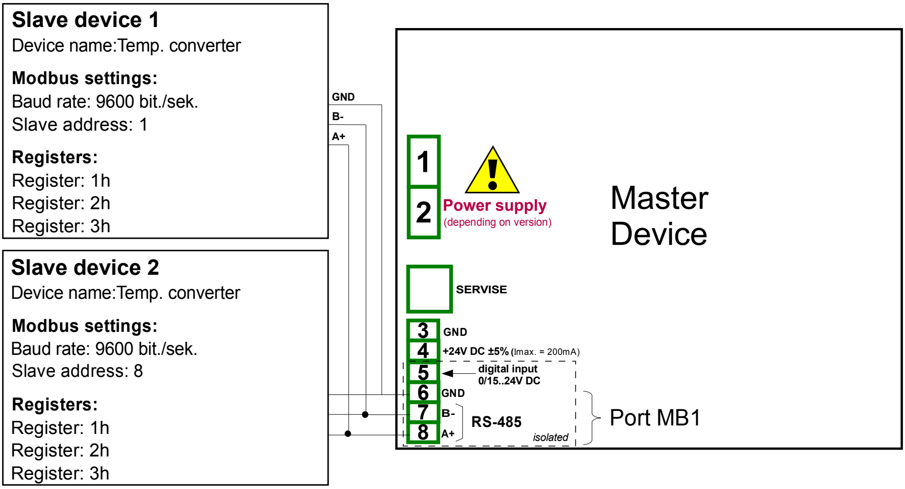
TOUCH THE SCREEN TO SHOW NAVIGATION KEYS

Fig. 7.68. The proposition of the result of the Example 4

Example 7.8.11.5*: Application of* ***Logical channel*** *in the* ***Modbus*** *mode (see* ***Chapter 7.15. MODBUS****).*

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

Solution:
First, configure the device in MASTER mode (like in **Example 7.15.4.1**), then configure it as follows and connect the SLAVE device as shown in **Fig. 7.69**.



Slave device 1
Device name:Temp. converter
Modbus settings:
Baud rate: 9600 bit./sek.
Slave address: 1
Registers:
Register: 1h
Register: 2h
Register: 3h

Slave device 2
Device name:Temp. converter
Modbus settings:
Baud rate: 9600 bit./sek.
Slave address: 8
Registers:
Register: 1h
Register: 2h
Register: 3h

Fig. 7.69. Schematic diagram for SLAVE devices connected to MultiCon

In first step you need to configure **Logical channels** which will read data from Slave devices. To do this:

- touch screen and press the **Menu** button,
- press the **Device configuration** button,
- enter **Logical channels** menu,
- using the arrows in the top navigation bar, select any **logical channel** such as 1,
- in **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 unit displayed on the screen,
- enter **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 **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 other parameters you should set like in previously configured logical channel,

In the next step you need to define **Group** the 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 **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 other parameters should be on default settings,

The last step is to define the **Initial view** displayed on the LCD screen the device starts. Initial View has to display a defined Group of logical channels: 1, 2 in values mode. To do this:

- enter **General settings** menu,
- for **Initial view** block parameters,
  - **Display mode** parameter set as **values**,
  - **Displayed group** parameter set as **Group 1: Temperature**,

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

Example 7.8.11.6**: Application of* **Logical channel** *in the* **Hardware input** *(see* ***8.6. TC4, TC8, TC12 – THERMOCOUPLE SENSOR MEASUREMENT MODULES*) mode and* ***Math function*** *mode (see* ***Chapter 7.8.6. Logical Channels - Math function mode****).*

Task:
In this example we calculate the mean value from logic channels 1÷8. In these channels are values from thermocouple sensors that are located around the furnace.

Solution:
Before measuring temperature, user should configure the device as follows and then connect the thermocouple sensors to the device. An example of the connection is shown in **Fig. 7.70** (see also **Appendix *8.6. TC4, TC8, TC12 – THERMOCOUPLE SENSOR MEASUREMENT MODULES***).

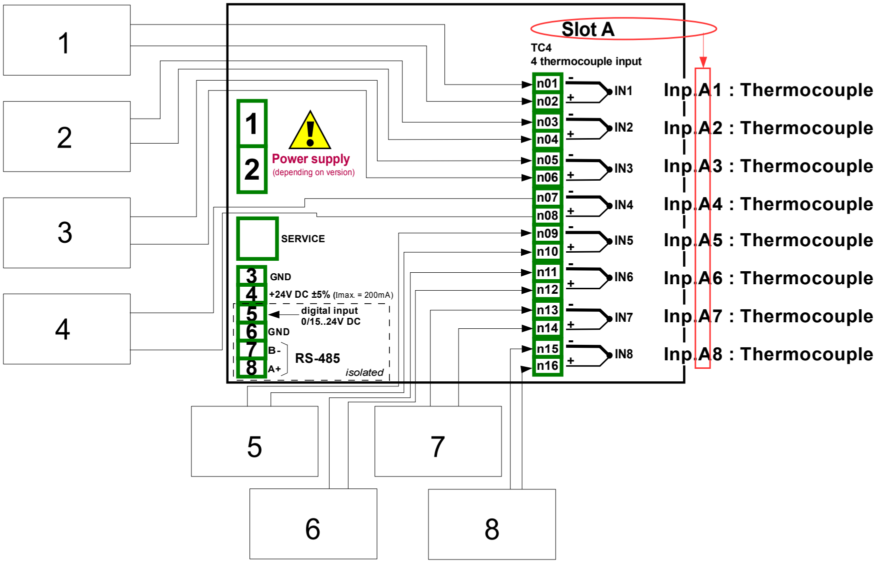


Fig. 7.70. Schematic diagram for the ***TC8*** *module*

To configure logical channels for reading temperatures from 1÷8 inputs (names **Temperature 1**, **Temperature 2**, etc.) should follow the same way as in **Example 7.8.11.2**.

For Logical channel in **Math function** mode calculated an average value:

- touch screen and press the **Menu** button,
- press the **Device configuration** button,
- enter **Logical channels** menu,
- using the arrows in the top navigation bar, select any **logical channel** (except 1÷8) such as **9**,
- in Name parameter write **Mean value**,
- Mode parameter set as **Math function**,
- enter **Function** submenu to select the appropriate function that allows the calculation of the mean value,
    - **Function** parameter set as **Mean X[i]**,

  - **Source X** parameter select **logical channels 1÷8**, which was defined as a reading of the thermocouples temperature,
  - **X error handling** parameter set as **errors forwarded to result**,
- for **Displaying** block parameters:
  - **Format** parameter set as **numeric**,
  - **Precision** parameter set as **0**,
  - in **Graph low** parameters write **-200°C**,
  - in **Graph high** parameter write **1370°C**,

The next step is to define Groups set up logical channels to view them in two windows on the screen. To do this:

- enter the **Group** menu, to enable the Group 1 if it is off,
- in **Name** write **Measure Temp. 1**,
- for **Channels** block parameters:
  - **Slot 1** parameter set as **logical channel 9** called the **Mean value**,
  - **Slot 2** parameter set as **logical channel 1** called the **Temperature 1**,
  - **Slot 3** parameter set as **logical channel 2** called the **Temperature 2**,
  - **Slot 4** parameter set as **logical channel 3** called the **Temperature 3**,
  - **Slot 5** parameter set as **logical channel 4** called the **Temperature 4**,
  - **Slot 6** parameter set as **logical channel 5** called the **Temperature 5**,
- all other parameters in this **Group** should be on default settings,

- using the arrows in the top navigation bar, select any **Group** (except 1) such as 2,
- in **Name** parameter write **Measure Temp. 2**,
- for **Channels** block parameters:
  - **Slot 1** parameter set as **logical channel 6** called the **Temperature 6**,
  - **Slot 2** parameter set as **logical channel 7** called the **Temperature 7**,
  - **Slot 3** parameter set as **logical channel 8** called the **Temperature 8**,
  - **Slot 4÷6** parameter set as **disabled**,
- all other parameters in this **Group** should be on default settings,

The last step is to define the **Initial view** displayed on the LCD screen when the device starts. Initial View has to display a defined group of logical channels in values mode. To do this:

- enter **General settings** menu,
- for **Initial view** block parameters,
  - **Display mode** parameter set as **values**,
  - **Displayed group** parameter set as **Group 1: Measure Temp. 1**,

Finally, you must exit the menu by pressing the button **Save changes**. The result of the changes will be visible after calling the first measurement. Use **[↓ GROUP]** or **[GROUP ↑]** button in the navigation bar to switch between display groups.

Example 7.8.11.7*: Application in* ***Math function*** *mode (see* ***Chapter 7.8.6. Logical Channels - Math function mode****).*

Task:
The task is to create a logical channel with the function returns a value of logical channel (**X [i]**) selected by the other logical channel (**Y**).
Suppose we have defined four logical channels 1, 5, 8, 12 in **Set point** mode. However, in the logical channel 2 will be the value derived from the temperature sensor. In the logical channel 3 create math function **X [i] selected by the Y**. Using the value of channel 2 we select value from the channels 1, 5, 8, 12.
The result of this function is placed in **Tab. 7.4**:

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. 7.4 Example of configure function ***X [i] selected by the Y***

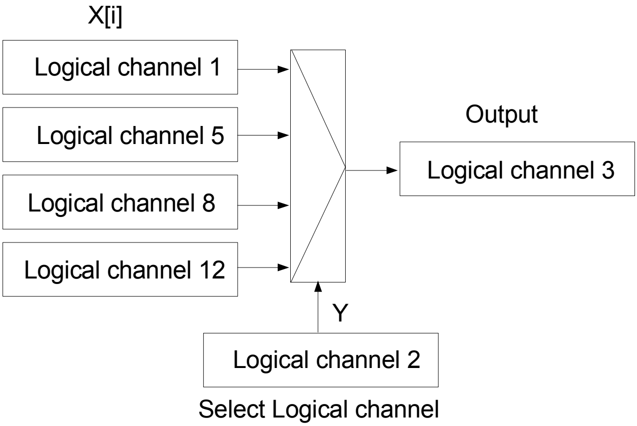


Fig. 7.71. The block diagram of a mathematical function ***X [i] selected by Y***

Solution:
We define a logical channels in **Set point value** mode:

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

- using the arrows in the top navigation bar, select **logical channel 1**,
- in **Name** parameter write **Function**,
- **Mode** parameter set as **Math function**,
- enter **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 a 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**,

***Example 7.8.11.8**: Application of **Logical channel** in the **Controller** mode (see **Chapter 7.8.7. Logical Channels - Controller mode**).*

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

Solution:
Before measuring and regulating the temperature, user should configure the device as follows and then connect the thermocouple sensor and Current converter to the device. An example of the connection is shown in **Fig. 7.72**.

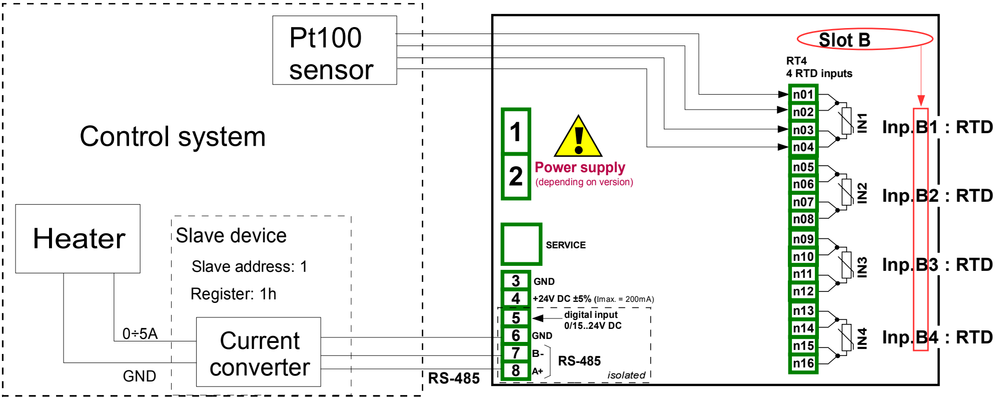


Fig. 7.72. Schematic diagram for the RT4 module and MB1 Modbus port

The first step is to configure appropriate parameters for the **Controllers** menu according to **Example 7.13.2.1**. Next, is to configure logical channels as follows:

For demanding object temperature value:

- touch screen and press the **Menu** button,
- press the **Device configuration** button,
- enter **Logical channels** menu,
- using the arrows in the top navigation bar, select any **logical channel** such as 1,
- in **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 PT100 sensor,

- touch screen and press the **Menu** button,
- press the **Device configuration** button,
- enter **Logical channels** menu,
- using the arrows in the top navigation bar, select any **logical channel** such as 2,
- in **Name** parameter write **Feedback**,
- Mode parameter set as Hardware input,
- **Source** parameters set as **Inp.A1 : RTD** (see **Fig. 7.72**),
- enter **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 Controller mode:

- touch screen and press the **Menu** button,
- press the **Device configuration** button,
- enter **Logical channels** menu,
- using the arrows in the top navigation bar, select any **logical channel** such as 32,
- in **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 signal from controller to Current converter (see **Fig. 7.72**),
  - in **Graph low** write **0**,
  - in **Graph high** write **10**,

For external output configuration (sending via RS-485) see **Chapter 7.11.4. Examples of external output configurations**.

***Example 7.8.11.9**: Application of Logical channel in the Profile/timer mode (see **Chapter 7.12. PROFILES/TIMERS**).*

Task:
Create the Profile/timer in the logical channel 1 in the edge (once) triggering mode. Trigger source is logical channel 2 which is connected to current input from UI4 module. 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. Idle value is 0 and looping is disabled.

Solution:
First configure the device as follows. Then connect current converter to the UI4 module as it is shown in **Fig. 8.37** (see also **Appendix 8.2. UI4, UI8, UI12, U16, U24, I16, I24 – VOLTAGE and CURRENT MEASUREMENT MODULES**).

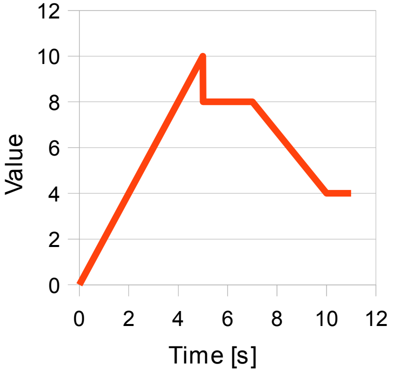


Fig. 7.73. Example of Profile waveform

For logical channel in Profile/timer mode:

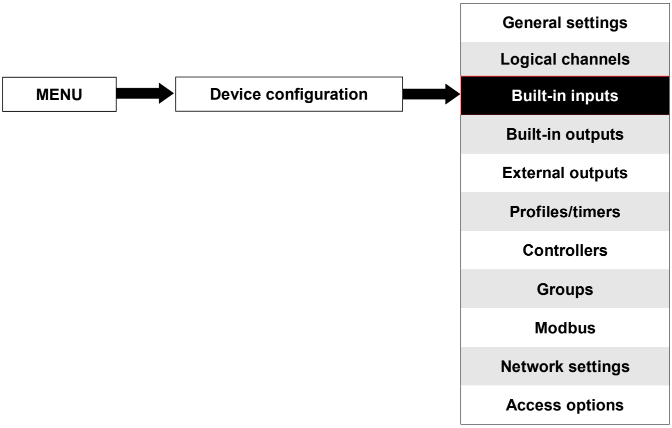
- touch screen and press the **Menu** button,
- press the **Device configuration** button,
- enter **Logical channels** menu,
- using the arrows in the top navigation bar, select any **logical channel** such as 1,
- in **Name** parameter write **Profile**,
- **Mode** parameter set as **Profile/timer**,
- **Source** parameter set as **P/T1:"Profile 1"**,
- press **Configure source** button,
  - **Mode** parameter set as **edge (once)**,
  - **Triggering source** parameter set as **Log.ch.2**,
  - in **Idle value** parameter write **0**,
  - press **Section list** button,
    - press green button on 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 **Final value** parameter write **10**,

        - press green button on the bottom left side of the screen,
        - **Section num.** parameter should be set as **2**,
            - in **Duration** parameter write **2**,
            - **Unit** parameter set as **second**,
            - **Shape** parameter set as **const.value**,
            - in **Final value** parameter write **8**,
        - press green button on the bottom left side of the screen,
        - **Section num.** parameter should be set as **3**,
            - in **Duration** parameter write **3**,
            - **Unit** parameter set as **second**,
            - **Shape** parameter set as **slope**,
            - in **Final value** parameter write **4**,
        - press green button on the bottom left side of the screen,
        - **Section num.** parameter should be set as **4**,
            - in **Duration** parameter write **1**,
            - **Unit** parameter set as **second**,
            - **Shape** parameter set as **const.value**,
            - in **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 **Hardware input** mode for **current input** is shown in **Example 7.8.11.1** (for this mode we configure logical channel 2 in this task).

Second method is shown in **Example 7.12.4.1**.

7.9. BUILT-IN INPUTS



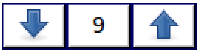
The **Built-in inputs** menu is directly related to the available inputs installed in the device. The basic version includes inputs:

- built-in **Digital input** is always marked as **Inp.X2:Digital 24V** - more about it see **Chapter 7.9.3. 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 7.9.4. Built-in inputs - Demo input numbered X3, X4, X5**,

Depending on customer's needs, input modules (description of available input modules is provided in **Appendix 8. APPENDICES** and the producer's website) can be installed in respective slots A, B or C (location of slot see **Fig. 4.8**).

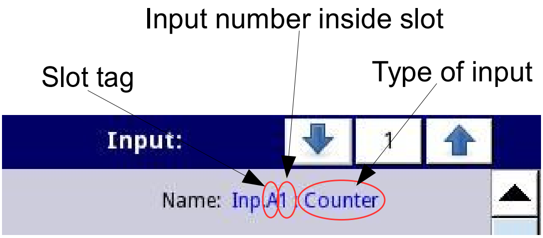
7.9.1. Built-in inputs - General settings

Available in the device in the basic configuration is **4 Built-in inputs** (**Digital input** and **3 Demo inputs**) and input modules 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 **Hardware input** mode in order to visualize the result, or use for further processing.



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.

Common parameter for all **Built-in inputs** is **Name** – each inputs has its own unique name and it can not be changed – description of this parameter see **Fig. 7.74**.



Name: Inp.A1 Counter

*Fig. 7.74. Description of **Name** parameter in **Built-in inputs** menu*

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

List of installed built-in input modules (slot tag and type of module) is in the **Device information** menu in **Hardware configuration** block (see **Fig. 7.25**),

In the device there are two ways to configure the hardware input:

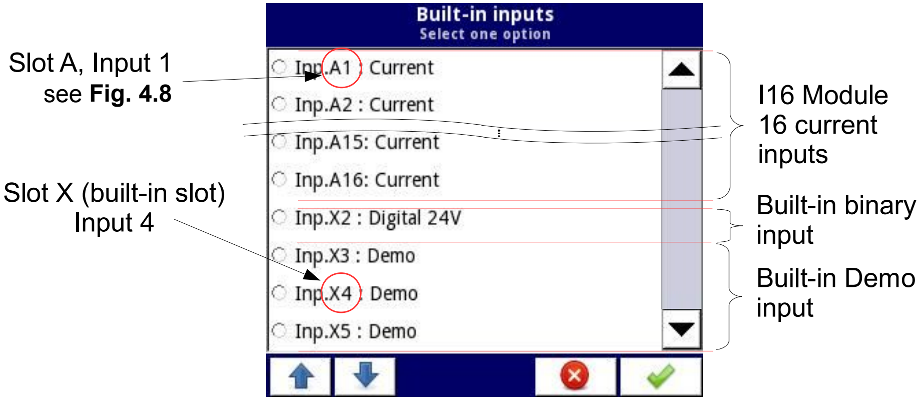
- using the **Configure source** button in the **Logical channels** menu in **Hardware input** mode,
- directly using the **Built-in inputs** menu.

The **Built-in inputs** can be (in the same order as list in the device - see **Fig. 7.25**):

a) installed **input modules** in the appropriate slots A, B or C (see **Fig. 4.8**) - the list of currently available modules is on the website,

b) built-in **digital input** is always designated as **Inp.X2: Digital 24V** (see **Chapter 7.9.3. Built-in inputs - Binary input Inp.X2 : Digital 24V**),

c) built-in **Demo inputs** numbered **X3**, **X4**, **X5**, (see **Chapter 7.9.4. Built-in inputs - Demo input numbered X3, X4, X5**),



○ Inp.A1 : Current
○ Inp.A2 : Current
○ Inp.A15: Current
○ Inp.A16: Current
○ Inp.X2 : Digital 24V
○ Inp.X3 : Demo
○ Inp.X4 : Demo
○ Inp.X5 : Demo

Fig. 7.75. The view of a sample list of available built-in inputs for a device

7.9.2. Built-in inputs - Input modules

Short description of configuration of the physical input is shown in **Fig. 4.9÷Fig. 4.21** and is dependent on specific measurement modules. In the **Built-in inputs** menu for the module the user can:

– change the ranges covered (depending on module), see **8. APPENDICES**,

– change the connection method - it depends on the module (see **Appendix 8. APPENDICES**), e.g. in the RTD module the user can select 2, 3 and 4-wire connections,

– change the type of reading of the input signal - depending on the module, e.g. a thermocouple module can read the temperature and voltage,

– change operation of the module - e.g. in counter module user can select function mode: add mode ("Function mode"="A+B"), subtract mode ("Function mode"="A-B"), quadrature mode 1 ("Function mode"="quad 1"), quadrature mode with increased resolution ("Function mode"="quad 4"),

Parameters common for built-in inputs modules:

– **Name** - each built-in input already has a name given by the device and user cannot change it, description of the **Name** parameter see **Fig. 7.74**,

– **Unit** – it is related to channel source data, for built-in modules default unit appears automatically,

– **Mode** – sets type of measurement; depending on module it sets measurement range or way of measurement,

– **Low Limit** – above this value in logical channel, low alarm state appears, displayed as **Lo**,

– **High Limit** – above this value in logical channel, high alarm state appears, displayed as **Hi**,

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

7.9.3. 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 digital input are included in **Chapter 3. TECHNICAL DATA**. This digital input has levels:

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

*Tab. 7.5 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 user cannot change it, for description of **Name** parameter see **Fig. 7.74**,

- **Filter time** – in which 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 value in Filter time parameter will be ignored. Filtration can be used if:
  - contact bounce occurs when switching,
  - user deliberately want to reduce the maximum frequency of the input.

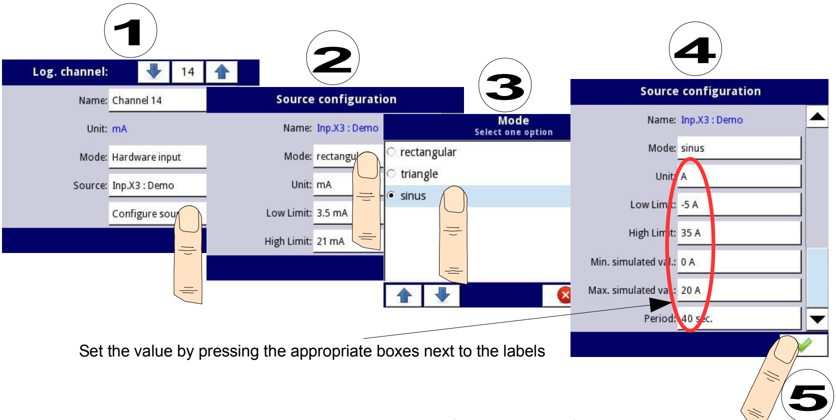
7.9.4. Built-in inputs - Demo input numbered X3, X4, X5

The device has 3 built-in simulation **Demo** inputs which can be defined by the user. The configuration contains parameters:

- **Mode** – defines shape of 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** – lowest generated signal level,
- **Maximum simulated value** – highest generated signal level,
- **Period** - duration (in seconds) of one cycle of the Demo signal selected in Mode parameter,
- **High state time** - this parameter is displayed only for the **rectangle** mode; it defines for how long (in seconds) generated signal will be in **Maximum simulated value** state,
- **Rise time** - this parameter is displayed only for the **triangle** mode; it defines how long (in seconds) generated signal will be rising from **Minimum simulated value** to **Maximum simulated value**,

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

Example configuration of a **Demo** for sinus mode shown in **Fig. 7.76**.



1. Log. channel: 14

Name: Channel 14
Unit: mA
Mode: Hardware input
Source: Inp.X3 : Demo
Configure sou

2. Source configuration

Name: Inp.X3 : Demo
Mode: rectangul
Unit: mA
Low Limit: 3.5 mA
High Limit: 21 mA

3. Mode
Select one option

○ rectangular
○ triangle
● sinus

4. Source configuration

Name: Inp.X3 : Demo
Mode: sinus
Unit: A
Low Limit: -5 A
High Limit: 35 A
Min. simulated val.: 0 A
Max. simulated val.: 20 A
Period: 40 sec.

Set the value by pressing the appropriate boxes next to the labels

Fig. 7.76. Example configuration of ***Demo***

7.9.5. Built-in inputs – Modules

7.9.5.1. Voltage and current measurement modules

UI modules are equipped in eight, sixteen or twenty four voltage or current inputs, but there are also modules where half of their inputs are voltage and other half are current, see **Appendix 8.2. UI4, UI8, UI12, U16, U24, I16, I24 – VOLTAGE and CURRENT MEASUREMENT MODULES**. To make sensor connection easier, inputs are grouped and all ground terminals are common but separated from power supply and other modules.

Voltage inputs 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. 7.74**.
- **Unit** – parameter which displays measurement unit - **V**,
- **Mode** – defines measurement range,
- **Low limit** – defines measurement level bellow which in logical channel will be displayed **Lo** state,
- **High limit** – defines measurement level above which in logical channel will be displayed **Hi** state,

Current inputs 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. 7.74**.
- **Unit** – parameter which displays measurement unit - **mA**,
- **Mode** – defines measurement range,
- **Low limit** – defines measurement level bellow which in logical channel will be displayed **Lo** state,
- **High limit** – defines measurement level above which in logical channel will be displayed **Hi** state,

7.9.5.2. Mixed UIN/UID modules

Mixed **UIN/UID** modules has sixteen or twenty two inputs which allows to measure current, voltage, resistance and temperature (using NTC sensors) and can be equipped with uninsulated digital inputs. Full hardware configuration description is in **Appendices 8.3. UI4N8, UI4D8, UI8N8, UI8D8 – MIXED UIN/UID MODULES**. To make sensor connection easier, inputs are grouped and all ground terminals are common but separated from power supply and other modules.

When NTC inputs are used, it is recommended to use shielded wires and connect shield to GND connector on the module.

Voltage inputs 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. 7.74**.
- **Unit** – parameter which displays measurement unit - **V**,
- **Mode** – defines measurement range,
- **Low limit** – defines measurement level bellow which in logical channel will be displayed **Lo** state,
- **High limit** – defines measurement level above which in logical channel will be displayed **Hi** state,

Current inputs 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. 7.74**,
- **Unit** – parameter which displays measurement unit - **mA**,
- **Mode** – defines measurement range,
- **Low limit** – defines measurement level bellow which in logical channel will be displayed **Lo** state,
- **High limit** – defines measurement level above which in logical channel will be displayed **Hi** state,

NTC (temperature or resistance measurement) inputs parameters are:
For temperature mode:

- **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. 7.74**,
- **Unit** – parameter which displays measurement unit - °**C**,
- **Mode** – defines measurement mode (temperature or resistance),
- **Low limit** – defines measurement level bellow which in logical channel will be displayed **Lo** state,
- **High limit** – defines measurement level above which in logical channel will be displayed **Hi** state,
- **NTC Equation** – allows to choose method of entering thermistor characteristic. There are two possibilities:
  - equation with Beta parameter, where:
    - A – resistance in reference temperature,
    - B – Beta parameter,
    - C – reference temperature in Centigrades,
  - equation based on Steinhart-Hart coefficients:

$$T=\frac{1}{A+B\cdot\ln(R)+C\cdot\ln(R)^3}$$

Measurement range which will be displayed is calculated based on equations above. In the second place **Low limit** and **High limit** parameters are taken into consideration when determining displayed range.

For resistance mode:

- **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. 7.74**,
- **Unit** – parameter which displays measurement unit - **Ω**,
- **Mode** – defines measurement mode (temperature or resistance),
- **Low limit** – defines measurement level bellow which in logical channel will be displayed **Lo** state,
- **High limit** – defines measurement level above which in logical channel will be displayed **Hi** state,

Digital inputs 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. 7.74**,
- **Mode** – defines input voltage ranges (TTL: „**0**" for 0÷0.8V; „**1**" for 2÷5,5V and HTL: „**0**" for 0÷4.2V; „**1**" for 11.5÷30V),

- **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,

7.9.5.3. Isolated current inputs module

IS modules are equipped in six isolated current inputs with different ground potentials, see **Appendix 8.4. IS6 – Isolated current inputs module**.

Current inputs 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. 7.74**.
- **Unit** – parameter which displays measurement unit - **mA**,
- **Mode** – defines measurement range,
- **Low limit** – defines measurement level bellow which in logical channel will be displayed **Lo** state,
- **High limit** – defines measurement level above which in logical channel will be displayed **Hi** state,

7.9.5.4. Analogue flowmeter modules

FI modules are equipped in two or four analogue flowmeters and same number of current inputs, see **Appendix 8.5. FI2. FI4, FT2, FT4 – FLOWMETER MODULES**. Analogue flowmeters are used to liquids, gases and bulk materials flow measurement, based on measured sensor current and calculating total flow in selected channel. Each flow channel is equipped in **IN[n]** input and common ground **GND**. All analogue channels are galvanically separated from the device.

The user can, thanks to **Scaling** parameter in Logical channel, see chapter **7.8.1. Logical Channels - General settings**, interpret measured current by the device at its discretion, which allows to cooperate with any kind of analogue sensors.

Current inputs are used to current measurement in typical ranges (0÷20 mA; 4÷20mA) and to facilitate their connection, they are grouped and they have common ground, but it is isolated from the supply voltage and other modules.

Analogue flowmeter 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. 7.74**.
- **Unit** – parameter which displays measurement unit - **mA**,
- **Base unit** – defines unit which is used by the totalizer (for example when flow measurement unit is m^3/s, the base unit is m^3),
- **Mode** – defines current range of selected input (0÷20mA, 4÷20mA) and the time base which is used during the measurement (pulses per second, pulser per minute, pulser per hour),
- **Low limit** – value which determines measured current, bellow which in logical channel will be displayed **Lo** state, but the actual flow value still will be used in balance calculation,
- **High limit** – value which determines measured current, above which in logical channel will be displayed **Hi** state, but the actual flow value still will be used in balance calculation,

- **Totalizer configuration** – submenu which contains totalizer settings,
  - **Reset now** – manual reset of the totalizer,
  - **Reset mode** – turns on and off reset from logical channel,
  - **Reset source** – allows to choose a logical channel from the list, which will reset the totalizer, when value in this channel will be greater than 0,
  - **Strobe mode** – turns on and off balance calculating from logical channel,
  - **Strobe source** – allows to choose logical channel from list, which will hold balance calculating in totalizer (this do not reset the totalizer and do not stops flow measurement); balance calculating is hold when value in this logical channel is equal or lower than zero,
  - **Counting direction** – turns on and off change of the counting direction,
  - **Direction source** – allow to choose a logical channel from the list, which will change the counting direction; counting direction will change when value in this channel will be greater than 0,

7.9.5.5. Pulse flowmeter modules

FT modules are equipped in two or four pulse flowmeters and same number of current inputs, see **Appendix 8.5. FI2. FI4, FT2, FT4 – FLOWMETER MODULES**. Pulse flowmeters are used to liquids, gases and bulk materials flow measurement, based on pulses from sensor and calculating total flow in selected channel. Each pulse channel is equipped in counting inputs: **Inp[n]1**, **Inp[n]2** and common ground **COM[n]**. All pulse channels are galvanically separated from the device and from themselves.

The user can, thanks to **Scaling** parameter in Logical channel, see chapter **7.8.1. Logical Channels - General settings**, interpret amount of pulses received by the device at its discretion, which allows to cooperate with any kind of pulse sensors.

Current inputs are used to current measurement in typical ranges (0÷20 mA; 4÷20mA) and to facilitate their connection, they are grouped and they have common ground, but it is isolated from the supply voltage and other modules.

Pulse flowmeter 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. 7.74**.
- **Unit** – parameter which displays the time base of the measurement,
- **Base unit** – defines unit which is used by the totalizer (for example when flow measurement unit is m^3/s, the base unit is m^3) ,
- **Mode** – defines the time base which is used during the measurement (pulses per second, pulser per minute, pulser per hour),
- **Low limit** – value which determines amount of pulses, bellow which in logical channel will be displayed Lo state, but the actual flow value still will be used in balance calculation,
- **High limit** – value which determines amount of pulses , above which in logical channel will be displayed Hi state, but the actual flow value still will be used in balance calculation,
- **Counting mode** – defines how the device should interpret signals on pulse inputs; it has following options:
  - **with direction input** – **Inp[n]1** is input which counts pulses and **Inp[n]2** sets the counting direction (low state - positive flow, counting up; high state - negative flow, counting down),
  - **quad 1** – **Inp[n]1** and **Inp[n]2** work as a quad inputs,
  - **quad 4** – **Inp[n]1** and **Inp[n]2** work as a quad inputs with increased resolution,

- **Filter** – defines maximal frequency (with 50% duty cycle) the device will measure,
  - **disable** – the device is measuring without the frequency limitation, however above 50kHz there can occurs errors in measurements and totalizer calculation,
  - **10 Hz**,
  - **25 Hz**,
  - **50 Hz**,
  - **100 Hz**,
  - **1 kHz**,
  - **5 kHz**,
  - **10 kHz**,
  - **50 kHz**,
- **Zero time** – maximal awaiting time for the next pulse. If this time is exceeded the device will display 0 value; it has following options:
  - **0.1 sec**.
  - **1 sec**.
  - **10 sec**.
- **Totalize configuration** – submenu which contains totalizer settings,
  - **Reset now** – manual reset of the totalizer,
  - **Reset mode** – turns on and off reset from logical channel,
  - **Reset source** – allows to choose a logical channel from the list, which will reset the totalizer, when value in this channel will be greater than 0,
  - **Strobe mode** – turns on and off balance calculating from logical channel,
  - **Strobe source** – allows to choose logical channel from list, which will hold balance calculating in totalizer (this do not reset the totalizer and do not stops flow measurement); balance calculating is hold when value in this logical channel is equal or lower than zero,
  - **Counting direction** – turns on and off change of the counting direction,
  - **Direction source** – allow to choose a logical channel from the list, which will change the counting direction; counting direction will change when value in this channel will be greater than 0,

7.9.5.6. Thermocouples sensor measurement modules

TC modules are equipped in four, eight or twelve thermocouple inputs and one input for connectors temperature measurement, see **Appendix 8.6. TC4, TC8, TC12 – THERMOCOUPLE SENSOR MEASUREMENT MODULES**. Primary destination of this module is temperature measurement using a thermocouple sensors, but it is also possible to measure voltage in typical thermocouple ranges.

TC modules 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. 7.74**.
- **Unit** – parameter which displays **°C** or **mV**, depending on **Mode** parameter settings,
- **Mode** – allows to set thermocouple which is used to measurements or voltage measurement range,
- **Low limit** – defines measurement level bellow which in logical channel will be displayed **Lo** state,
- **High limit** – defines measurement level above which in logical channel will be displayed **Hi** state,
- **Wire compensation** – menu which allows to compensate measurement errors, which can be caused by wrong sensor readings,

- **Compensation mode** – allows to choose parameter, which will be used in compensation process,
  - **disable** – compensation is not active,
  - **manual** – allows to compensate sensor constant offset,
    - **Compensation** – parameter which allows tu manually compensate sensor error, written here value will be added or subtracted from measured sensor value,
  - **automatic** – allows to compensate the difference between real temperature and that measured by the sensor,
    - **Actual temperature** – parameter in which user enters actual temperature near the sensor, which is measured by more reliable thermometer,
    - **Compensate** – button which activates the compensation; value which is written in Actual temperature parameter will be from now on used for compensation,
    - **Compensation** – read only parameter; displays voltage value calculated to compensate temperature measurement,

User should remember, that after changing the sensor, new settings in **Wire compensation** menu should be made.

Connectors temperature input 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. 7.74**.
- **Unit** – parameter which displays **°C**,
- **Mode** – parameter which displays "**Temp. of connectors**",
- **Low limit** – defines measurement level bellow which in logical channel will be displayed **Lo** state,
- **High limit** – defines measurement level above which in logical channel will be displayed **Hi** state,
- **Offset** – parameter which allows to add or subtract constant value to temperature measured by connectors temperature sensor,

7.9.5.7. RTD measurement modules

RT modules are equipped in four or six RTD inputs, see **Appendix 8.7. RT4 , RT6 – RTD MEASUREMENT MODULES**. These inputs are used for temperature measurement using RTD type temperature sensors (by two, three and four wire method) or for resistance measurement in ranges up to 3kΩ. RT modules cooperates with many kind of resistance thermometers and thanks to this, there is no need to change the module when sensor has been changed, user only needs to change settings in module.

RT modules 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. 7.74**.
- **Unit** – parameter which displays **°C** or **Ω**, depending on **Mode** parameter settings,
- **Mode** – defines which sensor is used for measurement, or what is the resistance measurement range and what is the method of these measurements (three wired method is assigned to separated option in Mode parameter e.g. Pt100 3-wire, while two and four wired method is assigned to different option in Mode parameter e.g. Pt100),

- **Low limit** – defines measurement level bellow which in logical channel will be displayed **Lo** state; for each sensor this limitation is different, but it can be changed in certain limits,
- **High limit** – defines measurement level above which in logical channel will be displayed **Hi** state; for each sensor this limitation is different, but it can be changed in certain limits,

7.9.5.8. Isolated universal inputs modules

UN modules are equipped in three or five universal inputs and one input for connectors temperature measurement, see **Appendix 8.8. UN3, UN5 – OPTOISOLATED UNIVERSAL INPUTS MODULES**. It is most advanced available module. With its help user can make many different kind of measurements in each channel. This module can measure: voltage, current, resistance, temperature using resistance sensors or thermocouples.

UN modules 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. 7.74**.
- **Unit** – parameter which displays **mA**, **V**, **°C, Ω** or **mV**, depending on **Mode** parameter settings,
- **Mode** – allows to set input type used to measurements,
- **Low limit** – defines measurement level bellow which in logical channel will be displayed **Lo** state,
- **High limit** – defines measurement level above which in logical channel will be displayed **Hi** state,
- **Wire compensation** – menu which allows to compensate measurement errors, which can be caused by wrong sensor readings,
  - **Compensation mode** – allows to choose parameter, which will be used in compensation process,
    - **disable** – compensation is not active,
    - **manual** – allows to compensate sensor constant offset,
      - **Compensation** – parameter which allows tu manually compensate sensor error, written here value will be added or subtracted from measured sensor value,
    - **automatic** – allows to compensate the difference between real temperature and that measured by the sensor,
      - **Actual temperature** – parameter in which user enters actual temperature near the sensor, which is measured by more reliable thermometer,
      - **Compensate** – button which activates the compensation; value which is written in **Actual temperature** parameter will be from now on used for compensation,
      - **Compensation** – read only parameter; displays voltage value calculated to compensate temperature measurement,

User should remember, that after changing the sensor, new settings in Wire compensation menu should be made.

Connectors temperature input 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. 7.74**.
- **Unit** – parameter which displays measurement unit - **°C**,
- **Mode** – parameter which displays **Temp. of connectors**,
- **Low limit** – defines measurement level bellow which in logical channel will be displayed **Lo** state,
- **High limit** – defines measurement level above which in logical channel will be displayed **Hi** state,
- **Offset** – parameter which allows to add or subtract constant value to temperature measured by connectors temperature sensor,

7.9.5.9. Optoisolated digital inputs modules

D modules are equipped with eight, sixteen or twenty four digital inputs, see **Appendix 8.9. D8, D16, D24 – OPTOISOLATED DIGITAL INPUTS MODULE**. This inputs can be used individually, but also in groups of four or all inputs available in module. Measurement results are presented in decimal system.

D modules 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. 7.74**.
- **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,

7.9.5.10. Optoisolated universal counters modules

CP modules can be equipped in two or four universal counters inputs (see **Appendix 8.10. CP2, CP4 – OPTOISOLATED UNIVERSAL COUNTERS MODULES**). Each counter can be configured individually and is equipped with two counting inputs: **Inp[n]1**, **Inp[n]2**, programming input **Prog[n]**, reset input **Res[n]**, where **n** means counter number from 1 to 4.

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. 7.74**,
- **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 both inputs 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,

7.9.5.11. Optoizolated hourmeters modules.

HM hourmeters modules developed for the MultiCon units. They allows to measure period of time between START and STOP signals, as well as sum of periods.

These modules are ideal solution to control working time of a machinery, duration of phenomena or for maintenance purposes. The HM2 and HM4 have 2 and 4 independent counters respectively. Each counter is equipped with 2 inputs - START/STOP and programmable, which can be set as asynchronous RESET, HOLD or used as independent digital input (see **Appendix 8.11. HM2, HM4 – OPTOIZOLATED HOURMETERs MODULES**).

With the HM module a Time Format functionality of the device is related (see **Chapter 7.8.1. Logical Channels - General settings**).

Time format is an advanced method of data presentation in format of time. User can easily get a desired format, by entering a "formatting string" composed of letters and colons, for example a string: "w:d:hh:mm" lets user to display data as number of weeks, days, hours and minutes, separated by colons. This mechanism allows also for dividing the result on two or three independent channels. The maximum displaying precision is 1/1000 of a second. Input data must be expressed in seconds.

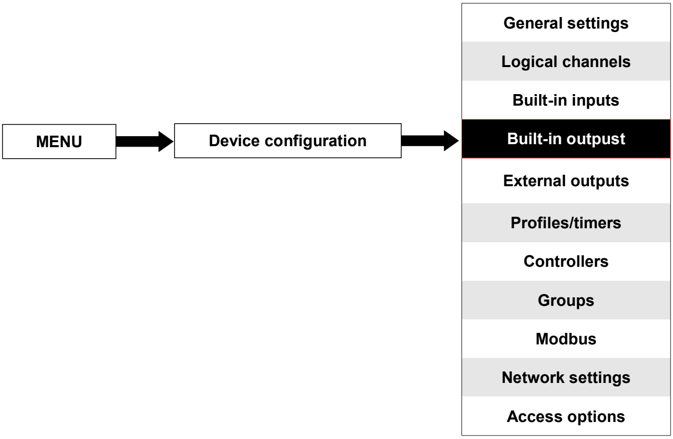
Hourmeters parameters are:

- **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. 7.74**,
- **Unit** – parameter which displays measurement unit: sec.,
- **Mode** – read only parameter, which displays measured physical quantity: **Time**,
- **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,
- **Satrt/Stop inp. trig.** - allows to define, when the device should start or stop counting,
  - **high level** – the device is counting time during high value on counting input,
  - **low level** – the device is counting time during low value on counting input,
  - **rising edge** – the device starts counting time when it detects rising edge on input signal. The device stops counting time when it detects another rising edge on this input.
  - **falling edge** – the device starts counting time when it detects falling edge on input signal. The device stops counting time when it detects another falling edge on this input.
- **Reset now** – button which allows user to manually reset the counter,
- **Reset mode** – it allows to choose 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 can be counter’s programmable input, but **Prog. Inp. mode** parameter has to be set as **reset** in order to work it properly,
  - **both** – both of options shown above are active, but **Prog. Inp. mode** parameter has to be set as **reset** in order to work it properly,
- **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 **PRG** input; it has following parameters:
  - **binary input** – signals on programmable input are not affecting counter's work, but they are measured and can be used and displayed by setting source for logical channels as one of five (for HM4) or three (for HM2) last inputs which module offers; for more details see bellow in this chapter,
  - **reset** – allows to asynchronously reset hourmeter, but **Reset mode** parameter has to be set as **external** or **both** in order to work it properly,
  - **inhibit** – allows to temporarily inhibit time counting,
- **Prog. inp. trig.** - allows to define when the device should reset or inhibit hourmeter (depending on **Prog. inp. mode** parameter settings),

  - **high level** – the device keeps counter in reset or inhibit state during high state on programmable input,
  - **low level** – the device keeps counter in reset or inhibit state during low state on programmable input,
  - **rising edge** – the device asynchronously resets the counter when rising edge occurs and **Prog. inp. mode** parameter is set as reset, or inhibits the counter when rising edge occurs and **Prog. inp. mode** parameter is set as inhibit, but in this case the counter will start counting again when on **Start/Stop** input new trigger occurs,
  - **falling edge** – the device asynchronously resets the counter when falling edge occurs and **Prog. inp. mode** parameter is set as reset, or inhibits the counter when falling edge occurs and **Prog. inp. mode** parameter is set as inhibit, but in this case the counter will start counting again when on **Start/Stop** input new trigger occurs,
- **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,

HM modules offers also possibility to measure programmable input state in binary format. To measure those inputs, user in **Logical channel**, in **Source** parameter should choose one of the last inputs which module offers. This measurement can be made regardless of the **Prog. Inp. mode** parfameter setting. In this case the parameters are identical to digital inputs modules, see **Chapter 7.9.5.9. Optoisolated digital inputs modules**.

7.10. BUILT-IN OUTPUTS



Build-in outputs menu is directly related to the available outputs installed in the device. The basic version includes outputs:

- built-in **Sound signal** output is always marked as **Out.X1: Sound signal** - more about the **Sound signal** output see **Chapter 7.10.2. Built-in Output - 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 7.10.2. Built-in Output - Relays, Sound signal, Virtual relays**,

Depending on customer's needs output modules (description of available output modules is provided in **Appendix 8. APPENDICES** and the producer's website) can be installed in respective slots A, B or C (location of slot see **Fig. 4.8**).

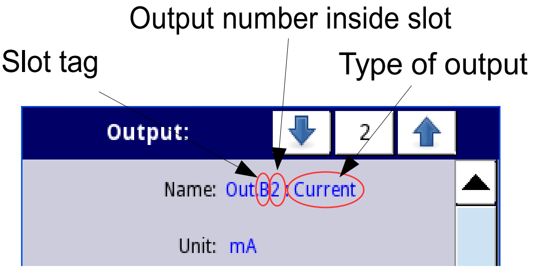
7.10.1. Built-in outputs - General settings

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

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

Parameters common for **Built-in outputs**:

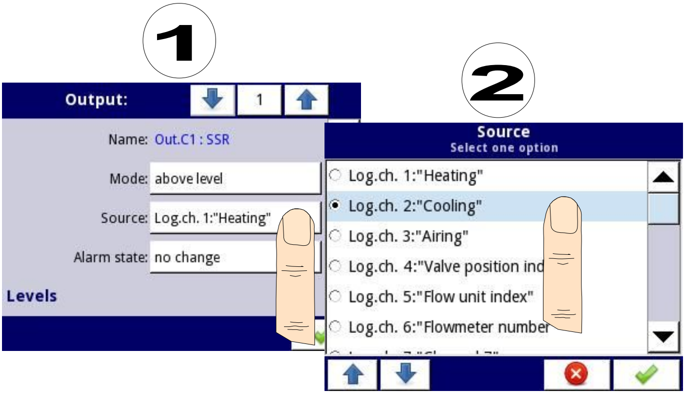
- **Name** - each output already has a name given by the device and user cannot change it, for description of **Name** parameter see **Fig. 7.77**,
- **Source** – this parameter contains a list of logical channels (up to 60), where the selected logical channel will be a data source for this built-in output (see **Fig. 7.78**).



Name: Out.B2 : Current

Unit: mA

*Fig. 7.77. Description of **Name** parameter in **Built-in outputs** menu*



Name: Out.C1 : SSR

Mode: above level

Source: Log.ch. 1:"Heating"

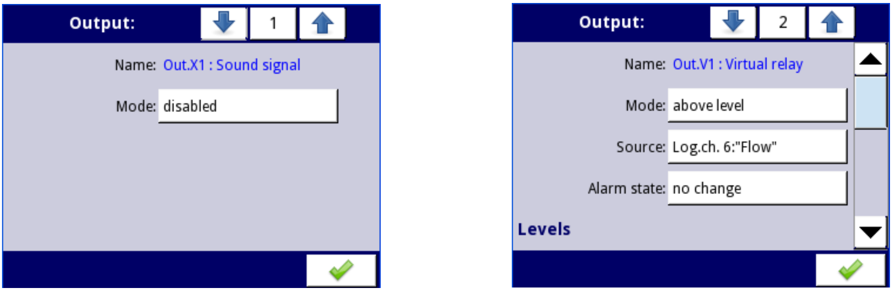
Alarm state: no change

Levels

○ Log.ch. 1:"Heating"
● Log.ch. 2:"Cooling"
○ Log.ch. 3:"Airing"
○ Log.ch. 4:"Valve position ind
○ Log.ch. 5:"Flow unit index"
○ Log.ch. 6:"Flowmeter number

Fig. 7.78. Data source settings for built-in output (for SSR output module)

List of build-in output modules (slot tag and type of module) is in **Hardware configuration** parameters block in **Device information** menu (see **Fig. 7.25** in **Chapter 7.4.**).



Name: Out.X1 : Sound signal

Mode: disabled

Name: Out.V1 : Virtual relay

Mode: above level

Source: Log.ch. 6:"Flow"

Alarm state: no change

Levels

Fig. 7.79. Main settings of disabled (left) and enabled (right) output

State of physical outputs can be used as source for **Logical Channels** (for details see **Chapter 7.8.3. Logical Channels - Hardware output monitor mode**)

7.10.2. Built-in Output - Relays, Sound signal, Virtual relays

The parameters of **Build-in outputs** for: Relays, Sound signal, Virtual relays are:

- **Name** - each output already has a name given by the device and user cannot change it, for description of **Name** parameter see **Fig. 7.77**,
- **Mode** - this parameter allows the user to select the method of operation of the output, **Mode** parameter has options (see **Fig. 7.80** and **Fig. 7.81**):
  - **disabled** - the built-in 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), otherwise the output is low state,
  - **below level** - the result is a high state when the input data (see **Source** parameter) is below the level (see **Level** parameter block), otherwise the output is low state,
  - **inside range** - the result is a high state when the input data (see **Source** parameter) will be within the range (see **Level** parameter block), otherwise the output is low state,
  - **outside range** - the result is a high state when the input data (see **Source** parameter) will be out of the range (see **Level** parameter block), otherwise the output is low state,
  - **PWM** - this option is visible only for SSR relay output type, PWM mode is discussed in **Chapter 7.10.3. Build-in output - PWM (Pulse-width modulation) mode for SSR relay output**,
- **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 **Fig. 7.78**).
- **Alarm state** – this parameter allows to choose type of output reaction in case of alarm state appearance; The **Alarm state** is when the value of Logical channel in which the data source for built-in output returns **-Err-** state or the state of the exceeding range: low **-Lo-** state and high **-Hi-** state; There are 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 alarm state the device immediately switches the output to low state,
  - **immediate ON** - means that in times of alarm state the device immediately switches the output to high state,
  - **timed OFF** - means that in times of alarm state the device switches the output to low state after time delay set in **Timing** parameter block,
  - **timed ON** - means that in times of alarm state the device switches the output to low state after time delay set in **Timing** parameter block,
  - for **PWM** mode in **SSR relay** module instead of **Alarm state** parameter is **Alarm level** parameter which allows the user to enter value at the output in times of alarm state (for more information see **Chapter 7.10.3. Build-in output - PWM (Pulse-width modulation) mode for SSR relay output**),
- **Levels** block parameter - these parameters allows the user to set range of changes of the output depending on the input signal, is discussed below in this **Chapter**,
- **Timing** parameter block - these parameters allows the user to set delay time change the output state and minimum duration of the output state, is 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'**.

The manufacturer discourages forcing relays state changes often than every second, because of their inertia. If user will do it anyway, the relay might not to react on signal changes.

Levels parameter block (see **Fig. 7.80**, **Fig. 7.81** and **Fig. 7.82**)

This parameters depends 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 constant signal level, exceeding which will cause output state change (for **Level mode**: **value**), or allows to choose logical channel which actual 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, high state on output appears,
  - **bellow level** – if **Source** value is lower than **Level** value, high state on output appears,
- **Lower level** and **Upper level** - these parameters define the range at which output state changes occurs (for **Level mode**: **value**), or they allows to choose a logical channels which actual values will be range for output state changes (for **Level mode**: **channel**); occurs for **Mode**:
  - **inside range** - if the input data is within the defined range at the output high state appears,
  - **outside range** - if the input data is outside the defined range at the output high state appears,
- **Hysteresis** - defining this parameter, the user can move the level for changes in output state (see **Fig. 7.80** and **Fig. 7.81**),
- **Alarm level** - this parameter is only visible for **PWM** mode, allows the user to enter value of fill of the pulse in times of alarm state (for more information see **Chapter 7.10.3. Build-in output - PWM (Pulse-width modulation) mode for SSR relay output**),

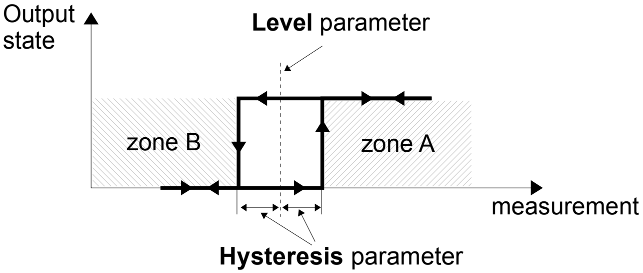


Fig. 7.80. One threshold control of the relay outputs

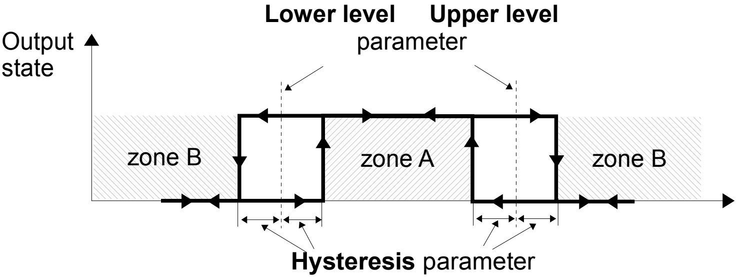
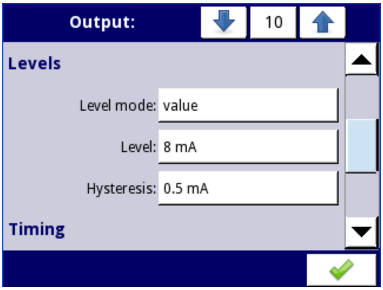


Fig. 7.81. Two threshold control of the relay outputs



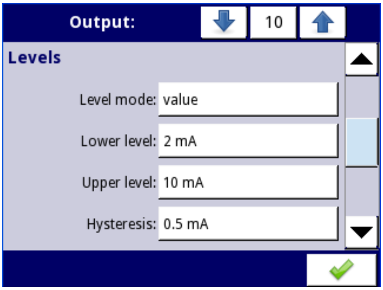
Levels

Level mode: value

Level: 8 mA

Hysteresis: 0.5 mA

Timing



Levels

Level mode: value

Lower level: 2 mA

Upper level: 10 mA

Hysteresis: 0.5 mA

*Fig. 7.82. **Levels** settings for **Mode**: **above** and **bellow level** (left) and **inside range** and **outside range** (right)*

Timing parameter block (see **Fig. 7.85**)
The parameters of this block include:

- **ON delay** - this parameter allows the setting of the time that must pass from the time of exceeds **Level** value until the output switch from low to high state (see **Fig. 7.83**),
- **OFF delay** - this parameter allows the user to set the time that must pass from the time of exceeds **Level** value until the output switch from high to low state (see **Fig. 7.83**),
- **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 **Fig. 7.84**,
- **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 **Fig. 7.84**,

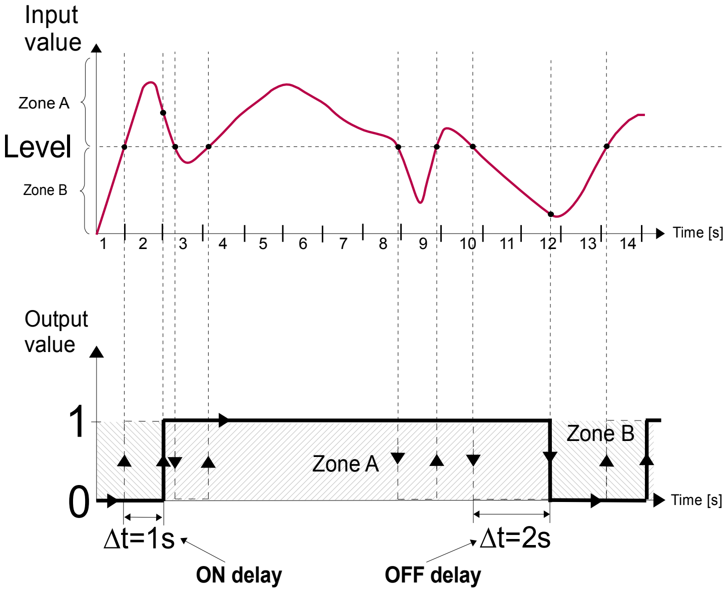


Fig. 7.83. Principle of relay output operation for sample timing settings: ***On delay****=1 sec.,* ***OFF delay****=2 sec.*

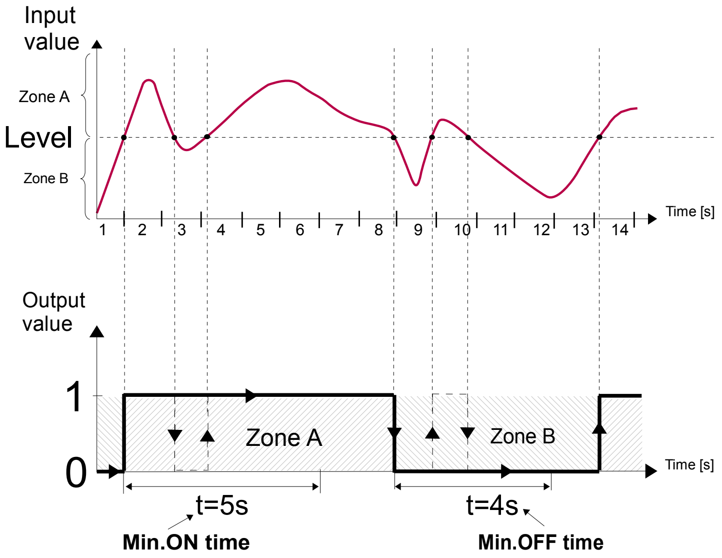
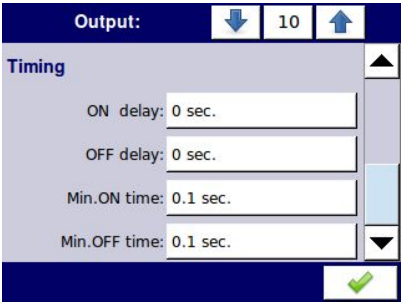


Fig. 7.84. Principle of relay output operation for sample timing settings: ***Min. ON time****=5 sec.,* ***Min. OFF time****=4 sec.*



ON delay: 0 sec.

OFF delay: 0 sec.

Min.ON time: 0.1 sec.

Min.OFF time: 0.1 sec.

Fig. 7.85. Timing settings for different modes

7.10.3. Build-in output - PWM (Pulse-width modulation) mode for SSR relay output

Technical specifications of SSR relay can be found in **Appendix 8.12. S8, S16, S24 - SOLID STATE RELAY DRIVERS MODULES**

Built-in outputs parameters in **PWM** mode are:

- **Name** – each outputs already has a name given by the device and user cannot change it - see **Fig. 7.77**,
- **Mode=PWM** – this parameter allows the user to select method of operation the output,
- **Source** – this parameter contains a **Logical channels** list, where the selected logical channel will be a data source for this built-in output (see **Fig. 7.78**),
- **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, is discussed below in this **Chapter**,
- **Timing** parameter block - these parameters allow the user to set the timing parameters of the output signal, is 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 **Fig. 7.86**),
- **Alarm level** - when the data source for the built-in output returns **Error** state or the range being exceeded: the low **-Lo-** state and high **-Hi-** state, **Alarm level** parameter for the PWM mode allows setting the duty cycle of the output signal in time of 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.

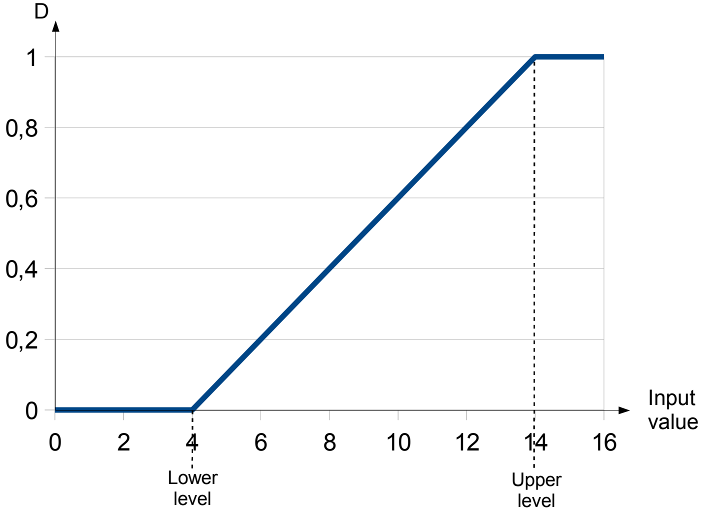


*Fig. 7.86. Normalized duration (**t_n**) of the output signal in PWM mode for parameters: **Lower level**=4, **Upper level**=14 (input signal lower than 4 has **t_n**=0, higher than 14 has **t_n**=1)*

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 high state (after switching to high state, switch back is possible after **Minimum On time**), see **Fig. 7.87**,
- **Minimum OFF time** - minimum duration of low state (after switching to low state, switch back is possible after **Minimum OFF time**), see **Fig. 7.87**,

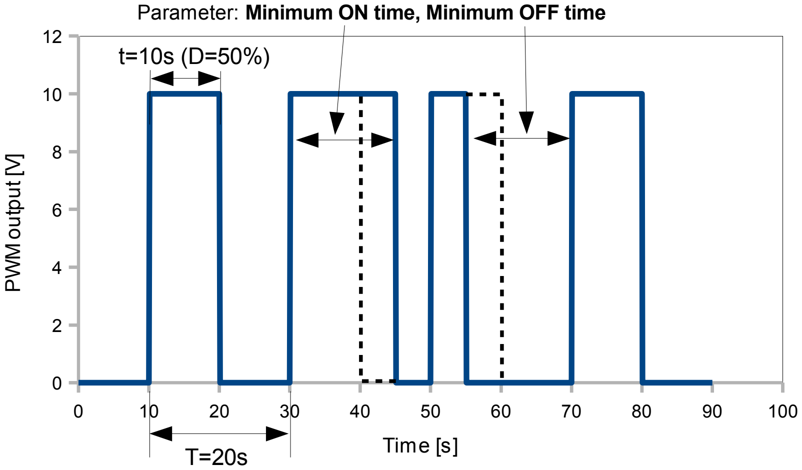
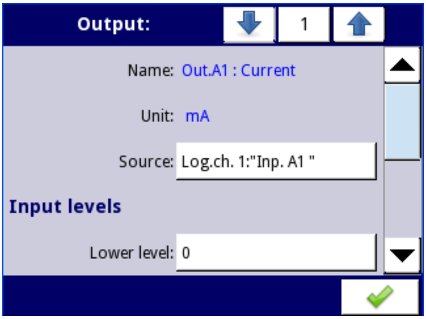


Fig. 7.87. Sample of output signal of SSR relay in PWM mode

7.10.4. Built-in output - Current output



Name: Out.A1 : Current

Unit: mA

Source: Log.ch. 1:"Inp. A1 "

Input levels

Lower level: 0

Fig. 7.88. Menu of the Passive current output

The parameters of built-in output for **Current outputs** module are:

- **Name** – each outputs already has a name given by the device and user cannot change it - see **Fig. 7.77**,
- **Unit** – defined permanently, it occurs only for 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 are calculated according to parameters gathered in **Input levels** and **Output levels** blocks,
- **Input levels** parameter block - determine the range of data source for this built-in output, this block has parameters:
  - **Lower level** and **Upper level** - these parameters limit the range of the input signal selected in **Source** parameter, below this range input signal will have **Lower level** value and above this range the signal will have **Upper level** value,

– **Output levels** parameter block - determine the range of output value, this block has the following parameters:
  - **Lower level** and **Upper level** - these parameters limit range of the output signal, below this range input signal will have **Lower level** value and above this range the signal will have **Upper level** value, relationship between Lower and Upper levels is linear and it is limited by defined range, see **Fig. 7.90**,
  - **Alarm level** – this parameter defines output value, when alarm state appears, but with maintaining hardware output limitation (**3÷22mA**). Alarm state is when 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__ describe the transfer equation (linear) - see **Fig. 7.89**. **Lower level** of the output defines the current which can be generated when the value of the input signal is equal to the **Lower level**. **Upper level** of the output defines the current which can be generated when the value of the input signal is equal to the **Upper level**.

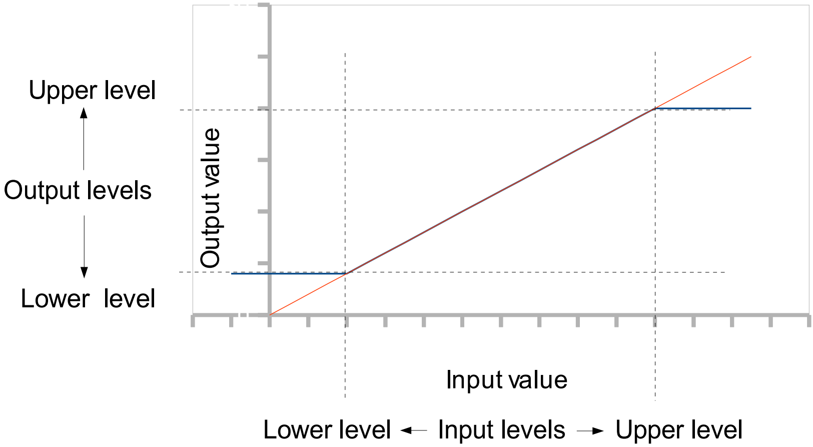
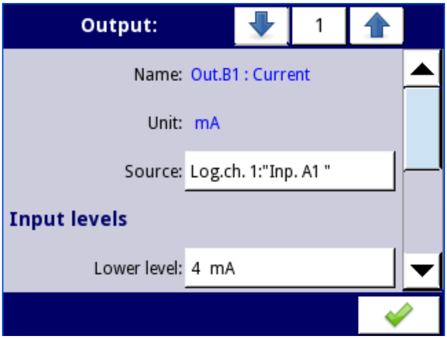


*Fig. 7.89. Input (red)-output (blue) characteristic of signal in **Current output** module*



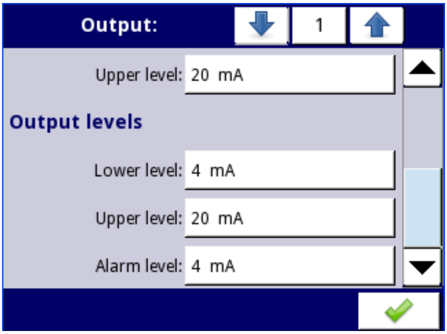
Name: Out.B1 : Current

Unit: mA

Source: Log.ch. 1:"Inp. A1 "

Input levels

Lower level: 4 mA



Upper level: 20 mA

Output levels

Lower level: 4 mA

Upper level: 20 mA

Alarm level: 4 mA

*Fig. 7.90. Parameters of the **Passive current** output*

7.10.5. Examples of build-in output configurations

Example 7.10.5.1: *Application of the output for **R45**, **R81** modules (see **Appendix 8.13. R45, R81, R65, R121 - RELAY MODULES**).*

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 **RT4** input module, and let the heater is controlled using an internal 5A relay (**R45** module).

Solution:
First configure the device as described below, then connect the object and power supply as shown on **Fig. 7.91**. See also **Appendix 8.7. RT4 , RT6 – RTD MEASUREMENT MODULES**.

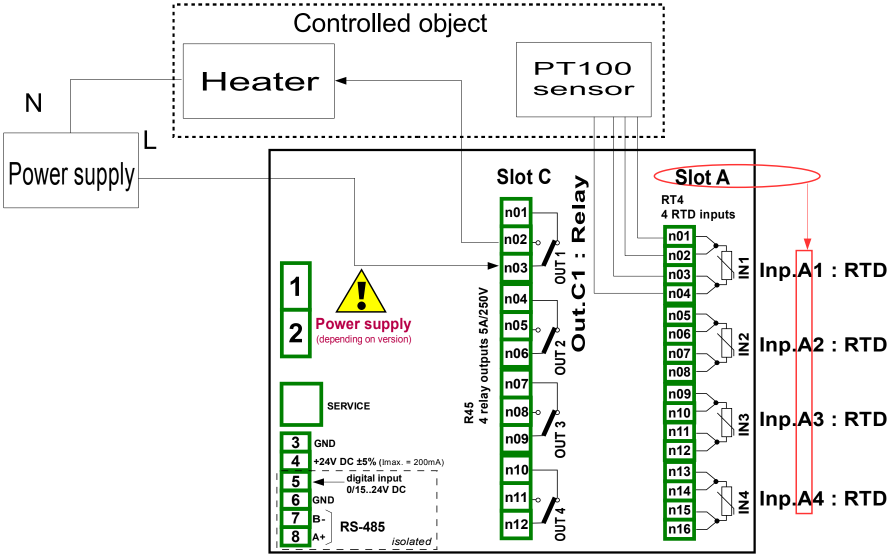


Fig. 7.91.Schematic diagram for the RTD input and Relay output modules

To control temperature in object shown on **Fig. 7.91**, it is necessary to configure one **Logical channel** and **Relay output**.

For reading PT100 sensor temperature:

- touch 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 **Name** parameter write **Feedback**,
- **Mode** parameter set as **Hardware input**,
- **Source** parameter set as **Inp.A1 : RTD** (see **Fig. 7.91**),
- enter **Configure source** submenu:
    - **Mode** parameter set as **Pt100**,

  - in **Low limit** parameter write **-100°C**,
  - in **High limit** parameter write **600°C,**
- for **Displaying** parameter bloc:
  - **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 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 heater. For this example it is the output named Out.C1 : Relay (see **Fig. 7.91**),
- **Mode** parameter set as **below level**, because we want the relay to switch on when the temperature will be 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 other parameters should be on default settings,

***Example 7.10.5.2**: Application of output for **IO** modules (see **Appendix 8.14. IO2, IO4, IO6, IO8 – PASSIVE CURRENT OUTPUT**).*

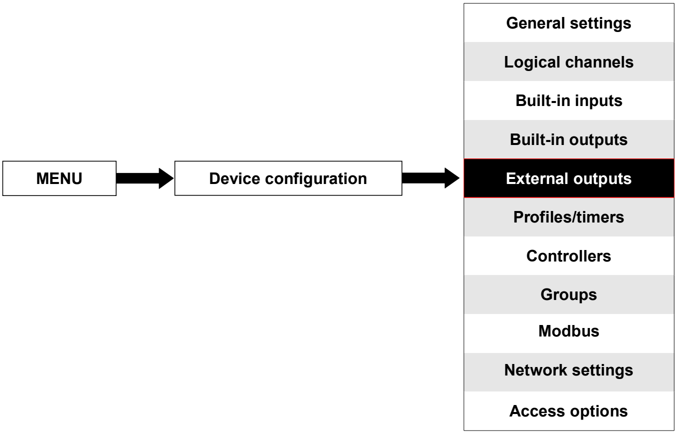
Task:
Assume that Logical channel 1 indicates pressure in range 100÷500 bars, and its result should be regenerated to current output in corresponding range 4÷20mA.

Solution:
For Current output configuration:

- touch 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 demanded range, e.g. **Out.C1 : Current**,
- **Source** parameter set as **Logical channel 1**, which returns pressure value in range 100÷500 bars.
- for **Input levels** parameter block:
    - in **Lower level** write **100 bars**,
    - in **Upper level** write **500 bars**,
- 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 Logical channel 1 will be in alarm state (**Err**, **Hi** or **Lo**), then on current output will be 4 mA,

User should remember that Current output is passive, which means it needs to be powered from current loop. **Fig. 8.29** Is showing how to do it. Output polarization in IO modules is irrelevant.

7.11. EXTERNAL OUTPUTS



The **External outputs** menu is related to sending the date to SLAVE device using Modbus communication protocol. In this menu it is determined what data will be send to SLAVE device while the configuration of Modbus in Master mode (for example baud rate, define the SLAVE device, active output register list and etc.) is defined in the **Modbus** menu (see **Chapter 7.15.3. Modbus - MASTER mode**).

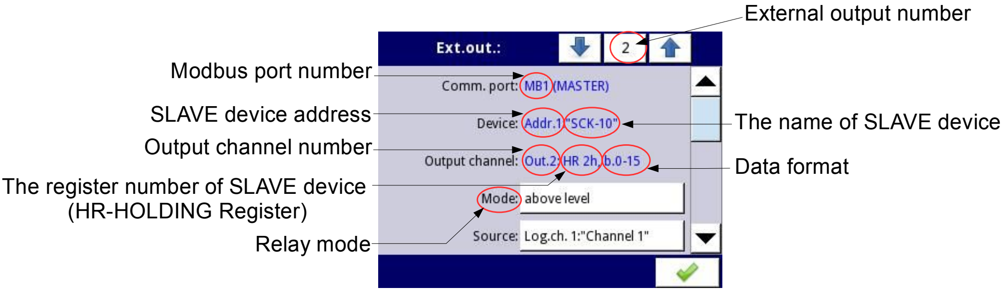
7.11.1. External outputs - General settings

In the device there are as many external outputs as will be defined in the **Modbus** menu (see **Chapter 7.15.3. Modbus - MASTER mode**). In case when the external outputs are not defined or inactive than in **External outputs** menu is an empty list.

External outputs can have a control type (control type setting, see **Chapter 7.15.3.2. Modbus MASTER - Device channels parameter block**):

- **as a relay**,
- **as a linear output**,

View of **External outputs** menu is created for two types of control: as a relay (digital output) and a linear output (analog output) is shown in **Fig. 7.92** and **Fig. 7.93**, respectively.


Ext.out.: 2

Comm. port: MB1 (MASTER)

Device: Addr.1 "SCK-10"

Output channel: Out.2 HR 2h, b.0-15

Mode: above level

Source: Log.ch. 1:"Channel 1"

*Fig. 7.92. View of **External outputs** menu for 'as a relay' type control*

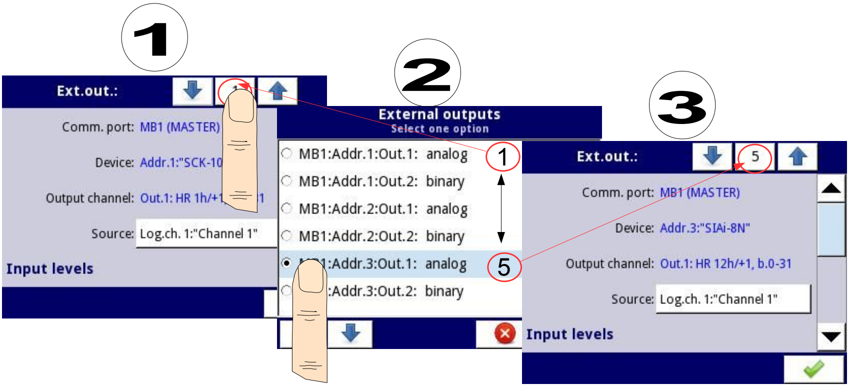

1
2
3
Ext.out.:
Comm. port: MB1 (MASTER)
Device: Addr.1:"SCK-10
Output channel: Out.1: HR 1h/+1
Source: Log.ch. 1:"Channel 1"
Input levels
External outputs
Select one option
○ MB1:Addr.1:Out.1: analog
○ MB1:Addr.1:Out.2: binary
○ MB1:Addr.2:Out.1: analog
○ MB1:Addr.2:Out.2: binary
● :Addr.3:Out.1: analog
○ :Addr.3:Out.2: binary
1
5
Ext.out.:
5
Comm. port: MB1 (MASTER)
Device: Addr.3:"SIAi-8N"
Output channel: Out.1: HR 12h/+1, b.0-31
Source: Log.ch. 1:"Channel 1"
Input levels

*Fig. 7.93. View of **External outputs** menu for 'as a linear output' type control*

Arrows placed in the upper right corner of the screen let you switch between a succession of external outputs. The middle button allows direct selection of a specific external output from the list.

The parameters that are common for the **External** outputs (see **Fig. 7.92** and **Fig. 7.93**) are:

- **Communication port** - this parameter is read only in **External outputs** menu, and indicates the Modbus port number, a description of Modbus ports configuration and view of the Modbus port connectors is located in **Chapter 7.15. MODBUS**,
- **Device** - this parameter is read only in **External outputs** menu, it shows address and name of the SLAVE device configured in Modbus menu (**Chapter 7.15.3. Modbus - MASTER mode**),
- **Output channel** - this parameter is read only in **External outputs** menu, it shows output channel number, type of register and data format configured in 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 external output from logical channel list (see **Fig. 7.94**),

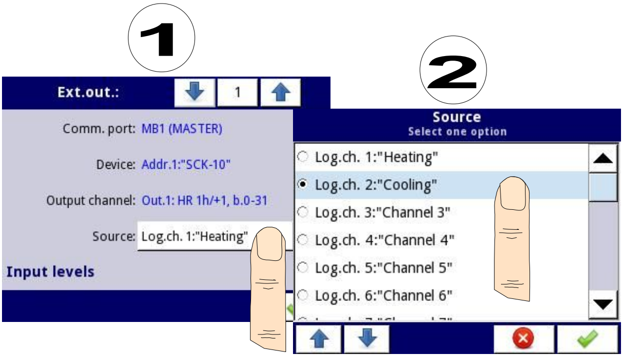


*Fig. 7.94. Sample selection of **Source** for External output*

7.11.2. External outputs - Control type: as a relay

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

The parameters of **External outputs** in the Control type **as a relay** are (see **Fig. 7.92**):

- **Comm.port** - this parameter is read only in **External outputs** menu, it is the Modbus port number for which the Modbus external output is configured; a description of Modbus ports configuration and indication of the Modbus port connectors is located in **Chapter 7.15. MODBUS**,
- **Device** - this parameter is read only in **External outputs** menu, it shows address and name of the SLAVE device configured in **Modbus** menu (**Chapter 7.15.3. Modbus - MASTER mode**),
- **Output channel** - this parameter is read only in **External outputs** menu, it shows output channel number, type of register and data format configured in 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 source signal (**Source** parameter); it has options (see **Fig. 7.80** and **Fig. 7.81**):
  - **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) will be within the range (see **Level** parameter block),
  - **outside range** - the result is a high state when the input data (see **Source** parameter) will be out of the range (see **Level** parameter block),
- **Source** – this parameter contains Logical channels list where the selected logical channel will be data source for the external output,
- **Alarm state** – it allows the user to choose external output reaction on alarm state. It occurs when the logical channel which is source for the external output, returns **Err**, **Hi** or **Lo** value. This parameter has 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 alarm state the device immediately switches the output to low state,
  - **immed. ON** - means that in times of alarm state the device immediately switches the output to high state,
  - **timed OFF** - means that in times of alarm state the device switches the output to low state after time delay set in **Timing** parameter block,
  - **timed ON** - means that in times of alarm state the device switches the output to low state after time delay set in **Timing** parameter block,
- **Levels** block parameter - these parameters allow the user to set range of changes of the output depending on the input signal, is discussed below in this **Chapter**,
- **Timing** parameter block - these parameters allow the user to set delay time change the output state and minimum duration of the output state, is discussed below in this **Chapter**,

Levels parameter block (see **Fig. 7.80**, **Fig. 7.81** and **Fig. 7.82**)

This parameters depends 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 high state at the output,
  - **below level** - below the level we get 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 high state,
  - **outside range** - if the input data is outside the defined range at the output we get high state,
- **Hysteresis** – this parameter defines offset for switch state on the output,

Timing parameter block (see **Fig. 7.85**)

The parameters of this block include:

- **ON delay** – it is the time that must pass from the time of switch request to actual switch from low state to high state (see **Fig. 7.83**),
- **OFF delay** - it is the time that must pass from the time of switch request to actual switch from high state to low state (see **Fig. 7.83**),
- **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. 7.84**,
- **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. 7.84**,

Note! If external output is active (see the parameter **Output active=yes** in the **Chapter 7.15.3.2. Modbus MASTER - Device channels parameter block**) for **Mode=disabled** the device send to Slave device value **'0'**.

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

The parameters of **External outputs** in the type of control **as a linear output** are (see **Fig. 7.93**):

- **Comm.port** - this parameter is read only in **External outputs** menu, it is the Modbus port number for which the Modbus external output is configured; a description of Modbus ports configuration and indication of the Modbus port connectors is located in **Chapter 7.15. MODBUS**,
- **Device** - this parameter is read only in **External outputs** menu, it shows address and name of the SLAVE device configured in **Modbus** menu (**Chapter 7.15.3. Modbus - MASTER mode**),
- **Output channel** - this parameter is read only in **External outputs** menu, it shows output channel number, type of register and data format configured in Modbus menu (each SLAVE device as defined in a specific address, has its output list individually numbered),
- **Source** - this parameter contains Logical channels list where the selected logical channel will be data source for the external output. Data from this source are calculated according to parameters in **Input levels** block and **Output levels** block,

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

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

User should remember that the relationship between input and output levels are described by linear transmission equation, and it is limited by the range, (see **Fig. 7.95**).

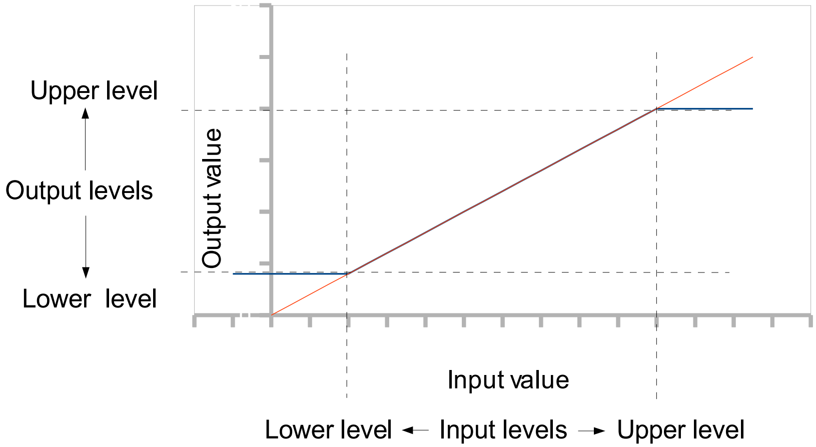


*Fig. 7.95. The relation between the input and output for **External output***

Note! If external output is active (see the parameter **Output active=yes** in the **Chapter 7.15.3.2. Modbus MASTER - Device channels parameter block**) for undefined **Source** parameter (e.g. a Logical channel in the **Mode=disabled**) or for undefined **Input** and **Output levels** parameter block (every parameter has value 0) the **CONTROLLER/DATA RECORDER** send to Slave device value **'0'**.

7.11.4. Examples of external output configurations

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

***Example 7.11.4.1**: Application of external output for protocol **Modbus** in the **MASTER** mode.*

Task:
The task is to send data located in Logical channel 1 to SLAVE device (address 5) without scaling and restrictions.

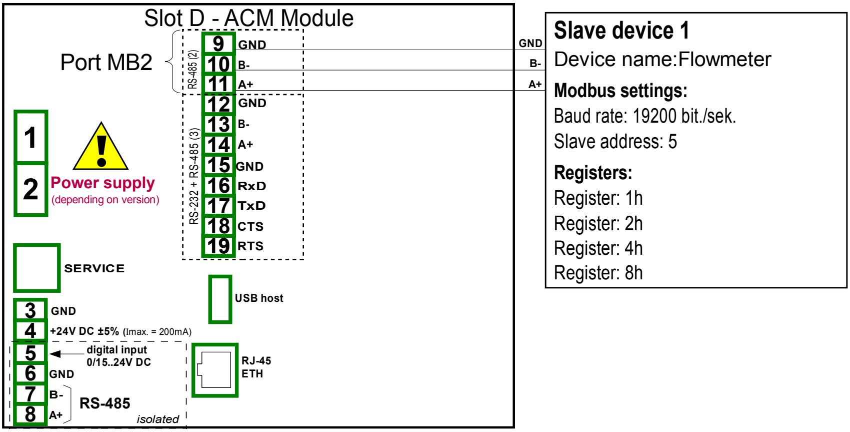


Fig. 7.96. Connection diagram for Modbus MB2

Solution:
First we need to configure settings in Modbus menu as it is shown in **Example 7.15.4.2**. Next:

We configure logical channels used to set values, which will be send by RS-485 to the output. To do this:

- touch 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 **Name** parameter write **Set point value**,
- **Mode** parameter set as **Set point value**,
- in **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 other parameters should be on default settings,

We define external outputs. To do this:

- touch 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 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 as **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 relay output),
- there are three constant parameters, which are consistent **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 value will be greater 0,
  - **Source** parameter set as logical channel namer **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 protection from fast switching on and off the valve,
- using the arrows in the top navigation bar, select **Ext.out.:3**, (it is relay output),
- there are three constant parameters, which are consistent **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 **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 other parameters should be on default settings,

In the next step you need to define Group of configured logical channels to display them in one window on the device screen . To do this:

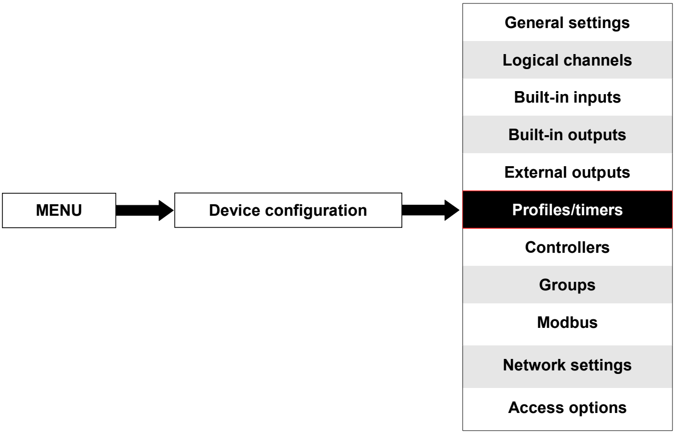
- touch screen and press the **Menu** button,
- press the **Device configuration** button,
- enter the **Groups** menu,
- in **Name** parameter write **Flowmeter**,
- for **Channels** block parameters:
  - **Slot 1** parameter set as **logical channel 1** named **Set point value**,
  - **Slot 2** parameter set as **logical channel 2** named **Valve position**,
  - **Slot 3** parameter set as **logical channel 3** named **Alarm state**,
  - all other parameters – **Slot 4÷6** set as disabled, because we want to display only three logical channels on the screen,
- all 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. Initial view has to display a defined Group 1 in values mode. To do this:

- enter **General settings** menu,
- for **Initial view** block parameters:
  - **Display mode** parameter set as **values**,
  - **Displayed group** parameter set as **Group 1: “Flowmeter”**,

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

7.12. PROFILES/TIMERS



Profiles/timers menu allow the user to define any profile/timer which can be used to control any process.

7.12.1. Profile/timer - General settings

In the device there are **8 independent settings of Profiles/timers** available. Configured **Profile/timer** can be used by any **Logical channel** switched to **Profile/timer** mode - **Fig. 7.79** (see also **Chapter 7.8.8. Logical Channels - Profile/timer mode**).

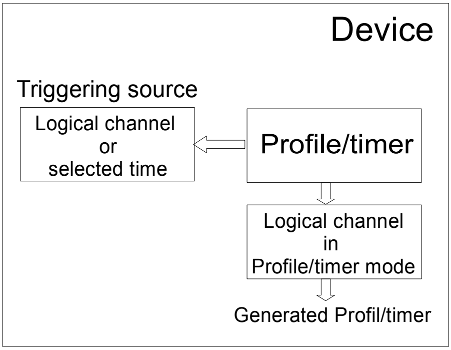
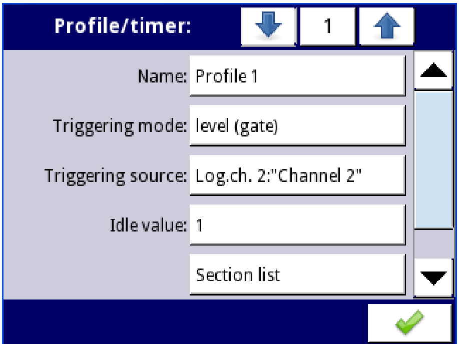


Fig. 7.97. Block diagram of the device configuration for generating Profiles/timers

The window with basic parameters of the **Profile/timer** shown on **Fig. 7.98**.



Profile/timer: 1

Name: Profile 1

Triggering mode: level (gate)

Triggering source: Log.ch. 2:"Channel 2"

Idle value: 1

Section list

Fig. 7.98. View of the configuration Profiles/timers window

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

Common parameters for **Profile/timer** are:

- **Name** – it gives the name to the Profile/timer,
- **Triggering mode** – using this parameter user select a way to generated Profile/timer, there are five modes of triggering:
  - **disabled**,
  - **level (gate)** – this means that **Profile** which was configured by user will be generated when source signal will have a value > 0, otherwise (if source value ≤ 0) defined Profile will not be generated. For more details see **Chapter 7.12.2. Profiles/timers - Triggering mode: level (gate), edge (once), edge (retrig.)**,
  - **edge (once)** – this means that which was configured by user **Profile** will be triggered by rising edge signal (from values ≤ 0 to the value > 0) come from 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 7.12.2. Profiles/timers - Triggering mode: level (gate), edge (once), edge (retrig.)**,
  - **edge (retrig.)** - this means that configured by user **Profile** will be triggered by rising edge signal (from values ≤ 0 to the value > 0) that comes from source signal. However, in this mode, unlike the **edge (once)** mode a defined **Profile** will be generated from the beginning every time when the Triggering source signal will generate an edge, whether that Profile had been completed or not. For more details see **Chapter 7.12.2. 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 7.12.3. Profiles/timers - Triggering mode: on time**,
- **Pause mode** – way to pause generated Profile/timer,
  - **disabled** – Profile/timer can not be paused,
  - **high level** – Profile/timer will be paused, when channel's value in **Pause source** parameter will be greater than zero (>0) and will be resumed when this value will be lower or equal zero (≤0). If Profile/timer is paused and new trigger will be

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

  - **low level** – Profile/timer will be paused, when channel's value in **Pause source** parameter will be lower or equal zero (≤0) and will be resumed when this value will be greater than zero (>0). If Profile/timer is paused and new trigger will be detected, the Profile/timer will remain in pause state. Additionally, when in **Section list** parameter, **Shape** of **Section num. 1** is set as **const.value**, then Profile/timer will set itself on **Final value** of this section, and when in **Section list** parameter, **Shape** of **Section num. 1** is set as **slope**, then Profile/timer will set itself on its **Idle value**.
  - **rising edge** – Profile/timer will be paused, when the device detects channel's value change from lower or equal zero (≤0) to greater than zero (>0) in Pause source parameter, and will be resumed when the device detects another change like that in this channel. If Profile/timer is paused and new trigger will be detected, the Profile/timer will react on this trigger like it would not be paused.
  - **falling edge** – Profile/timer will be paused, when the device detects channel's value change from greater than zero (>0) to lower or equal zero (≤0) in **Pause source** parameter, and will be resumed when the device detects another change like that in this channel. If Profile/timer is paused and new trigger will be detected, the Profile/timer will react on this trigger like it would not be paused.
- **Pause source** – this parameter is not visible when **Pause mode** parameter is set as **disabled**. It contains the list of Logical channels, where selected one will be pause source for the Profile/timer,
- **Idle value** – the value before and after generating the defined the Profile,
- **Section list** – submenu where user can define Profile/timer consisting 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 following options:
  - **disabled** – the Profile is generated only once,
  - **counted** – allows to generate Profile specified number of times defined in **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 infinite repeating of generated **Profile**,
- **Loop count** – this parameter is visible only when **Looping** parameter is set on **counted**. It allows the user to enter number of generated Profile repetitions,
- **Looping source** – this parameter is visible only when **Looping** parameter is set on **from logical channe**l. It allows the user to select the logical channel from the list, which value will be compared to the number of Profile cycle. If the cycle value is equal or greater than value in the logical channel, next repetition of the Profile will not start,
- **Return to position** – this parameter is invisible only when **Looping** parameter is set as **disabled**. It allows user to select a section from the list, from which any next generated Profile should starts. In case when **Section list** in empty, this parameter is also empty,

Section list Sub-menu
It allows the user to configure Profile/timer consisted up to 100 sections and for each of them user can set: Duration, Unit, Shape and Final value.

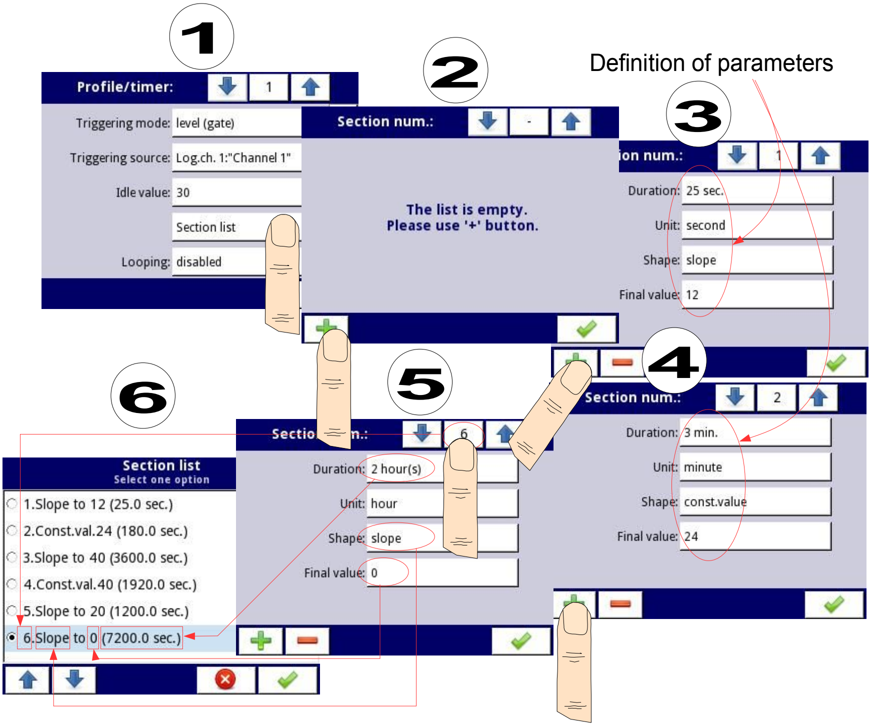
+ This button allows the addition of a new section to list.

− This button allows the removal of the section from the list.

⬇ 3 ⬆ Arrows placed in the upper right corner of the screen allow switching between sections. Middle button allows direct selection of a specific section.

The **Section list** parameters are:

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



1
Profile/timer: ⬇ 1 ⬆
Triggering mode: level (gate)
Triggering source: Log.ch. 1:"Channel 1"
Idle value: 30
Section list
Looping: disabled

2
Section num.: ⬇ - ⬆
The list is empty.
Please use '+' button.

3
ion num.: ⬇ 1 ⬆
Duration: 25 sec.
Unit: second
Shape: slope
Final value: 12

4
Section num.: ⬇ 2 ⬆
Duration: 3 min.
Unit: minute
Shape: const.value
Final value: 24

5
Sectio n.: ⬇ 6 ⬆
Duration: 2 hour(s)
Unit: hour
Shape: slope
Final value: 0

6
Section list
Select one option
○ 1.Slope to 12 (25.0 sec.)
○ 2.Const.val.24 (180.0 sec.)
○ 3.Slope to 40 (3600.0 sec.)
○ 4.Const.val.40 (1920.0 sec.)
○ 5.Slope to 20 (1200.0 sec.)
● 6.Slope to 0 (7200.0 sec.)

Fig. 7.99. Sample of Profile/timer configuration

<u>Comments for **Looping** parameter</u>

If the user select: **counted** or **infinite**, repeats of the Profile/timer the user has:

- 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** this section. This is shown in **Fig. 7.100** (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 **Fig. 7.100** (profile guided fine line).

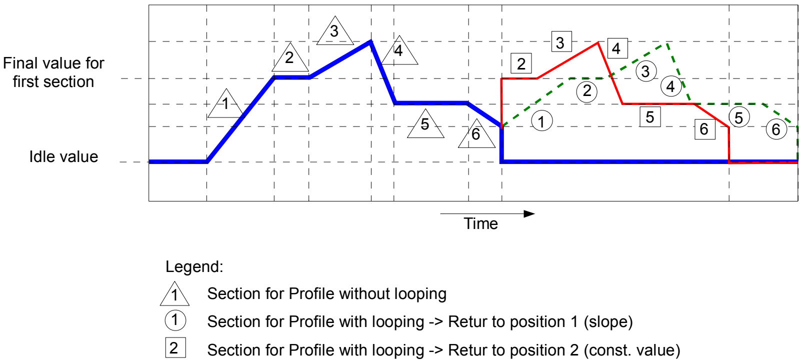


Fig. 7.100. Sample of Profile/timer with description of parameters

<u>7.12.2. Profiles/timers - Triggering mode: level (gate), edge (once), edge (retrig.)</u>

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 Profile/timer, there are five modes of triggering:
  - **disabled**,
  - **level (gate)** – this means that configured by user **Profile** is generated when source signal will have a value **> 0**, otherwise (if source value **≤ 0**) defined Profile will not be generated. See **Fig. 7.101** and example **a)** in **Fig. 7.102**,
  - **edge (once)** – this means that configured by user **Profile** will be triggered by the rising edge (from values **≤ 0** to the value **> 0**) of source signal. After the rising edge the **Profile** will be generated in whole (once), regardless of further changes to the signal source, see **Fig. 7.101** and example **b)** in **Fig. 7.102**,
  - **edge (re-triggering)** – this means that configured by user **Profile** will be triggered by the rising edge (from values **≤ 0** to the value **> 0**) of source signal. However, in this mode, unlike the **edge (once)** mode the defined **Profile** will be generated from the beginning every time when the **Triggering source** signal will generate an edge, whether that Profile had been completed or not. See **Fig. 7.101** and example **c)** in **Fig. 7.102**,

– **Triggering source** – allows to choose 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** – submenu in which the user can define up to 100 sections generated by the Profile/timer. For more information see **Chapter 7.12.1. Profile/timer - General settings**,
– **Looping** – each Profile/timer has an ability to repeat its generated signal, this parameter has options:
  - **disabled** – the profile is not repeated,
  - **counted** – allows to generate the profile specified number of times defined in **Loop count** parameter,
  - **from logical channel** - allows to run the profile number of times equals the value in the selected logical channel determined by the **Looping source** parameter,
  - **infinite** – allows to infinite repeating of generated **Profile**,

– **Loop count** – this parameter is visible only if **Looping** parameter is set as **counted**, defines the number of repetitions of generated Profile,
– **Looping source** – this parameter is visible only if **Looping** parameter is set as **from logical channel**. It allows to choose a logical channel from the list, which value will be compared with the number of already made cycles of Profile,
– **Return to position** – this parameter is invisible if **Looping** parameter is set as **disabled**, allows to select a fixed position from which every next generated Profile has to start,

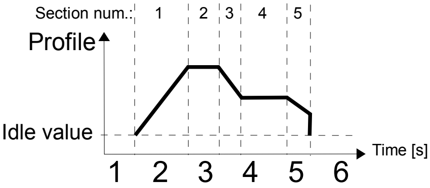


Fig. 7.101. Sample of Profile/timer

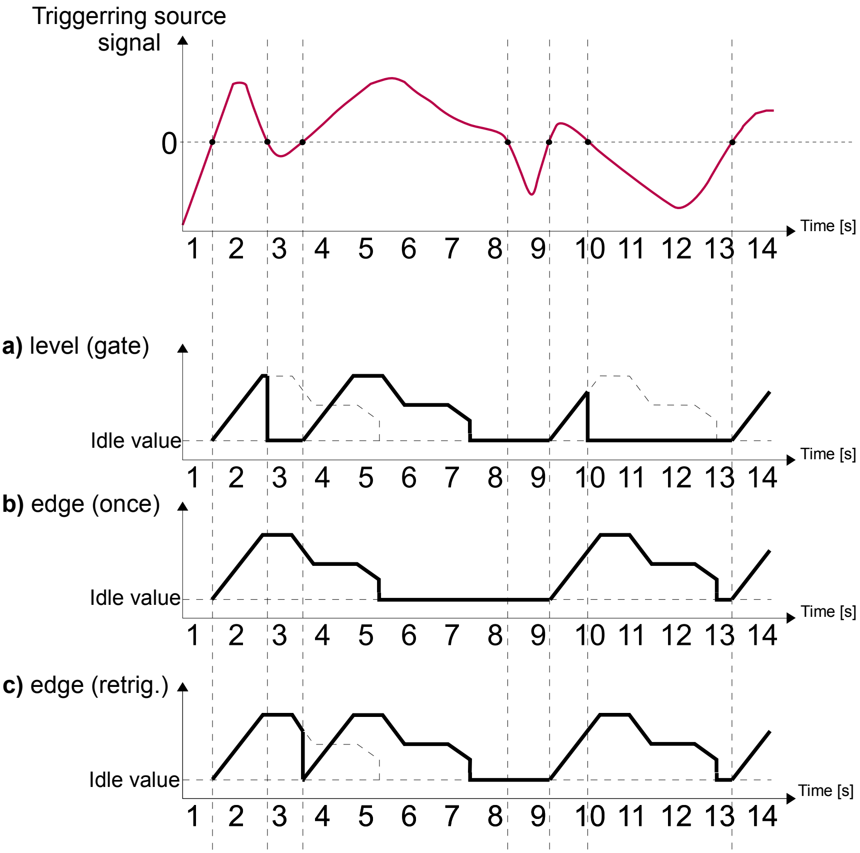


Fig. 7.102. Samples of ***Profile*** *output waveforms defined in* ***Fig. 7.101****, triggered by signal selected in* ***Triggering source*** *parameter for* ***Triggering mode****:* ***level (gate), edge (once), edge (retrig.)***

7.12.3. Profiles/timers - Triggering mode: on time

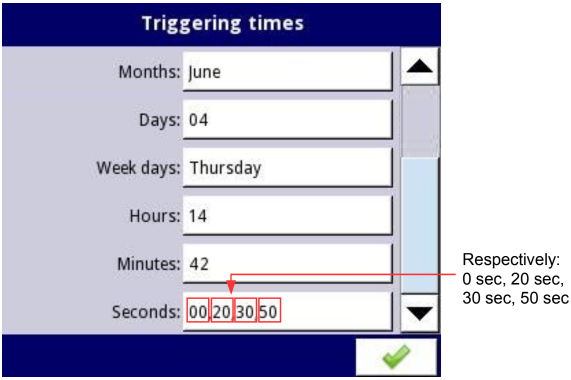
The parameters of **Profiles/timers** for **triggering mode**: **on time** are:

- **Name** – it gives the name to the Profile/timer,
- **Triggering mode=on time** – allows to trigger the Profile at demanded time defined in **Triggering times** parameter,
- **Triggering times** – submenu with the parameters which defines times of Profile triggering. This 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** – submenu in which the user can define up to 100 sections generated by the Profile/timer. For more information see **Chapter 7.12.1. Profile/timer - General settings**,
- **Looping** – each Profile/timer has an ability to repeat its generated signal, this parameter has options:
  - **disabled** – the profile is not repeated,
  - **counted** – allows to generate the profile specified number of times defined in **Loop count** parameter,
  - **from logical channel** – allows to run the profile number of times equals the value in the selected logical channel determined by the **Looping source** parameter,
  - **infinite** – allows to infinite repeating of generated **Profile**,
- **Loop count** – this parameter is visible only if **Looping** parameter is set as **counted**, defines the number of repetitions of generated Profile,
- **Looping source** – this parameter is visible only if **Looping** parameter is set as **from logical channel**. It allows to choose a logical channel from the list, which value will be compared with the number of already made cycles of Profile,
- **Return to position** – this parameter is invisible if **Looping** parameter is set as **disabled**, allows to select a fixed position from which every next generated Profile has to start,

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

In **Fig. 7.103** and **Fig. 7.104** shown examples of Profile/timer: triggering times configuration and output waveform. Operation of Profile in **'on time'** mode is similar to **edge (once)** mode - see **Chapter 7.12.2. 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 time generating the Profile.



Triggering times

Months: June

Days: 04

Week days: Thursday

Hours: 14

Minutes: 42

Seconds: 00 20 30 50

Respectively:
0 sec, 20 sec,
30 sec, 50 sec

*Fig. 7.103. Samples of time settings for **Profile/timer***

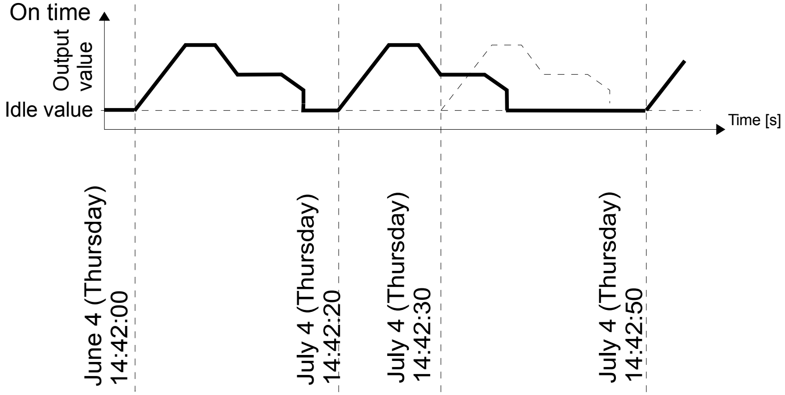


*Fig. 7.104. Waveform for **'on time'** triggering mode and time parameters in accordance with **Fig. 7.103***

7.12.4. Examples of Profile/timer configurations

***Example 7.12.4.1**: Application of the Profiles/timers.*

Task:

The task is to create a Profile timer in logical channel 1 in **edge (once)** mode, which source is logical channel 2 which is connected to current input of **UI4** module. Profile consists of four segments:

1. ramp from 0 to 10 during 5 seconds,
2. const.value 8 during 2 seconds,
3. ramp from 8 to 4 during 3 seconds,
4. const.value 4 during 1 second,

Looping is disabled and 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. 7.57** (see also **Appendix 8.2. UI4, UI8, UI12, U16, U24, I16, I24 – VOLTAGE and CURRENT MEASUREMENT MODULES**),

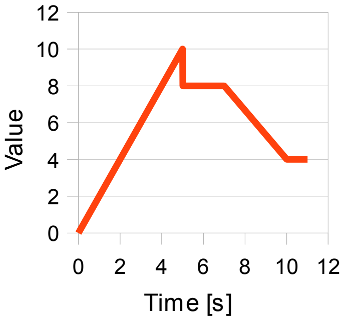


Fig. 7.105. An example of the waveform

In the first step you need to configure the Profile/timer. To do this:

- touch 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 **Name** parameter write **Profile 1**,
- **Triggering mode** parameter set as **edge (once)**,
- **Triggering source** parameter set as **Logical channel 2**,
- in **Idle value** parameter write **0**, because we want at the beginning the Profile has 0 value on the output,
- 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 that Profile will be executed only once after triggering from logical channel 2,
- at the end leave the **Profiles/timers** menu,

In the next step you need to define a **Logical channel**:

- touch 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 **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 logical channel (**Triggering source**) in **Hardware input** mode for **current input** is shown in **Example 7.8.11.1** (we configure this way the logical channel 2).

The second method is described in **Chapter 7.8.11. Examples of Logical Channels configuration** in the ***Example 7.8.11.9***.

***Example 7.12.4.2**:* Application of the Profiles/timers.

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 consist of two sections:

1. const.value 1 during 0,2 second,
2. const.value 0 during 1,8 second,

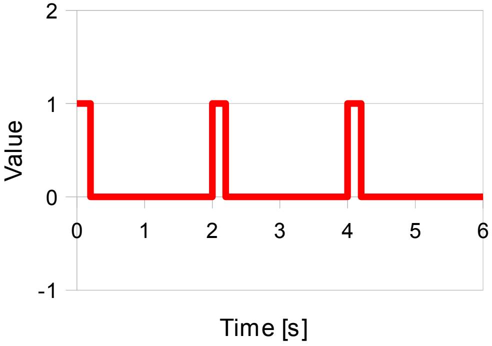


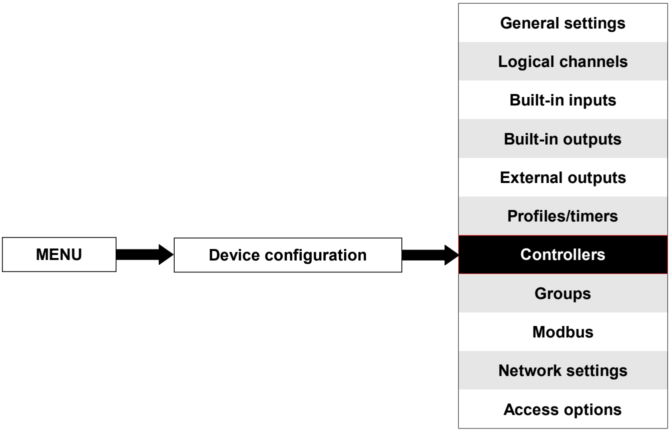
Fig. 7.106. An example of the waveform

Solution:

To complete the task a logical channel in Profile/timer Mode will be needed:

- touch 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 **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 on the bottom left side of the screen,
  - enter the **Days** menu and press the icon on the bottom left side of the screen,
  - enter the **Week days** menu and press the icon on the bottom left side of the screen,
  - enter the **Hours** menu and press the icon on the bottom left side of the screen,
  - enter the **Minutes** menu and press the icon on the bottom left side of the screen,
  - enter the **Seconds** menu and press the icon on 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**,

7.13. CONTROLLERS



MENU → Device configuration → Controllers

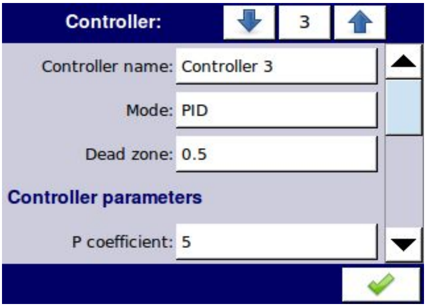
General settings
Logical channels
Built-in inputs
Built-in outputs
External outputs
Profiles/timers
Controllers
Groups
Modbus
Network settings
Access options

Although most controlling processes can be realised using simple ON - OFF mode, there is sometimes necessity to use more advanced methods to control the actuators. The MultiCon 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 is 8 independent settings of PID type controllers available.

7.13.1. 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 **Controller** mode - see **Chapter 7.8.7. Logical Channels - Controller mode**.

The window with basic parameters of the **Controller** shown on **Fig. 7.107**.



Controller: 3

Controller name: Controller 3

Mode: PID

Dead zone: 0.5

Controller parameters

P coefficient: 5

Fig. 7.107. Main configuration of an Controller profile

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

The parameters of **Controllers** are:

- **Name** – it gives the name to the Controller,
- **Mode** – in this parameter user can select control mode which is used to controller calculation (algorithm), there are 3 options:
  - **PD** - proportional–derivative mode,
  - **PI** - proportional–integral mode,
  - **PID** - proportional–integral–derivative mode,
- **Dead zone** - this parameter determines how much the process variable must change in relation to its value in the previous cycle before it will be noticed by the controller, it means that the output of the controller will be changed if the difference between **Set point channel** value and **Feedback channel** value (more about **Set point channel** and **Feedback channel** parameters see **Chapter 7.8.7. Logical Channels - Controller mode**) exceeds the **Dead zone** value,
- **Controller parameters** parameter block - this block allows the user to set **PID** coefficients:
  - **P coefficient** – this parameter is visible always, allows to set value of the proportional coefficient,
  - **I coefficient** – this parameter is available for **PI** and **PID** mode and allows to set the integral coefficient,
  - **D coefficient** - this parameter is available for **PD** and **PID** mode and allows to set the derivative coefficient,
  - **Differentiated signal** - this parameter is available for **PD** and **PID** mode and allows the selection of the option (see **Fig. 7.108** and **Fig. 7.109**):
    - **feedback (measured)** - in this option the value of **Feedback channel** is directly sent to **D term**, which allows the fast response of the device to fast changes with the controlled object,
    - **error (deviation)** - in this option the value of **Feedback channel** is send to **D term** after calculation of error output and checking exceeds the range of **Dead zone**, this option is set for slow changes controlled object,
- **Controller output** parameter block - this block has parameters:
  - **Offset** - value of this parameter causes offset of controller output value,
  - **Low output limit** – defines low limit of the controller output signal value,
  - **High output limit** – defines high limit of the controller output signal value,
- **Initial conditions** parameter block has following options:
  - **Init type** – has two options:
    - **No init (internal state 0)** – during start of the controller its output is in 0 state,
    - **Value** – allows to set controllers output state during its start depending on **Low output limit** and **High output limit** parameters,
  - **Initial output** – parameter is visible for **Init type: Value**; defines controllers output state, where **0%** corresponds to **Low output** limit and **100%** corresponds to **High output limit**,

Note! After offset the output signal, the output value is limited to the range set in **Low output limit** and **High output limit** parameters (see **Fig. 7.109**),

Fig. 7.108 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 a **Set point channel** and the **Feedback channel**, which store the data required to control the object. Respectively, **Set point channel** contains a destination value of the process, while the **Feedback channel** includes the value of feedback coming from the object controlled. MultiCon uses data collected from these channels and the corresponding Controller controls the object.

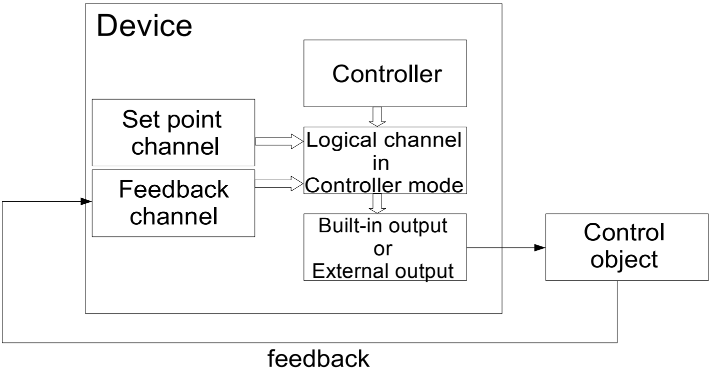


Fig. 7.108. Block diagram of the control loop of the object by MultiCon

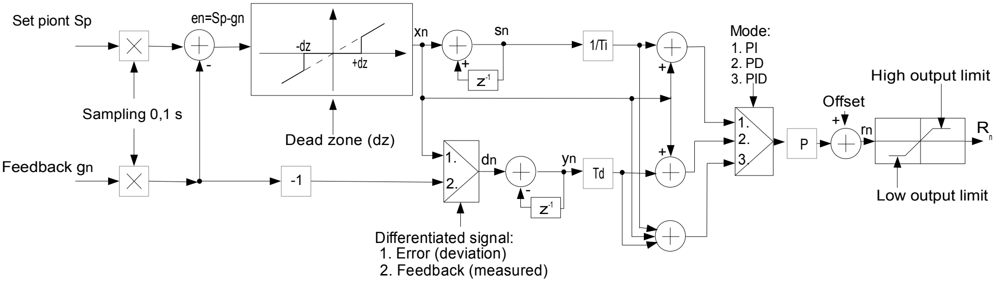


Fig. 7.109. Block diagram of the ***Controller*** *implemented in the device*

Formula for **Controller** output:

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

7.13.2. Examples of Controller configurations

Example 7.13.2.1: *Application of the 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. A logical channels with set point value and temperature from PT100 sensor are connected to the controller. After pre analize the sensor type and the transmittance of the object, we set the controller coefficients on values: P=0.3, I=0.5, D=0.02. We do not want that the controller respond for deviation lower than 2°C. 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. Example of this connections are shown on **Fig. 7.110**.

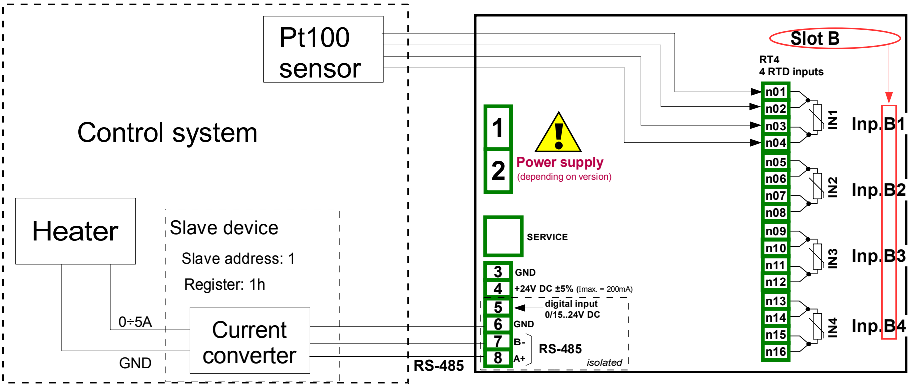


Fig. 7.110. The connection scheme for RT4 module and Modbus MB1 port

In the first step we define **Controller** settings:

- touch screen and press **Menu** button,
- press the **Device configuration** button,
- enter the **Controllers** menu,
- using the arrows tin the top navigation bar, select any **Controller** such as **1**,
- in **Controller name** parameter write **Controller 1**,
- **Mode** parameter set as **PID**,
- in **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**,
  - in **High output limit** write **20**,

After leaving the configuration save settings. In next steps we should set Logical channels (examples of logical channel configuration are in **Chapter 7.8.11. Examples of Logical Channels configuration**, especially **Example 7.8.11.8**), set Modbus for Slave device 9examples for Modbus configuration are in **Chapter 7.15.4. Modbus - Example of Modbus protocol configuration in the device**, examples of external outputs configuration which will communicate with Slave device via RS-485 are in **Chapter 7.11.4. Examples of external output configurations**).

Example 7.13.2.2: *Application of the Controllers in cooperation with heater which is controlled by SSR output.*

Task:

The task is to configure the controller which will control the heater using SSR output in order to obtain a constant object temperature. A logical channels with set point value and temperature from thermocouple type K sensor are connected to the controller. Due to fact that choosing of PID controller is not matter of the task, we can use following demonstration values: P=5, I=10, D=1. We do not want that the controller respond for deviation lower than 2ºC. On 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. Example of this connections are shown on **Fig. 7.111**.

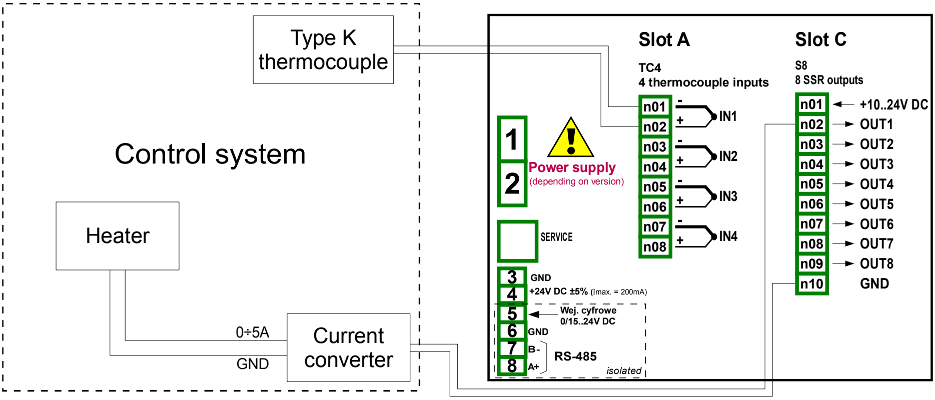


Fig. 7.111. The connection scheme for RT4 module and Modbus MB1 port

Logical channel operating as PID controller generates "analog" output signal. This signal can be limited to range suitable for used output. On the other hand all built-in outputs has settings allowing to fit them to source signal (see **7.10.3. Build-in output - PWM (Pulse-width modulation) mode for SSR relay output** or **7.10.4. Built-in output - Current output**). It is user's responsibility to choose correct limitation of PID output (as well as other PID settings), but important is to set limitation range wide enough to obtain smooth controlling of actuator, and narrow enough to make PID as fast as possible. We can assume that PID output range is set as 0-100. To set this, user should enter **Controller** menu, select requested controller and in **Controller output** parameters block set:

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

Other PID controller settings depends only on application.

Lets assume that user has to use TC signal (K) for temperature measurement (module TC4 installed in slot A) and SSR output driver (module S8 installed in slot C) to control a heater keeping set temperature of the object. Lets assume that temperature setpoint can be set between 120 and 200 $^{\circ}$C.

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

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

for **Logical channel 1** :

Name: Temperature,
Mode: Hardware input,
Source: Inp. A1 (we assume that built-in input is connected and set properly for TC type K),
Precision: 0,
Graph Lo: 0,
Graph Hi: 300,

all 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 other parameters should be on default settings,

for **Logical channel 3**:

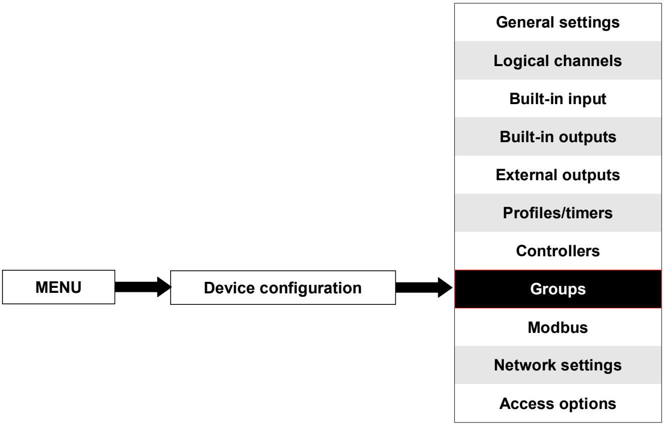
Name: PID controller,
Mode: Controller,
Controller num: 1 PID:"Controller 1",
Set point channel: Log.ch. 2:"Desired temp",
Feedback channel: Log.ch. 1:"Temperature",
Precision: 0,
Graph Lo: 0,
Graph Hi: 300,

all other parameters should be on default settings,

The last thing to do is set of SSR output.
Enter to the **Device configuration** menu, then **Built-in outputs** and select Output: **Out.C1 : SSR**. In this output set parameters as follows:

Mode: PWM
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,

7.14. GROUPS



Groups are the sets of 1-6 **Logical Channels** collected together for clearance. To see detailed definition of **Group** see **Chapter 5.1. UNDERSTANDING CONTROLLER/DATA RECORDER**. If the MultiCon has license for data logging then each Group is able to log the data coming from Logical channel included in this Group.

7.14.1. Groups - General settings

Arrows placed in the upper right corner of the screen allow switching between groups to configure settings of group parameters. The middle button allows direct selection of specific group from the list.

The parameters of **Groups** are:

– **Group** - each group can be:
  - **disabled** - after selecting this option, the other parameters are not visible and this group is not visible on the measuring screen,
  - **enabled** - for this option the group is active,

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

– **Charts** – sets the chart type whet 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 phasor amplitude value and the logical channel which is set in **Slot 2 (4, 6)** is responsible for phasor phase value. User should remember that **Graph low** and **Graph high** parameters should be adjusted to values that may occur in the logical channel, because minimal and maximal phasor value corresponds to values in these parameters. Phase of the phasor is module 360 function of the logical channel.

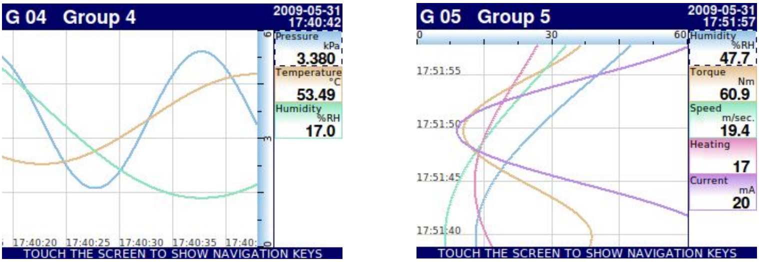


*Fig. 7.112. Examples of **Horizontal** and **Vertical Charts** Group views*

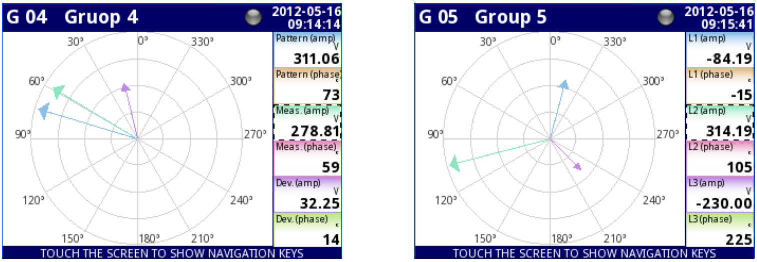


*Fig. 7.113 Examples of **Phasor Charts** Group views*

– **Bars** – sets the bars type when group is displayed in this mode,
- **horizontal** - horizontal direction of bars position,
- **vertical** - vertical direction of bars position,

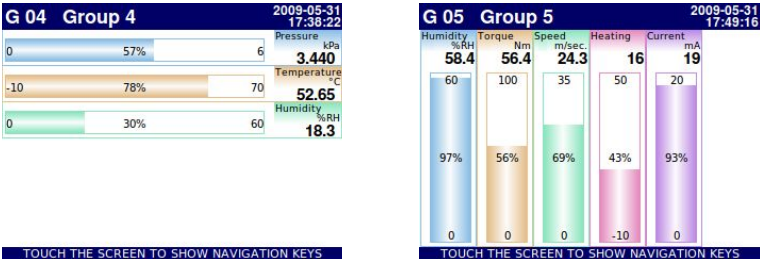


*Fig. 7.114. Examples of **Horizontal** and **Vertical Bars** Group views*

– **Line width** – sets the width of the line:
- **1 pixel** - the char line one pixel width,
- **2 pixels** - the char line two pixels width,
- **3 pixels** - the char line three pixels width,

– **Time scale** – time window which displays chart which contains samples from 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**

Each group displays samples in time window (defined in **Time scale** parameter in **Groups** menu) (see **Fig. 7.115**).
When **Time scale** is set as **30 min. or more**, the device has possibility to display all samples which were measured to present moment and can fit in this **Scale** (regardless if the device was on or off, see **Fig. 7.116**). Samples which were not measured will be:

displayed as "0", when time when the device was off does not fits between moving vertical markers on the screen

repeated on screen, until new samples after turning on the device occurs, when time, when the device was off does fits between moving vertical markers on the screen

When **Time scale** is set as **less then 30 min.** the device has possibility to display only samples which were measured from moment when the device was turned on to present moment (but they must fits in the **Scale**, see **Fig. 7.117**).

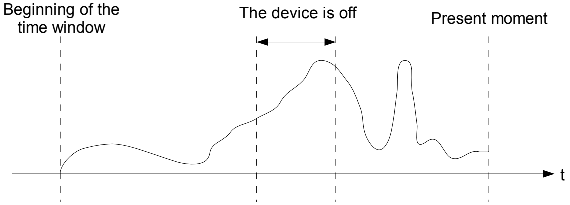


Fig. 7.115. Example of displayed real signal

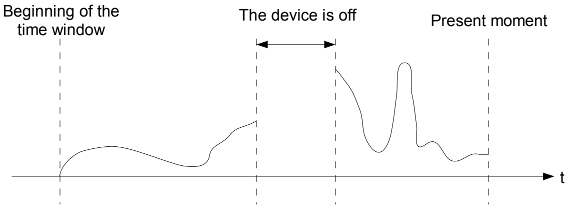


*Fig. 7.116. Example of displayed signal when **Time sample** ≥ 30min*

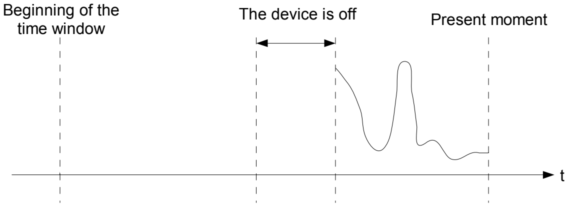
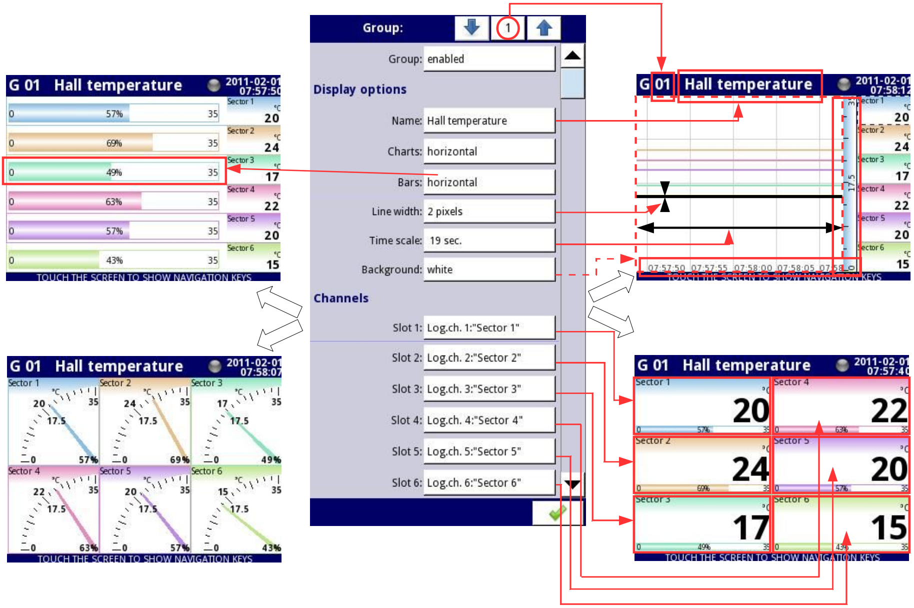


*Fig. 7.117. Example of displayed signal when **Time sample** < 30min*

– **Background** - this parameter has option:
  - **white** - the background of window displaying the chars is white,
  - **black** - the background of window displaying the chars is black,

– **Channels** parameter block - this block defines the number and location of Logical channels that are displayed in the Group, includes the parameters:
  - **Slot 1÷6** - in each slot user can select a option (see **Fig. 7.118÷Fig. 7.120**):
    - ◦ **disabled** – disabled position is skipped which reduces the number of position to deploy in the display window,
    - ◦ **empty** – the empty position remain 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** – user can select one of all available Logical channels which will be displayed in the specific location on the screen,
  - **Style** – defines color which will be used to draw logical channel selected in this **Slot**. User can specify font color in channel (except value color) and drawing color of all elements connected with channel visualisation (see **Fig. 7.121**). Using black **X** buttons user can return to default settings.

– **Logging options** parameter block - logging options are available only in the device having a license for logging the data (for more information about the logging license is in **Chapter 7.4. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION**); parameters of this block shown and described in the **Chapter 7.14.2. Groups - Logging options**.



Group: 1

Group: enabled

Display options

Name: Hall temperature

Charts: horizontal

Bars: horizontal

Line width: 2 pixels

Time scale: 19 sec.

Background: white

Channels

Slot 1: Log.ch. 1:"Sector 1"

Slot 2: Log.ch. 2:"Sector 2"

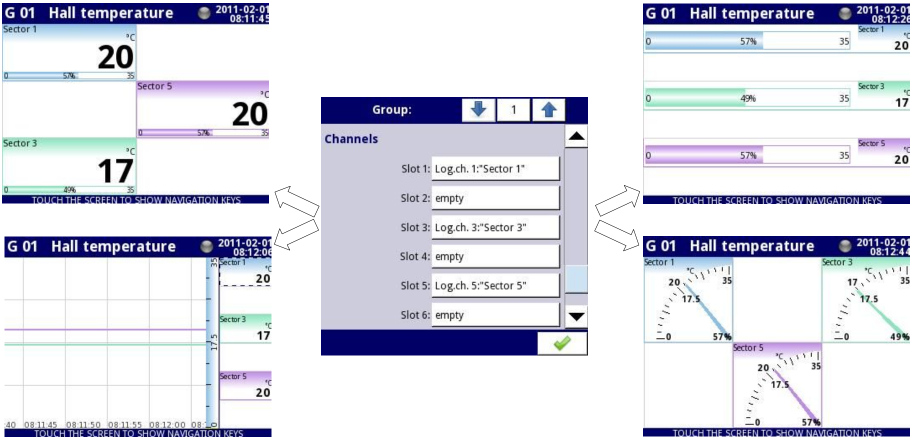
Slot 3: Log.ch. 3:"Sector 3"

Slot 4: Log.ch. 4:"Sector 4"

Slot 5: Log.ch. 5:"Sector 5"

Slot 6: Log.ch. 6:"Sector 6"

Fig. 7.118. Sample of **Group** parameters settings - all Slot set to **Logical channel**



Group: 1

Channels

Slot 1: Log.ch. 1:"Sector 1"

Slot 2: empty

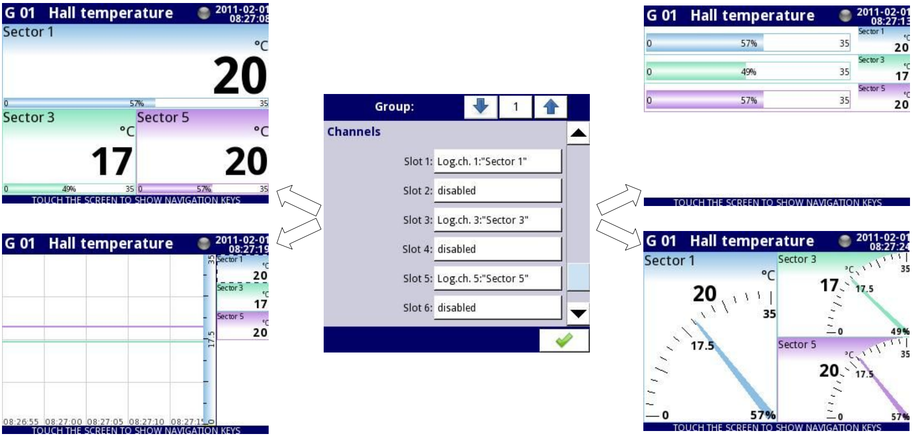
Slot 3: Log.ch. 3:"Sector 3"

Slot 4: empty

Slot 5: Log.ch. 5:"Sector 5"

Slot 6: empty

Fig. 7.119. Sample of **Group** parameters settings - Slots set to **Logical channel** and set to **empty**



Group: 1

Slot 1: Log.ch. 1:"Sector 1"

Slot 2: disabled

Slot 3: Log.ch. 3:"Sector 3"

Slot 4: disabled

Slot 5: Log.ch. 5:"Sector 5"

Slot 6: disabled

Fig. 7.120. Sample of **Group** parameters settings - Slots set to **Logical channel** and set to **disabled**

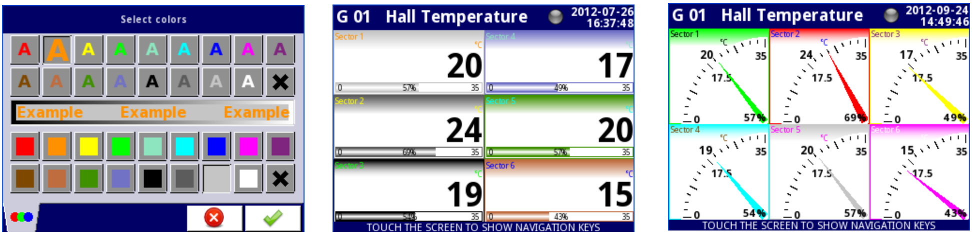


Fig. 7.121. Style color select screen and examples of views with changed styles

7.14.2. Groups - Logging options

Logging options are available only in the device having a license to log the data (for more information about the logging license see **Chapter 7.4. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION**). To log the data from the Logical Channel should be:

- **Logical channel** attached to **Group** using the **Channels** parameter block (see **Chapter 7.14.1. Groups - General settings**),
- **enable** the data logging by setting the options in block of parameters -> **Logging options**,
- after exiting the menu accept the changes by **writing configuration**,
- received logging data files can be sent to flash drive (form more information about files management see **Chapter 7.3. FILES MANAGEMENT**)

Each **Group** has its own data logging options and the device can log the 10 independent Groups of Logical channels at the same time .

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. 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.

Logging options parameter block

- **Mode** – defines way of triggering group logging:
  - **disabled** - logging of selection Group is disabled,
  - **always** - logging is continuous in time,
  - **from log.channel** - this option activate new parameter - **Triggering source** which enables logging the data when the value of **Triggering source > 0**,
- **Triggering source** - this parameter is visible for **Mode: from log.channel**, in this parameter user can choose a logical channel from list and selected one will be triggering source for this group; logging is active when value in selected channel is **> 0**,
- **Description** - user can set a description of a data logging file by pressing the button next to the **Description** label and then setting any text,
- **Base period** and **Base unit** - this parameters set duration of the sample of data logging, 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**,
- **Alternative mode** - this mode allows the user to log data in special situation where a deeper analysis is required (for example in critical state of object),this parameter has the following options:
  - **disabled** - alternative logging of selection Group is disabled,
  - **from logical channel** - this option activate 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 value of this source **> 0** than the **data logging** for alternative mode is **enabled**,
- **Alternative period and Alternative unit** - this parameters set duration of the sample of data logging for alternative mode, these parameters have 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**,

For any changes to the settings of the configuration of logging Group (e.g. a change in the logging parameters, changing parameters of **Display options** parameter block or change parameters of Logical channel included in logging Group) creates a new logging file. If user shuts down the device or changes other parameters independent of logging Group new logging file is not created.

On time of saving, reverting configuration or loading configuration from pendrive, the logging is hold what will cause absence of samples in logging files.

Accurate description of data structure which are saved during logging is in chapter **8.16. DATA FORMAT**.

7.14.3. Groups - Examples of visualisations of groups

Example 7.14.3.1: *Single channel - one big needle.*

Task:

The task is to create Group with only one logical channel. The value of that channel should be displayed in whole screen in needle mode.

Solution:

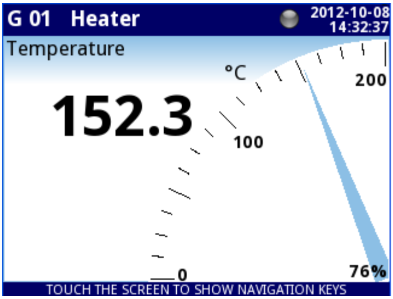
Let's assume that we have configured logical channel in Hardware input Mode for RTD module. The logical channel is named Temperature and it displays heater temperature (example of logical channels configuration are shown in **Chapter 7.8.11. Examples of Logical Channels configuration**, especially see **Example 7.8.11.3**).
We need to define Group with logical channel in order to display it on the device screen. To do that:

- enter **Groups** menu and turn on **Group 1** if it is turned off,
- in **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 other parameter should be on default settings,

Next we need to define Initial view. To do this:

- enter **General settings** menu,
- for **Initial view** parameter block:
    - **Display mode** parameter set as **needles**,
    - **Displayed group** parameter set as **Group 1: Heater**,

Finally leave this menu and press Save settings button. Result of just made changes will be visible after first measurement is called. Example of this view is posted in **Fig. 7.122**.


G 01 Heater

2012-10-08 14:32:37

Temperature

152.3 °C

Fig. 7.122. Single big needle example

Example 7.14.3.2*: Three channels view - one bigger, two smaller*

Task:
The task is to visualize three logical channels which will present 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**,

(example of logical channel configuration are shown in **7.8.11. Examples of Logical Channels configuration**)

We need to define Group with this logical channels to display them in one window on the screen. To do that:

- enter **Groups** menu and turn on **Group 4** if it is turned off,
- in **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 other parameter should be on default settings,

Next we need to define Initial view. To do this:

- enter **General settings** menu,
- for **Initial view** parameter block:
    - **Display mode** parameter set as **needles**,
    - **Displayed group** parameter set as **Group 4L Group 4**,

Finally leave this menu and press Save settings button. Result of just made changes will be visible after first measurement is called. Example of this view is posted in **Fig. 7.123**.

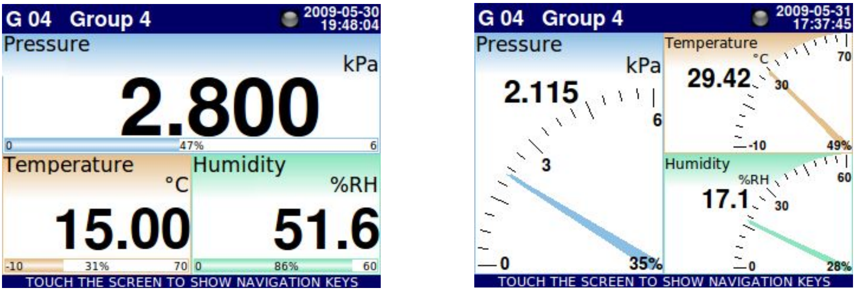
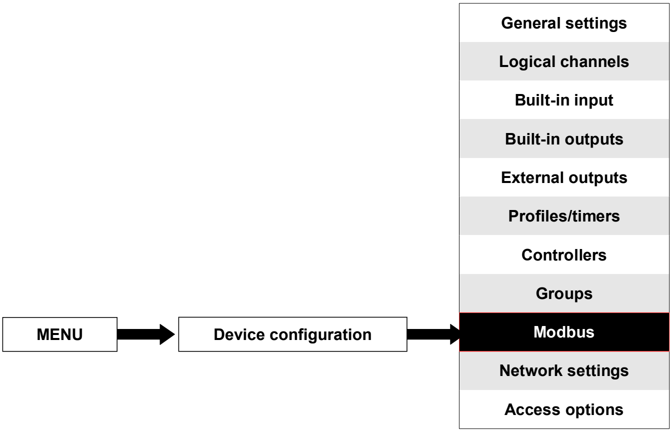


Fig. 7.123. Example of three channels presentation with emphasis of Pressure

Switching between modes it is noticeable that position of particular channels can slightly vary. It is caused by aspect of different modes – their position is selected in that way to obtain elements of particular data panels as big as possible.

7.15. MODBUS



The basic version of MultiCon has one RS-485 port built-in. The communication ability can be increased by installing a communication module into slot D of the device (**Fig. 7.124**). This module offers 2 additional serial ports (one RS-485, and one RS-485/RS-232) 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 Slave or Master (except port 4 which can be only SLAVE) mode.

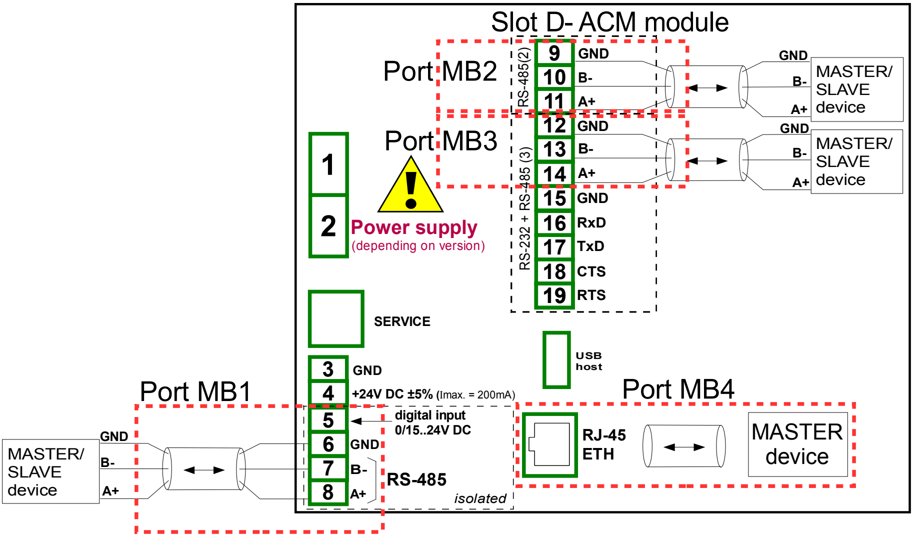
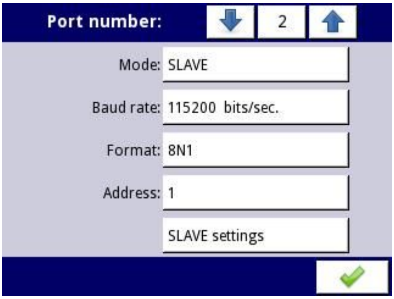


Fig. 7.124. Serial communication ports available in the device

7.15.1. Modbus - General settings

Window with basic parameters of communication interfaces is shown in **Fig. 7.125**.



Port number: 2

Mode: SLAVE

Baud rate: 115200 bits/sec.

Format: 8N1

Address: 1

*Fig. 7.125. Configuration parameters for **SLAVE** mode*

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.

Parameter common for all Modbus protocol modes is:

- **Mode** - which has option:
  - **disabled** – the selected Modbus port is inactive,
  - **SLAVE** – this device is SLAVE device, see **Chapter 7.15.2Modbus - SLAVE mode**,
  - **MASTER** – this device is MASTER device and manages the Slave devices, see **Chapter 7.15.3. Modbus - MASTER mode**

7.15.2. Modbus - SLAVE mode

The parameters of the Modbus RTU protocol (serial communication) for 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. 7.6**.

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. 7.6 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 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 7.15.2.3. Modbus SLAVE - The Modbus protocol handling** for detailed description of MODBUS protocol handling in MultiCon 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 type, data format and registers activity which are available to the MultiCon 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 **SLAVE settings** menu:

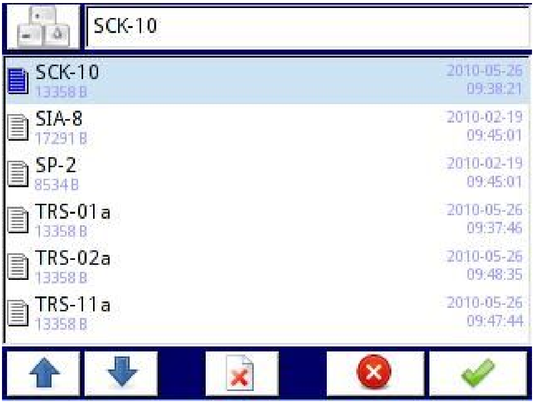
- **Load device template** - this button allows the loading of a template with predefined blocks of input registers, for more information see **Chapter 7.15.2.1. Modbus SLAVE - Modbus Templates for SLAVE mode**,
- **Save device template** - this button allows saving of templates of predefined blocks of input registers, for more information see **Chapter 7.15.2.1. Modbus SLAVE - Modbus Templates for SLAVE mode**,
- **Device channels** parameters block - this block of parameters allows the user to define 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,

7.15.2.1. 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 for loading a template with predefined blocks of output registers. Pressing this button invokes a file selection window. Templates can be imported into the MultiCon using a **File Management** menu,
- **Save device template** - this button allows the user to save in templates the predefined blocks of input registers. Saved template can be used for:
  - fast copying/moving of the SLAVE device settings to another port (using **Load device template** button),
  - easy exchange of templates between different MulitCon devices (using **File management** menu),

An example of Modbus template selection window is shown in **Fig. 7.126**. Below this figure additional icons are described.



SCK-10

SCK-10 13358 B 2010-05-26 09:38:21

SIA-8 17291 B 2010-02-19 09:45:01

SP-2 8534 B 2010-02-19 09:45:01

TRS-01a 13358 B 2010-05-26 09:37:46

TRS-02a 13358 B 2010-05-26 09:48:35

TRS-11a 13358 B 2010-05-26 09:47:44

Fig. 7.126. Template selection window

This button invokes software keyboard window allowing write or search the template name.

This button allows the user to delete selected template.

⬆ ⬇ Navigation keys allows the user to select appropriate template.

7.15.2.2. Modbus SLAVE - Device channels for SLAVE mode

Device channels parameter block has 2 buttons:

- **Output list** – submenu where the user can configure registers of available logical channels. In this submenu the user decide which registers are read only or read and write.
- **Avail. output blocks** – 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:

⬇ 3 ⬆ 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.

<u>Output list</u> <u>submenu</u>

This menu consists of following fields:

- **Value register** – submenu where the details of the Modbus register can be set,
    - **Register type**, a user can select two types:
        - **HOLDING** - holding registers of SLAVE device compatible with the Modbus protocol
        - **INPUT** - input registers of 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 parameters see **Chapter 7.15.2.4. 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, 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** means: **A** - most significant byte of high words (word = 2 bytes), **B** - least significant byte of the high words, **C** - most significant byte of low words, **D** - least significant byte of the 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 consist of following fields:

- **Block type** - read only parameter, 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 be also used for 32-bit registers reading/preseting. In a such 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,

ℹ **HOLDING** registers can be set as **read** or **read and write**. That is why when **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.

7.15.2.3. Modbus SLAVE - The Modbus protocol handling

Parameters for Modbus RTU implemented in MultiCon

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

7.15.2.4. Modbus SLAVE - List of registers

The device parameters and measurement result are available via 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 03h function, and written by 06h (single registers) or 10h (group of the registers) accordingly to Modbus RTU and TCP/IP specification.

Register	*Write*	*Range*	*Register description*
20h	No	0÷199	Address of device
21h	No	2050h, 2060h	Device identification code **2050h** - device in big housing (5.7” display), **2060h** - device in small housing (3.5” display).
Measurements results (floating point format) [1]			
200h	Yes	0÷0FFFFh	Measurement result for logical channel 1 (high word)

Register	Write	Range	Register description
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 not available (e.g. logical channel not configured)
203h	Yes	0÷6	Decimal point for logical channel 1
Register from 204h to 2F0h			Measurement results, 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)
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 not available (e.g. logical channel not configured)
403h	No	0÷6	Decimal point for logical channel 1
Register from 404h to 4F0h			Measurement results, status and decimal point for Logical Channels 2÷60

[1] IEEE 754 standard, Float point format represents data as precision as possible. Integer 32 represents value with constant precision,selected 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 pint is 0.000 then integer represents integer part of the value contained in float registers and multiplied by 1000 (e.g.: float is 1.2345, D.P = 0.0, then Integer = 1234)

Tab. 7.7 List of register available in the device

7.15.2.5. Modbus SLAVE - Transmission errors handling

If during reading or writing one of registries an error occurs 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, 06h and 10h are available),
02h - illegal register address
03h - illegal data value

7.15.2.6. Modbus SLAVE- Example of query/answer frames

The examples concern a unit with address 1. All 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, to which the command refers to
COUNT H,L	Higher and lower part of registry counter number, to which 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	60	A1	AC

DATA H,L - identification code (2060h)

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 being read registers (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 available.

7.15.3. Modbus - MASTER mode

The parameters of a **Modbus** protocol for **MASTER** mode are:

- **Mode = MASTER**,
- **Baud rate** - this parameter allows the user to select 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 the **Tab. 7.8**.

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. 7.8 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 operations with Slave devices. If the Master device requests all Slave devices in shorter time than the interval, then new requests will begin after Interval elapses.
- **Slave device** - this Button enters submenu allows to define the list of Slave devices connected to the current serial port of MultiCon and configure registers for read and/or write. See below for more informations about this menu.

SLAVE device menu

MultiCon allows to address as many as 255 slave devices on the addresses from 1 to 255. To define external data source, first an address must be chosen, next further parameters of the SLAVE device (having this address) set.

Arrows placed in the upper right corner of the screen allow switching between addresses of SLAVE devices. The middle button allows direct selection of specific address from the list.

If the particular address is not used, then a *short menu* is displayed:

- **Device type –** turns on and off selected address,
  - **not present** - means that this address is not used (in other words - there is no SLAVE device with this address connected),
  - **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 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 using a **File Management** menu and also created by the user using **Save device template** button after fully configuration of a slave device.

If the **Device type** is set to **defined** or a Template has been loaded, then *extended SLAVE device menu* is shown. This menu as following fields:

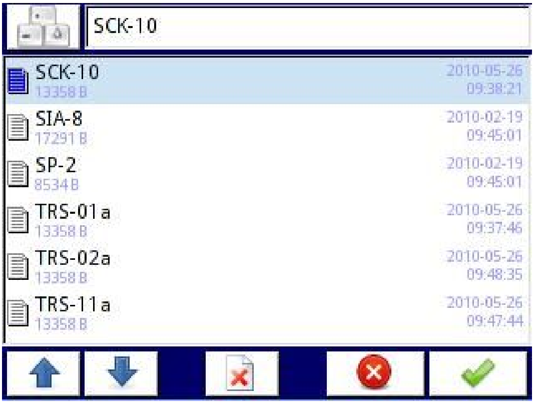
- **Device type = defined**,
- **Device name** - to create or change the name of a SLAVE device, press the button next to the **Device name** label and enter the name using displayed editor,
- **Device templates** parameter block – see **Chapter 7.15.3.1** for extended description,
- **Device channels** parameter block - see **Chapter 7.15.3.2** for extended description,
- **Register blocks** parameter block - see **Chapter 7.15.3.4** for extended description,

7.15.3.1. 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 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 using a **File Management** menu and also created by the user using **Save device template** button after fully configuration of a slave device
- **Save device template** - allows the user to save a configured SLAVE device as a template for further usage. Saved template can be used for:
  - fast copying/moving of the SLAVE device to another address (use **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 devices (using **File Management** menu).

An example of Modbus template selection window is shown in **Fig. 7.127**. Below this figure additional icons are described.



SCK-10

SCK-10	13358 B	2010-05-26 09:38:21
SIA-8	17291 B	2010-02-19 09:45:01
SP-2	8534 B	2010-02-19 09:45:01
TRS-01a	13358 B	2010-05-26 09:37:46
TRS-02a	13358 B	2010-05-26 09:48:35
TRS-11a	13358 B	2010-05-26 09:47:44

Fig. 7.127. Template selection window

This button invokes software keyboard window allowing write or search the template name.

This button allows the user to delete selected template.

↑ ↓ Navigation keys allows the user to select appropriate template.

7.15.3.2. Modbus MASTER - Device channels parameter block

This block includes following buttons:

- **Input list** – submenu , where the user can define logical channels with chosen write registers,
- **Output list** – submenu, where the user can define logical channels with chosen read registers,

Both these submenus has basic icons presented below. Their functions are as follow:

+ This button allows the user to add a new Input/Output to Input list / Output list,

− This button allows the user to delete the Input/Output from Input list / Output list

↓ 3 ↑ Arrows placed in the upper right corner of the screen allows user to switch between Inputs/Outputs. The middle button moves directly to specific Input/Output channel selected from the list.

Input List submenu.

When at least one Input Channel is added to the **Input list**, then an Input channel menu is displayed. This menu consist of following fields:

- **Channel value** parameters block composed of buttons:
  - **Value register** – menu where the details of the Modbus register serving data of input being can be set - see **Chapter 7.15.3.3. Modbus MASTER - Register settings**.
  - **Decimal point** – list of available decimal point positions, in a last place there is an option: *** exp (-point register).** 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 decimal point position. Using this parameter, the **Channel value** is being displayed according to formula:

$$(data\ of\ \ Value\ register)\cdot 10^{(-Decimal\ \ point\ register)}$$

See Chapter **7.15.3.3. Modbus MASTER - Register settings** for more info.

- **Channel status '-HI-'** parameter block – allows to define when a status **-HI-** should be displayed (returned) in a place of numerical value of Input Channel,
  - **-HI- state** – defines condition for **-HI-** state appearance:
    - **never** – do 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 to **'-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** - allow 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,

When a **Logical channel** is configured to **Modbus** mode then while reading the registers from 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 an 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 **7.11. EXTERNAL OUTPUTS**),
  - **yes** – the output channel is defined and visible in the **External output** menu (see Chapter **7.11. EXTERNAL OUTPUTS**).

- **Control type**
  - **as a relay** - the output has two state, 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 of the settings in **Output register** and settings the parameters in the **External outputs** menu (see **Chapter 7.11. EXTERNAL OUTPUTS**),
- **Output register** submenu - menu where user can set details of the Modbus register to be written - see **Chapter 7.15.3.3. Modbus MASTER - Register settings**.

7.15.3.3. 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), a 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, 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 means: A - most significant byte of high words (word = 2 bytes), B - least significant byte of the high words, C - most significant byte of low words, D - least significant byte of the low words
  - **ABCD (standard)**
  - **CDAB**,
  - **DCBA**,
  - **BADC**,

- **Data shift** - values can be bit moved to the right any integer value in the range (parameter does not exists 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 is means that the entire value of register is visible, while the 0x0 mask (no mask) means that the value is zero,

7.15.3.4. 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 manual mode.

Register blocks group has following fields:

- **Blocks config. mode**
  - **automatic** - the device automatically creates a blocks of registers to be read using the list defined in **Device channels** sub-menu. Then the **Block list** is informal only and cannot be edited.
  - **manual** - user must create a list of registers blocks using **Block list** parameter
- **Maximum block size** - occurs only for the **Blocks config.mode = automatic.** This parameter allows user to limit number of data registers to be read at once. It can be very useful when SLAVE devices has a limitation of max. number of registers read in a single frame.
- **Block list** – occurs only for **manual** mode, parameter description bellow in this chapter,
- **Avail. Input blocks** – read only submenu, allows to preview automatically or manually created input blocks,
- **Avail. Output blocks** – read only submenu, allows to preview automatically or manually created output blocks,

Block list submenu

➕	This button appears only for manual configuration mode and allows the user to add a new block of registers to list of register blocks.
➖	This button appears only for manual configuration mode and allows the user to remove the block of registers from the list of register blocks.
⬇ 3 ⬆	Arrows placed in the upper right corner of the screen allow switching between register blocks. Middle button allows direct selection of a specific register block.

To 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 be also used for 32-bit registers reading/preseting. In a such 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 the **Chapter 7.15.3.3. 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 single block with one 16-bit register the parameter **First register** and **Last register** must be the same number register,

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 **Device channels** parameter block of the device (see above in this Chapter). If user creates a block of registers to read / preset in which there were not registers appearing on the Input list and Output list in the **Device channels** parameter this device in the case of:

- **read** - the read whole register block and registers undefined in Input list menu will not be visible in the Logical channel in Modbus mode which cannot be read either a data from this registers and to use this registers to control and regulating process,

- **preset** - will send the frame to preset the entire block of registers and registers which not defined in the **Output list** will not appear on the **External output** menu

Warning! in this case to Slave devices will be sent to a random value of these registers, uncontrolled by the user,

7.15.4. Modbus - Example of Modbus protocol configuration in the device

Modbus protocol sends data in floating point format. In this connection user should keep in mind that transmitted and received values using Modbus protocol, can be slightly different. This situation can happened when decimal value can not be represented as floating point numbers, or when the least significant digit is eight orders of magnitude smaller than most significant digit.

Example 7.15.4.1**:** Input configuration of Modbus protocol in MASTER mode

Task:
The task is to configure a logical channel to read Slave registers (eg temperature converter). We read register 1 which contains temperature, register 2 is a status register, register 3 is a decimal point register. The devices has addresses 1 and 8. To set Modbus configuration for the same devices, 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 (in accordance with **Fig. 7.128**).

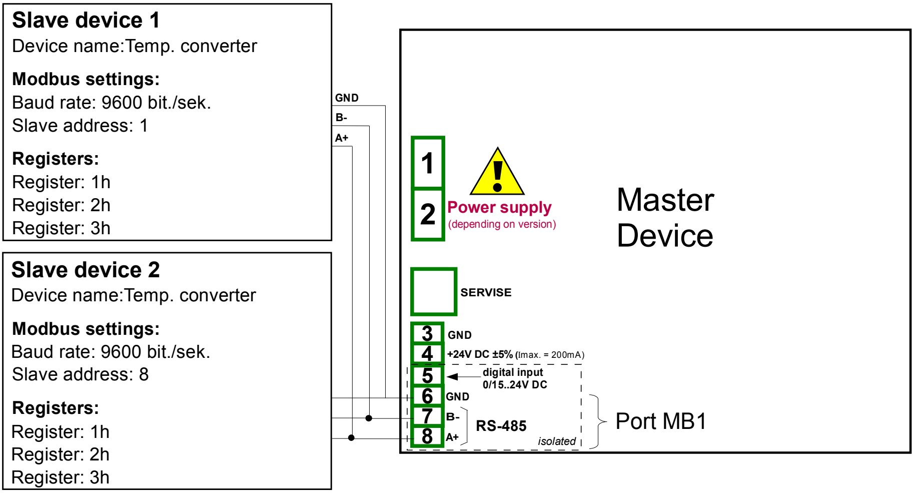


Fig. 7.128. Connection diagram for the Modbus port MB1

Description of the register of SLAVE device shown in the **Tab. 7.9**.

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. 7.9 Description of the registers of SLAVE device

In first step:

- touch screen and press the **Menu** button,
- press the **Device configuration** button,
- enter the **Modbus** menu,
- using the arrows in upper right corner choose **Port number**: **1** (**MB1** in accordance with **Fig. 7.128**),
- **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 **Slave devices** button:
    - using the arrows in upper right corner select Slave device **Modbus address**, in this case it is **1**,
    - **Device type** parameter set as **defined**,
    - in **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 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 format depends on SLAVE device register format,
            - **Data shift: 0**, because we do not want to move data read from Slave device,
            - **Data mask: 0xFFFF**, which reads the whole value of register, exit the **Value register** parameter menu,

  - in the **Decimal point** parameter select option: * exp (-point register) causes that will be appear a new parameter to appear - **Decimal point register** that causes an automatic change of decimal point, depending on 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 the interest to us only the first 2 bits changing register value, the value of which determines the decimal point, exit after the settings this menu,

- **-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 frame (using the Modbus function 03h) with up to 5 registers,

- move to the **Device templates** parameter block and press **Save device template** button,
  - in the **Save device template** menu, enter the template name, eg **Temperature converter** (stored template can be used to configure the device to a different address),

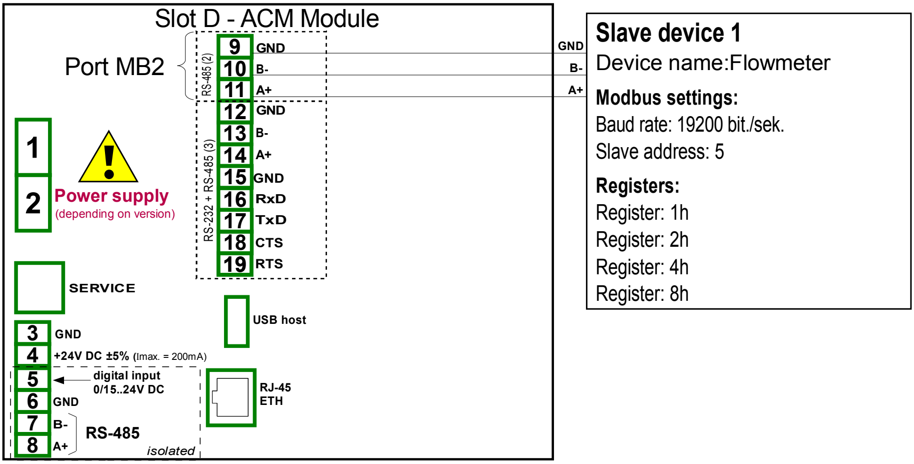
- using arrows in the upper right corner of the screen, go to the **Modbus address**: **8**,
- press the **Load device template** button and in the template list select the template named **Temperature converter**, in such a fast way to have set all the parameters for the **SLAVE device 2** identical to **SLAVE device 1**,
- exit from the **SLAVE devices** menu,

Example of Logical channel in Modbus mode is shown in **Example 7.8.11.5**.

***Example 7.15.4.2*: Configuration of the Modbus Input in the MASTER mode.**

Task:
The task is to configure Slave device (eg 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 ACM module (see **Appendix 8.15. COMMUNICATION MODULES**), in which the port MB2 is used.

Solution:
First we need to configure the device according to the following recommendations, then connect the device according to **Fig. 7.129**.



Slave device 1
Device name:Flowmeter
Modbus settings:
Baud rate: 19200 bit./sek.
Slave address: 5
Registers:
Register: 1h
Register: 2h
Register: 4h
Register: 8h

Fig. 7.129. Connection diagram for the Modbus port MB2

16-bits Slave device registers description is shown if **Tab. 7.10**.

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,

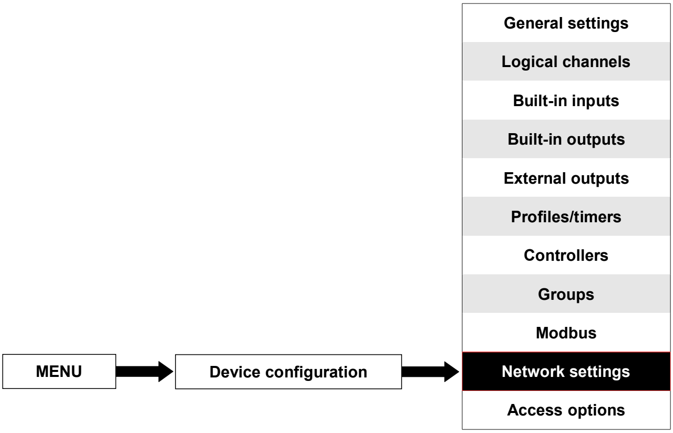
Tab. 7.10 Description of 16-bits Slave device registers

In first step:

- touch screen and press the **Menu** button,
- press the **Device configuration** button,
- enter the **Modbus** menu,
- using the arrows in upper right corner choose **Port number**: **1** (**MB1** in accordance with **Fig. 7.129**),
- **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 **Slave devices** button:
    - using the arrows in upper right corner select **Modbus address** of Slave device, in this case: **5**,
    - **Device type** parameter set as **defined**,
    - in **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 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 register format in Slave device),
            - **32-bit reading** parameter set as **two 16-bit registers**,
            - **Ordering** parameter set as **CDAB**, because first the younger word part is send (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 register format in 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 register format in 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 registers format in 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 registers format in 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 registers format in Slave device),
    - in **First register** parameter write **8h**,
    - in **Last register** parameter write **8h**,
- move to the **Device templates** parameter block and press **Save device template** button,
    - in the **Save device template** menu, enter the template name, eg **Write do device** (stored template can be used to configure the device to a different address),

7.16. NETWORK AND REMOTE DISPLAY SETTINGS



This button is visible only for device which is equipped in ACM or ETU module.

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 and to display any data with an external PC.

Parameters of the **Network settings** menu are:

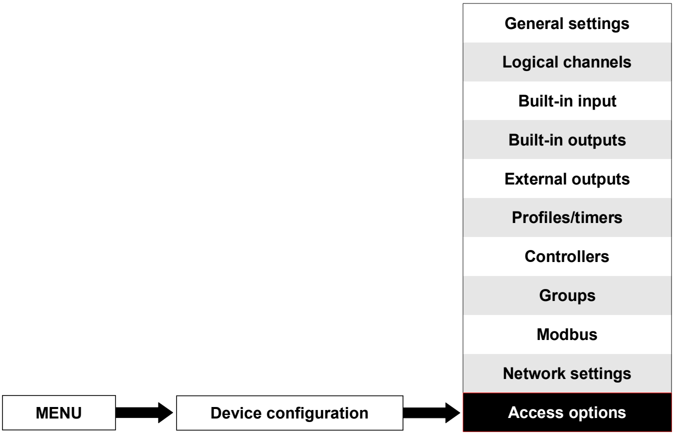
- **DHCP** (Dynamic Host Configuration Protocol) - allows a device to be configured automatically, eliminating the need for intervention by a network administrator,
  - **disabled** - **DHCP** is disabled, the user needs to manual enter an **IP address** and **Subnet mask** in the following fields, and 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 user set this option other parameters in this menu is invisible,
- **IP address** - this parameter is visible only if **DHCP** parameter is set as **disabled**, the user may enter a static IP address,
- **Subnet mask** - this parameter is visible only if **DHCP** parameter is set as **disabled**, sets a range of IP addresses that can be accessed,
- **Default gateway** - this parameter is visible only if **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.

- **Remote display** parameter block - this parameters let the user configure the MultiCon with an external PC to display the screen of MultiCon on the PC monitor, parameters of this block are:
  - **IP address** - IP address of external PC,
  - **Screen number** – can bet set from 0-9 in the MultiCon allowing the PC to display up to 10 devices' screens,

If **DHCP** parameter is set as **disabled** the parameters **IP address** and **Subnet mask** must be configured correctly, depending on the local network settings which will work with the device. User should contact with network administrator in case of errors in communication.

The actual network settings are visible in the **Device Information** menu (see **Chapter 7.4.**).

7.17. ACCESS OPTIONS



To prevent accidental or unauthorized change of the settings, user can set single or **MultiLevel Access** protection. Both of them can be activated in **Access options** menu.

Single-level access protection requires to enter password (see **Fig. 7.130**) in order to login as the Admin and obtain access to all settings of the device.

MultiLevel Access protection requires the user to go through the Authentication procedure (see **Fig. 7.131**) in order to login and getting access to appropriate permission. It is done by choosing user (see **Fig. 7.132**) and passing his password.

Password can be activated in two different ways depending on level access protection. In the **Access options** menu by pressing button next to **Access password** label (for single-level mode) or in **Users list** menu (for **MultiLevel Access** mode) by typing user's password in text box (see **Fig. 7.1**, **Fig. 7.2**, **Fig. 7.3**). After closing the text box, in place of entered password 8 asterisks will be displayed independently to password length.

Removing the password is even easier, it is done by leaving the text box empty. After that, field next to **Access password** label will also be empty.

When the user passes through login process successfully on the information bar opened padlock with the letter “A” (if single level mode is set or the user is logged as Administrator in **MultiLevel Access** mode) appears (see **Fig. 7.134a**). Otherwise a number of user will be displayed (from 2 to 16) instead of opened padlock (see **Fig. 7.134b**). 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 incorrect password or attempt to enter to restricted function will cause denial (see **Fig. 7.135**).

At a time only one user can be logged in. The user can be logged out manually by touching the padlock icon on the information bar and confirming logout (see **Fig. 7.136**) or automatically after one minute for single level mode or after time set by the Administrator (in both cases the time is measured from the last user's interaction with the device).

For **MultiLevel Access** mode there is a possibility to login by connecting your own USB flash drive to the device, later called Hardware key. The Hardware key allows user to login without using password but only by plugging the key into the device. Removing the hardware key causes immediate logout. One key can be assigned to one user only and also user can only have one key. Hardware key options are available with a device having Access Dongle licence activated only (for more information about the license key see **Chapter 7.4. DEVICE INFORMATION, LICENCE, FIRMWARE UPDATE, REMOTE DISPLAY, EXPORT MANUAL AND RENEW CONFIGURATION**).

Access options parameters are:

- **Access mode** – allows to select protection mode from unauthorized usage,
  - **single level** – protects whole menu from unauthorized changes with one password. User which is currently logged in, can freely configure the device until 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 single-level access mode (unless it's left empty),
- **Access timeout** – time in minutes which will elapse from last interaction with the device to automatic logout. Value of 0 means that the user will be logout immediately after closing protected menu. Each new attempt to entry into the protected menu will require a password once again; Available only for **MultiLevel Access** mode,
- **Users list** – menu containing Administrator's and user's access settings; Visible only for **MultiLevel Access** mode,
  - **Name** – used during authentication process,
  - **Password** – used for authentication selected 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 hardware key (currently connected flash disk) to selected user,
  - **Remove dongle** – button which removes earlier assigned hardware key,

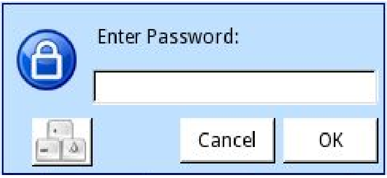
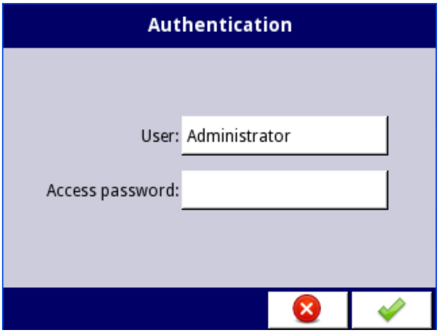


Fig. 7.130. Single-level password entering window

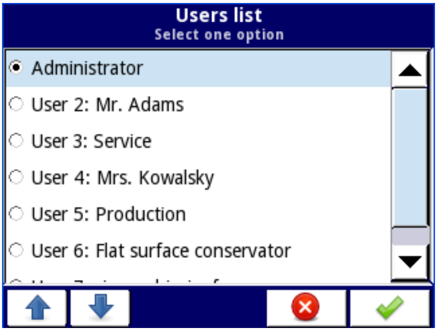


Authentication

User: Administrator

Access password:

Fig. 7.131. ***MultiLevel Access*** *password entering window*

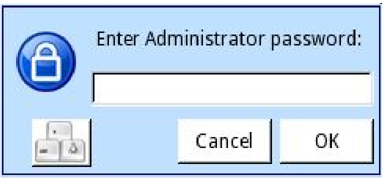


Users list

Select one option

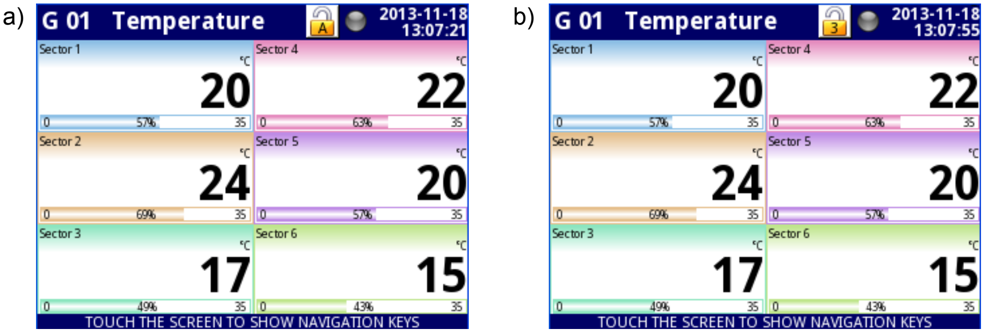
- ● Administrator
- ○ User 2: Mr. Adams
- ○ User 3: Service
- ○ User 4: Mrs. Kowalsky
- ○ User 5: Production
- ○ User 6: Flat surface conservator

Fig. 7.132. User selection window



Enter Administrator password:

Fig. 7.133. Administrator password window



a)

G 01 Temperature A 2013-11-18 13:07:21

Sector 1	°C 20	0	57%	35
Sector 2	°C 24	0	69%	35
Sector 3	°C 17	0	49%	35
Sector 4	°C 22	0	63%	35
Sector 5	°C 20	0	57%	35
Sector 6	°C 15	0	43%	35

TOUCH THE SCREEN TO SHOW NAVIGATION KEYS

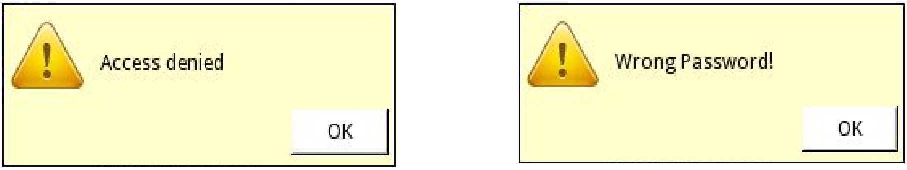
b)

G 01 Temperature 3 2013-11-18 13:07:55

Sector 1	°C 20	0	57%	35
Sector 2	°C 24	0	69%	35
Sector 3	°C 17	0	49%	35
Sector 4	°C 22	0	63%	35
Sector 5	°C 20	0	57%	35
Sector 6	°C 15	0	43%	35

TOUCH THE SCREEN TO SHOW NAVIGATION KEYS

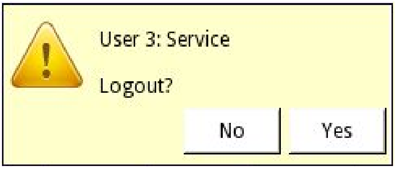
Fig. 7.134. Example of screen view with user logged in as:
a) Administrator b) User number 3



Access denied

Wrong Password!

Fig. 7.135. Access denied and Wrong password window



User 3: Service

Logout?

Fig. 7.136. User logout window

Button which calls the keyboard allowing to type user's password

Settings of single-level access protection are stored in main configuration file. They are also copied to another devices together with main configuration file.

Settings of **MultiLevel Access** protection have to be saved in independent file. They are not stored in main configuration file. They also cannot be removed by restoring default configuration as it is shown in **Chapter 7.4. 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 external memory as independent permission file (see **Chapter 7.3. FILES MANAGEMENT**).

8. APPENDICES

All connections must be made while power supply is disconnected !

8.1. PS3, PS4, PS42 - POWER SUPPLY MODULE

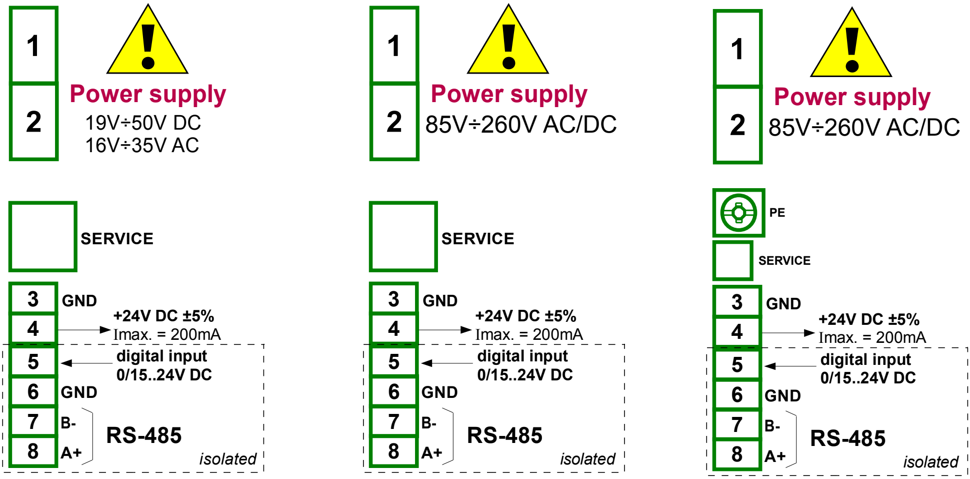


Fig. 8.1 Power supply module: PS3 (left side), PS4 and PS42

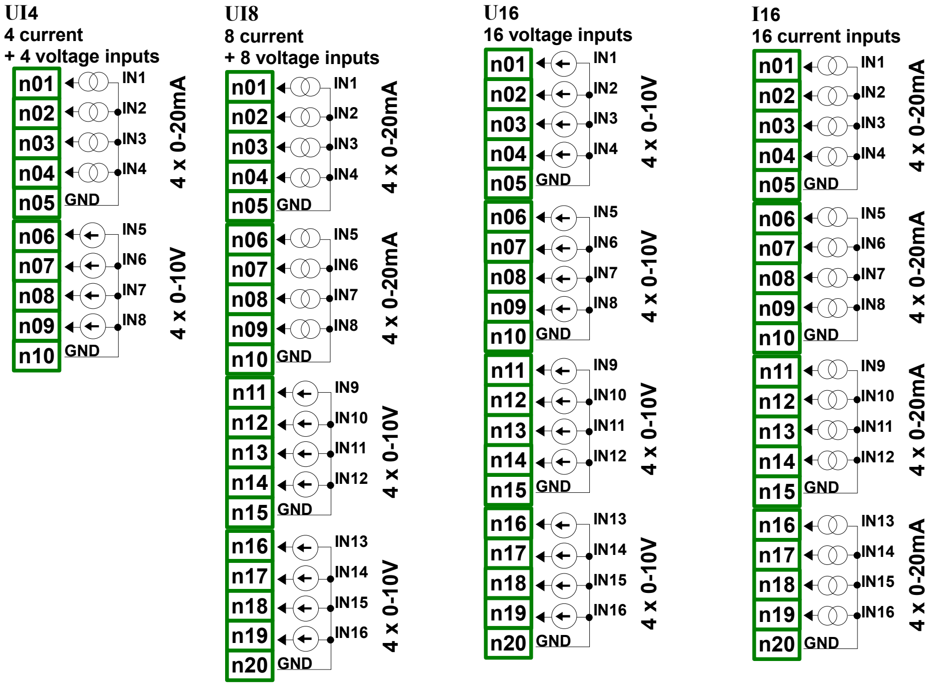
If the **UN3** or **UN5** module is installed, the +24V DC output is available only in devices equipped with **PS42** power supply module. In other cases there is **no +24V DC output** available and these terminals remain not used.

Most important parameters of PS modules:

	PS3	**PS4**	**PS42**
Number of inputs/outputs	5	5	5
Power supply	19V...**24**...50V DC 16V...**24**...35V AC	85V...**230**...260 AC/DC 50 60Hz	85V...**230**...260 AC/DC 50 60Hz
USB SERVICE	Service port		
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/sec. ÷ 115200 bit/sec.		
Weight	70g	69g	81g

Tab. 8.1 Technical data for the power supply module PS3, PS4 and PS42

8.2. UI4, UI8, UI12, U16, U24, I16, I24 – VOLTAGE AND CURRENT MEASUREMENT MODULES

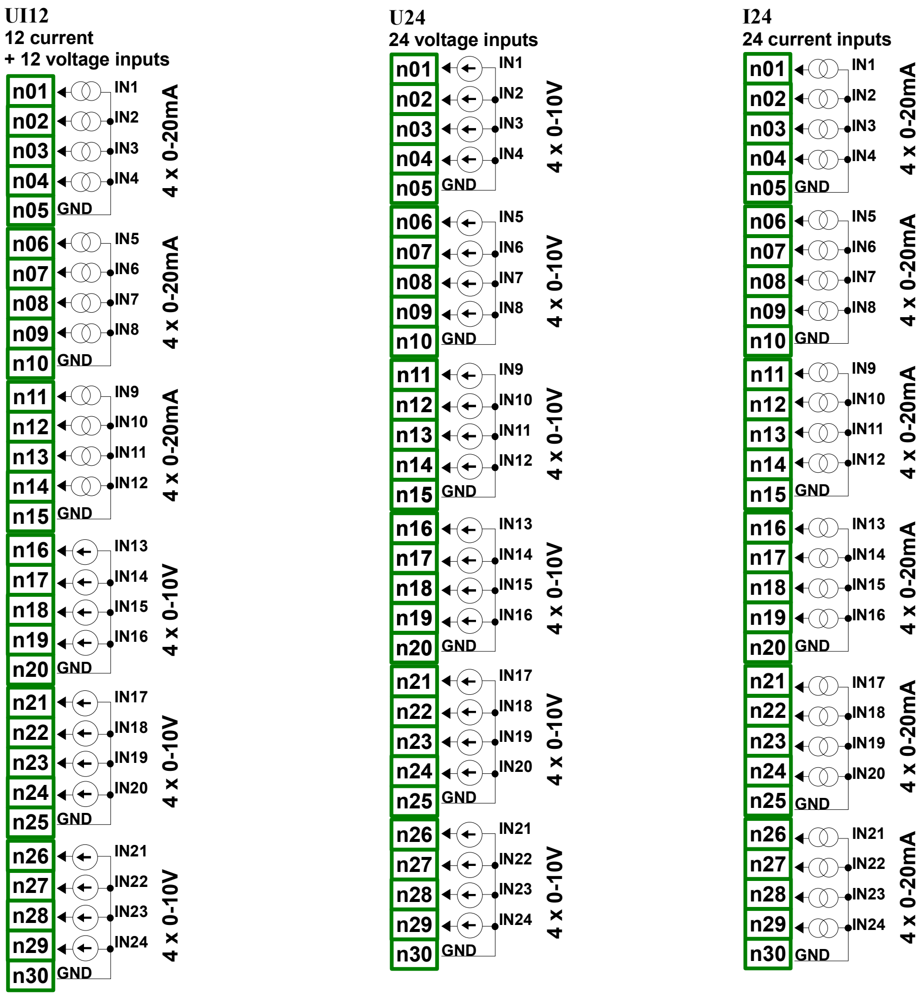

UI4
4 current
+ 4 voltage inputs

UI8
8 current
+ 8 voltage inputs

U16
16 voltage inputs

I16
16 current inputs

Fig. 8.2. Voltage and current measurement modules



UI12
12 current
+ 12 voltage inputs

U24
24 voltage inputs

I24
24 current inputs

Fig. 8.3. Voltage and current measurement modules contd.

The **Fig. 8.2** and **Fig. 8.3** shows terminals placement of UI modules. Inputs are gathered into groups to make connections easier. All ground terminals of a particular module are common, but separated from power supply and other modules. If it is necessary to measure Voltages with different ground potentials, several UI modules have to be installed into **CONTROLLER/DATA RECORDER** unit.

Most important parameters of UI modules:

	UI4		**UI8**		**U16**		**I16**	
Number of inputs	4xU + 4xI		8xU + 8xI		16xU		16xI	
Hardware limitations								
voltage inputs	0V ÷ 12V		0V ÷ 12V		0V ÷ 12V		-	
current inputs	0mA ÷ 24mA		0mA ÷ 24mA		-		0mA ÷ 24mA	
Hardware resolution								
voltage inputs	1mV		1mV		1mV		-	
current inputs	1μA		1μA		-		1μA	
Precision								
voltage inputs	0,1% @ 25˚C		0,1% @ 25˚C		0,1% @ 25˚C		-	
current inputs	0,1% @ 25˚C		0,1% @ 25˚C		-		0,1% @ 25˚C	
Covered CONTROLLER/DATA RECORDER measurement ranges *	type	max.	type	max.	type	max.	type	max.
	0÷5V	0÷6V	0÷5V	0÷6V	0÷5V	0÷6V	-	-
	1÷5V	0÷6V	1÷5V	0÷6V	1÷5V	0÷6V	-	-
	0÷10V	0÷12V	0÷10V	0÷12V	0÷10V	0÷12V	-	-
	2÷10V	0÷12V	2÷10V	0÷12V	2÷10V	0÷12V	-	-
	0÷20mA	0÷22mA	0÷20mA	0÷22mA	-	-	0÷20mA	0÷22mA
	4÷20mA	4÷22mA	4÷20mA	4÷22mA	-	-	4÷20mA	4÷22mA
Temperature stability	50 ppm/˚C		50 ppm/˚C		50 ppm/˚C		50 ppm/˚C	
Internal impedance								
voltage inputs	50kΩ		50kΩ		50kΩ		-	
current inputs	type 100Ω		type 100Ω		-		type 100Ω	
Protection								
voltage inputs	no		no		no		-	
current inputs	50mA auto-reset fuse		50mA auto-reset fuse		-		50mA auto-reset fuse	
Sampling period	10ms		10ms		10ms		10ms	
Isolation strength	1 min. @ 500V AC		1 min. @ 500V AC		1 min. @ 500V AC		1 min. @ 500V AC	
Weight	32g		43g		42g		42.5g	

* Measurement ranges are limited by software upon hardware inputs ability, check current list of **CONTROLLER/DATA RECORDER** measurement ranges at producer's website

Tab. 8.2 Technical data for the module UI4, UI8, U16, I16

Most important parameters of UI modules contd.:

	UI12		**U24**		**I24**	
Number of inputs	12xU + 12xI		24xU		24xI	
Hardware limitations **voltage inputs** **current inputs**	0V ÷ 12V 0mA ÷ 24mA		0V ÷ 12V -		- 0mA ÷ 24mA	
Hardware resolution **voltage inputs** **current inputs**	1mV 1μA		1mV -		- 1μA	
Precision **voltage inputs** **current inputs**	0,1% @ 25˚C 0,1% @ 25˚C		0,1% @ 25˚C -		- 0,1% @ 25˚C	
Covered CONTROLLER/DATA RECORDER measurement ranges *	type	max.	type	max.	type	max.
	0÷5V 1÷5V 0÷10V 2÷10V 0÷20mA 4÷20mA	0÷6V 0÷6V 0÷12V 0÷12V 0÷22mA 4÷22mA	0÷5V 1÷5V 0÷10V 2÷10V - -	0÷6V 0÷6V 0÷12V 0÷12V - -	- - - - 0÷20mA 4÷20mA	- - - - 0÷22mA 4÷22mA
Temperature limitations	50 ppm/˚C		50 ppm/˚C		50 ppm/˚C	
Internal impedance **voltage inputs** **current inputs**	50kΩ type 100Ω		50kΩ -		- type 100Ω	
Protection **voltage inputs** **current inputs**	no 50mA auto-reset fuse		no -		- 50mA auto-reset fuse	
Sampling period	10ms		10ms		10ms	
Isolation strength	1 min. @ 500V AC		1 min. @ 500V AC		1 min. @ 500V AC	
Weight	62g		62.5g		63g	

* Measurement ranges are limited by software upon hardware inputs ability, check current list of **CONTROLLER/DATA RECORDER** measurement ranges at producer's website

Tab. 8.3 Technical data for the module UI12, U24, I24

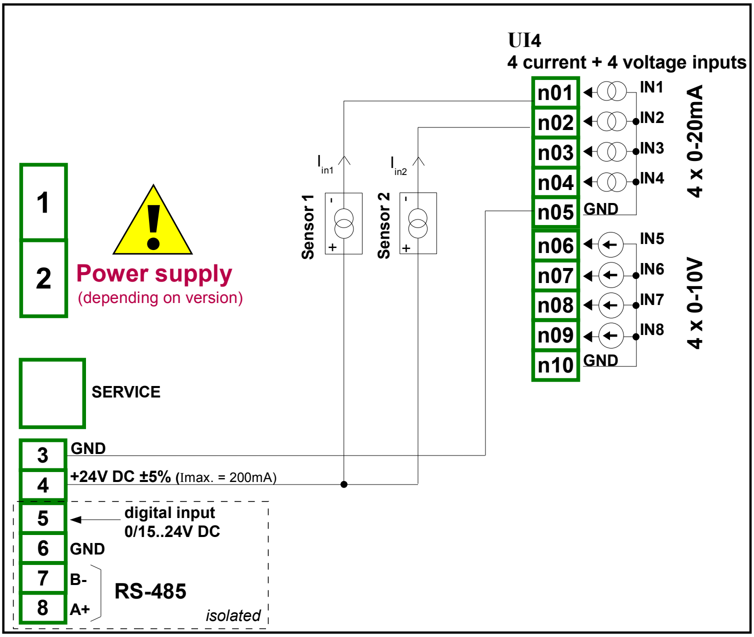


Fig. 8.4. Connections for 2 - wire sensor (current)

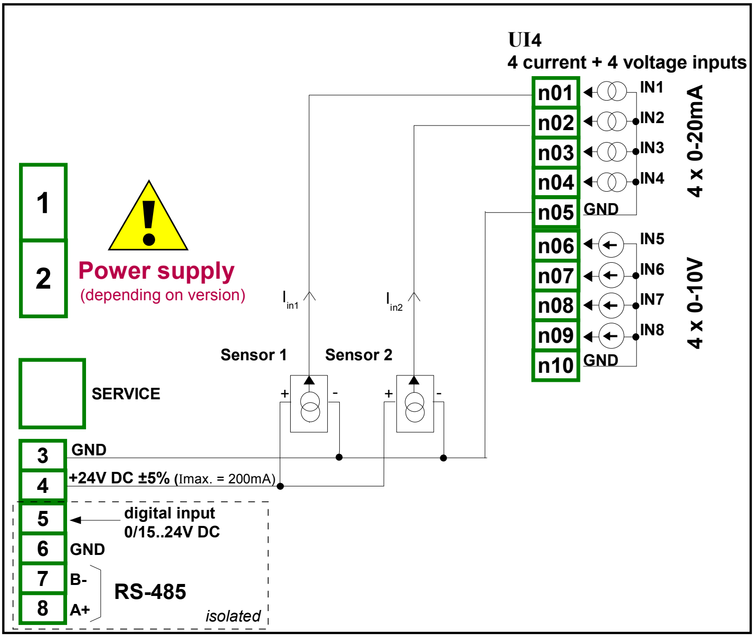


Fig. 8.5. Connections for 3 - wire sensor (current)

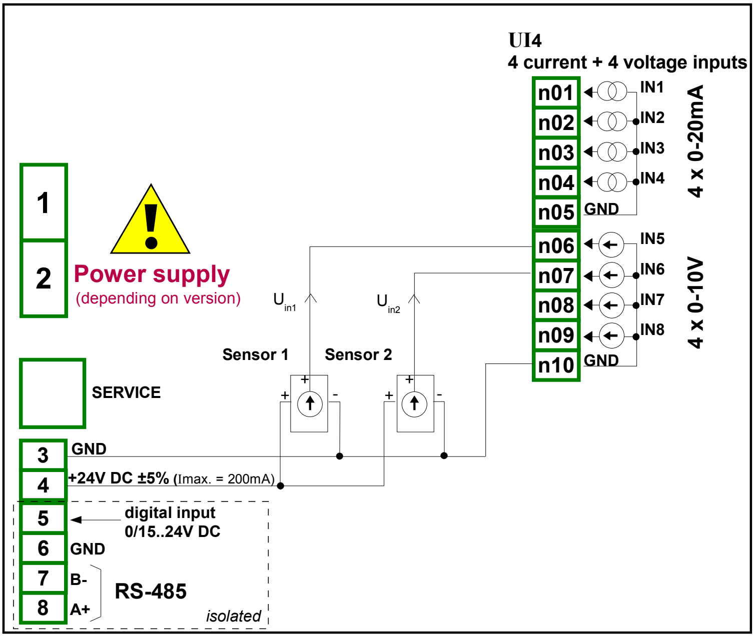


Fig. 8.6. Connections for 3 - wire sensor (voltage)

8.3. UI4N8, UI4D8, UI8N8, UI8D8 – MIXED UIN/UID MODULES

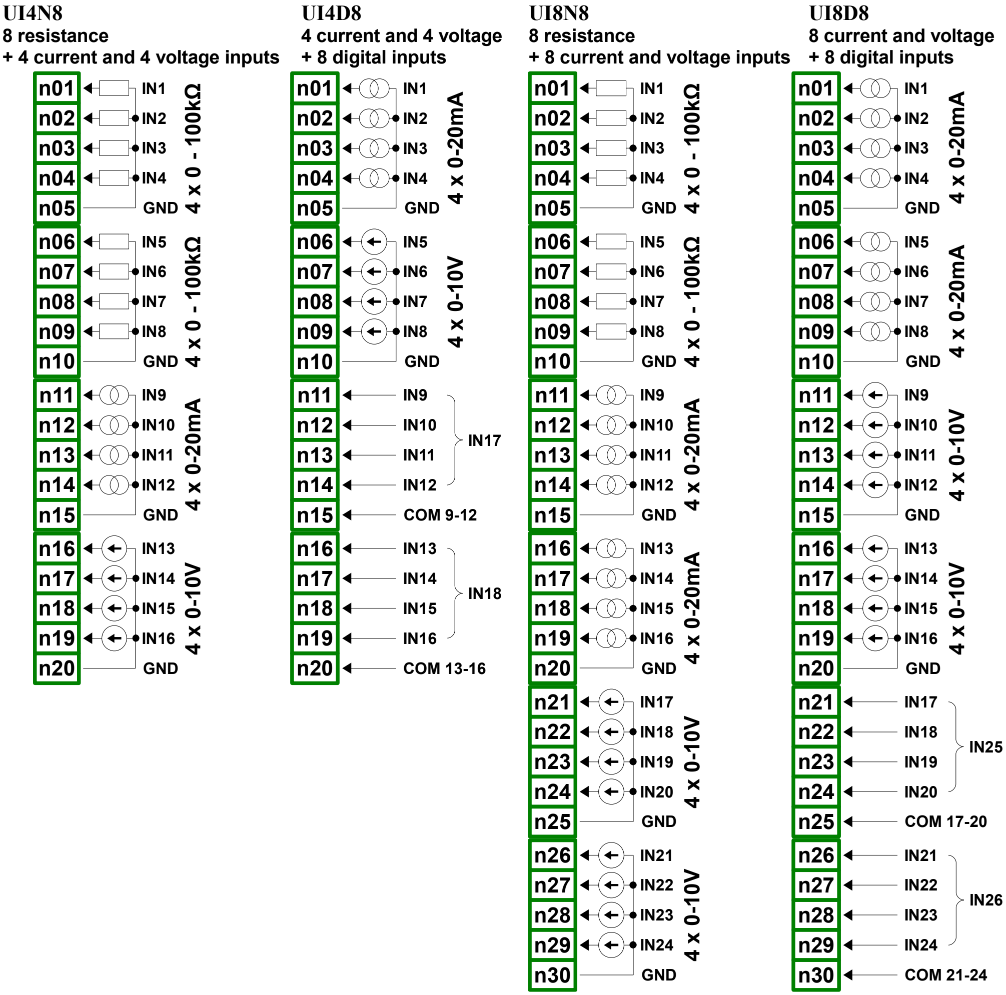


Fig. 8.7. Mixed measurement modules

The **Fig. 8.7** shows terminals placement of UIN/UID modules. Inputs are gathered into groups to make connections easier. All ground terminals of a particular module are common, but separated from power supply and other modules. If it is necessary to make measurements with different ground potentials, several UIN/UID modules have to be installed into **CONTROLLER/DATA RECORDER** unit.

Most important parameters of UIN/UID modules:

	UI4N8		UI4D8		UI8N8		UI8D8	
Number of inputs	8x NTC (resistive) + 4x I + 4x U		4x I + 4x U + 8x D		8x NTC (resistive) + 4x I + 4x U		4x I + 4x U + 8x D	
Hardware limitations								
voltage input	0 ÷ 12V		0 ÷ 12V		0 ÷ 12V		0 ÷ 12V	
current input	0 ÷ 24mA		0 ÷ 24mA		0 ÷ 24mA		0 ÷ 24mA	
resistance input	0 ÷ 110kΩ		-		0 ÷ 110kΩ		-	
digital input	-		0 ÷ 32V		-		0 ÷ 32V	
Hardware resolution								
voltage input	1mV		1mV		1mV		1mV	
current input	1µA		1µA		1µA		1µA	
resistance input	4Ω		-		4Ω		-	
digital input	-		-		-		-	
Precision								
voltage input	0.1% @ 25˚C		0.1% @ 25˚C		0.1% @ 25˚C		0.1% @ 25˚C	
current input	0.1% @ 25˚C		0.1% @ 25˚C		0.1% @ 25˚C		0.1% @ 25˚C	
resistance input	0.1% @ 25˚C		-		0.1% @ 25˚C		-	
digital input	-		2% @ 25˚C		-		2% @ 25˚C	
Covered CONTROLLER/DATA RECORDER measurement ranges*	type	max.	type	max.	type	max.	type	max.
voltage input	0÷5V 1÷5V 0÷10V 2÷10V	0÷6V 0÷6V 0÷12V 0÷12V	0÷5V 1÷5V 0÷10V 2÷10V	0÷6V 0÷6V 0÷12V 0÷12V	0÷5V 1÷5V 0÷10V 2÷10V	0÷6V 0÷6V 0÷12V 0÷12V	0÷5V 1÷5V 0÷10V 2÷10V	0÷6V 0÷6V 0÷12V 0÷12V
current input	0÷20mA 4÷20mA	0÷20mA 4÷20mA	0÷20mA 4÷20mA	0÷22mA 4÷22mA-	0÷20mA 4÷20mA	0÷20mA 4÷20mA	0÷20mA 4÷20mA	0÷22mA 4÷22mA-
resistance input	0÷100kΩ	0÷110kΩ	-	-	0÷100kΩ	0÷110kΩ	-	-
TTL digital input	-	-	Lo: 0÷0.8V Hi: 2÷5.5V	-	-	-	Lo: 0÷0.8V Hi: 2÷5.5V	-
HTL digital input	-	-	Lo: 0÷4.2V Hi: 11.5÷30V	-	-	-	Lo: 0÷4.2V Hi: 11.5÷30V	-
Temperature stability	50 ppm/˚C		50 ppm/˚C		50 ppm/˚C		50 ppm/˚C	
Input impedance								
voltage input	61kΩ		61kΩ		61kΩ		61kΩ	
current input	100Ω		100Ω		100Ω		100Ω	
resistance input	121kΩ		-		121kΩ		-	
digital input	-		80kΩ		-		80kΩ	
Protection								
voltage input	protection resistor		protection resistor		protection resistor		protection resistor	
current input	50mA auto-reset fuse		50mA auto-reset fuse		50mA auto-reset fuse		50mA auto-reset fuse	
resistance input	protection resistor		-		protection resistor		-	
digital input	-		protection resistor		-		protection resistor	
Sampling period	ca 100ms		ca 100ms		ca 100ms		ca 100ms	
Isolation strength	1 min. @ 500V AC		1 min. @ 500V AC		1 min. @ 500V AC		1 min. @ 500V AC	
Weight	51.5g		51.5g		63.5g		63.5g	

* Measurement ranges are limited by software upon hardware inputs ability, check current list of **CONTROLLER/DATA RECORDER** measurement ranges at producer's website

Tab. 8.4 Technical data for the UI4N8, UI4D8 UI8N8 and UI8D8 modules

8.4. IS6 – ISOLATED CURRENT INPUTS MODULE

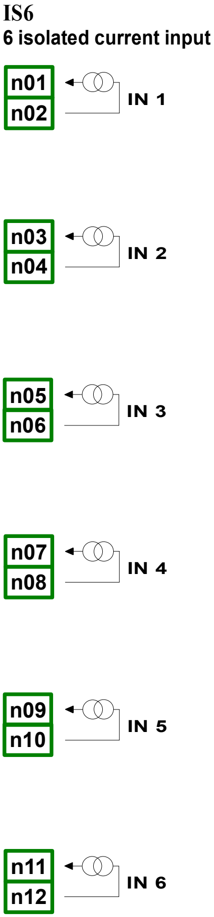


Fig. 8.8.Isolated current input module

Most important parameters of IS6 module:

	IS6
Number of inputs	6xI
Hardware limitations **current inputs**	3mA ÷ 30mA
Hardware resolution **current inputs**	1µA
Precision **current inputs**	0.25% @ 25°C

	IS6	
Covered CONTROLLER/DATA RECORDER measurement ranges *	type	max.
	4÷20mA	3÷30mA
Internal impedance **current inputs**	type 1750Ω @ 4mA; type 400Ω @ 20mA	
Temperature stability	65 ppm/°C	
Input voltage dropout	<9V	
Signal polarization	irrelevant (absolute value measurement)	
Protection **current inputs**	50mA auto-reset fuse	
Sampling period	100ms	
Isolation strength	1 min. @ 500V AC	
Weight	39g	

* Measurement ranges are limited by software upon hardware inputs ability, check current list of **CONTROLLER/DATA RECORDER** measurement ranges at producer's website

Tab. 8.5 Technical data for the IS6 module

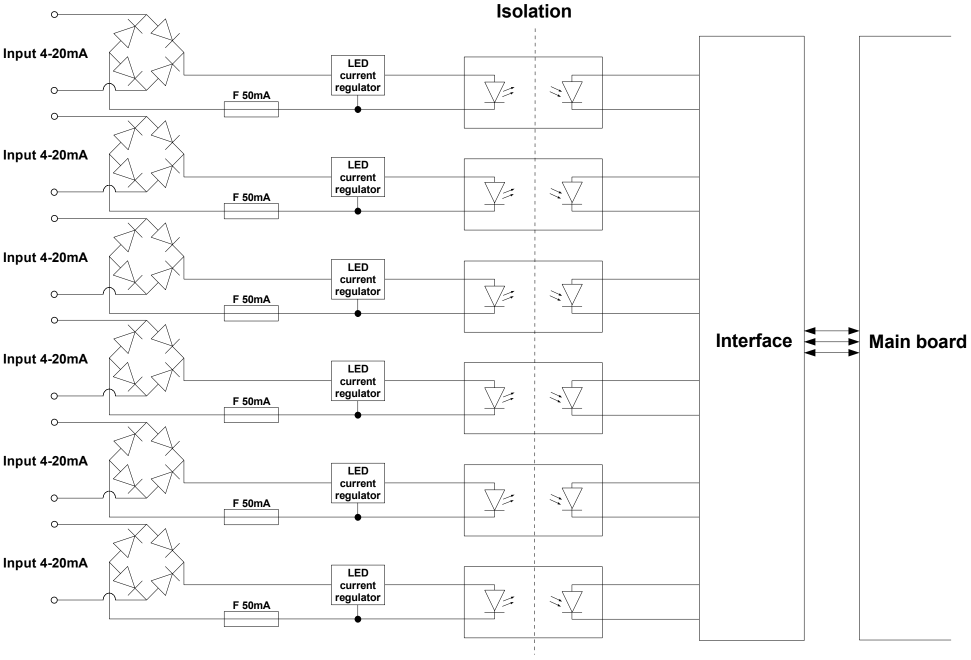


Fig. 8.9. Internal structure of the optoisolated current input module

8.5. FI2. FI4, FT2, FT4 – FLOWMETER MODULES

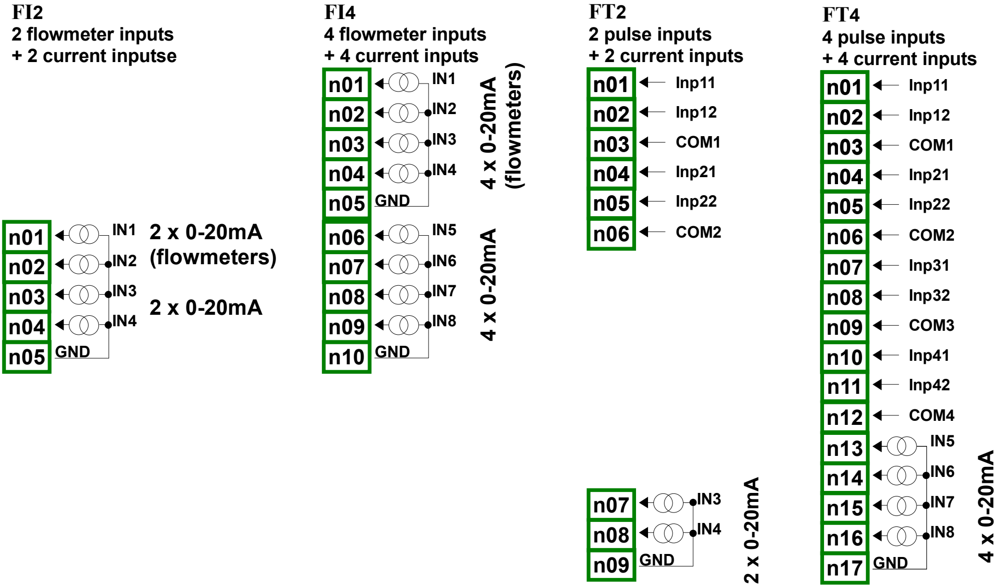


Fig. 8.10 Flowmeter modules

Most important parameters of F modules:

	FI2		FI4		FT2		FT4	
Number of inputs	2xF (flowmeters) + 2xI		4xF (flowmeters) + 4xI		2xT (tachometers) + 2xI		4xT (tachometers) + 4xI	
Hardware limitations **current inputs**	0mA ÷ 24mA		0mA ÷ 24mA		-2mA ÷ 30mA		-2mA÷ 30mA	
Hardware resolution **current inputs**	1µA		1µA		1µA		1µA	
Precision **current inputs**	0,1% @ 25˚C		0,1% @ 25˚C		0,1% @ 25˚C		0,1% @ 25˚C	
Covered CONTROLLER/DATA RECORDER measurement ranges *	type	max.	type	max.	type	max.	type	max.
	0÷20mA 4÷20mA	0÷22mA 4÷22mA	0÷20mA 4÷20mA	0÷22mA 4÷22mA	0÷20mA 4÷20mA	0÷20mA 4÷20mA	0÷20mA 4÷20mA	0÷22mA 4÷22mA
Temperature stability	50 ppm/˚C		50 ppm/˚C		50 ppm/˚C		50 ppm/˚C	
Internal impedance **current inputs**	type 100Ω		type 100Ω		type 100Ω		type 100Ω	
Protection **current inputs**	50mA auto-reset fuse		50mA auto-reset fuse		50mA auto-reset fuse		50mA auto-reset fuse	

	FI2	**FI4**	**FT2**	**FT4**
Input signals voltage levels: **logical LOW state** **logical HIGH state**	- -	- -	| Uin | < 1V | Uin | ≥ 10V	| Uin | < 1V | Uin | ≥ 10V
Max input voltage	5V	5V	30V	30V
Sampling period	50ms	50ms	50ms	50ms
Max. input frequency	-	-	50kHz	50kHz
Totalizer capacity	1 000 000 000 000	1 000 000 000 000	1 000 000 000 000	1 000 000 000 000
Isolation strength	1 min. @ 500V AC	1 min. @ 500V AC	1 min. @ 500V AC	1 min. @ 500V AC
Weight	36g	42.5g	42g	49g

* Measurement ranges are limited by software upon hardware inputs ability, check current list of **CONTROLLER/DATA RECORDER** measurement ranges at producer's website

Tab. 8.6 Technical data for the module FI2, FI4, FT2, FT4

8.6. TC4, TC8, TC12 – THERMOCOUPLE SENSOR MEASUREMENT MODULES

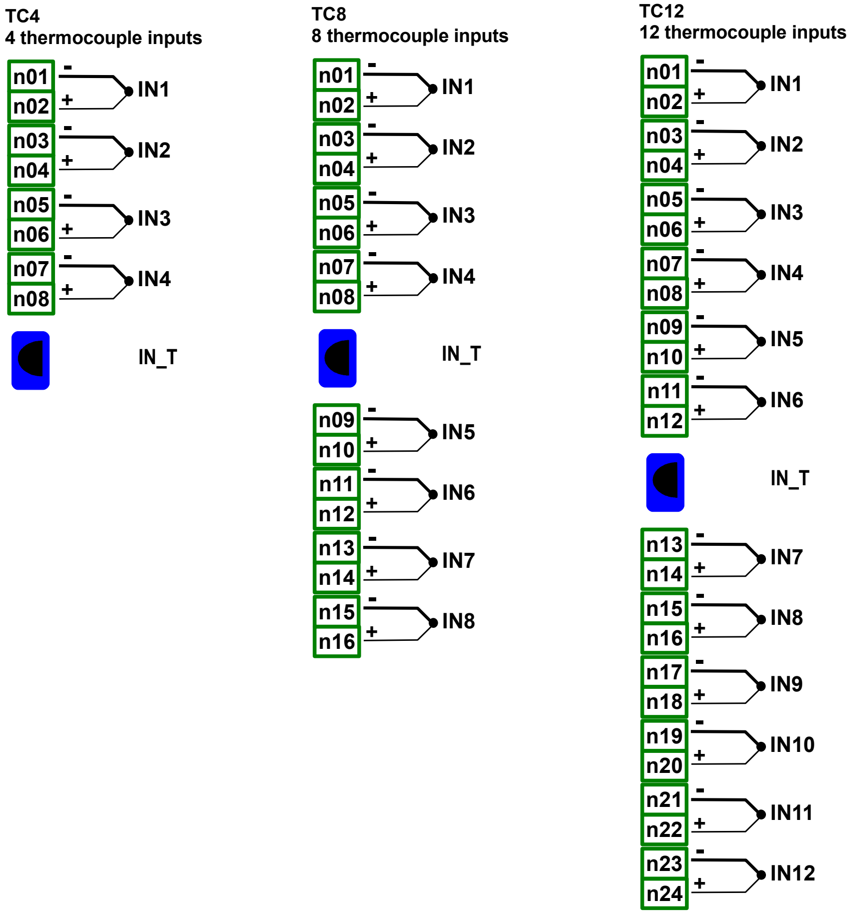


Fig. 8.11 Thermocouple module TC4, TC8, TC12

Most important parameters of TC4, TC8 and TC12 modules:

	TC4	**TC8**	**TC12**
Number of inputs	4	8	12
Hardware limitations	-30mV ÷ 30mV -120mV ÷ 120mV	-30mV ÷ 30mV -120mV ÷ 120mV	-30mV ÷ 30mV -120mV ÷ 120mV
Hardware resolution **range ± 30mV** **range ± 120mV**	1μV 4μV	1μV 4μV	1μV 4μV
Voltage measurement precision	0,1% @ 25°C	0,1% @ 25°C	0,1% @ 25°C
Permissible Long time overload	20%	20%	20%
Permissible voltage difference between channels **	0.5V	0.5V	0.5V
Covered CONTROLLER/DATA RECORDER measurement ranges * **thermocouple** **voltage**	K, S, J, T, N, R, B, E, L(GOST) -10÷25mV ±25mV -10÷100mV ±100mV		
Temperature stability	50 ppm/°C	50 ppm/°C	50 ppm/°C
Input impedance	typ. 1MΩ	typ. 1MΩ	typ. 1MΩ
Sampling period	385ms	385ms	385ms
Weight	32g	42g	52g

* Measurement ranges are limited by software upon hardware inputs ability, check current list of **CONTROLLER/DATA RECORDER** measurement ranges at producer's website

** Hi and Lo terminals of all inputs are pulled up/down by 470kW resistor to internal supply/GND. It is strongly recommended not to connect Lo or Hi terminals of different inputs together, but to connect every sensor using individual wires.

Tab. 8.7 Technical data for the module TC4, TC8, TC12

type	range	resolution	full range	full range accuracy		limited range	limited range accuracy	
	[mV]	[μV]	[°C]	[°C]	[%]	[°C]	[°C]	[%]
K	±120	4	-200÷1370	±8	±0.51	-100÷1370	±4	±0.27
S	±30	1	-50÷1768	±7.5	±0.41	0÷1768	±6	±0.34
J	±120	4	-210÷1200	±6.3	±0.44	-100÷1200	±3	±0.23
T	±30	1	-200÷400	±1.9	±0.31	-100÷400	±1	±0.20
N	±120	4	-200÷1300	±12	±0.80	-100÷1300	±5.9	±0.42
R	±30	1	-50÷1768	±8.6	±0.47	0÷1768	±5.5	±0.31
E	±120	4	-200÷1000	±4.7	±0.40	-100÷1000	±2.7	±0.25
L	±120	4	-200÷800	±4.4	±0.44	-100÷800	±2.5	±0.28
B	±30	2	250÷1820	±15	0,96	480÷1820	±6	0,45

Tab. 8.8 Measurement accuracy for the module TC4, TC8, TC12

type	norm
K	PN-EN 60584-1:1997
S	PN-EN 60584-1:1997
J	PN-EN 60584-1:1997
T	PN-EN 60584-1:1997
N	PN-EN 60584-1:1997
R	PN-EN 60584-1:1997
E	PN-EN 60584-1:1997
L	GOST R 8.585:2001
B	PN-EN 60584-1:1997

Tab. 8.9 Thermocouple norm for the module TC4, TC8, TC12

8.7. RT4 , RT6 – RTD MEASUREMENT MODULES

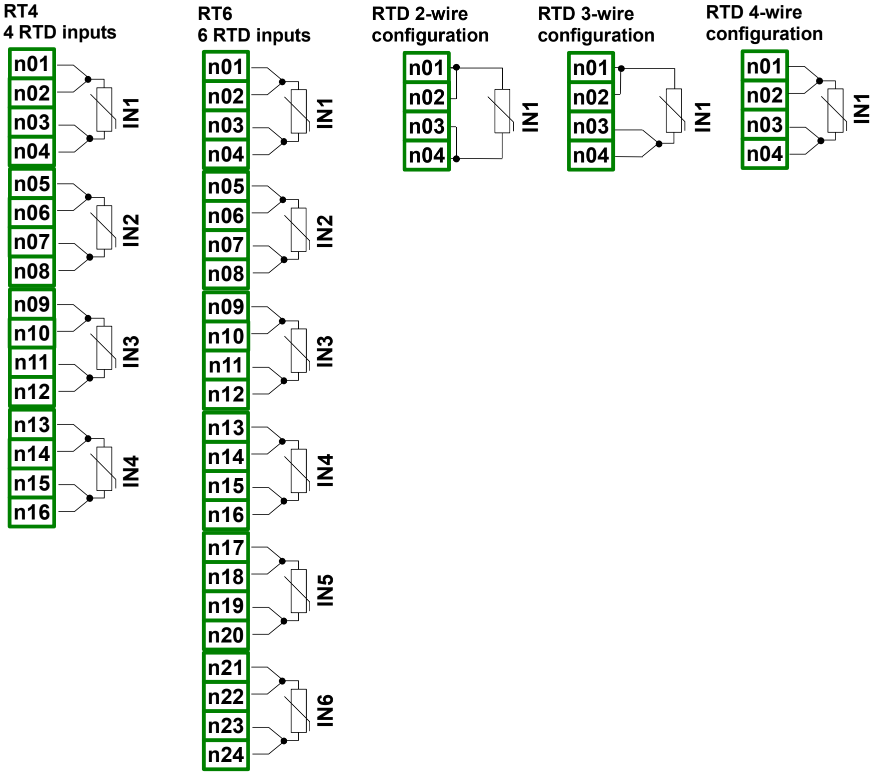


Fig. 8.12 RTD modules

Most important parameters of RT4 and RT6 modules:

	RT4	**RT6**
Number of inputs	4	6
Hardware measurement ranges	0÷325Ω, 0÷3250Ω	0÷325Ω, 0÷3250Ω
Hardware resolution **range ±325Ω** **range ±3250Ω**	0.01Ω 0.1Ω	0.01Ω 0.1Ω
Covered CONTROLLER/DATA RECORDER measurement ranges * **RTD** **resistance**	Pt100, Pt'100, Pt'50, Pt500, Pt'500, Pt1000, Cu50, Cu'50, Cu100, Cu'100 0-300Ω 0-3kΩ	Pt100, Pt'100, Pt'50, Pt500, Pt'500, Pt1000, Cu50, Cu'50, Cu100, Cu'100 0-300Ω 0-3kΩ

	RT4	RT6
Resistance measurement precision	0,1% @ 25˚C	0,1% @ 25˚C
Temperature stability	50 ppm/˚C	50 ppm/˚C
Connection method	2, 3 and 4 wire (switched manually)	2, 3 and 4 wire (switched manually)
Measurement wires resistance	max. 20Ω (every wire)	max. 20Ω (every wire)
Sampling period	1s	1s
Weight	42g	57g

Due to the real measurement accuracy it is recommended to set the temperature display accurate to one decimal place.
Check current list of **CONTROLLER/DATA RECORDER** measurement ranges at producer's website

Tab. 8.10 Technical data for the RT4 and RT6 modules

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. 8.11 Measurement accuracy for the RT4 and RT6 modules

type	norm
Pt100	PN-EN 60751:2009
Pt500	PN-EN 60751:2009
Pt1000	PN-EN 60751:2009
Pt'50	GOST 6651-94 (W_{100}=1,3910)
Pt'100	GOST 6651-94 (W_{100}=1,3910)
Pt'500	GOST 6651-94 (W_{100}=1,3910)
Cu50	GOST 6651-94 (W_{100}=1,3910)
Cu100	GOST 6651-94 (W_{100}=1,3910)
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. 8.12 RTD norm for the RT4 and RT6 modules

8.8. UN3, UN5 – OPTOISOLATED UNIVERSAL INPUTS MODULES

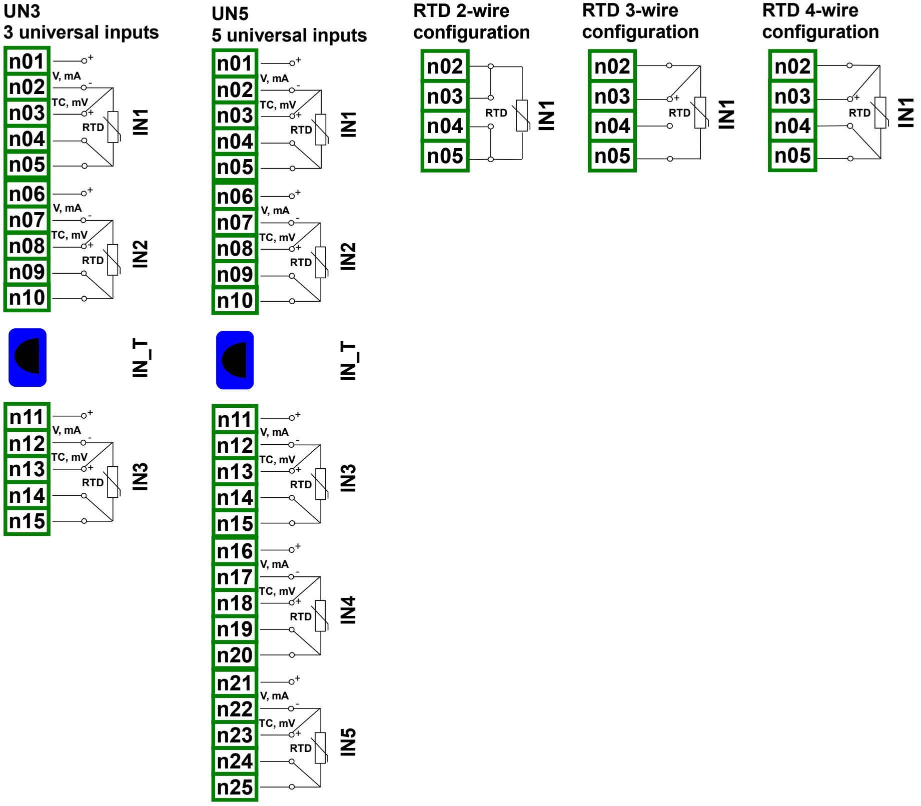


Fig. 8.13. Universal input modules

UN3 and **UN5** are modules with 5 and 3 universal inputs:

- – current measurements,
- – voltage measurements,
- – temperature measurements (TC or RTD).

Most important parameters of UN3 *and* UN5 *modules:*

<table>
<tr><th></th><th colspan="2">UN3</th><th colspan="2">UN5</th></tr>
<tr><td>Number of inputs</td><td colspan="2">3</td><td colspan="2">5</td></tr>
<tr><td>Hardware limitations</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td><b>current inputs</b></td><td colspan="2">-2mA ÷ 30mA</td><td colspan="2">-2mA ÷ 30mA</td></tr>
<tr><td><b>voltage inputs</b></td><td colspan="2">-1V÷ 12V
-15÷30mV
-15÷120mV
-50÷650mV</td><td colspan="2">-1V÷ 12V
-15÷30mV
-15÷120mV
-50÷650mV</td></tr>
<tr><td><b>thermocouple inputs</b></td><td colspan="2">-15mV ÷ 30mV
-15mV ÷ 120mV</td><td colspan="2">-15mV ÷ 30mV
-15mV ÷ 120mV</td></tr>
<tr><td><b>RTD inputs</b></td><td colspan="2">0÷325Ω,
0÷3250Ω,</td><td colspan="2">0÷325Ω,
0÷3250Ω,</td></tr>
<tr><td>Hardware resolution:
<b>current inputs</b></td><td colspan="2">1µA</td><td colspan="2">1µA</td></tr>
<tr><td><b>voltage inputs</b>
for range -1÷12 V
for range -15÷30 mV
for range -15÷120 mV
for range -50÷650 mV</td><td colspan="2">1 mV
2 µV
4 µV
40 µV</td><td colspan="2">1 mV
2 µV
4 µV
40 µV</td></tr>
<tr><td><b>thermocouple inputs</b>
range -15mV ÷ 30mV
range -15mV ÷ 120mV</td><td colspan="2">2µV
4µV</td><td colspan="2">2µV
4µV</td></tr>
<tr><td><b>RTD inputs</b>
range 0÷325Ω
range 0÷3250Ω</td><td colspan="2">0,01Ω
0,2Ω</td><td colspan="2">0,01Ω
0,2Ω</td></tr>
<tr><td>Permissible Long time overload</td><td colspan="2">20%</td><td colspan="2">20%</td></tr>
<tr><td>Covered CONTROLLER/DATA RECORDER measurement ranges *</td><td>type</td><td>max</td><td>type</td><td>max</td></tr>
<tr><td><b>current inputs</b></td><td>0÷20mA
4÷20mA</td><td>0÷22mA
4÷22mA</td><td>0÷20mA
4÷20mA</td><td>0÷22mA
4÷22mA</td></tr>
<tr><td><b>voltage inputs</b></td><td>0÷5V
1÷5V
0÷10V
2÷10V
-10÷25mV
-10÷100mV
0÷600mV</td><td>0÷6V
0÷6V
0÷12V
0÷12V
-10÷30mV
-10÷120mV
-50÷650mV</td><td>0÷5V
1÷5V
0÷10V
2÷10V
-10÷25mV
-10÷100mV
0÷600mV</td><td>0÷6V
0÷6V
0÷12V
0÷12V
-10÷30mV
-10÷120mV
-50÷650mV</td></tr>
</table>

	UN3	**UN5**
thermocouple inputs	K,S,J,T, N, R, B, E, L(GOST)	K,S,J,T, N, R, B, E, L(GOST)
RTD inputs	Pt100, Pt'100, Pt'50, Pt500, Pt'500, Pt1000, Cu50, Cu'50, Cu100, Cu'100, Ni100, Ni500, Ni1000	Pt100, Pt'100, Pt'50, Pt500, Pt'500, Pt1000, Cu50, Cu'50, Cu100, Cu'100, Ni100, Ni500, Ni1000
resistance input	0÷300Ω 0÷3000Ω	0÷300Ω 0÷3000Ω
Connection method in **RTD** mode	2, 3 and 4 wire (switched manually)	2, 3 and 4 wire (switched manually)
Input impedance		
current inputs	<65Ω (typ. 30Ω)	<65Ω (typ. 30Ω)
voltage inputs	>100kΩ (while maintaining correct polarization)	>100kΩ (while maintaining correct polarization)
thermocouple inputs	>1,5MΩ	>1,5MΩ
Precision		
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	0,1% @ 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
Sampling period in mode:		
current inputs:	535ms	535ms
voltage inputs:	535ms	535ms
thermocouple inputs:	535ms	535ms
RTD inputs:	1410ms	1410ms
resistance inputs:	1410ms	1410ms
Isolation strength	1 min. @ 500V AC	1 min. @ 500V AC
Weight	44g	73g

Tab. 8.13 Technical data for the UN3 and UN5 modules

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

Tab. 8.14 Thermocouple measurement accuracy for the UN3 and UN5 modules

type	norm
K	PN-EN 60584-1:1997
S	PN-EN 60584-1:1997
J	PN-EN 60584-1:1997
T	PN-EN 60584-1:1997
N	PN-EN 60584-1:1997
R	PN-EN 60584-1:1997
E	PN-EN 60584-1:1997
L	GOST R 8.585:2001
B	PN-EN 60584-1:1997

Tab. 8.15 Thermocouple norm for the UN3 and UN5 modules

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. 8.16 RTD measurement accuracy for the UN3 and UN5 modules

type	norm
Pt100	PN-EN 60751:2009
Pt500	PN-EN 60751:2009
Pt1000	PN-EN 60751:2009
Pt'50	GOST 6651-94 (W_{100}=1,3910)
Pt'100	GOST 6651-94 (W_{100}=1,3910)
Pt'500	GOST 6651-94 (W_{100}=1,3910)
Cu50	GOST 6651-94 (W_{100}=1,3910)
Cu100	GOST 6651-94 (W_{100}=1,3910)
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. 8.17 RTD norm for the UN3 and UN5 modules

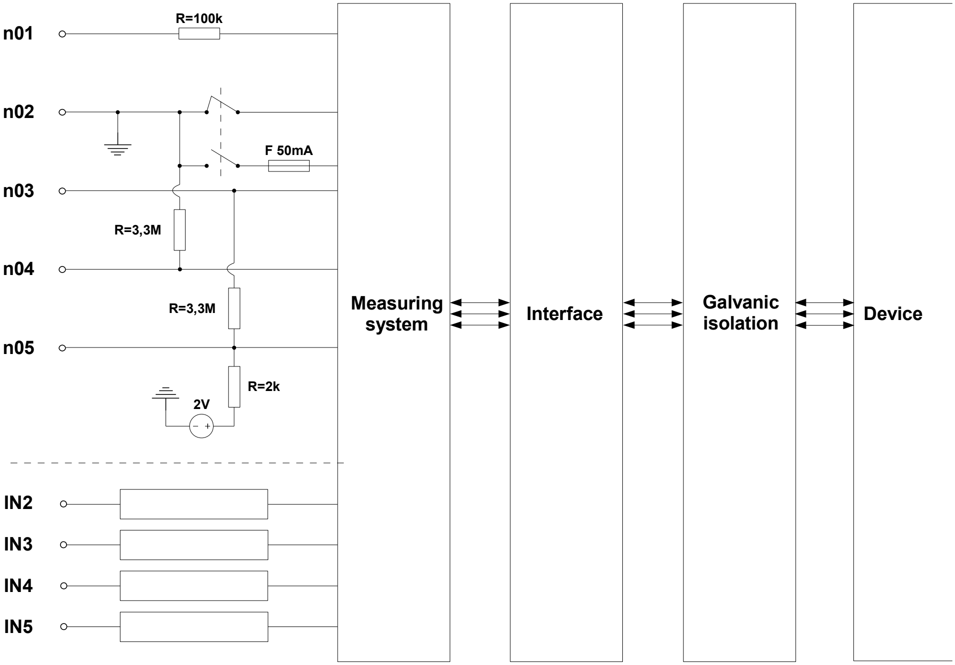


Fig. 8.14. Internal structure of the optoisolated Universal input module

8.9. D8, D16, D24 – OPTOISOLATED DIGITAL INPUTS MODULE

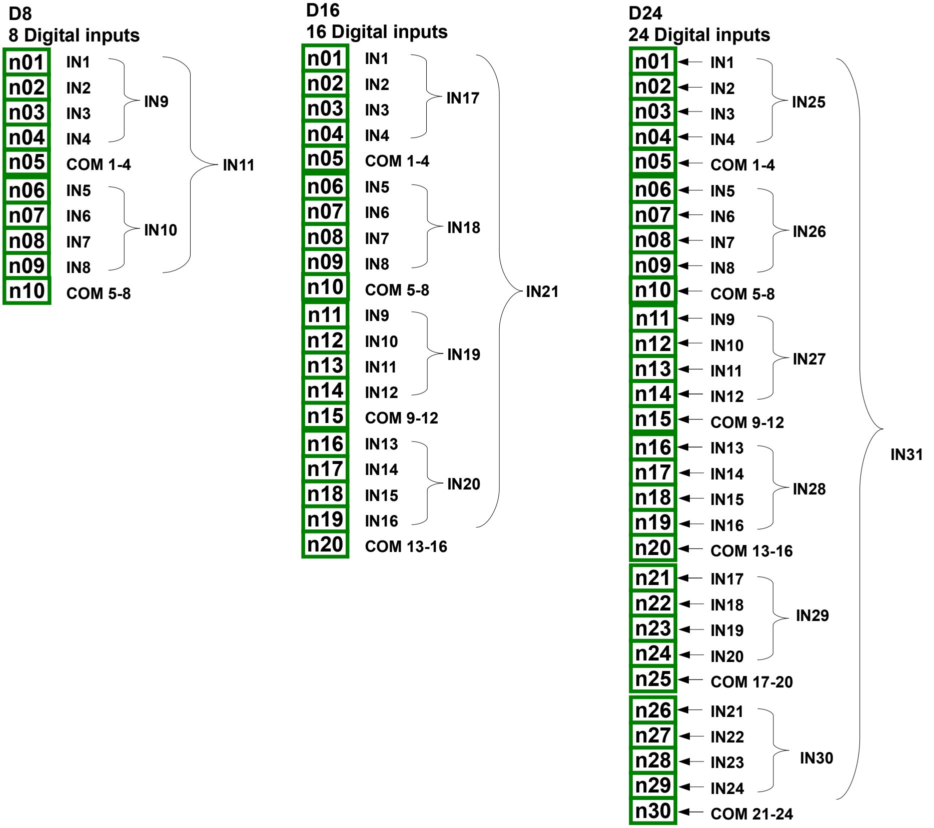


Fig. 8.15 Digital input module D8, D16, D24

D8, D16, D24 are modules with 8, 16 and 24 digital inputs respectively. Inputs are divided into groups of four input every. Every group has own common terminal, and is optically isolated from others groups and **CONTROLLER/DATA RECORDER** GND signal as well.

Most important parameters of D8, D16 and D24 modules:

	D8	D16	D24
Number of inputs	8 (2 groups 4 inputs each, optoisolated from others signals)	16 (4 groups 4 inputs each, optoisolated from others signals)	24 (6 groups 4 inputs each, optoisolated from other signals)
Input signals voltage levels: **logical LOW state** **logical HIGH state**	 | Uin | < 1V | Uin | > 4V	 | Uin | < 1V | Uin | > 4V	 | Uin | < 1V | Uin | > 4V

	D8	D16	D24
Max input voltage	30V	30V	30V
Input current consumption	about 15mA @24V about 5mA @10V about 2mA @5V	about 15mA @24V about 5mA @10V about 2mA @5V	about 15mA @24V about 5mA @10V about 2mA @5V
Input signals representation	8 single bits: ***IN1-IN8*** two 4-bit groups: ***IN9-IN10*** 1 byte: ***IN11***	16 single bits: ***IN1-IN16*** four 4-bit groups: ***IN17-IN20*** 1 integer: ***IN21***	24 single bits: ***IN1-IN24*** six 4-bit groups: ***IN25-IN30*** 1 integer: ***IN31***
Sampling frequency	3kHz	3kHz	3kHz
Isolation strength	1 min. @ 500V AC	1 min. @ 500V AC	1 min. @ 500V AC
Weight	30g	40g	58g

Tab. 8.18 Technical data for the module D8, D16, D24

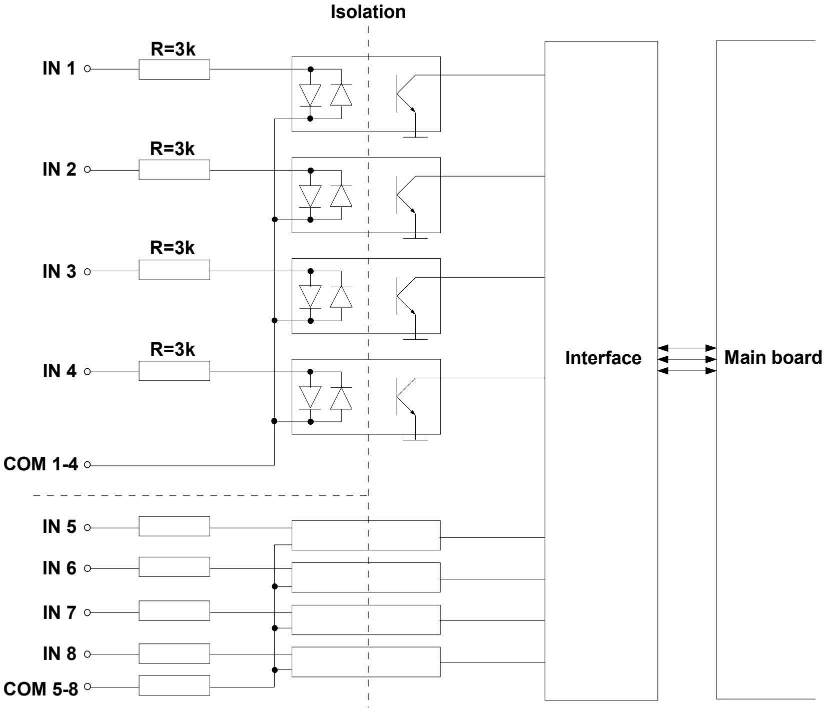


Fig. 8.16. Internal structure of the optoisolated Digital inputs module

8.10. CP2, CP4 – OPTOISOLATED UNIVERSAL COUNTERS MODULES

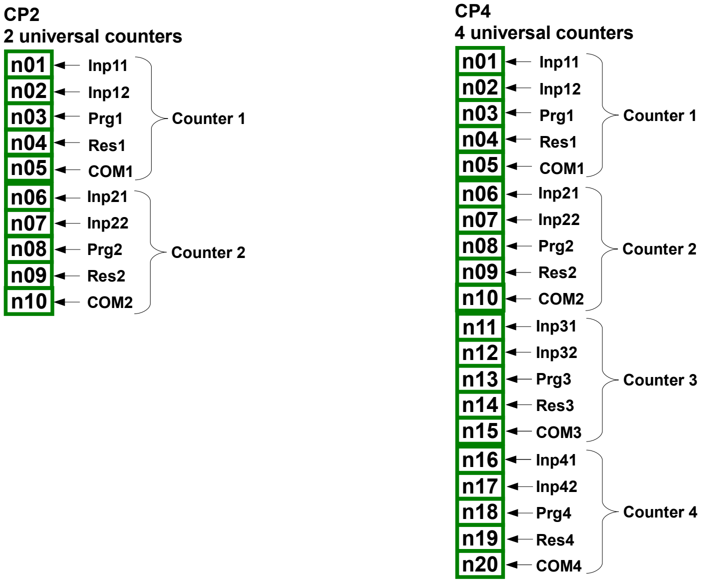


Fig. 8.17. Optoisolated counter modules

CP2 and **CP4** are modules of universal 4-input counters (**Fig. 8.32**). Each counter has common terminal (**COM**) and is optically isolated from other counters and **CONTROLLER/DATA RECORDER** GND signal as well, see **Fig. 8.18**.

Most important parameters of CP2 and CP4 modules:

	CP2	***CP4***
Number of inputs	2 groups of counter inputs (2 groups 4 inputs, every group optoisolated from the other signals)	4 groups of counter inputs (4 groups 4 inputs, every group optoisolated from the other signals)
Terminal description	**Inp[n]1, Inp[n]2** - counting inputs, pulse, **Prg[n]** - programmable inputs, **Res[n]** - reset inputs, **COM[n]** - common inputs, **[n]**=1, 2.	**Inp[n]1, Inp[n]2** - counting inputs, pulse, **Prg[n]** - programmable inputs, **Res[n]** - reset inputs, **COM[n]** - common inputs, **[n]**=1, 2, 3, 4.
Input signals voltage levels: **logical LOW state** **logical HIGH state**	 | Uin | < 1V | Uin | ≥ 10V	 | Uin | < 1V | Uin | ≥ 10V
Max input voltage	30V	30V

	CP2	**CP4**
Max input frequency for quad signals* **	5kHz	5kHz
Max input frequency for other signals **	10kHz	10kHz
Counter capacity	4 503 599 627 370 495 pulses (52 data bits)	4 503 599 627 370 495 pulses (52 data bits)
Input current consumption	about 14mA @24V about 6mA @10V	about 14mA @24V about 6mA @10V
Insulation strength (between individual measurements inputs)	2kV	2kV
Protection	50mA auto-reset fuse	50mA auto-reset fuse
Isolation strength (between internal circuits and measurements inputs)	1 min. @ 500V AC	1 min. @ 500V AC
Weight	35g	42g

*For signals with frequency greater than 3kHz it is recommended to keep duty cycle around 50% and phase shift 90°.
**Shift between input signals edges has to be at least 100µs.

Tab. 8.19. Technical data for the CP2 and CP4 modules

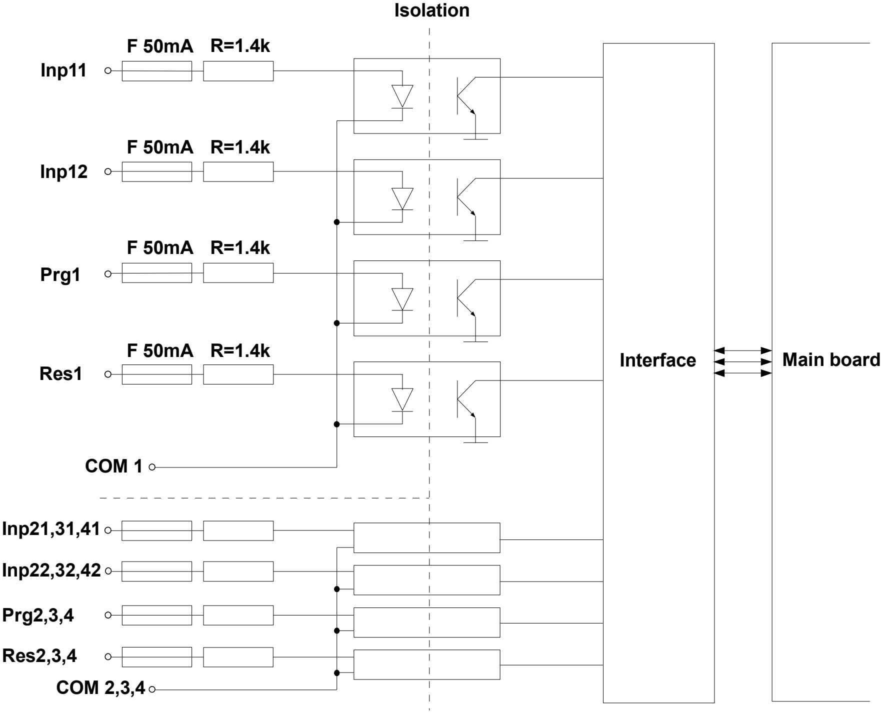


Fig. 8.18. Internal structure of the optoisolated Universal counters modules

8.11. HM2, HM4 – OPTOIZOLATED HOURMETERS MODULES

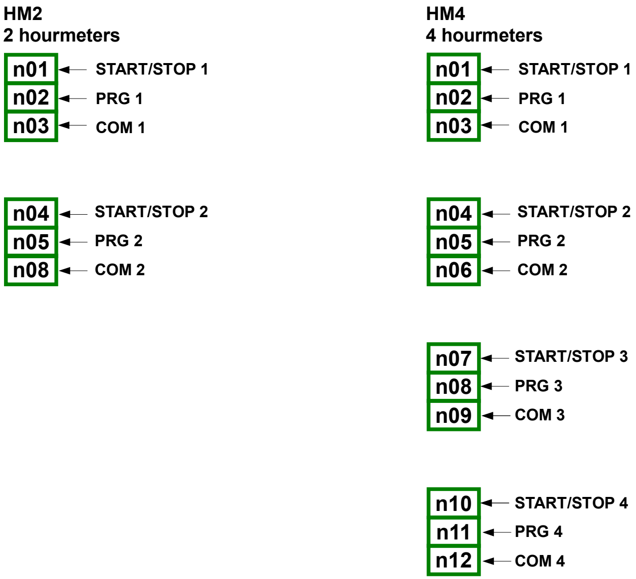


Fig. 8.19. Optoizolated hourmeters modules

Most important parameters of HM2 and HM4 modules:

	HM2	***HM4***
Number of inputs	2 hourmeters	4 hourmeters
Terminal description	**START/STOP [n]** – input which starts and stops time counting, **PRG[n]** - programmable inputs, **COM[n]** - common inputs, **[n]**=1, 2.	**START/STOP [n]** – input which starts and stops time counting, **PRG[n]** - programmable inputs, **COM[n]** - common inputs, **[n]**=1, 2, 3, 4.
Input signals voltage levels: **logical LOW state** **logical HIGH state**	| Uin | < 1V | Uin | ≥ 10V	| Uin | < 1V | Uin | ≥ 10V
Max input frequency	1kHz	1kHz
Max input voltage	30V	30V
Input current consumption	about 14mA @24V about 6mA @10V	about 14mA @24V about 6mA @10V
Insulation strength (between individual measurements inputs)	2kV	2kV
Measurement accuracy	± 30ppm @ +25°C	± 30ppm @ +25°C

	HM2	HM4
Temperature stability	± 50ppm/K	± 50ppm/K
Counting range	0 ÷ 10E+9 sec.	0 ÷ 10E+9 sec.
Counting resolution	0,001 sec.	0,001 sec.
Protection	50mA auto-reset fuse	50mA auto-reset fuse
Isolation strength (between internal circuits and measurements inputs)	1 min. @ 500V AC	1 min. @ 500V AC
Weight	28g	32,3g

Tab. 8.20. Technical data for the HM2 and HM4 modules

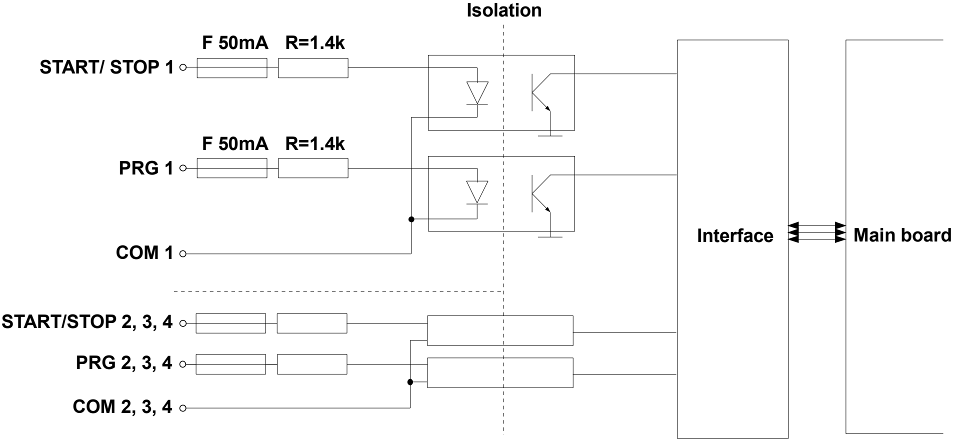


Fig. 8.20. Internal structure of the optoisolated hourmeters modules

8.12. S8, S16, S24 - SOLID STATE RELAY DRIVERS MODULES

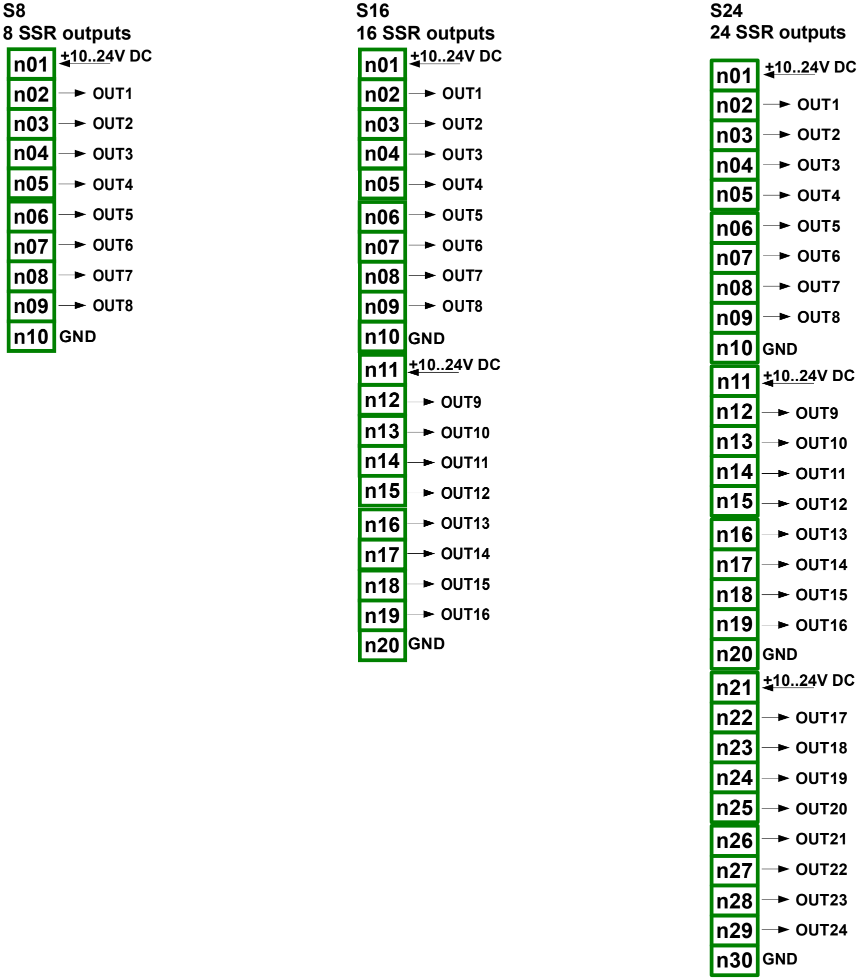


Fig. 8.21 SSR modules

Most important parameters of S8, S16 and S24 modules:

	S8	**S16**	**S24**
Static parameters			
Number of outputs	8	16 (in 2 groups with separate supply)	24 (in 3 groups with separate supply)

Max current source per output:
while **powered internally**
while **powered externally**
	10mA,
	sum limited to 50mA
	100mA,
	sum limited to 500mA
		10mA,
		sum limited to 50mA for a group
		100mA,
		sum limited to 500mA for a group
			10mA,
			sum limited to 50mA for a group
			100mA,
			sum limited to 500mA for a group

Output High Level voltage (Iout = 5mA)
while **powered internally**
while **powered externally**
	≥ 8V
	≥ (Vext. - 0.5V)
		≥ 8V
		≥ (Vext. - 0.5V)
			≥ 8V
			≥ (Vext. - 0.5V)

Output Low Level voltage (Iout = 5mA)	<1V	<1V	<1V

Overload protection
while **powered internally**
while **powered externally**
	Internal fuse 50mA
	Internal fuse 500mA
		Internal fuse 50mA (per group)
		Internal fuse 500mA (per group)
			Internal fuse 50mA (per group)
			Internal fuse 500mA (per group)

Maximum external supply of output *	30 V	30 V	30 V
Dynamic parameters (set individually for every output)			
PWM period **	0.1 ÷ 1600 sec.	0.1 ÷ 1600 sec.	0.1 ÷ 1600 sec.
PWM resolution	0.1 sec.	0.1 sec.	0.1 sec.
PWM internal frequency**	5kHz***	5kHz***	5kHz***
Pulse - duty factor	0 ÷ 100%	0 ÷ 100%	0 ÷ 100%
Pulse - duty factor resolution	15 bits ** 0x8000 means 100%	15 bits** 0x8000 means 100%	15 bits** 0x8000 means 100%
Lo state minimum time limit	0 ÷ 800 sec.	0 ÷ 800 sec.	0 ÷ 800 sec.
Hi state minimum time limit	0 ÷ 800 sec.	0 ÷ 800 sec.	0 ÷ 800 sec.
Isolation strength	1 min. @ 500V AC	1 min. @ 500V AC	1 min. @ 500V AC
Weight	32g	42g	69g

* Minimum external supply voltage is 10V, if external supply is less than 10V then outputs are powered internally

** PWM internal frequency and PWM period limit Pulse - duty factor real resolution. For example, if PWM period is 0.1 sec then real resolution of Pulse - duty factor is about 9 bits (0.1 * 5kHz = 500 levels). If PWM period is longer than 6.55 seconds then Pulse - duty factor resolution is full 15 bits (6.56 * 5kHz > 32768 levels).

*** PWM output quantization: 20µs

Tab. 8.21 Technical data for the module S8, S16

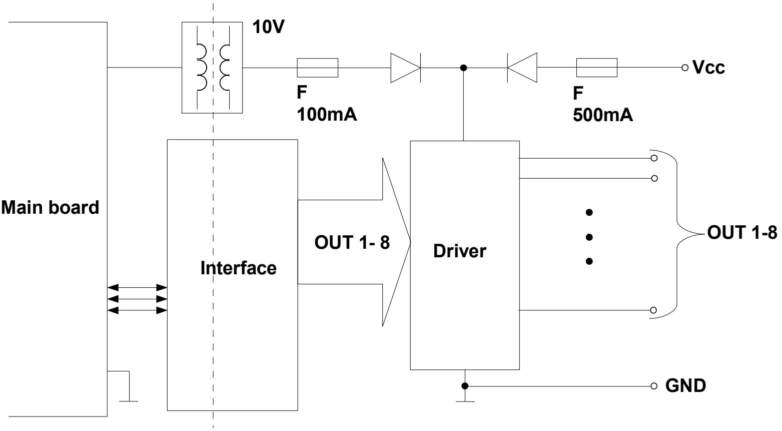


Fig. 8.22. Internal structure of the SSR output module (8 outputs)

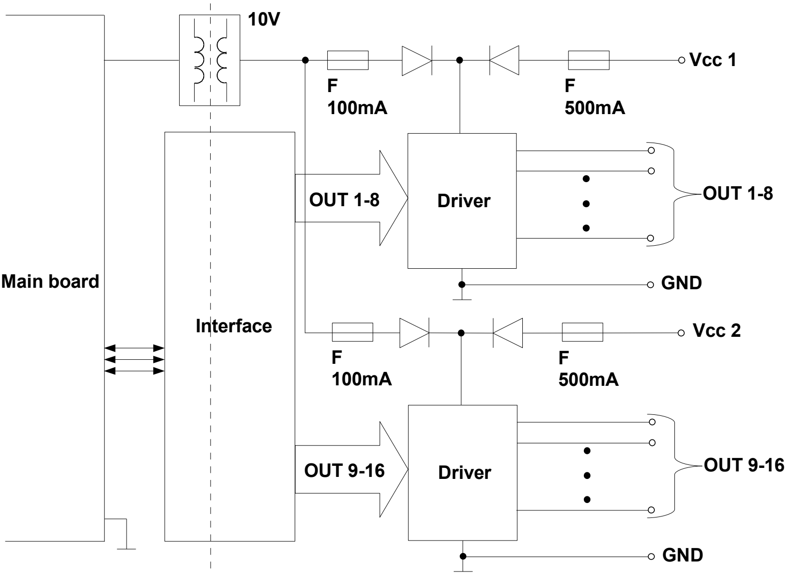


Fig. 8.23. Internal structure of the SSR output module (16 outputs)

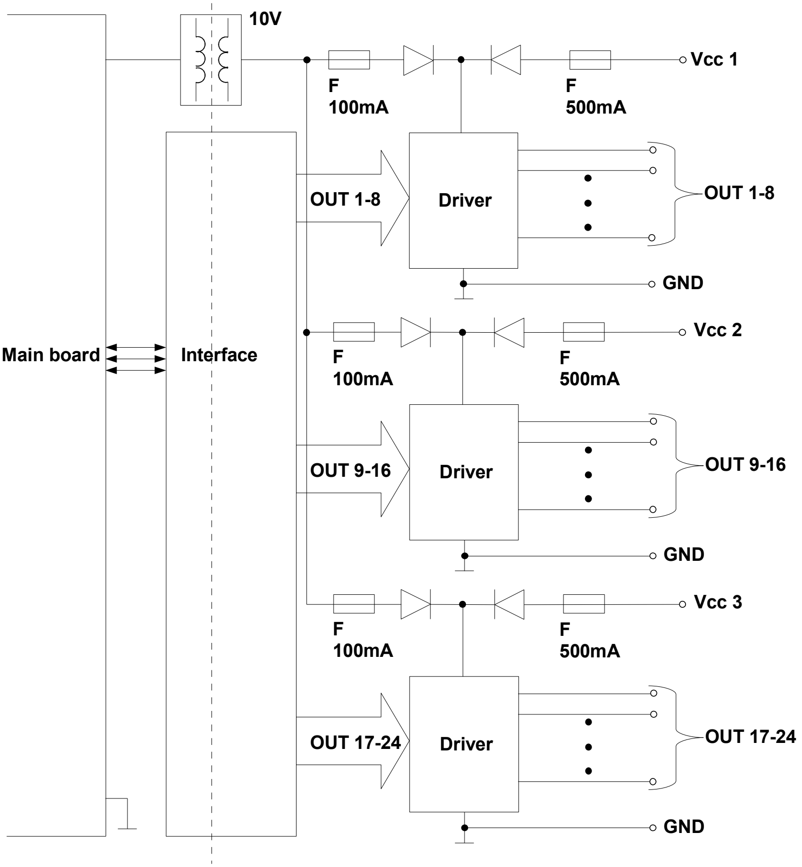


Fig. 8.24. Internal structure of the SSR output module (24 outputs)

8.13. R45, R81, R65, R121 - RELAY MODULES

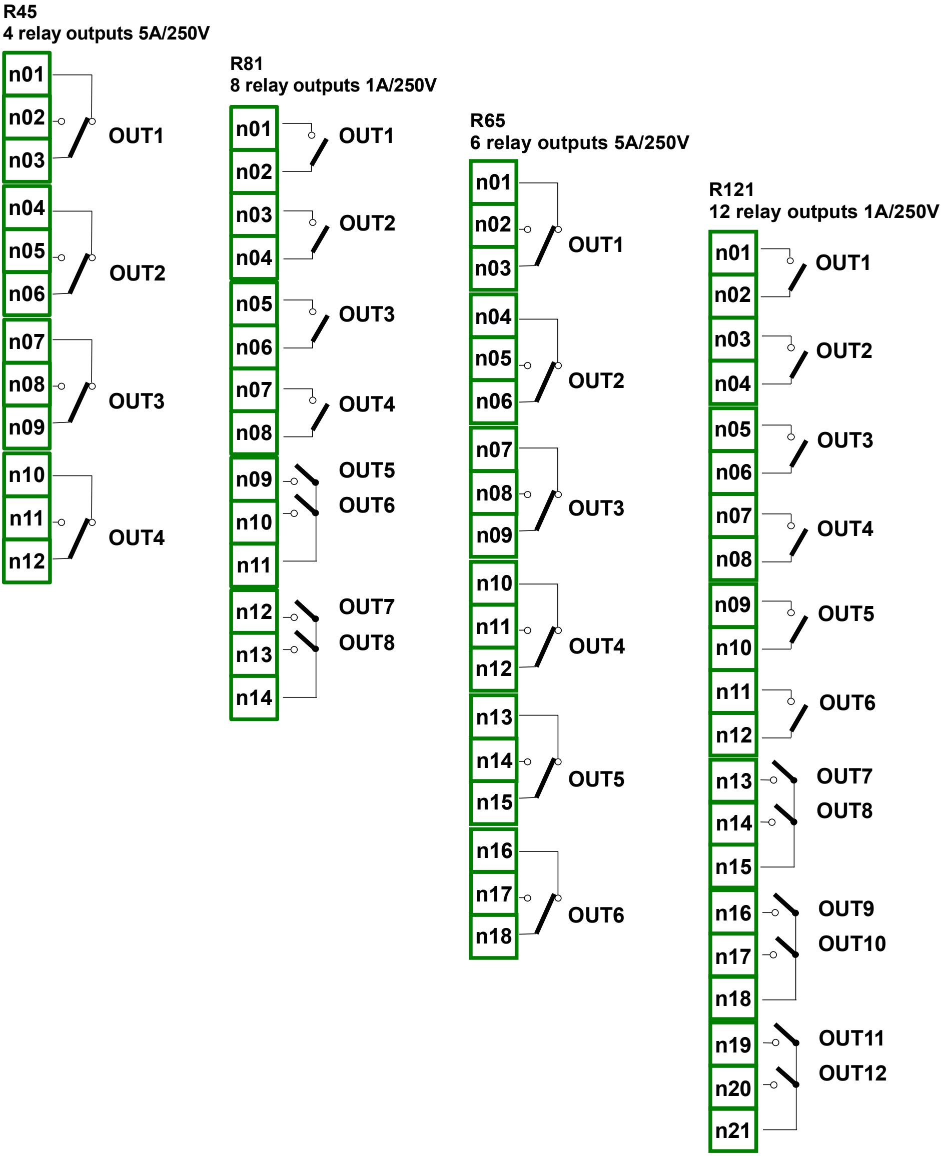


Fig. 8.25. Relay output modules

Most important parameters of R45, R81, R65 and R121 modules:

	R45	**R81**	**R65**	**R121**
Number of relays	4 SPDP (Switchable)	8 SPST (N.O.)	6 SPDP (Switchable)	12 SPST (N.O.)
Max. load per relay	5A, cos j = 1 (resistive load)	1A, cos j = 1 (resistive load)	5A, cos j = 1 (resistive load)	1A, cos j = 1 (resistive load)
Max. voltage switched by relay	250V AC	250V AC	250V AC	250V AC
Isolation strength between relays	1 min. @ 630V AC	1 min. @ 630V AC	1 min. @ 630V AC	1 min. @ 630V AC
Insulation strength (relay to CONTROLLER/DATA RECORDER supply)	1 min. @ 1000V AC	1 min. @ 1000V AC	1 min. @ 1000V AC	1 min. @ 1000V AC
Weight	50g	74g	133g	110g

Tab. 8.22 Technical data for the module R45, R81, R65 and R121

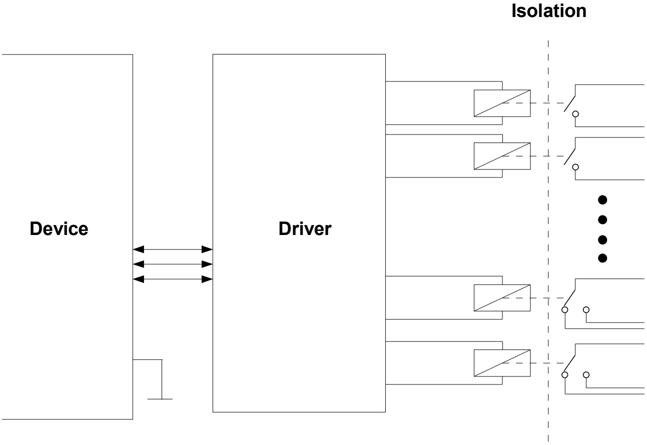


Fig. 8.26. Internal structure of the Relay outputs module

8.14. IO2, IO4, IO6, IO8 – PASSIVE CURRENT OUTPUT

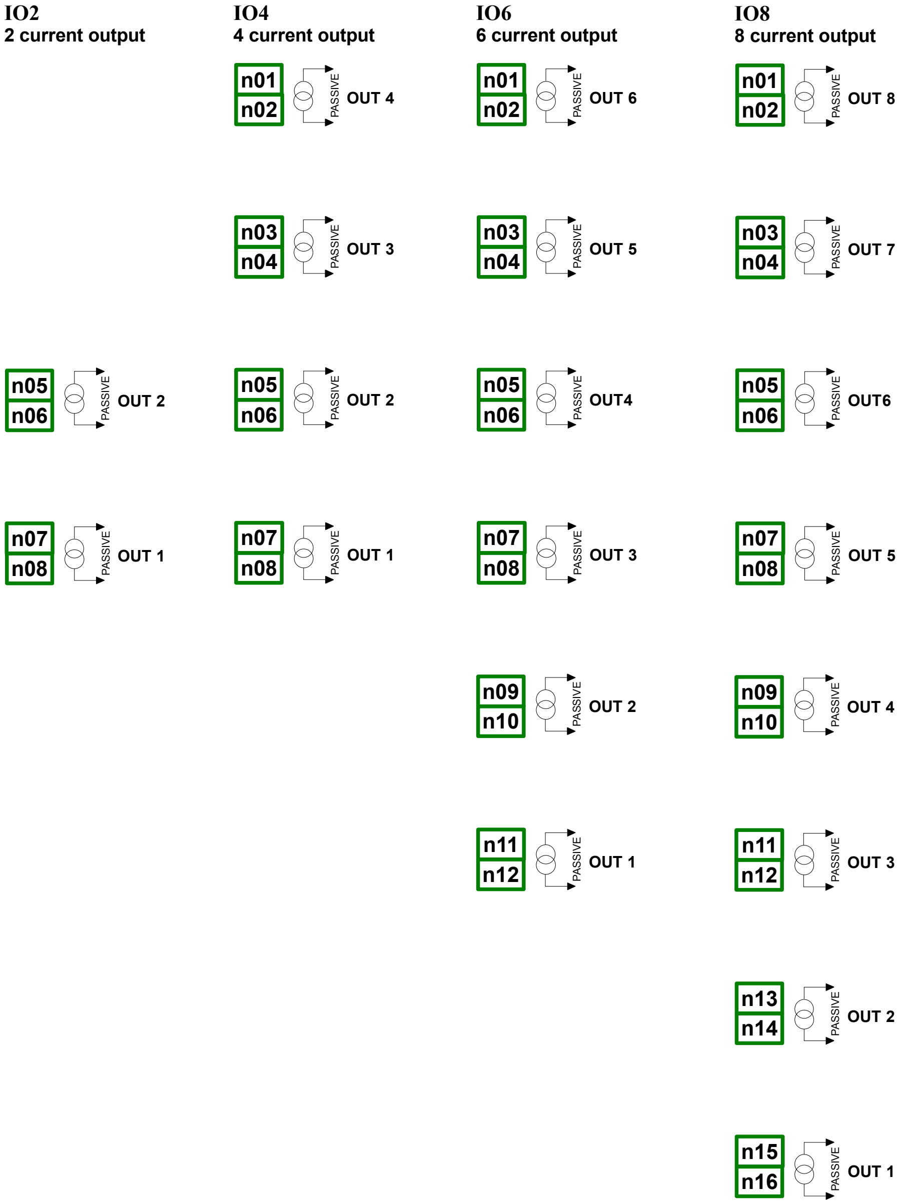


Fig. 8.27 Passive current output module IO2, IO4, IO6 and IO8

As the output is passive type, it is required to power the current loop. Note that polarisation of IO2, IO4, IO6 and IO8 outputs does not matter.

Most important parameters of IO2, IO4, IO6 and IO8 modules:

	IO2	**IO4**	**IO6**	**IO8**
Number of outputs	2	4	6	8
Output type	Passive current output	Passive current output	Passive current output	Passive current output
Nominal analogue range	4-20mA	4-20mA	4-20mA	4-20mA
Hardware output limitation	3-25mA	3-25mA	3-25mA	3-25mA
Output voltage dropout	max. 9V	max. 9V	max. 9V	max. 9V
Overload protection	Internal resettable fuse 50mA	Internal resettable fuse 50mA	Internal resettable fuse 50mA	Internal resettable fuse 50mA
Loop Supply Range	9-30V	9-30V	9-30V	9-30V
Output current precision	0.1% @25ºC, 50ppm/ºC	0.1% @25ºC, 50ppm/ºC	0.1% @25ºC, 50ppm/ºC	0.1% @25ºC, 50ppm/ºC
Resolution	12 bit	12 bit	12 bit	12 bit
Isolation strength	1 min @ 500V AC	1 min @ 500V AC	1 min @ 500V AC	1 min @ 500V AC
Weight	23g	30g	38g	53g

Tab. 8.23 Technical data for the module IO2, IO4, IO6 and IO8

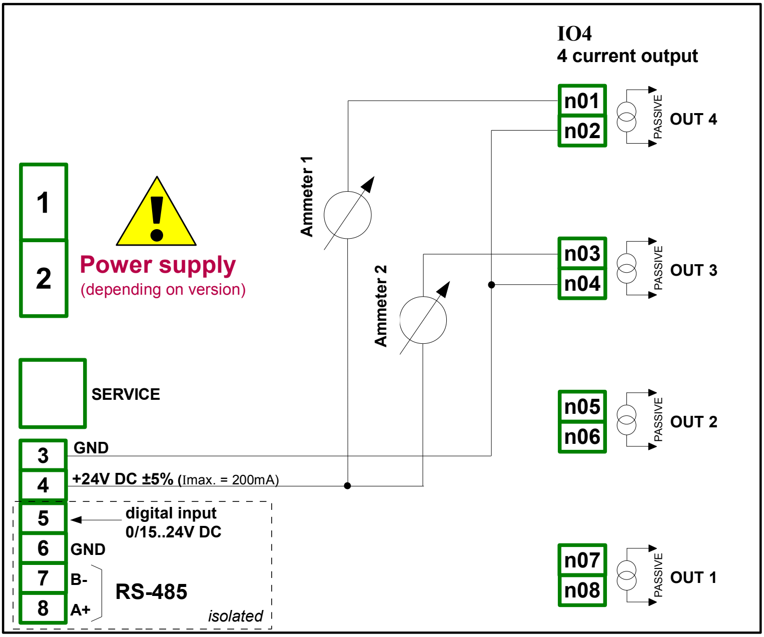


Fig. 8.28. Connections for the Passive current output from GND side

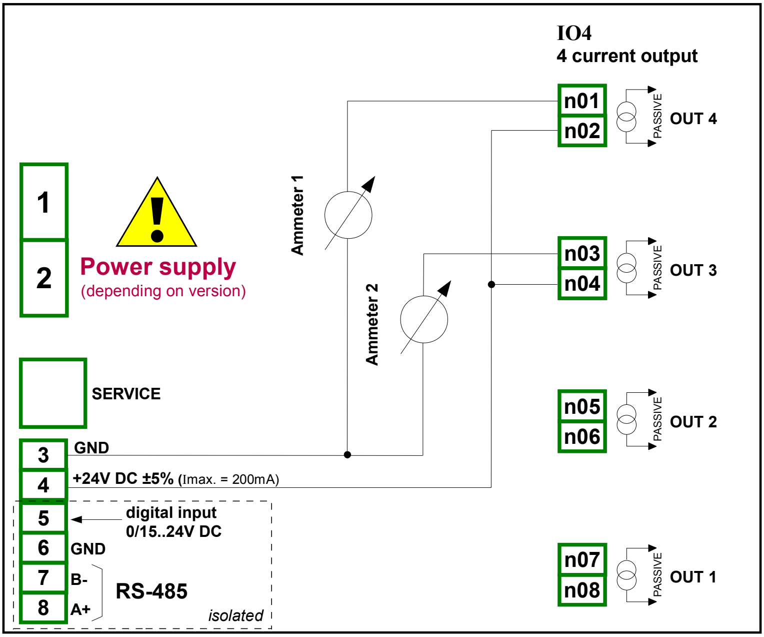


Fig. 8.29. Connections for the Passive current output from Power supply side

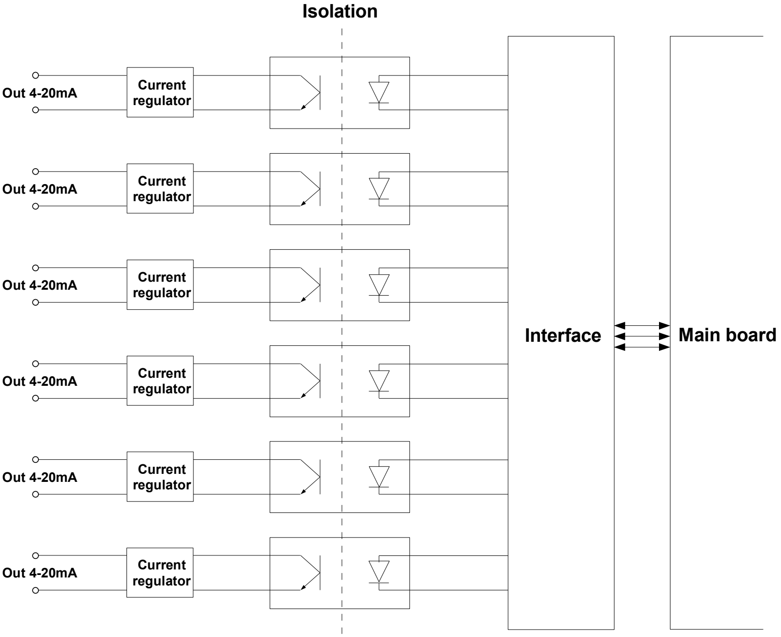


Fig. 8.30. Internal structure of the Passive current outputs module

8.15. COMMUNICATION MODULES

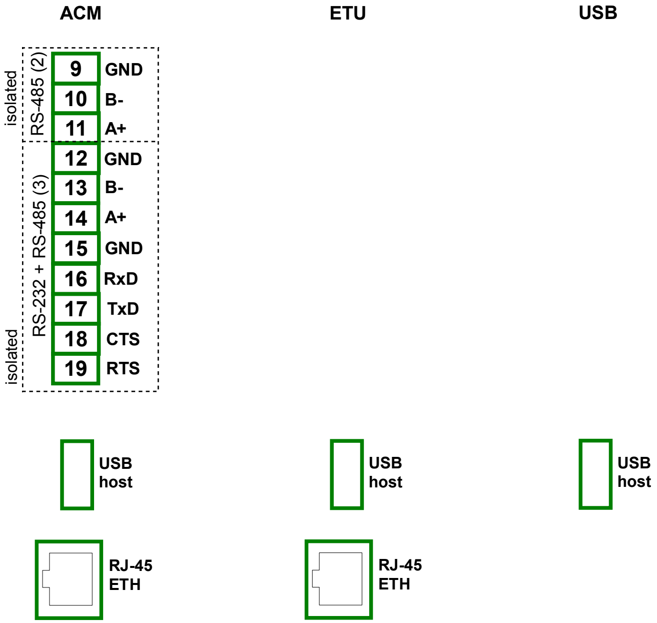


Fig. 8.31. Communication modules: ACM, ETU and USB (back - USB host only)

Most important parameters of ACM, ETU and USB modules:

	ACM	ETU	USB (back)
Number of inputs/outputs	4	2	1
Input/output type	RS-485, RS-232+RS-485, USB host, RJ-45 ETH	USB host, RJ-45 ETH	USB host
Hardware output limitation	USB host: max current output 100mA	USB host: max current output 100mA	USB host: max current output 100mA
Baudrate	RS-485 [bit./sec.]: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, RJ-45 ETH:10Mb/sec. USB host 12Mb/sec.	RJ-45 ETH:10Mb/sec. USB host 12Mb/sec.	USB host 12Mb/sec.
Data format	RS-232/485: 8N1, 8N2, 8E1, 8E2, 8O1, 8O2	-	-
Weight	48g	43,5g	21g

Tab. 8.24 Technical data for the communication module

8.16. DATA FORMAT

Introduction

Every data obtained from device have 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 delivered PC software, file structure **must be the same** as described below,
- If not stated otherwise, all data is a text in which subsequent columns are separated by 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 is an end of 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 other value than 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
```

First column is channel ID (channel number – 1 in decimal notation). In 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 following columns:

- channel ID (channel number – 1)
- name
- selected data source index (see **Mode** parameter in **Chapter 7.8.1. 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 7.8.1. Logical Channels - General settings**)
- ON label
- OFF label
- channel **-HI-** level
- channel **-LO-** level
- number precision — text data describing selected number precision (e.g.: 0.00 mean 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 have name Channel C1 it is in mode 1 (Hardware input), have 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 user's text editor. It is recommended that the user enables the visibility of special characters.

General data structure

Data obtained from device by copying them on drive, have fixed structure.

In flash drive's main directory is a folder which name is the same as device's serial number. All **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. Name of the folder is the same as internal log name from file **main_log.bin_logs**.

Every data folder contains 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 data file the same as 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 informations about logging functionality. There can be following types of data:

- **const_dev_info** – information specific to this device.
- **logs** – data describing current and past logs stored on 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 log,
- **event_log** — reserved for future use,
- **dev_settings** — reserved for future use,

Log data format description

Contains information about specific log. There can be following types of data:

- **group_settings** — group settings of this log. This is one line in 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 log,
  - 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 samples files created in this log in 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 channels that are being logged in following format:
  - channel index,
  - name,
  - selected data source index (see **Mode** parameter in **Chapter 7.8.1. 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 7.8.1. 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 selected number precision (e.g.: 0.00 mean two decimal digits),

part data format description

Specific log data. There can be 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 time of sample number **S_{id}** use below procedure:

- Find **R_{v}** and maximum **R_{id}** where **$R_{id} \le S_{id}$**,
- Find **T_{t}** and maximum **T_{id}** for **$T_{id} \le S_{id}$**,
- Calculate sample time (**S_{t}**) using given equation:

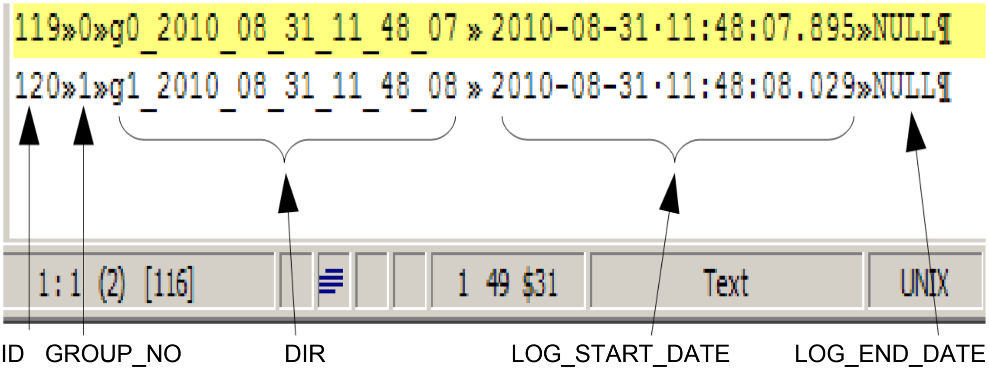
$$S_t = T_t + R_v \cdot (S_{id} - T_{id})$$

Detailed description of files

File **main_log.bin_logs** contains:

- **ID** – unique decimal number identifying log,
- **GROUP_NO** – number of a logged group (correspond 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 schema bellow: “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 particular file is not closed (logging is not finished),



```
119»0»g0_2010_08_31_11_48_07»2010-08-31·11:48:07.895»NULL¶
120»1»g1_2010_08_31_11_48_08»2010-08-31·11:48:08.029»NULL¶
```

Fig. 8.32. ***main_log_bin_logs*** *file view example*

File **main_log.bin_event_log** – reserved for future usage,

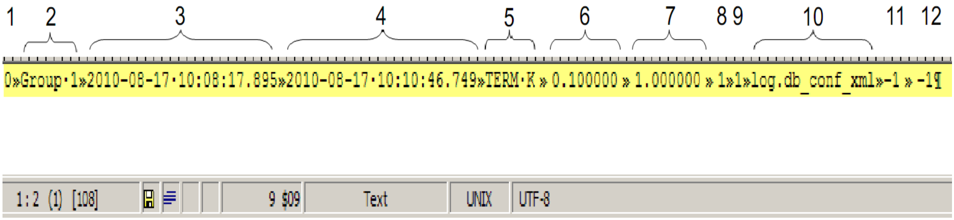
File **main_log.bin_dev_settings** – reserved for future usage,

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 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,



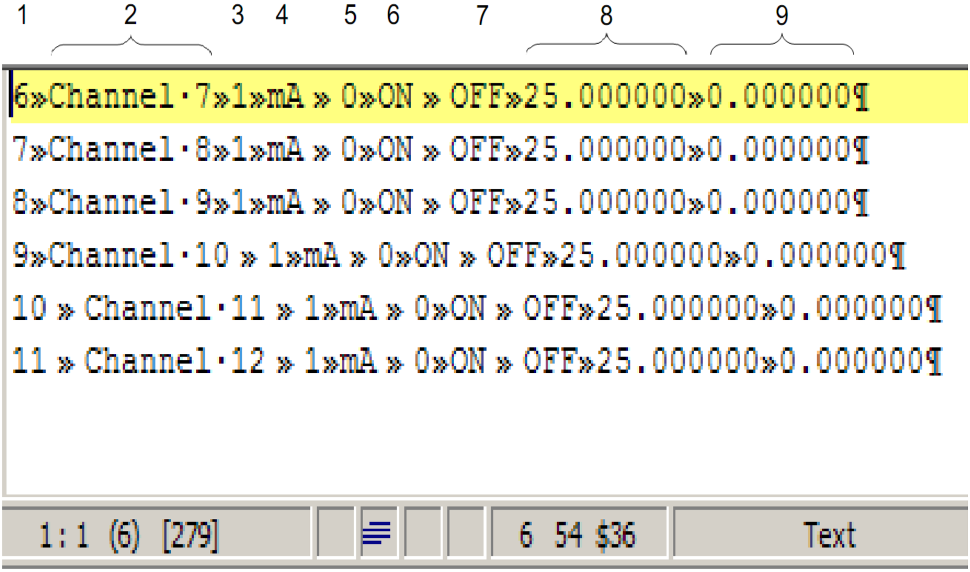
```
0»Group·1»2010-08-17·10:08:17.895»2010-08-17·10:10:46.749»TERM·K » 0.100000 » 1.000000 » 1»1»log.db_conf_xml»-1 » -1¶
```

*Fig. 8.33. **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 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,



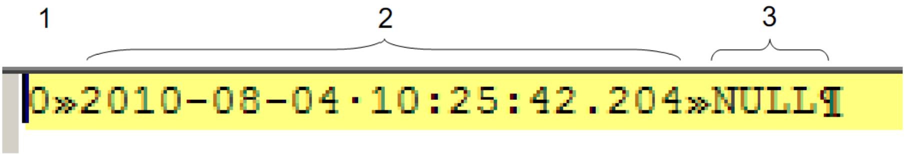
```
6»Channel·7»1»mA » 0»ON » OFF»25.000000»0.000000¶
7»Channel·8»1»mA » 0»ON » OFF»25.000000»0.000000¶
8»Channel·9»1»mA » 0»ON » OFF»25.000000»0.000000¶
9»Channel·10 » 1»mA » 0»ON » OFF»25.000000»0.000000¶
10 » Channel·11 » 1»mA » 0»ON » OFF»25.000000»0.000000¶
11 » Channel·12 » 1»mA » 0»ON » OFF»25.000000»0.000000¶
```

*Fig. 8.34. **log.db_chan_conf** file view example*

File log.db_log_files – informations about files with log samples,

Due to security of data, samples are stored in a separate files which changes from time to time. File **log.db_log_files** contains an information about such files. Every line describes other file.

1) ID – unique identifier of the file,
2) file start date,
3) file end date or NULL if file is not finished,



```
0»2010-08-04·10:25:42.204»NULL¶
```

*Fig. 8.35. **log.db_log_files** file view example*

Files with samples **partX.db_***

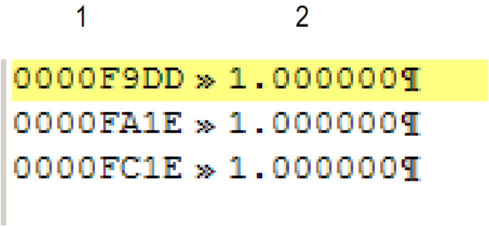
These files contains a logged samples. Names of these files are created according to a rule:

partX.db_SUFFIX, where:

X – unique identifier of a file (see **log.db_log_files**),
SUFFIX – can be: **raster**, **samples**, **timestamp**,

If file has suffix **raster**, the it contains info about time period between samples.

1) **IDR** – hexadecimal id of the sample for which raster becomes valid,
2) **period** – floating point number which expresses period in seconds,

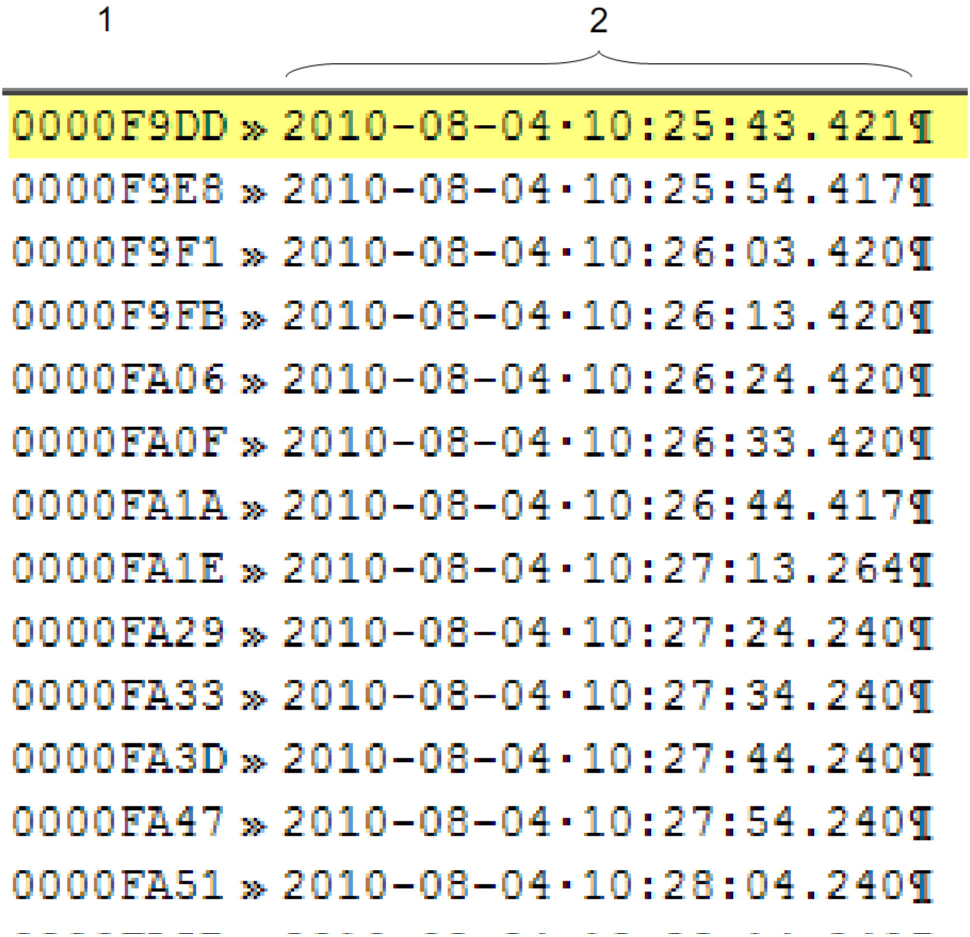


```
0000F9DD » 1.000000¶
0000FA1E » 1.000000¶
0000FC1E » 1.000000¶
```

*Fig. 8.36. **partX.db_raster** file view example*

If file has suffix **timestamp**, then it contains info about samples synchronisation.

1) **IDT** – hexadecimal id of the sample for which synchronisation data is stored,
2) **T** – date (**timestamp**),



```
0000F9DD » 2010-08-04·10:25:43.421¶
0000F9E8 » 2010-08-04·10:25:54.417¶
0000F9F1 » 2010-08-04·10:26:03.420¶
0000F9FB » 2010-08-04·10:26:13.420¶
0000FA06 » 2010-08-04·10:26:24.420¶
0000FA0F » 2010-08-04·10:26:33.420¶
0000FA1A » 2010-08-04·10:26:44.417¶
0000FA1E » 2010-08-04·10:27:13.264¶
0000FA29 » 2010-08-04·10:27:24.240¶
0000FA33 » 2010-08-04·10:27:34.240¶
0000FA3D » 2010-08-04·10:27:44.240¶
0000FA47 » 2010-08-04·10:27:54.240¶
0000FA51 » 2010-08-04·10:28:04.240¶
```

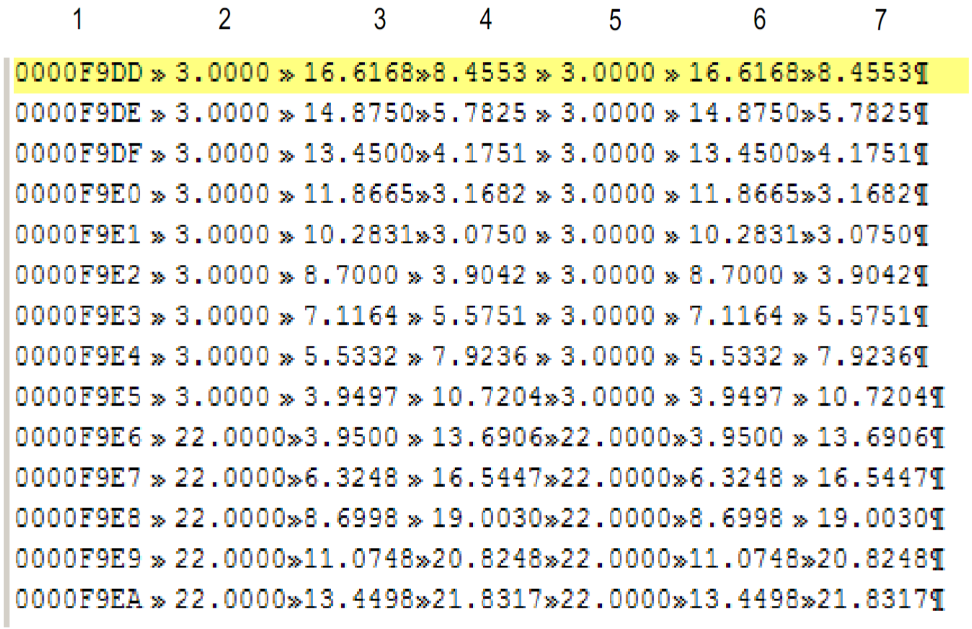
Fig. 8.37. ***partX.db_timestamp*** *file view example*

If 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. 8.38. **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 following schema:

1) if a directory with the name equal to device's serial number does not exist, then it is created,
2) in such directory, files witch information about logs are created (description bellow),
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\
```

8.17. DIRECT ACCESS TO LOG FILES USING HTTP PROTOCOL.

Request format:

To request specified data the user must perform CGI query on address specified below using 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 parameters id_first and id_last. These parameters limit the amount of data processed and sent to given Id range.

Example:

To limit the amount of returned data to 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 decimal or hexadecimal notation (can be mixed – one can be decimal one 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 firs 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 bellow.

Obtaining measurements (cmd=current):

Current measurements can be obtained by using value **current** of **cmd** parameter. This feature should not be used as a replace of recording data functionality because it causes high device load. Use this feature to sporadic device measurement check.
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 value **settings** of **cmd** parameter. This feature should not be used as a replace of recording data functionality because it causes high device load. Use this feature to sporadic device settings check.
Example of query viewing all channels settings:

http://192.168.1.176/cgi-bin/data.cgi?cmd=settings

If 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 valid data query is performed but data requested not exists on the device (example: requesting data of non existent log) the error page *"404: Not found"* will be returned.

If invalid query is performed the error page *"400: Bad request"* with inappropriate error message will be returned.

Obtaining general device logging information (cmd=main_log):

To obtain current device logging information a **main_log** parameter can be used. There must be specified additional parameter called **file**. Values must be one of 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 informations (cmd=log):

To obtain specified log information a **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 **part** parameter can be used. There must be specified additional parameters:

1) **log_name** – is the name of log,
2) **part_no** – is the number of data file,
    - **samples** – returns recorded channels data,
    - **timestamp** – returns recorded time stamps,
    - **raster** – returns samples interval,

If there is no **id_first** nor **id_last** specified the whole data file will be returned. When requesting **samples**, file size can be even **1MB**.