

# User manual

30.7.2013  
Firmware V1.0-1.5

## 6821

### 2 channel transmitter



**Nokeval**

# Introduction

The 6821 is a rail-mounted two-channel measurement unit for temperature sensors and other electrical inputs. The input channels are individual and can be used for different signals. The unit has two analog outputs or alternatively one analog and one serial output. The serial output accepts Nokeval SCL and Modbus RTU commands. Up to four logical alarms can control two alarm relays. The inputs are galvanically isolated from the outputs and the supply voltage, but not from each other.

There are four built-in inter-channel functions: average, difference, minimum, and maximum. More mathematical and conditional and timed operations may be realized with a simple programming language called ELo.

The front panel has a four-digit display and four push buttons that can be used to monitor the readings and to change the settings. The settings can also be edited on a computer using the RS-485 serial connection. The ELo program can be edited on a personal computer only, not on the front panel.

## How to use this manual

The transmitter consists of several quite independent blocks like the two inputs, analog outputs, serial communications and so on. That is why this manual is also divided in chapters, one chapter concerning one block.

First read through the chapter "General" to find out how to mount the transmitter and to open the transmitter case etc and how to get started with the configuration settings, either with the front panel or with a PC software. Then advance to the chapter "Power supply". To get the transmitter to measure something, read the chapter "Inputs". To get an analog output, read the chapter "Analog output", and so on.

## Table of contents

Introduction.....	2
General .....	3
Power supply .....	7
Front panel .....	8
Inputs.....	12
Analog outputs .....	17
Alarms and relays .....	19
ELo program .....	21
Serial communications .....	24
Specifications.....	31

## Manufacturer

**Nokeval Oy**  
Yrittajakatu 12  
FIN-37100 Nokia  
Finland

Tel +358 3 3424800  
Fax +358 3 3422066  
WWW: [www.nokeval.com](http://www.nokeval.com)  
Sales & inquiries: [sales@nokeval.com](mailto:sales@nokeval.com)  
Technical support: [support@nokeval.com](mailto:support@nokeval.com)

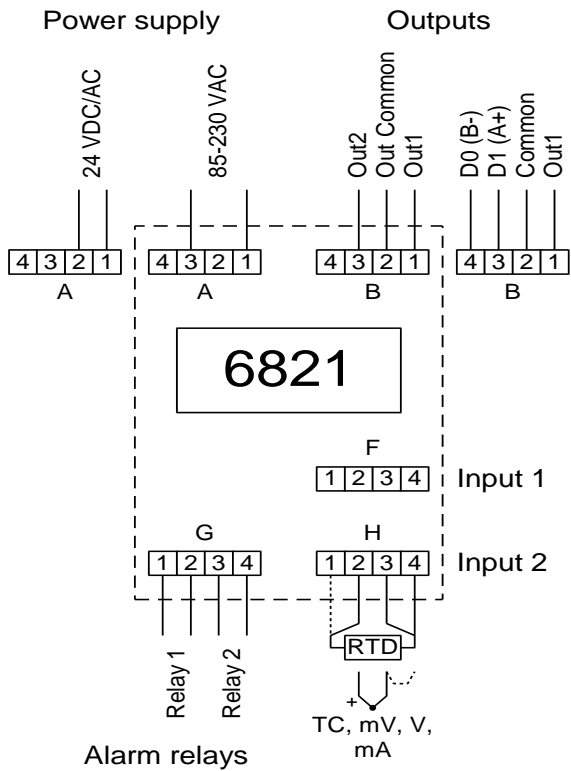
# General

## Mounting

This transmitter is intended to be mounted on a 35 mm DIN rail. It should be installed on a wall as on the front cover picture. Other positions will affect the flow of the cooling air and ruin the thermocouple accuracy. A small air gap to the next instrument on the rail is recommended.

## Connections

The connections are explained in the chapters Inputs, Analog outputs, Alarms & relays, Power supply, and Serial communications.



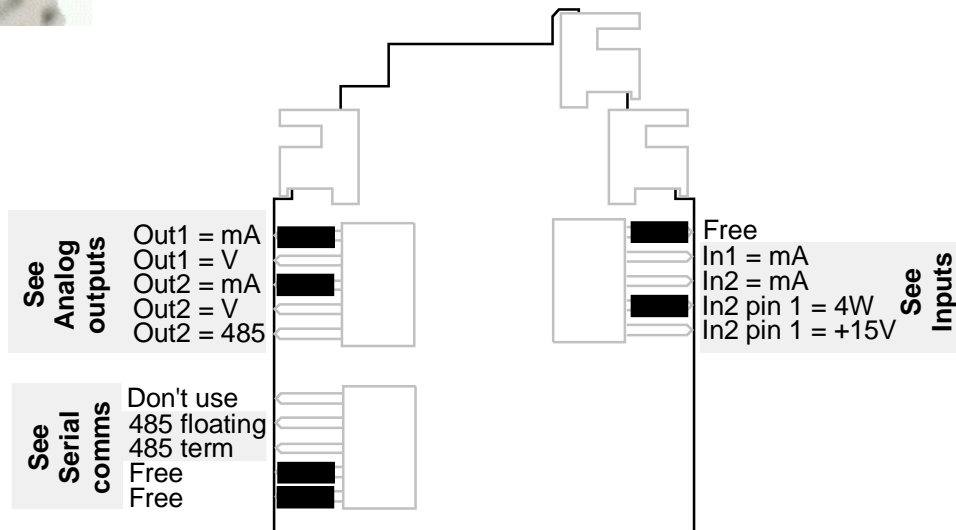
## Jumpers



To access the jumpers, use a small screwdriver to click the four locks (see photo). Then pull the top and bottom parts of the case apart. The power supply must be disconnected to avoid electric shocks.

The picture shows the jumper locations and the default setup. The jumpers labeled “Free” can be taken off and used where needed.

The jumper positions are explained in the chapters Inputs, Analog outputs, and Serial communications, pages 12, 17, and 24 respectively.

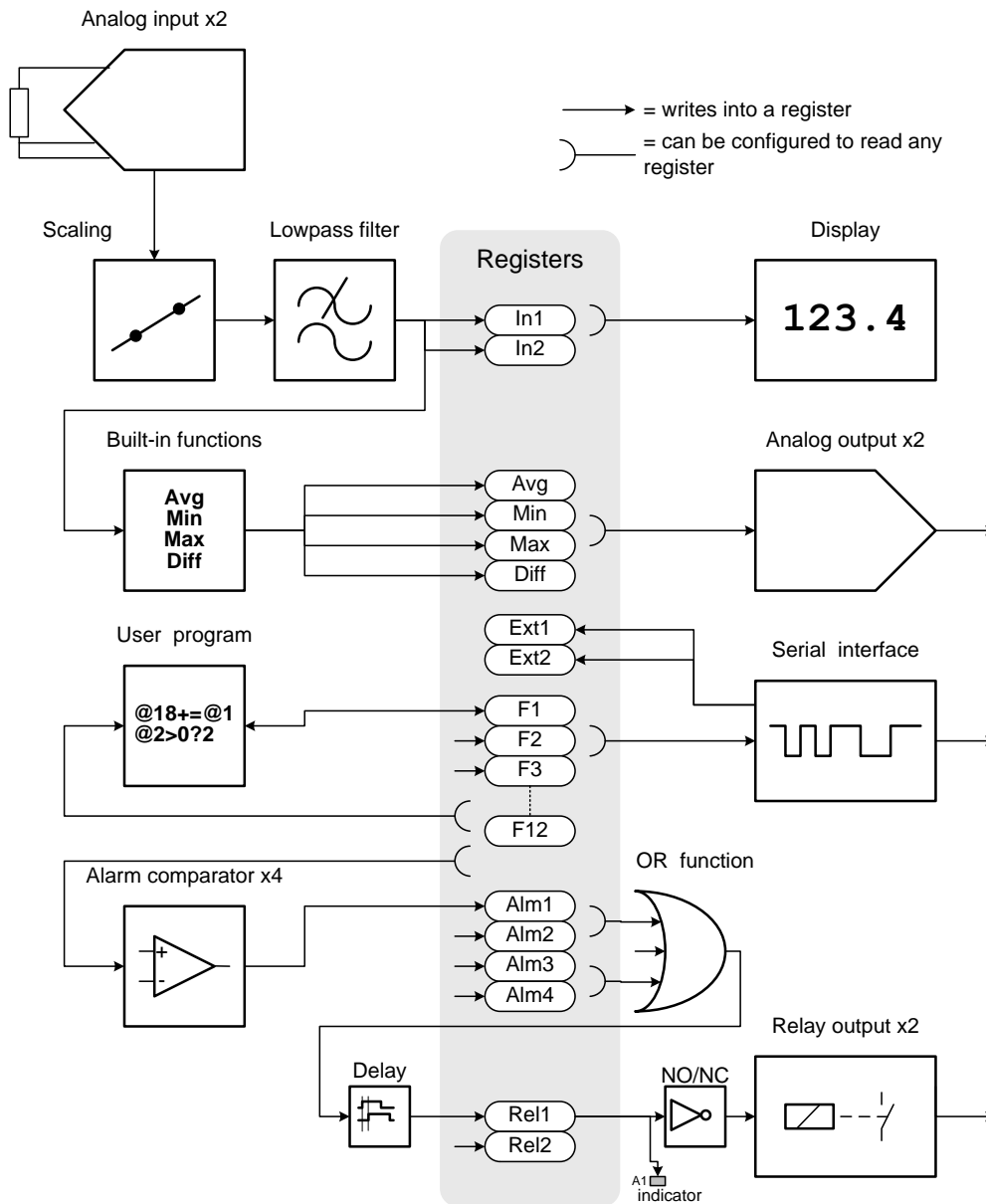


## Configuration settings

There are a plenty of settings that are used to define the operation of the inputs, the outputs, etc. There are two ways to access them:

- On the front panel, see chapter Front panel, page 8.
- Using a PC and a serial communications, see chapter Serial communications, section PC configuration, page 24.

# Registers



The function blocks of this transmitter – inputs, outputs, etc – are quite independent from each other. The information between them is exchanged in so called registers. The registers can be seen in the block diagram above on the gray background. Most of the registers are controlled by some function block. For example, the register "In1" is controlled by the first input – there is the input 1 reading.

Registers F1-F12 are controlled by the user program written in the ELO language.

Any register can be used as a "source" for another block. For example, the analog output 1 can be set to read the register In1 or any other register. The symbol ) represents this in the block diagram.

The register values can be viewed with a monitor function, either on the front panel or the PC configuration program Mekuwin.

Register	Name	Explanation
1	In1	Input 1 scaled reading
2	In2	Input 2 scaled reading

3	Avg	Average of the readings, (In1+In2)/2
4	Min	Smaller of the readings In1, In2
5	Max	Greater of the readings
6	Diff	Difference In1-In2
7	Isens	RTD excitation current in mA, normally about 0.25
8	CJ	Internal cold junction temperature in °C
9	Cycle	The measured time to get all the channels, in seconds
10	Out1	Analog output 1 signal in mA or V
11	Out2	Analog output 2 signal in mA or V
12	Alm1	Logical alarm 1 state (0=passive, 1=active)
13	Alm2	Logical alarm 2 state (0=passive, 1=active)
14	Alm3	Logical alarm 3 state (0=passive, 1=active)
15	Alm4	Logical alarm 4 state (0=passive, 1=active)
16	Rel1	Relay 1 state (0=passive, 1=active)
17	Rel2	Relay 2 state (0=passive, 1=active)
18	F1	Free floating point register for ELo programs
19	F2	Free floating point register
20	F3	Free floating point register
21	F4	Free floating point register
22	F5	Free floating point register
23	F6	Free floating point register
24	F7	Free floating point register
25	F8	Free floating point register
26	F9	Free floating point register
27	F10	Free floating point register
28	F11	Free floating point register
29	F12	Free floating point register
30	Ext1	Modbus controlled register
31	Ext2	Modbus controlled register
32	Dch	What to display: 0=Display/Src1, 1=Display/Src1, etc
33	Keys	Front panel key state, see below

### Keys register

Keys register gives a sum of the codes of the keys currently pressed:

Key	Code
▲	2
▼	4
*	8
▶	16

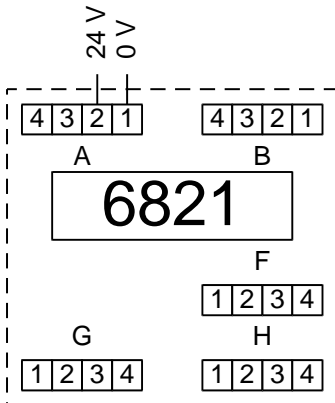
# Power supply

The transmitter 6821 is available for two supply voltage ranges, that are nominally called “24VDC” and “230VAC”.

## 6821-24VDC

This model accepts supply voltage of 20 to 28 V DC or AC. The supply is connected in connector A terminals 1 and 2. Either polarity will do.

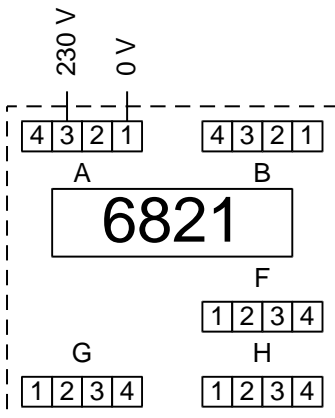
This transmitter will consume less than 200 mA. However at power-up it will need 500 mA.



## 6821-230VAC

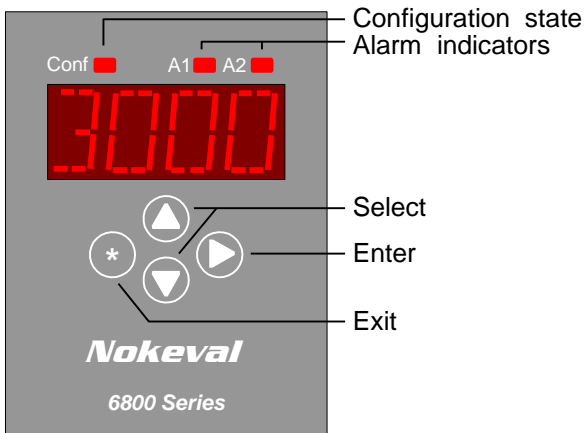
This model accepts supply voltage of 85 to 260 V DC or AC. The supply is connected in connector A terminals 1 and 3. The protective earth is not used.

This unit has an internal pre-fuse. If an external one is used, it should be at least 500 mA T.



When wiring, a special attention has to be paid to ensure that accidentally disconnected wires can't touch other connectors and cause a risk of an electric hazard.

# Front panel



The front panel can be used to monitor the operation of the transmitter and to change the settings. It has several states of operation:

- Normal state – displaying the readings.
- Configuration state – changing the settings.
- Monitor state – displaying the readings and other variable data.

## Normal state

After power-up, the front panel is in the normal state, displaying the input 1 reading (unless otherwise set). The "channel" being displayed can be selected using the ▲▼ keys. While changing the channel, the channel name is displayed until the key is released. Which channels can be selected, depends on the settings (Display/Src1...4).

The ► key can be used to access and change the value of pre-selected registers in order to control the ELo program (see the Display/Ed settings).

A1 and A2 indicate the states of the alarm relays. They are lit if any of the alarms selected for the relay is active. The relay NO/NC setting does not affect the indicators.

## Configuration state

### Entering

Press the \* and ▲ keys simultaneously for two seconds in the normal state to enter the configuration state. When entered, the Conf led will light. If configuration password is set, it must be entered (Cod.0 displayed). In case the password is not known, switch the power off, hold the \* and ► keys pressed and switch the power on again – "PWDC" is displayed briefly. This will also set the serial settings to their default values.

### Navigating

The menu is organized hierarchically. You can move within one menu using the ▲▼ keys and enter a submenu with the ► key. Returning from a submenu is done with the \* key. See the menu chart on page 11.

### Editing

To see or edit a setting value, press ► key.

**Most data types** are edited simply with ▲▼ keys, finally exiting with \* key.

**Floating point values**, such as scaling and the low pass filter, are edited with ▲▼► keys: select digit to edit (blinking) with ► and change it with ▲▼. When the decimal point is blinking, it can be moved with ▲▼. The first digit can be replaced by a minus sign.



A **password** is set as follows: push ▲ to select Set (means password will be used), then push ► to enter the new password. Cod.0 is shown. The password is a sequence of six key presses using all the four keys. Enter the same password twice; if they match, Set is shown again and you can exit with \*. If they didn't match, Off is shown. Redo from start. To disable a password, push ▼ to select Off and exit with \*.

The **math program** can't be edited with the front panel. An RS-485 connection to a computer is required.

### Exiting

When all settings are done, exit from the main level of the configuration menu with the \* key. Two options are shown: Save to keep the settings made, and Undo, to discard all changes. Select ▲▼ Save or Undo and push ►.

## Monitor state

Monitoring can be used to examine the internal readings, called registers. The built-in monitoring is started by pressing \* and ▼ together. Select the item using ▲▼ keys, and finally exit with \*.

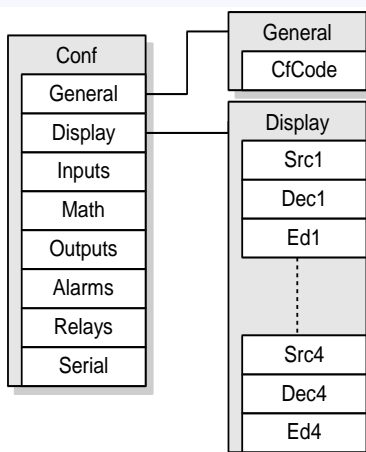
The registers are explained on page 5.

The last item is **Diag**. It can be used to see the diagnostic messages. Push ►. If nothing happens, then there are no messages. If happens, try ▲▼ to see if there are several messages. Exit with \*.

There are three possible diagnostic messages:

- **Sensor Fault:** some channel has improper connection, over ranging, or a broken sensor or wires.
- **AD Error:** the A/D converter is not working. The unit needs service.
- **Math Error:** there is an error in the math program.

## Settings



In the configuration settings menu, there are two submenus concerning the front panel: General, and Display.

### CfCode (General)

This defines a password for the settings. If this is set, the settings can't be accessed without knowing the password.

To change the password on the front panel, advance to the CfCode setting and push ► to edit the value. Options Off and Set are shown; select Set with the ▲▼ keys. Then push ►, and Cod.0 should be displayed. Now enter six key presses using any of the four keys. Then enter the same sequence another time. If they matched, Set is displayed again and you can exit with \*. Otherwise Off is displayed.

### Src1...Src4 (Display)

Display submenu is used to define which readings (or registers) you want to see on the display. Up to four registers can be selected in Src1..Src4 (the registers are described on page 5). If you do not need all the four registers, set the rest to Off. After power-up, the register selected in Src1 is displayed, and the others can be viewed using the ▲▼ keys.

### Dec1...Dec4 (Display)

Defines how many digits after the decimal point is displayed on the registers defined in Src1...Src4. Can be set between -2 and 3. Negative values mean, that the corresponding number of last digits is rounded to zero. Examples with a reading of 12.34:

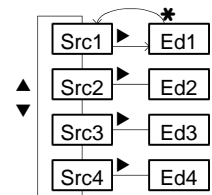
- Dec=2: "12.34"
- Dec=1: "12.3"
- Dec=0: "12"
- Dec=-1: "10" (rounded to nearest 10)

### Ed1...Ed4 (Display)

If the register selected in Src1 is being displayed, and the user pushes the > key, then the register selected in Ed1 is displayed and its value can be changed. Likewise while Src2 is displayed, the register selected in Ed2 can be edited. This way the user can change register values manually and affect the ELo program operation or control an analog output manually.

Other registers than F1...F12 should not be selected in Ed. If the editing is not desired, set the Ed registers to Off.

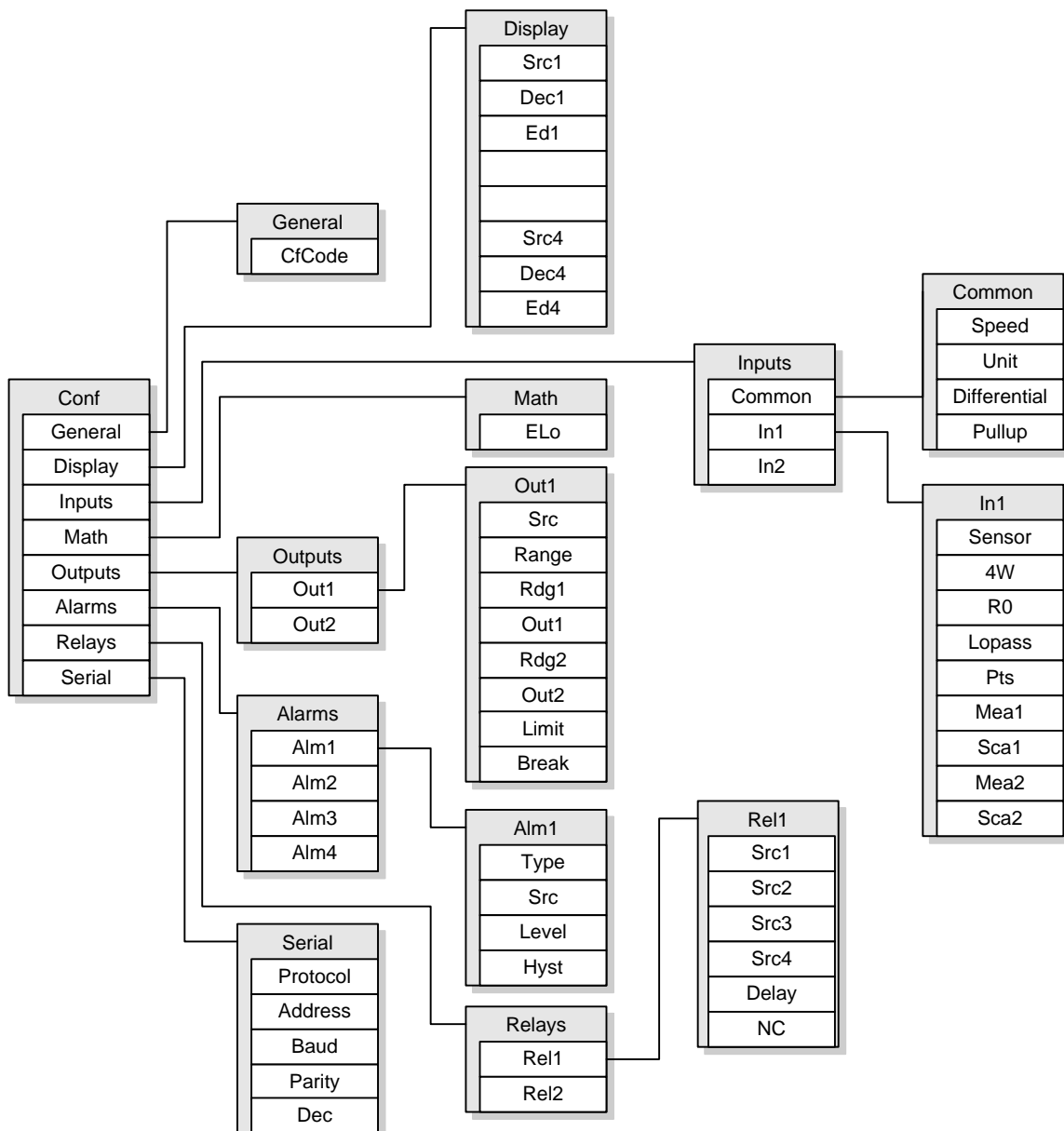
When a register value is changed this way, it is stored in the non-volatile EEPROM memory and restored when the transmitter is powered up next time.



## Menu tree

The complete configuration menu is shown here to aid navigating. The menus are explained in the relevant chapters:

- General and Display: page 9.
- Inputs: page 15.
- Math: page 21.
- Outputs: page 17.
- Alarms and Relays: page 20.
- Serial: page 24.

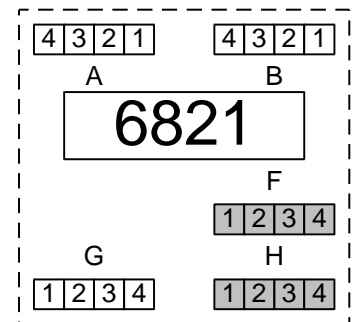


# Inputs

## Connections and jumpers

The input signals are connected in connectors F and H: F is the input 1 and H is the input 2. The connections are illustrated in the next sections.

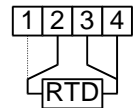
In addition, jumper settings inside the case might need adjusting. The factory setting of the jumpers is suitable for all input signals except the mA signal.



- **In1 = mA:** If closed, an internal 50 ohm shunt resistor is connected between the terminals 2 and 3 to enable the mA measurement on the input channel 1. The jumper must be open for all the other input signals.
- **In2 = mA:** The same for the input channel 2.
- **In2 pin1 = 4w:** If closed, the terminal H1 will function as a fourth RTD wire on input 2, allowing a four-wire connection.
- **In2 pin1 = +15V:** If closed, the terminal H1 will provide a 15 V supply for an external transmitter.
- **Free:** These jumper locations are not used, and the jumper may be taken off and used where needed.

## RTD and resistance inputs

Three-wire connection: If the sensor has two wires the same color, connect these wires in terminals 3 and 4, and the third wire in terminal 2.

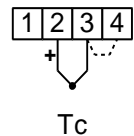


Four-wire connection: Connect the other end of the sensor in terminals 1 and 2, and the other in terminals 3 and 4. If using input 2, open the case and make sure that the jumper "In2 pin1" is set to "4w", not "+15V", otherwise you might burn your sensor.

Also make sure that the mA jumper is not closed on that input channel.

## Thermocouple inputs

Connect the positive wire (K type: green or brown) of the thermocouple in terminal 2, and the negative (white or blue) in terminal 3. If the wires are long, it is also advisable to link the terminals 3 and 4 together.



If you have two thermocouples and they are electrically connected to each other (mounted on the same metallic object), then use the 3-to-4 link on the other channel only, so that the other one is a fully differential input.

Make sure that the mA jumper is not closed on that channel.

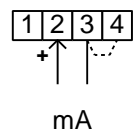
## Voltage inputs

Connect the positive wire in terminal 2, and the negative wire in terminal 3. If the wires are long, it is also advisable to link the terminals 3 and 4 together.

mV Make sure that the mA jumper is not closed on that channel.

## Current inputs

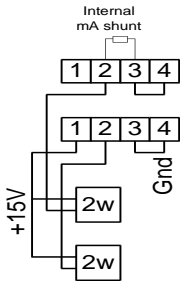
Connect the positive wire in terminal 2, and the negative wire in terminal 3. If the wires are long, it is also advisable to link the terminals 3 and 4 together.



Close the mA jumper on that channel.

### Transmitter supply

This transmitter can provide a 15 V 50 mA supply for an external transmitter. This voltage is output at connector H terminal 1 (+) and terminal 4 (ground). To switch on the voltage supply, the jumper “In2 pin1 = +15V” has to be closed and the jumper “In2 pin1 = 4w” opened. After this, an RTD in four-wire connection must not be connected in connector H, otherwise it will get 15 volts and probably get damaged.



The picture shows an example of connecting two external 2-wire transmitters to this transmitter. Terminal H pin 1 provides the 15 V supply for the transmitters. The current is returned to the terminal 2, which is the positive mA input. The current passes the internal shunt resistor and comes out of the terminal 3, which must be connected to terminal 4 to allow the current to return to the internal ground.

### Potential equalization

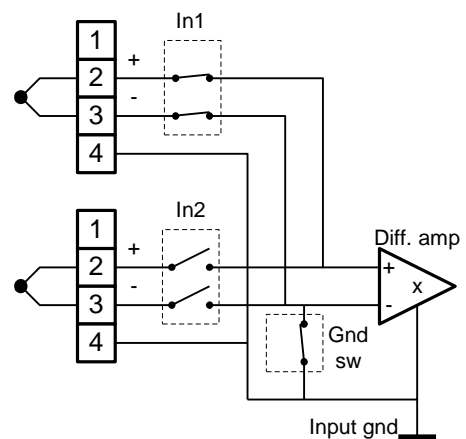
The input circuitry is galvanically isolated from the power supply and output circuits. However the input channels are not isolated from each other. Moreover there is a differential amplifier in each input that requires that the potential of the inputs (terminals 2 and 3) is near the input circuitry ground (terminal 4).

With RTD's, the differential amplifier input is tied to the input ground via the sensor connections and there is no matter. With the other inputs, the differential amplifier input must be tied to the internal ground somehow. There are three different ways:

#### Internal grounding switch

The input terminals 2 and 3 are connected through semiconductor switches to the differential amplifier feeding the analog-to-digital converter. The other channel is separated by open semiconductor switches. In the picture a thermocouple is connected on the both channels as an example. There is no external connection in the terminal 4, the input ground.

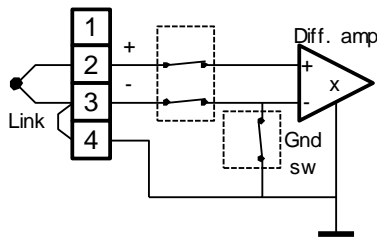
The internal grounding switch is enabled by setting Differential to No in the Inputs / Common menu. This switch will pull the input circuitry ground potential to the negative line potential on the channel being measured.



The potential of the other input can be several volts different than this channel, since it is separated by the switches. However the switches will not tolerate infinitely voltage. The rule is that the both lines (+ and -) of an input must be within 10 V in respect to the negative line of the other input. A difference of more than 30 V may damage the inputs.

This is the easiest way to connect, and the only that will allow several volts between the negative lines of the inputs.

### External linking



If the input sensors are galvanically isolated from each other, then it is recommended to tie them both to the input circuitry ground. Connect a piece of wire between the terminals 3 and 4. This ties them to each other too.

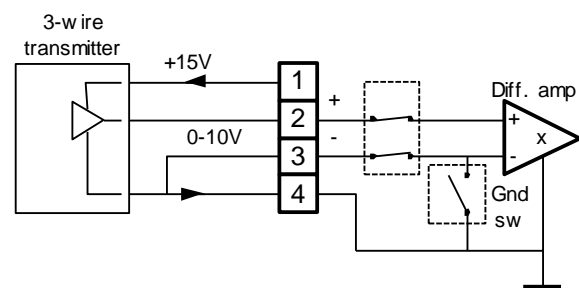
It does not matter if the internal grounding switch is open or closed, since the differential amplifier negative input is externally tied to the input ground. So the Differential setting in Inputs / Common can be in any position.

This connection method will give the best performance against disturbances and overvoltage.

### Other external path

The potential equalization does not have to be a link on the terminals – it can be a longer path as far as it keeps the potential between terminals 3 and 4 small enough. If an external path exists, then it is best to open the internal grounding switch by setting the Differential setting to On in the Inputs / Common menu.

An example of this could be a voltage-output three-wire transmitter. If it is connected with three wires, the lowest wire will have two purposes: it is both the return of the supply current and the negative signal wire. The return current will cause a voltage difference between the wire ends, making an error in the voltage signal. This can be avoided by using separate wires for the supply current return and the signal, as in the picture. The differential amplifier measures the voltage between terminals 2 and 3 using the wires carrying no current.



The differential amplifier inputs (terminals 2 and 3) can tolerate voltage in respect to terminal 4 as follows:

- mV and Tc ranges: -0.15...+0.95 V both.
- 1 and 2.5V and mA ranges: -2...+4 V both.
- 10 V range: negative terminal  $\pm 1$  V, positive -6...+11 V.

Another example of the external path: Two thermocouples connected to the same metal object. In this case, the negative wires are not allowed to be connected together at the transmitter terminals. The link between terminals 3 and 4 can't be used. Instead, have the link on the other input only – it will keep the differential amplifier inputs near ground on both inputs.

## Operation

Several analogue switches are used to connect the A/D converter and the sensor current supplies to the channel being measured. In addition to the two external channels, there are two internal channels: the cold junction temperature (for thermocouples) and the RTD excitation current measurement.

The result is processed according to the sensor type selected. For the temperature sensors, a temperature reading in °C or °F is obtained; for other input types, a mV/V/mA reading is obtained. This transmitter provides a possibility for a free two-point scaling. It can be used to cancel sensor errors or to convert the input signal to “engineering units”. A first-order low pass filter can be applied to remove noise and disturbances.

This transmitter calculates always four inter-channel readings: the average, the smaller, the greater, and the difference of the inputs.

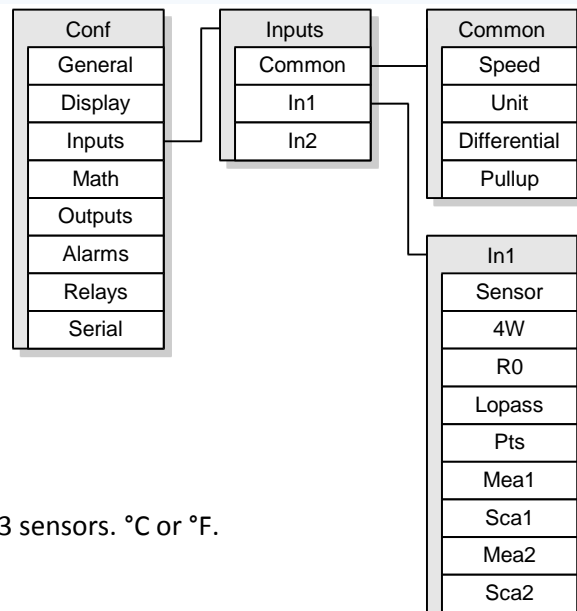
### Fault detection

If the transmitter finds the sensor faulty, the corresponding In1 or In2 register will be set to a NaN (not-a-number) value to indicate an exception. An analog output configured to follow that register will then indicate fault by going to a maximum value. Every logical alarm following the In1 or In2 register will activate.

If either sensor is faulty, the Diff and Avg registers will also go to NaN. The Min and Max registers will ignore the faulty channel, but if both of the inputs are faulty, then Min will go to 100000 and Max to -100000.

## Settings

The configuration menu is divided in several submenus. The input settings are in a submenu called Inputs, which is further divided in Common, In1, and In2 submenus. In1 and In2 submenus are identical.



### Speed (Common)

Measurement speed selection. "Normal" speed is intended for normal use, and the specifications are valid for that. The higher speeds will increase noise. The "Slow" speed can be used, when increased accuracy is needed.

### Unit (Common)

Measurement unit with thermocouple and Pt/Ni/Cu/KTY83 sensors. °C or °F.

### Differential (Common)

Internal grounding of the differential inputs. Not applicable to resistance inputs, like Pt100. This is discussed in detail on page 13.

- **No:** The transmitter uses a semiconductor switch to ground the negative terminal of the channel being measured.
- **Yes:** The inputs are differential (or floating), and the potential equalisation must be done externally.

### Pullup (Common)

Sensor/wire break sensing. If enabled, a weak current (1.5 µA) is fed to the thermocouple sensor to detect faults.

### Sensor (In1, In2)

Input range and sensor selection.

- **Off:** Channel not used. It is recommended to set unused channels to off, in order to speed up the measurement cycle.
- **55mV, 100mV, 1V, 2.5V, and 10V:** Voltage inputs. Can measure negative voltage too, however the 2.5V range reaches only -1 V. The reading is in millivolts or volts according to the name of the range.
- **20mA and 50mA:** Current inputs. Can measure also negative current. The appropriate jumper inside the case has to be closed.

- **400ohm, 4000ohm, and 40000ohm:** Resistance inputs. The resistor is connected in three-wire or four-wire connection. The reading is in ohms.
- **Pt, Ni, Cu, and KTY83:** Resistance thermometers (RTD's). The nominal resistance is set in R0 (see below). The reading is in Celsius or Fahrenheits.
- **TcB-TcT:** Thermocouples. The result is in Celsius or Fahrenheits.

#### 4W (In1, In2)

RTD connection mode. Visible on the RTD and ohm inputs only.

- **No:** Three-wire RTD connection.
- **Yes:** Four-wire RTD connection.

#### R0 (In1, In2)

The nominal resistance of an RTD. With Pt and Ni sensors, this is the resistance at 0°C. With a Pt100 set R0=100. With Cu and KTY83, the nominal resistance is given at 25°C. This setting is not visible on other than RTD inputs.

If the real, calibrated resistance of the sensor at the nominal temperature is known, it can be fed here, in order to cancel the sensor error.

#### Lopass (In1, In2)

First-order lowpass filter for the reading. Attenuates noise and disturbances. Set the time constant (to 63% of step change) in seconds, or 0 to disable.

#### Pts (In1, In2)

Number of the scaling points. The scaling means converting the reading to represent some other (engineering) reading. The scaled value is used on the display, serial output, analog outputs, and alarms.

- **0:** No scaling.
- **1:** One point offset correction. The reading corresponding to Mea1 is scaled to be Sca1 when displayed, using appropriate offset value.
- **2:** Two point scaling. Readings from Mea1 to Mea2 are scaled to be Sca1 to Sca2. Any values can be used, these have not to be the end points.

#### Mea1, Sca1, Mea2, and Sca2 (In1, In2)

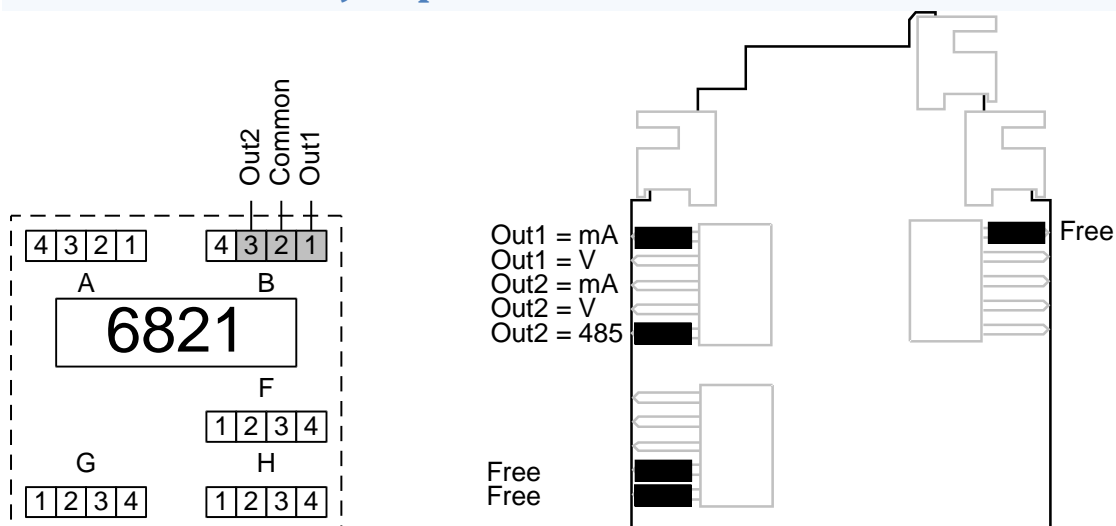
Scaling points. Visibility of these settings depends on the Pts setting. Unscaled reading Mea1 is converted to Sca1, and Mea2 to Sca2.

These scaling points can be conveniently used to **calibrate a sensor-transmitter pair** in a thermal bath. First set the scaling off by setting Pts=0. Apply one or two known temperatures to the sensor and write down the displayed and the real temperatures. Then set Pts to 1 or 2 depending on the number of calibration points, and write the first reading in Mea1 and the real temperature in Sca1. And the same with Mea2 and Sca2 if two points are to be calibrated.



# Analog outputs

## Connections and jumpers



The two analog outputs are provided in the connector B.

There are five jumpers affecting the analog outputs. The first output Out1 can be selected among the mA and V output signals – the corresponding change must be done in the configuration settings too.

The second output can be selected among mA, V, or RS-485 serial outputs.

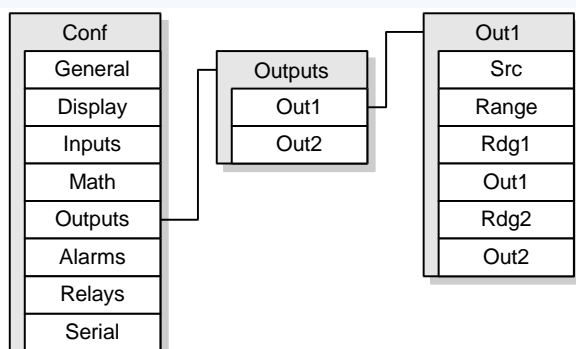
The factory setting is two mA outputs and no serial communications.

## Operation

The analog output can be programmed to follow either input, any built-in function, or any register including the results of an ELo program. A free two-point scaling is provided to convert the reading to a physical signal in mA or V.

When the output is configured to follow an input (e.g. the In1 register) and the sensor is broken, the output will indicate fault as defined by the Break setting. Technically, when the register value is a NaN, the output will indicate fault.

## Settings



The Outputs submenu is further divided in two identical submenus, Out1 and Out2 for two analog output channels.

### Src

The register that the output is taken from. If you want this output to follow the input 1 reading, select In1. If you want to follow the difference In1-In2, select Diff etc. More about the registers on the page 5.

## Range

Analog output range mA or V. The jumpers inside have to be set correspondingly.

## Rdg1, Out1, Rdg2, and Out2

Output scaling. The reading Rdg1 corresponds to the output signal Out1 (in mA or V), and Rdg2 to Out2. These have not to be the end points, since the transmitter is able to extrapolate.

Example: Reading 0-6 (bar) is wanted to give output of 4-20 mA. Settings:

Range = mA

Rdg1 = 0 (bar)

Out1 = 4 (mA)

Rdg2 = 6 (bar)

Out2 = 20 (mA)

## Limit

- Yes: The output is limited between Out1 and Out2 and will not overrange.
- No: The output is not limited and will overrange as much as it physically can.

This setting does not affect during a fault; the Break setting overrides this. Available in firmware V1.5 and newer; V1.0-1.4 will not limit.

## Break

The output behavior when the source is indicating fault (NaN), e.g. when the sensor is faulty. Available in firmware V1.5 and newer; V1.0-1.4 will behave according to the Max option.

- Min: The output will go to 0 mA or 0 V.
- NamurLo: The output will go to 3.5 mA.
- Out1: The output will go to Out1, i.e. the low end of the scaled range.
- Out2: The output will go to Out2.
- Max: The output will go to its maximum value, approx. 22.5 mA or 11 V.

# Alarms and relays

## Connections

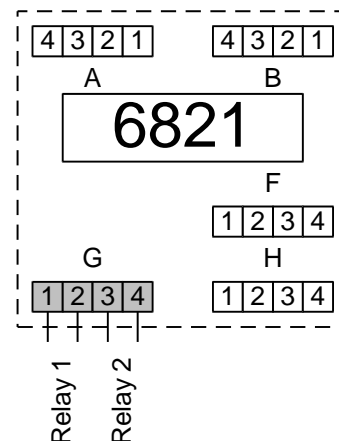
There are two alarm relays inside the transmitter. The contacts of the first relay are connected in connector G terminals 1 and 2, and the second relay in terminals 3 and 4. These have no internal connection elsewhere: they are “potential free”.

The relays can be used to control a 230 VAC line voltage. If one relay is connected to the line voltage, then the other must not be used for SELV circuits for electrical safety reasons.

When there are line voltages present, the wires must be tied e.g. with cable ties so that an accidentally disconnected wire can't cause a hazard.

Controlling a heavily inductive load will shorten the life of the relays. An external snubber circuit or a varistor is then recommended.

When this unit has no power supply, the relay 2 contacts will be closed and relay 1 contacts open.



## Operation

There are four independent logical alarm comparators that are used to examine a single register (e.g. the input 2 reading) whether it is above or below a limit. The result is either “false” (0) or “true” (1). These do not control any relay yet.

After the logical alarms have been configured, the relays can be programmed to follow those logical alarms. Up to four logical alarms can be defined for a relay. If any of them is “true”, the relay will activate.

The relays have a NO/NC (normally open/closed) selection. It affects the relay coil only. The front panel indicators A1 and A2 tell whether the relay is active, that is if any of the logical alarms selected is “true” and the delay has expired, but are not affected by the NO/NC selection.

When the register selected for the logical alarm has a NaN value (Not-a-number used to express a fault), the logical alarm will always be “true”. In practice, if the sensor fails, all the associated alarms will activate.

## Alarm settings

There are four identical logical alarms, one in each Alm1...Alm4 submenu. The first alarm controls the register Alm1 and so on. The relays following these alarms are set up in the Rel1 and Rel2 submenus.

### Type (Alarms/Alm1...Alm4)

Alarm type:

- **Off:** This alarm is not used.
- **Lo:** Low level alarm. Activates if the reading specified in the Src setting goes below the Level setting.
- **Hi:** High level alarm.

### Src (Alarms/Alm1...Alm4)

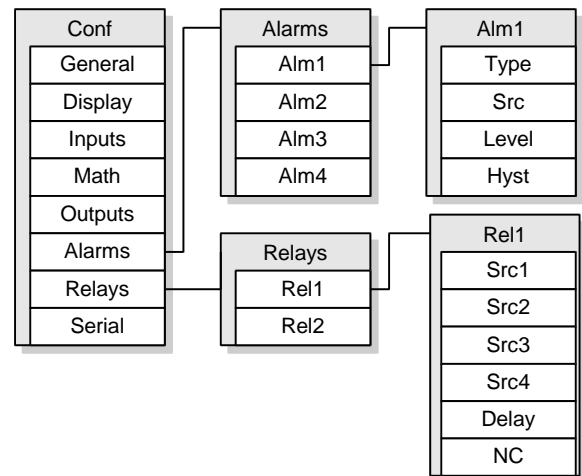
This defines the register that this alarm investigates. E.g. to have a high level alarm when input 1 exceeds 50, select Src=In1 and Level=50.

### Level (Alarms/Alm1...Alm4)

Alarm level. The alarm will activate when the reading goes past this limit.

### Hyst (Alarms/Alm1...Alm4)

Alarm hysteresis. When the alarm has activated, the reading must come an amount defined here back from the Level to be deactivated. E.g. if Level=50 and Hyst=5 and Type=Hi, the alarm will activate at 50 and deactivate at 45. Always positive.



## Relay settings

### Src1...Src4 (Relays / Rel1...Rel2)

Sources for one relay. Select up to four registers that are examined. If any of them has a positive value, this relay is activated. Normally these are one of Alm1...Alm4 registers or Off. Example: If you want this relay to pull when Alarm 1 is activated, select Src1=Alm1 and rest of Srcs Off.

### Delay (Relays / Rel1...Rel2)

Activation and deactivation delay. An alarm must be continuously active for the time specified in Delay in order to cause a real, common alarm.

The delay time is given in seconds. The maximum delay time is 3495 seconds.

### NC (Relays / Rel1...Rel2)

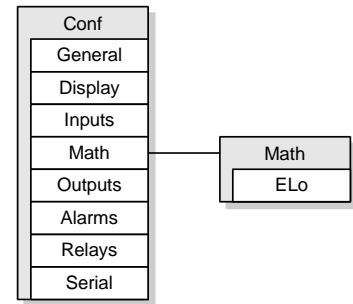
Inverting the relay operation. Affects only the relay coil, but not the LED indicators A1 and A2.

- **No:** Normal operation, normally open: the relay pulls when an alarm is active.
- **Yes:** Inverse operation, normally closed: the relay releases when an alarm is active.

# ELo program

If the capabilities offered by the transmitter are not enough, they can be extended by writing a custom program inside the transmitter. With the ELo programming language, it is possible to do simple calculation, conditional execution, and timing. The program can handle both floating-point values and truth values, e.g. the states of the alarms.

The program can't be entered nor edited on the front panel – an RS-485 connection and a computer is needed. The maximum length is 320 characters, each line feed consuming one character.



## Registers

The unit has so called registers. They are containers, where a variable data is stored. Some registers are used by the unit to store its readings, and some are free to be used by the user program. See page 5 for more information on the registers.

The registers F1-F12 are provided to be used freely by the user program. Other registers than F1-F12 should not be written to. Note that these registers can be accessed from the front panel too, see Ed settings on page 10. Unless used as an “Ed” register, these are initialized to zero at power-up.

In the language, the registers are referenced by using their name or index: e.g. the register F1 is accessed with an expression F1 or alternatively @18 (F1 is the 18<sup>th</sup> register). Before the firmware V1.3, the names can't be used. The names are case-sensitive.

The register Intv or @0 is a special register: every time the user program is executed, it tells the time in seconds from the previous execution. This can be used while generating timings or totalizing a reading. See the examples. The program is executed every time the A/D converter gets a new reading for any channel, normally approx 12 times/sec.

## Program structure

The program consists of lines. Every line has one simple command. The command can change a register or cause a conditional jump inside the program. The program is entered in the configuration settings, Math submenu.

The program run time is limited to 200 operations (or lines) to avoid jamming the whole transmitter due to an erratic program. If the transmitter detects any errors in the program, the Math error diagnostic message will be active (it can be examined in the built-in or Mekuwin Monitor mode).

A lengthy program will slow down all the functions of the transmitter. Using register indices (e.g. @18) instead of the names (F1) gives a faster execution.

During each execution, the register values are not written nor read by the transmitter, thus it is safe to use the same register for intermediate and final value.

### Data moving and math commands:

dest=src	Copies a value from src to dest. E.g. F1=3.14 will place a value 3.14 in register F1. (For firmware V1.0-1.2, use @18=3.14).
----------	--

<code>dest=src1+src2</code>	Sums src1 and src2 value and places the result in dest register. E.g. <code>F1=In1+10</code> will add register In1 contents to constant value 10 and place the result in register F1.
<code>dest=src1-src2</code>	Subtracts.
<code>dest=src1*src2</code>	Multiplies.
<code>dest=src1/src2</code>	Divides.
<code>dest=srcSQ</code>	Calculates a square root. E.g. <code>F1=In1SQ</code>
<code>dest=src1&amp;src2</code>	Bitwise AND. If applied to a floating-point value, it is first converted to a 8-bit integer by rounding towards zero.
<code>dest=src1 src2</code>	Bitwise OR.
<code>dest=src1^src2</code>	Exclusive OR.
<code>dest+=src</code>	Sums src and dest and places the result in dest. Exactly same as writing <code>dest=dest+src</code> .
<code>destX=src</code>	The same for other operators, e.g. <code>F1*=10</code> .

### Jumps and conditional jumps

<code>?lines</code>	Jumps given number of lines forward (+) or backward (-). E.g. <code>?-2</code> will execute the command that is two lines above this line. <code>?2</code> will skip the next line.
<code>x==y?lines</code>	Jumps given number of lines if x equals y. E.g. <code>Alm1==0?3</code> .
<code>x!=y?lines</code>	Jumps if x is not equal to y.
<code>x&lt;y?lines</code>	Jumps if x is less than y.
<code>x&lt;=y?lines</code>	Jumps if x is less or equal to y.
<code>x&gt;=y?lines</code>	Jumps if x is greater or equal to y.
<code>x&gt;y?lines</code>	Jumps if x is greater than y.

### NaN

The floating point registers can contain a Not-a-number or NaN value instead of a normal numeric value to indicate an exception or error. The transmitter itself will set the In1, In2, Diff, and Avg registers to NaN in a case of sensor fault, see Fault detection on page 15. The ELo program can use expressions like `In1==NaN` and `In1!=NaN` to check whether a register contains a NaN. Likewise, the ELo program can set a register to NaN, e.g. `F1=NaN`. Other comparisons with a NaN value will give an undefined result.

### References

<code>1</code>	decimal constant (allowed characters plus, minus, point, digits -1.234 0...9)
<code>In1</code>	Register In1. Requires firmware V1.3.
<code>@1</code>	Register 1, i.e. In1.
<code>@F1</code>	Register indexed by the value of the register F1. E.g. if F1 contains a value of 2, this refers to register In2. Firmware V1.3.
<code>@@18</code>	Register defined by register 18 value

## Examples

### Channels multiplied

```
F1=In1*In2 //The result is placed in register F1.
```

### Peak hold (result in F1)

```
In1<F1?2 //If current rdg is smaller, skip the next line
F1=In1 //Store the new peak
```

### Prevent negative (result in F1)

```
F1=In1 //Take a copy
F1>=0?2 //If not negative, skip the next line
F1=0 //Was negative, have a zero
```

### Clock

```
F1+=Intv //Register F1 will indicate seconds
F1<60?2 //If it is still smaller than 60, skip the next line
F1-=60 //Otherwise decrease by 60 (to approx. 0)
```

### Polynomial $y = 30x^3 - 20x^2 + 10x - 5$

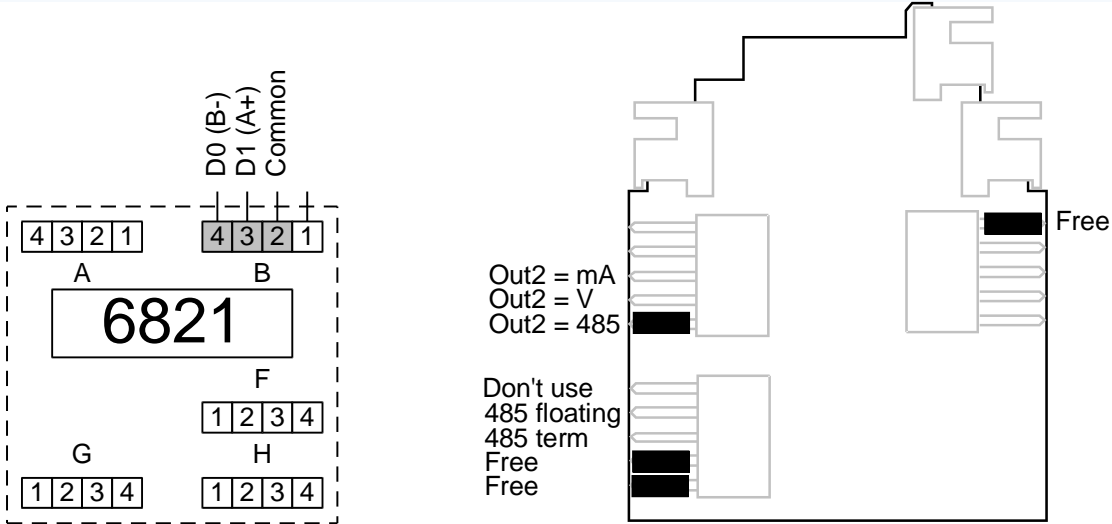
```
F1=30 //x = In1, y = F1
F1*=In1
F1+=-20
F1*=In1
F1+=10
F1*=In1
F1+=-5
```

### Derivator from In1 with about 5 sec gate time

```
F4+=Intv //Timing: increasing by 1.0 per second
F4<5?5 //If not a long time enough, then exit.
F1=In1-F2 //Difference between current and previous
F1/=F4 //...divided by the time difference = derivative!
F2=In1 //”Previous” reading stored for the next evaluation
F4=0 //Timer reset
```

# Serial communications

## Connections and jumpers



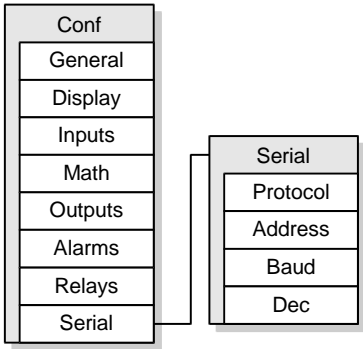
The RS-485 serial communications is alternative to the second analog output – they can't be used at the same time. The selection between them is made by jumpers inside the case, see page 4.

The RS-485 bus is connected in the connector B. Terminal 2 is the internal ground or common. Terminal 3 is the more positive line on the idle, called D1 or +, and terminal 4 is the more negative line, D0 or -. All the three lines should be connected 1:1 to the other devices on the bus. The actual data lines D1 and D0 should use a twisted pair cable with 100-120 Ω impedance. A shielded cable, grounded at one point only, should be used in a noisy environment.

If the RS-485 device that acts as a master does not have a common terminal available, then the “485 floating” jumper should be closed to do the potential equalization using the D1 data line.

The “485 term” jumper should be closed, if this is the last device on the bus. It enables an AC termination: 1 nF + 110 ohms. The jumpers labeled as “Free” can be taken off and used where needed.

## Settings



Serial communications settings. Note that the changes here do not affect until the configuration state has been exit.

### Protocol

Serial protocol:

- **SCL:** Nokeval SCL protocol, slave.
- **Modbus:** Modbus RTU protocol, slave.

### Address

Serial bus address. Valid choices for SCL are 0-123 and for Modbus 1-247. In addition, this device will always answer at address 126 when SCL protocol is used. Modbus will accept General call address 0 but not respond to it.

### Baud

Baud rate selection. Options 1200, 2400, 4800, 9600, 19200, 38400, and 57600 bits per second.



## Parity

Modbus parity. Options 8E1, 8O1, 8N2, and 8N1. Firmware 1.0-1.4 provides 8E1 only and has not this setting.

## Dec

The input readings and other registers can be read as 16-bit integers at Modbus Input registers starting at 1000. This setting defines, how many digits is after the decimal point. E.g. Dec=2: reading 3.1415 is given 314 as an integer.

## PC configuration

All the settings of the transmitter can be accessed from a PC. A free software “Mekuwin” is used, available at the Nokeval WWW site. (Alternatively, with the Modbus protocol, the settings can be read and changed using the standard Holding register functions.)

Of course, an RS-485 connection to the PC is needed, for example a Nokeval DCS770 USB-RS-485 converter.

Connect the transmitter to the PC, start Mekuwin, and make sure both ends have the same serial parameters: the protocol, the baud rate, and the address. Mekuwin Slot parameter, when applicable, should be 0.

## SCL protocol

A more detailed description of the Nokeval SCL protocol can be downloaded from Nokeval WWW site.

This unit understands the following SCL commands:

### TYPE ?

Returns the model name and software version “6821 V1.5” without the quotation marks.

### SN ?

Returns the serial number, e.g. “A123456”.

### MEA CH 1 ?

Returns the scaled reading of input 1. All the registers can be read this way. The response consists of characters -.0123456789. The scientific notation (e.g. 1E3) is never used. In case of invalid reading (e.g. sensor break), the response consists of dashes “-----”. With over/underflow, the response is “^^^^^” or “uuuuu”.

A positive reading is always represented with six significant digits and a negative reading with five, e.g. pi would be represented “3.14159”.

A couple of seconds after power-up, before the first reading is finished, the transmitter will return NAK 0 message to MEA commands, in order to prevent the logger software from logging invalid readings.

### MEA SCAN 1 2

Returns the readings on the registers 1...2 separated by one space. See MEA CH for data representation.

### OUT CH 1 100.5

Sets the value of the Ext1 register to 100.5. The value can contain characters -.0123456789. Only the registers Ext1 and Ext2 can be controlled this way.

### OUT SCAN 1 2 55 66

Controls both of the Ext registers.

**MN xxxxx**

Commands used by the Mekuwin configuration software.

## Modbus protocol

Supported commands:

- **3 Read Holding Registers:** reading the settings.
- **4 Read Input Registers:** reading the readings.
- **6 Write Single Register:** changing the settings and the Ext registers.
- **16 Write Multiple registers:** changing the settings.
- **17 Report Slave ID:** checking the device type.
- **109 Meku:** Mekuwin configuration software uses this.

When the settings are changed by writing a Holding register, the settings are changed to the non-volatile EEPROM memory immediately.

The maximum Modbus frame length is 150 bytes. This sets the limit to the number of registers accessed with commands 3, 4, and 16.

The command 17 will return 0x11 <byte count> 0x00 0xFF, followed with "6821 V1.5 A123456", for example.

When the serial connection settings are changed, the changes do not affect until the transmitter is powered down. This is to prevent breaking the connection while making the changes.

### Data types

- **BOOL:** Off/on setting. 0=False, 1=True in the lower (rightmost) byte.
- **BYTE:** One byte setting. Only the lower (rightmost) byte of the Modbus register is used.
- **WORD:** 16-bit setting.
- **ENUM:** Option list setting. The options listed in section Enum tables.
- **CODE:** Password 12 bits. 0=not used.
- **FLOAT:** 32-bit floating point number IEEE 754. Least significant word first (LSWF, little-endian).
- **STRINGZ:** Zero-terminated string.

Within one Modbus register, the data is represented the most significant byte first (MSBF, big-endian).

### Input registers

Floating Register	Integer register	Name	Type	Values
0..1	1000	In1	FLOAT	Signed
2..3	1001	In2	FLOAT	Signed
4..5	1002	Avg	FLOAT	Signed
6..7	1003	Min	FLOAT	Signed
8..9	1004	Max	FLOAT	Signed
10..11	1005	Diff	FLOAT	Signed
12..13	1006	Isens	FLOAT	Unsigned
14..15	1007	CJ	FLOAT	Signed
16..17	1008	Cycle	FLOAT	Unsigned
18..19	1009	Out1	FLOAT	Signed
20..21	1010	Out2	FLOAT	Signed
22	1011	Alm1	BOOL	
23	1012	Alm2	BOOL	
24	1013	Alm3	BOOL	
25	1014	Alm4	BOOL	
26	1015	Rel1	BOOL	
27	1016	Rel2	BOOL	
28..29	1017	F1	FLOAT	Signed

30..31	1018	F2	FLOAT	Signed
32..33	1019	F3	FLOAT	Signed
34..35	1020	F4	FLOAT	Signed
36..37	1021	F5	FLOAT	Signed
38..39	1022	F6	FLOAT	Signed
40..41	1023	F7	FLOAT	Signed
42..43	1024	F8	FLOAT	Signed
44..45	1025	F9	FLOAT	Signed
46..47	1026	F10	FLOAT	Signed
48..49	1027	F11	FLOAT	Signed
50..51	1028	F12	FLOAT	Signed
52..53	1029	Ext1	FLOAT	Signed
54..55	1030	Ext2	FLOAT	Signed
56	1031	Dch	BYTE	Unsigned 0..3
57	1032	Keys	BYTE	Unsigned

The **readings** can also be read as 16-bit signed **integers** from register 1000 on. The readings have as many digits after the decimal point as is set with the Dec setting in the Serial settings. E.g. Dec=1: divide the integer by 10 ( $=10^1$ ) to get the actual reading.

### Holding registers

**Configuration settings** are in Holding registers 0...268. Limit, Break, and Parity settings are not accessible because they have been added later (V1.5) and would have broken the compatibility to previous versions.

Register	Name	Type	Values
0	Conf\General\CfCode	CODE	
1	Conf\Display\Src1	ENUM	See table E1
2	Conf\Display\Dec1	BYTE	Signed -2...3
3	Conf\Display\Ed1	ENUM	See table E1
4	Conf\Display\Src2	ENUM	See table E1
5	Conf\Display\Dec2	BYTE	Signed -2...3
6	Conf\Display\Ed2	ENUM	See table E1
7	Conf\Display\Src3	ENUM	See table E1
8	Conf\Display\Dec3	BYTE	Signed -2...3
9	Conf\Display\Ed3	ENUM	See table E1
10	Conf\Display\Src4	ENUM	See table E1
11	Conf\Display\Dec4	BYTE	Signed -2...3
12	Conf\Display\Ed4	ENUM	See table E1
13	Conf\Inputs\Common\Speed	ENUM	See table E2
14	Conf\Inputs\Common\Unit	ENUM	See table E3
15	Conf\Inputs\Common\Differ	BOOL	
16	Conf\Inputs\Common\Pullup	BOOL	
17	Conf\Inputs\In1\Sensor	ENUM	See table E4
18	Conf\Inputs\In1\4W	BOOL	
19..20	Conf\Inputs\In1\R0	FLOAT	Unsigned
21..22	Conf\Inputs\In1\Lopass	FLOAT	Unsigned
23	Conf\Inputs\In1\Pts	BYTE	Unsigned 0...2
24..25	Conf\Inputs\In1\Mea1	FLOAT	Signed
26..27	Conf\Inputs\In1\Sca1	FLOAT	Signed
28..29	Conf\Inputs\In1\Mea2	FLOAT	Signed
30..31	Conf\Inputs\In1\Sca2	FLOAT	Signed
32	Conf\Inputs\In2\Sensor	ENUM	See table E4
33	Conf\Inputs\In2\4W	BOOL	
34..35	Conf\Inputs\In2\R0	FLOAT	Unsigned
36..37	Conf\Inputs\In2\Lopass	FLOAT	Unsigned
38	Conf\Inputs\In2\Pts	BYTE	Unsigned 0...2
39..40	Conf\Inputs\In2\Mea1	FLOAT	Signed
41..42	Conf\Inputs\In2\Sca1	FLOAT	Signed
43..44	Conf\Inputs\In2\Mea2	FLOAT	Signed

45..46	Conf\Inputs\In2\Sca2	FLOAT	Signed
47..206	Conf\Math\ELo	STRINGZ	Len=320
207	Conf\Outputs\Out1\Src	ENUM	See table E1
208	Conf\Outputs\Out1\Range	ENUM	See table E5
209..210	Conf\Outputs\Out1\Rdg1	FLOAT	Signed
211..212	Conf\Outputs\Out1\Out1	FLOAT	Signed
213..214	Conf\Outputs\Out1\Rdg2	FLOAT	Signed
215..216	Conf\Outputs\Out1\Out2	FLOAT	Signed
217	Conf\Outputs\Out2\Src	ENUM	See table E1
218	Conf\Outputs\Out2\Range	ENUM	See table E5
219..220	Conf\Outputs\Out2\Rdg1	FLOAT	Signed
221..222	Conf\Outputs\Out2\Out1	FLOAT	Signed
223..224	Conf\Outputs\Out2\Rdg2	FLOAT	Signed
225..226	Conf\Outputs\Out2\Out2	FLOAT	Signed
227	Conf\Alarms\Alm1\Type	ENUM	See table E6
228	Conf\Alarms\Alm1\Src	ENUM	See table E1
229..230	Conf\Alarms\Alm1\Level	FLOAT	Signed
231..232	Conf\Alarms\Alm1\Hyst	FLOAT	Signed
233	Conf\Alarms\Alm2\Type	ENUM	See table E6
234	Conf\Alarms\Alm2\Src	ENUM	See table E1
235..236	Conf\Alarms\Alm2\Level	FLOAT	Signed
237..238	Conf\Alarms\Alm2\Hyst	FLOAT	Signed
239	Conf\Alarms\Alm3\Type	ENUM	See table E6
240	Conf\Alarms\Alm3\Src	ENUM	See table E1
241..242	Conf\Alarms\Alm3\Level	FLOAT	Signed
243..244	Conf\Alarms\Alm3\Hyst	FLOAT	Signed
245	Conf\Alarms\Alm4\Type	ENUM	See table E6
246	Conf\Alarms\Alm4\Src	ENUM	See table E1
247..248	Conf\Alarms\Alm4\Level	FLOAT	Signed
249..250	Conf\Alarms\Alm4\Hyst	FLOAT	Signed
251	Conf\Relays\Rel1\Src1	ENUM	See table E1
252	Conf\Relays\Rel1\Src2	ENUM	See table E1
253	Conf\Relays\Rel1\Src3	ENUM	See table E1
254	Conf\Relays\Rel1\Src4	ENUM	See table E1
255..256	Conf\Relays\Rel1\Delay	FLOAT	Unsigned
257	Conf\Relays\Rel1\NC	BOOL	
258	Conf\Relays\Rel2\Src1	ENUM	See table E1
259	Conf\Relays\Rel2\Src2	ENUM	See table E1
260	Conf\Relays\Rel2\Src3	ENUM	See table E1
261	Conf\Relays\Rel2\Src4	ENUM	See table E1
262..263	Conf\Relays\Rel2\Delay	FLOAT	Unsigned
264	Conf\Relays\Rel2\NC	BOOL	
265	Conf\Serial\Protocol	ENUM	See table E7
266	Conf\Serial\Address	BYTE	0...255
267	Conf\Serial\Baud	ENUM	See table E8
268	Conf\Serial\Dec	BYTE	Signed -2...3

**Ext registers** can be accessed with Modbus and control e.g. an analog output through them. They can be accessed using floating point or integer format. The integers are treated as is, not using the Dec setting in the serial settings.

Register	Name	Type	Values
2000..2001	Ext1	FLOAT	Signed
2002..2003	Ext2	FLOAT	Signed
3000	Ext1	WORD	Signed
3001	Ext2	WORD	Signed

**Input readings** can be read from Holding registers 5000 onwards in floating point format the same way they are read from Input register 0 onwards. They are available in integer format from register 6000 onwards.

## Enum explanations

Table E1

Value	Src
0	Off
1	In1
2	In2
3	Avg
4	Min
5	Max
6	Diff
7	Isens
8	CJ
9	Cycle
10	Out1
11	Out2
12	Alm1
13	Alm2
14	Alm3
15	Alm4
16	Rel1
17	Rel2
18	F1
19	F2
20	F3
21	F4
22	F5
23	F6
24	F7
25	F8
26	F9
27	F10
28	F11
29	F12
30	Ext1
31	Ext2

32	Dch
33	Keys

Table E2

Value	Speed
0	Slow
1	Normal
2	Fast
3	Super

Table E3

Value	Unit
0	°C
1	°F

Table E4

Value	Sensor
0	Off
1	55mV
2	100mV
3	1V
4	2.5V
5	10V
6	20mA
7	50mA
8	400ohm
9	4000ohm
10	40000ohm
11	Pt
12	Ni
13	Cu
14	KTY83
15	TcB
16	TcC
17	TcD

18	TcE
19	TcG
20	TcJ
21	TcK
22	TcL
23	TcN
24	TcR
25	TcS
26	TcT

Table E5

Value	Range
0	mA
1	V

Table E6

Value	Type
0	Off
1	Lo
2	Hi

Table E7

Value	Protocol
0	SCL
1	Modbus

Table E8

Value	Baud
0	1200
1	2400
2	4800
3	9600
4	19200
5	38400
6	57600

# Specifications

## Environment

Storage temp	-30...+70 °C
Oper. temperature	-10...+60 °C
Humidity	Non-condensing
Altitude	<2000 m from sea level
Pollution degree	2
Protection class	IP20

## Inputs

### Pt100

Range	-200...+700 °C
Accuracy	0.05% rdg + 0.2°C (3W) 0.05% rdg + 0.1°C (4W)
Thermal drift	0.01°C / °C
Sensor current	0.25 mA, multiplexed

### Ni100

Range	-60...+180 °C
Accuracy	0.05% rdg + 0.1°C

### Cu10

Range	-200...+260 °C
Accuracy	1°C

### KTY83

Range	-55...+175 °C
-------	---------------

### PtXXX, NiXXX, CuXXX

Range	Same as Pt100, Ni100...
-------	-------------------------

## Thermocouples

TC	range	lin.error
B	400...1700°C	±0.3°C
C	0...2300°C	±0.5
D	0...2300°C	±1
E	-100...900°C	±0.2
G	1000...2300°C	±2
J	-160...950°C	±1
K	-150...1370°C	±0.5
L	-150...900°C	±0.5
N	0...1300°C	±0.1
R	0...1700°C	±0.5
S	0...1700°C	±0.5
T	-200...400°C	±1

Thermal drift	0.02°C / °C (ref 25°C)
Accuracy	0.05% rdg + 0.5°C + lin.error + thermal drift

### mV

Ranges	±55 and ±100 mV
Accuracy	0.1% rdg + 0.01 mV
Load	>1 Mohm

### V

Ranges	1V (-1...+1 V) 2.5V (-1...+2.5 V) 10V (-10...+10 V)
Accuracy	0.05% rdg + 0.01 V
Load	~800 kohm (1, 2.5V) >1 Mohm (10V range)
Thermal drift	50 ppm/°C

### mA

Range	±20 mA
Accuracy	0.005 mA
Thermal drift	50 ppm/°C
Load	50...80 ohm

### Ohm

Ranges	0...400 ohm 0...4000 ohm 0...40000 ohm
--------	--

## Common

A/D conversion	16 bits (±32767)
Speed	All channels in 0.25 sec.
Overvolt protect	±28 V except the mA shunt
Transmitter supply	+15 V 50 mA
Warm-up time	30 min (Tc inputs), 5 min (other inputs)

## Analog outputs

### mA output

Range	0...20 mA or less
Accuracy	0.008 mA
Load	0...600 ohm
Thermal drift	1 µA/°C

### V output

Range	0...10 V or less
-------	------------------

Accuracy	0.005 V
Thermal drift	2 mV/°C

## Serial communications

Connection	RS-485
Protocols	Nokeval SCL slave, Modbus RTU slave
Baud rates	1200, 2400, 4800, 9600, 19200, 38400, 57600
Bits	SCL: 8N1, Modbus: 8E1, 8O1, 8N2, or 8N1.
Min response time	3.5 characters
Max response time	SCL: typ 3, max 25 ms. Modbus: typ 5, max 15 ms (after changing settings, 300 ms for the next command)
Reading all chs	SCL: 100 ms @57600 baud Modbus: 30 ms @57600
Termination	Jumper selectable: None or 110 ohm + 1 nF

## Alarms

Response	Same as meas. cycle + definable delay
Relays	2 A, 250 VAC
Device unpowered	Relay 1 open, 2 closed.

## Power supply

### 24V model

Voltage range	24 V DC/AC ±15%
Max current	200 mA

### 230V model

Voltage range	85...265 V AC/DC
Frequency	45...65 Hz or DC
Max power	5 W
Protection	Reinforced insulation

## Other

Weight	250 g
Mounting	35 mm DIN rail
Connectors	2.5 mm <sup>2</sup> , detachable
Galvanic isolation	Inputs together. Analog and serial output together. These groups and power supply isolated from each other.
Power-up time	1.5 sec

## Regulations

### EMC immunity

EN 61326

### EMC emissions

EN 61326 class B

### Electrical safety

EN 61010-1

## Dimensions

