Nokeval

RMC685

User Manual

For firmware V1.1-1.2

M-Files ID	Version	Date (d.m.yyyy)	Article
1351	24	29.6.2016	93863

CONTENTS

Introduction 3
Warnings 3
Trademarks 3
Manufacturer 3
Quick set-up 4
Installing 6
Maintenance 6
Operation 7
Configuring 8
Input 10
Digital input 15
Analog output 16
User interface 17
Table function 23
ELo program 24
Serial bus RS-485 27
SCL protocol 28
Modbus protocol 29
Nopsa language 33
Ring buffer 33
Agents 34
Specifications 35

INTRODUCTION

The RMC685 and RTC685 are single-channel DIN rail mounted industrial transmitters for temperature sensors and for signal conversion. A thermocouple, an RTD, or an analog process signal (mA or V) can be connected to the input. Various processing options are provided: filtering, table linearization, and custom functions.

An analog output (mA or V) is provided, along with an RS-485 serial bus. The serial bus supports the Nokeval SCL and the Modbus RTU protocols.

The model RMC685 is equipped with a small display and a joystick to enable local configuration and for troubleshooting. RTC685 lacks the display. The devices can also be configured with a computer and a programming cable, or remotely over the RS-485 bus.

Each functional block is independent and interacts the other functions minimally unless configured to do so, which obviates the need to study and configure all the functions before using the device.

The device has two galvanic isolations. The supply is isolated, and the input is isolated. The remaining ports share a common ground.

WARNINGS

 Although the input is galvanically isolated and type-tested with a 1 kV voltage, the isolation is only functional. For safety, the input or output signals must not be connected to a voltage potential more than 50 V AC or 120 V DC with respect to earth.

 Connecting a four-wire RTD sensor in a transmitter configured for mA input and having the external transmitter supply enabled may result in damage for the sensor, as the sensor may not tolerate 15 volts. The transmitter is factory-configured for an RTD sensor.

TRADEMARKS

This device uses the FreeRTOS real-time operating system version 6.0.0. The source code of the RTOS is available on request – contact support@nokeval.com.

MANUFACTURER

Nokeval Oy
Rounionkatu 107
FI-37150 Nokia
Finland

Tel +358 3 3424800 (Mo-Fr 8:30-16:00 EET)
WWW http://www.nokeval.com/
email sales@nokeval.com,
support@nokeval.com

QUICK SET-UP

Connections overview

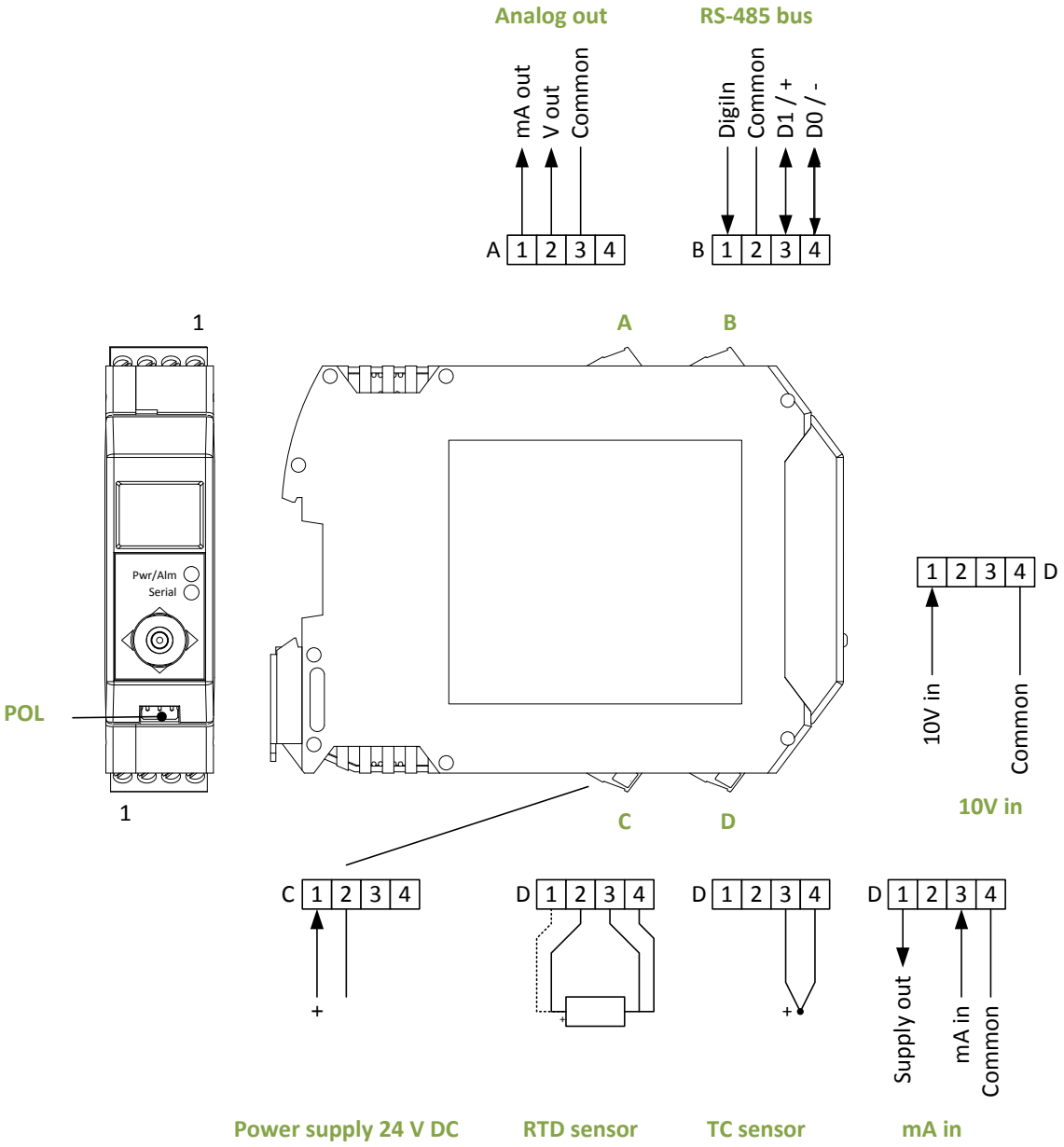


Figure 1

Quick configuration

The most typical configuration settings can be set at the quick configuration menu. For advanced functionality, refer to the later chapters.

To enter the quick configuration menu, press the joystick inwards one second (RMC685 only). Then move the joystick up or down to find “QConf”. Press the joystick inwards or to right to enter the quick configuration menu.

Use joystick up-down movements to browse the items, and if a value is not what desired, press the joystick inwards. While the value is blinking, use the joystick movements to edit the value, and finally press the joystick inwards to finish the editing. For a detailed explanation of editing the values, refer to page 18.

When all the settings are done, press the joystick two times to left to return to the normal display.

Item	Description	Factory default
Sensor	Input signal type. • 0-10V • 0-20mA • 4-20mA • Pt100 • TcJ (thermocouple J) • TcK (thermocouple K) • TcN (thermocouple N) • TcS (thermocouple S)	Pt100
Wires	Number of Pt100 sensor wires. Visible with a Pt100 sensor only.	3
InLo	Input low end scaling. The displayed engineering value at the low end of the input signal, e.g. at 4 mA signal on 4-20 mA range. Not visible on temperature sensors.	0
InHi	Input high end scaling. The displayed engineering value at the high end of the input signal, e.g. at 20 mA signal on 4-20 mA range. Not visible on temperature sensors.	100
Dec	Number of digits to show after the decimal point. Affects the display only.	1
OutRng	Analog output signal type and range. • 0-10V • 0-20mA • 4-20mA.	4-20mA
OutLo	Output low end scaling. The displayed engineering value that will make the output signal to be at its low end, e.g. 4 mA on 4-20 mA range.	0
OutHi	Output high end scaling.	100

A mixed use of the quick and the full configuration menu is not recommended, since using the quick configuration will affect (and possibly overwrite) multiple settings in the full menu. Changing the settings in the quick configuration menu will affect the following settings: Input/Sensor, Input/R0, Input/Wires, Output/Range, Output/Lo, Output/Hi, UI/Screens/1/Upper/Src, UI/Screens/1/Upper/Text, UI/Screens/1/Lower/Src, UI/Screens/1/Lower/Dec, UI/Screens/1/Lower/Text.

INSTALLING

Mounting

The transmitter is intended to be mounted on a standard 35 mm DIN rail. Hook the groove in the back of the transmitter to the outer lip of the rail and tilt the device until the lock engages. To remove, pull the lock downwards with a flat screwdriver.

The mounting position is free, but for maximum accuracy with thermocouples, an “upright” position is recommended: connectors A-B on top. With thermocouples, avoid heat sources near the input connector.

Jumpers

This device does not have user-configurable jumpers.

Connections

The connector blocks are detachable for easier wiring. Use a flat-bladed screwdriver to carefully lever them off. When levering the connector D, take care not to damage the blue cold junction temperature sensor behind the connector block.

The connections are explained in detail in the relevant chapters. A quick overview is in Figure 1 on page 4.

Power supply

This device requires a 24 V DC supply connected to connector C. The positive line is connected to terminal C1 and negative in C2. Reverse polarity will not damage the device. The supply is galvanically isolated from the other ports.

MAINTENANCE

Cleaning

The plastic parts can be cleaned with a soft cloth and soap water or isopropyl alcohol. The cloth must be damp but not wet. With the model RMC685, it is especially important to avoid water entrance between the LCD glass and the plastic front panel.

Recalibration

When maximal accuracy is important, it is recommended to recalibrate the transmitter annually at the manufacturer.

Calibration certificate

A calibration certificate for a new or recalibrated device can be downloaded for free at Nokeval web site based on the serial number.

OPERATION

Registers

The continuously varying values, e.g. measured input reading, table output, state of the digital input, and the results of the ELo program, are collected in a 1-dimensional table, called the table of registers.

The different operational blocks of the transmitter are configured to watch one or more registers. The analog output is most often configured to watch, or follow, the measured input reading, but can be configured to follow any other register for advanced functionality.

Here is a complete list of the registers:

Register	Number	Description
In	1	Measured, scaled and filtered input reading.
CJ	2	Measured cold junction temperature.
DigiIn	3	State of the digital input. 0=passive, 1=active.
Table	4	Output of the (linearization) table.
Out	5	Analog output in mA or V.
Setp1	6	Setpoint value 1.
Setp2	7	Setpoint value 2.
F1…F12	8…19	ELo registers.
Ser1…Ser2	20…21	Serial bus controlled registers.
Screen	22	Currently displayed screen number 1…4.
Keys	23	State of the joystick.

The current register values can be viewed with the Monitor function (RMC685 only): Press ✱ the joystick inwards one second and browse ▲▼ to Monitor. Enter with ▶. Browse the items with ▲▼. Finally exit with ◀.

Self-diagnostics messages

The transmitter may detect some internal or external problems. The messages can be viewed in the diagnostics menu. Using the integral display, see Self-diagnostics on page 20. Using Mekuwin, refer to its documentation. The messages are:

Message	Description	Corrective actions
EEPROM	The internal non-volatile memory storing the configuration settings and the factory calibration values is either faulty or the contents have been corrupted.	Check all the configuration settings and save them to the EEPROM. Cycle the power off-on. If the message persists, the device needs service.
ADC error	The microcontroller does not get a response from the A/D converter.	Check that the supply voltage is within specifications. Try to disconnect the input connections. The device needs service.
Sensor fault	The input signal is not what it should be.	Check the condition of the sensor. Check the input wiring. Check the input settings.
Math error	The ELo program has an error.	Go to the configuration menu and check the Math/Error and Math/ErrLine values to find the error.

CONFIGURING

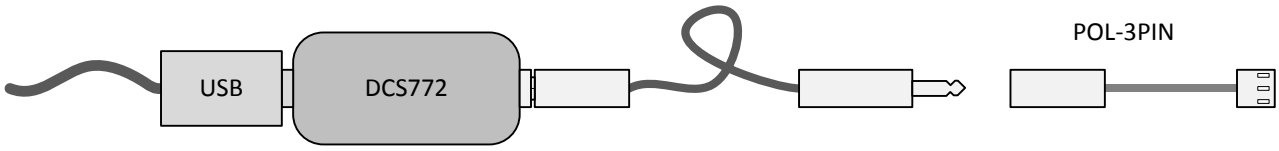
This device can be configured using the integral display and joystick, or with an aid of a computer.

With the integral display (RMC685 only)

Using the display and the joystick is described in Configuration state, page 17.

With a programming cable

The programming cable to be used is Nokeval DCS772, connected to a USB port. Currently the PCB version must be V1.3 or newer (since the older ones lack the echo suppression which is required with the Modbus protocol). In addition, an adapter POL-3PIN is needed. The adapter is connected to the 3.5 mm plug of the programming cable, and the other end to the RMC685 connector marked “POL” in Figure 1 on page 4.



A free Microsoft Windows software Nokeval Mekuwin is used. About using Mekuwin, refer to its documentation. The contents of the target device configuration menu are described in the relevant sections of this manual.

The communications parameters are always:

Protocol	Modbus RTU
Baud	9600
Parity	8E1
Address	1

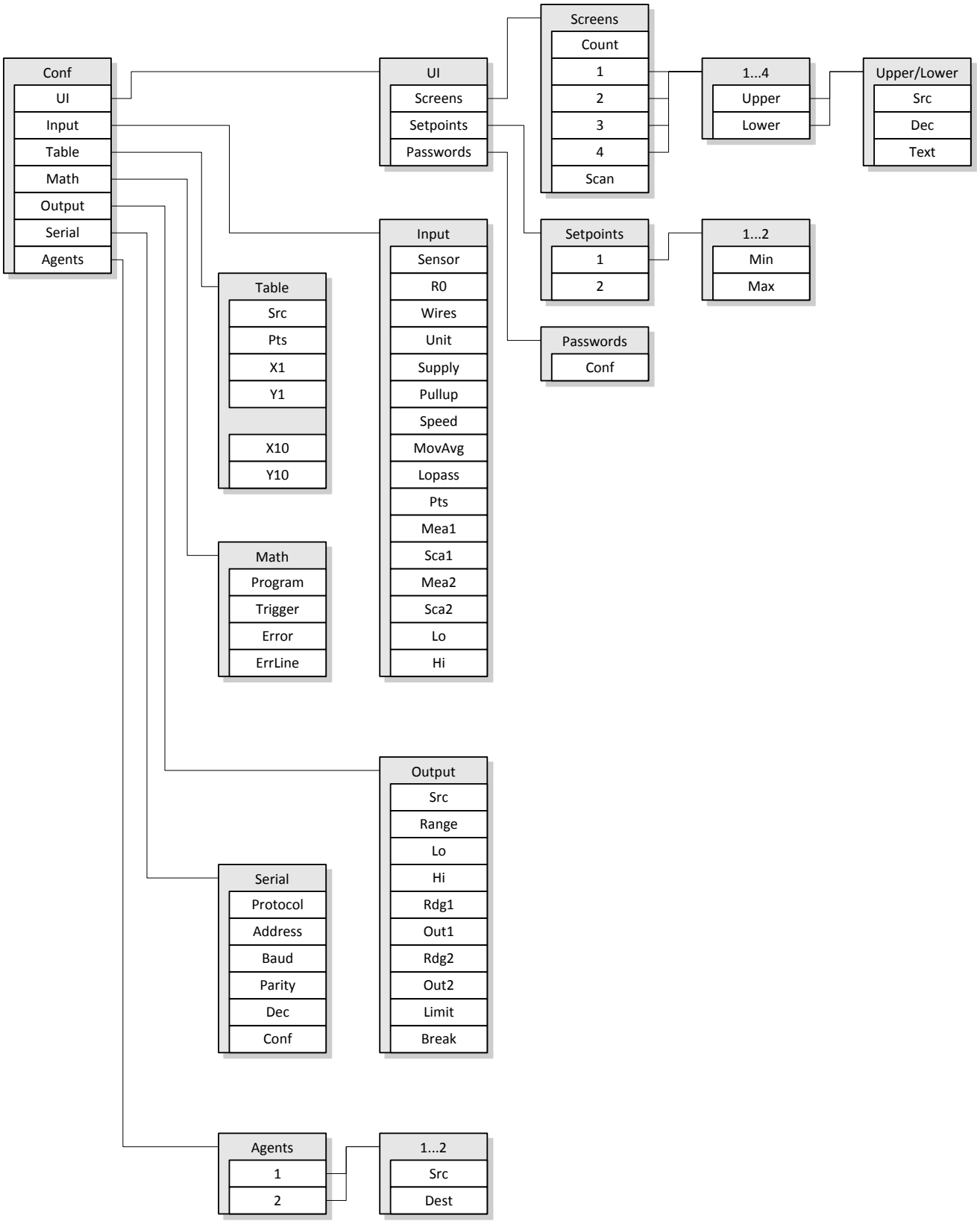
Over the serial bus

Mekuwin can be used over the RS-485 bus. The Mekuwin communications parameters must match the selections made in the device (Serial menu). Both Nokeval SCL and Modbus RTU protocols can be used.

Moreover, it is possible to change the configuration settings remotely by writing the appropriate Modbus Holding registers. This is allowed only if the Serial/Conf setting is enabled in the transmitter configuration menu.

Configuration menu

The contents of the configuration menu are described in the relevant chapters. A quick overview is shown here. Some of the settings will be hidden when not used.



INPUT

General

The configuration settings affecting the input are collected in the Input submenu of the configuration menu.

The final scaled and filtered input reading is placed in the ***In*** register to be available for the other functions, including the analog output, the integral display, and the serial bus.

RTD sensors including Pt100

Connections

Before connecting an RTD sensor, make sure that the transmitter is not configured for mA input, because selecting a mA range and enabling the transmitter supply at terminal D1 could damage the sensor.

- Two-wire RTD: connect the sensor in terminals D2 and D4. Please note that the wiring resistance will cause significant error in the measurement, and a short cable must be used in this configuration.
- Three-wire RTD: connect one end of the sensor in terminals D3 and D4 (these wires usually share the same color), and the other end in terminal D2.
- Four-wire RTD: connect one end of the sensor in terminals D3 and D4, and the other end in D1 and D2. Four-wire measurement yields the best accuracy and is recommended whenever possible.

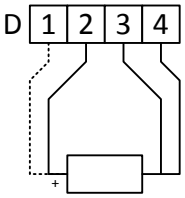


Figure 2

The maximum sensor cable length is 30 m. A shielded cable earthed at one point near the transmitter is recommended in noisy environments.

Settings

In the Input submenu of the configuration menu, set ***Sensor*** to Pt, Ni, Cu, KTY83, or NTCLE3977.

Set the ***R0*** setting to correspond to the nominal resistance of the sensor in ohms. For Pt100 and Ni100, the correct value is 100. For Pt1000, enter 1000.

If the precise resistance of the sensor is known (e.g. it is calibrated and marked on the sensor), it can be entered in R0 to reduce the sensor error.

For Pt and Ni sensors, the nominal resistance is the resistance at 0 °C. For Cu, KTY83, and NTCLE3977 sensors, it is the resistance at 25 °C.

The ***Wires*** setting must match the number of wires used to connect the sensor.

Select the measurement ***Unit***: °C, °F, or K.

The input scaling is usually disabled by setting ***Pts*** to 0. The scaling options can be used to remove the measurement errors as described in Sensor error correction on page 14.

See also Common input settings on page 13.

Thermocouples

Connections

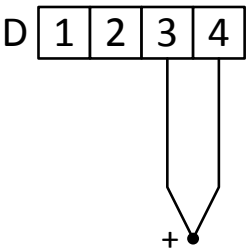


Figure 3

Connect the positive wire of the thermocouple sensor in terminal D3 and the negative wire in D4. Please note that if an extension cable is used, it must be intended for the thermocouple used, and the cable must be connected with correct polarity. With thermocouple K, the positive wire is usually green or brown.

The maximum cable length is 30 m. A shielded cable earthed at one point near the transmitter is recommended in noisy environments.

Settings

In the Input submenu of the configuration menu, select the correct thermocouple type in the ***Sensor*** setting. The thermocouple options begin with Tc, e.g. TcK. The thermocouples supported are B, C, D, E, G, J, K, L, N, R, S, and T. The measurement ranges for each of these are specified on page 35.

Select the measurement ***Unit***: °C, °F, or K.

See also Common input settings on page 13.

mA input

Connections

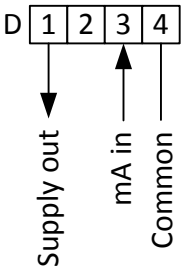


Figure 4

An active mA signal is connected to terminals D3 (positive) and D4. There is a 50 Ω shunt resistor between these inside the transmitter, accompanied with a self-resetting fuse, which will increase the total resistance to approx. 70 Ω. When the transmitter is powered off, the current cannot flow through it.

This transmitter can supply an external transmitter with 15 V ±10%, max 50 mA. The supply must be enabled with a configuration setting ***Supply***. This supply is available between terminals D1 (positive) and D4. The supply shares the negative terminal of the mA input.

An external two-wire (passive) transmitter can be connected between terminals D1 (positive) and D3. The supply has to be engaged with the ***Supply*** configuration setting. The external transmitter must not require more than 12 V supply voltage.

The maximum cable length is 30 m.

Settings

When the input signal is a standard 0-20 or 4-20 mA signal, select the appropriate range in the ***Sensor*** setting in the Input submenu of the configuration menu. Scale the signal with the ***Lo*** and ***Hi*** settings. The scaling is used to convert the raw mA reading to a meaningful engineering value, e.g. a pressure reading. The engineering value corresponding to the low end of the signal (4 mA on the 4-20 mA range) is entered in the Lo setting, and the high end in the Hi setting. Example: a pressure transmitter gives a signal of 4-20 mA corresponding to a pressure range 0 to 6 bar: set Lo=0 and Hi=6.

For more exotic signals, select the smallest sufficient range among 24mA, 1.5mA, and 0.18mA. Each is capable to measure negative current too. If a scaling is desired, use the Pts and the Mea/Sca settings as described in Advanced scaling on page 13.

If the supply for an external transmitter is required, switch on the ***Supply*** setting.

See also Common input settings on page 13.

Fault detection

On the 4-20mA range, an input signal below 3.68 or above 20.8 mA for more than 30 samples (4 seconds on Normal speed) is considered to indicate fault to conform to Namur NE 43. Dashes will be displayed, and the analog output will behave as configured with its Break setting.

Voltage inputs

Connections

For ranges 9mV, 70mV, 290mV, and 1100mV, use terminal D3 for the positive wire, and terminal D4 for the other.

For ranges ±1100mV, 0-10V, and 11 V, use terminal D1 (positive) and D4. Before connecting a signal in this configuration, make sure that the transmitter is not configured for a mA input with the supply option enabled to prevent a damage that could be caused by the 15 V transmitter supply.

The maximum cable length is 30 m. With small signals and also in extremely noisy environments, a shielded cable earthed at one point near this transmitter is recommended.

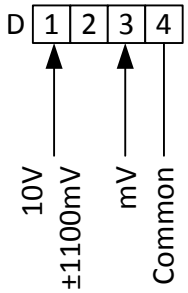


Figure 5

Settings

If using the standard signal of 0-10 V, set the Sensor setting in the Input submenu of the configuration menu to 0-10V. Scale the signal with the ***Lo*** and ***Hi*** settings. The scaling is used to convert the raw voltage reading to a meaningful engineering value, e.g. a pressure reading. The engineering value corresponding to 0 V signal is entered in the Lo setting, and the 10 V value in the Hi setting. Example: a pressure transmitter gives a signal of 0-10 V corresponding to a pressure range 0 to 6 bar: set Lo=0 and Hi=6.

With other than 0-10 V signal, select one of the ranges 9mV, 70mV, 290mV, 1100mV, ±1100mV, 11V in the Sensor setting. Select the smallest sufficient range in order to obtain the best accuracy. All the ranges can measure negative voltage at least to -70 mV; the ±1100mV and 11V ranges can measure -1100 and -11V respectively. The reading is expressed in the units corresponding to the name of the range (volts in 11V range, millivolts otherwise). If desired, use the two-point scaling (Pts and Mea/Sca settings) to scale the voltage to an engineering value as described in Advanced scaling on page 13.

See also Common input settings on page 13.

Resistance input and potentiometer

Refer to the RTD connections on page 10.

The resistance input can be used to measure an RTD that is not directly supported by this transmitter. The linearization table function may be used to handle the sensor non-linearity.

Another use of the resistance input is to read a potentiometer in a variable resistance configuration. The potential divider connection, also known as slidewire, is not supported by this transmitter. Select the smallest sufficient ohm range in the Sensor setting, and use the two-point scaling to convert the ohms to a meaningful position reading as described in Potentiometer teaching on page 14.

Common input settings

Sensor fault detection

With mV, thermocouple, RTD and ohm inputs, this transmitter is able to intermittently inject a small 100 nA current to the sensor wires in order to detect a failed sensor or wiring. The sensor fault detection is enabled by setting the ***Pullup*** setting on. It is generally recommended to keep this setting on, but when requiring the lowest noise or using a high-impedance voltage source, the setting should be set off.

On the 4-20mA range, an input signal below 3.68 or above 20.8 mA for more than 30 samples (4 seconds on Normal speed) is considered to indicate fault to conform to Namur NE 43.

When an input fault is detected, the *In* register will indicate fault instead of having a reading. The other functions of the transmitter then behave as they are programmed to do in such a situation. The display will show dashes (-----) etc.

Measurement speed

The ***Speed*** setting defines the A/D conversion rate and some internal settling times. The noise and accuracy specifications are given for the Normal speed. A lower noise can be achieved by selecting Slow.

Correspondingly selecting a faster speed will increase the noise (flicker) of the input reading, and can also reduce the accuracy overall. The faster speeds should be used only when really necessary.

Approximate update rate in Hz on different speed settings:

Slow	1.9
Normal	7.8
Brisk	15.6
Fast	50
Super	100

The speed, especially the highest speeds, depends on the other functions of the transmitter.

Please note that the analog output of RMC685 is not fast, and the fastest measurement speeds can be utilized via the serial bus only.

Filters

Noise and short disturbances in the input reading can be attenuated with two kinds of filters. The filters can be used simultaneously.

The moving average filter has a finite impulse response. It remembers the selected amount of latest readings and averages them. The number of samples is configured in the MovAvg setting. Due to the limited memory, the maximum number of samples is 20. When this kind of filtering is not desired set MovAvg to 1.

The 1st order lowpass filter has infinite impulse response, which means that after a step change in input signal the filtered reading will approach the new value forever in theory. First fast, then slower and slower. The main advantage is that, unlike the moving average filter, the lowpass filter can be configured to attenuate very strongly. Also in a control loop, the 1st lowpass is easier to handle than the moving average. The time constant is given in seconds. A setting of 0 disables the filter.

Advanced scaling

On non-standard input ranges the input can be scaled to an engineering value using the two-point free scaling.

To enable the two-point scaling, set the ***Pts*** setting to 2.

Enter the first “raw” or unscaled reading in Mea1 setting, and the corresponding engineering or scaled reading in Sca1. Do the same for the Mea2/Sca2 pair. Linear interpolation is used between these points, and linear extrapolation outside.

Example: input signal is 1-5 V, corresponding to a flow of 0 to 10 m^3/h:

- Sensor = 11V (the smallest sufficient range)
- Mea1 = 1 (volts)
- Sca1 = 0 (m^3/h)
- Mea2 = 5 (volts)
- Sca2 = 10 (m^3/h)

The scaling can also be performed with the Table function, either alone or combined with this scaling.

Sensor error correction

To enhance the measurement accuracy of the combination of this transmitter and the sensor, a one- or two-point calibration can be performed e.g. in a thermal bath with an accurate reference meter.

To do a one-point correction, set ***Pts*** to 1. Give a Lock command (see below) for the ***Mea1*** setting, which will make the transmitter to copy the current uncorrected input reading to Mea1. Then enter the reference meter reading in ***Sca1***. The Mea1 and Sca1 values should be quite close to each other when correcting a small sensor error. The transmitter will add a value of Sca1 minus Mea1 to every reading - it is an offset correction.

A two-point correction is done similarly to the one-point correction, but Pts is set to 2 and the operation is performed on Mea2/Sca2 pair in addition to Mea1/Sca1. The transmitter will then use linear interpolation and extrapolation to correct the reading.

If more than two points are needed, use the Table function instead of the input scaling. Its operation principle is very similar, but it provides up to 10 points. All the other functions (display, analog output, etc.) must then be changed to follow the Table register instead of the In register.

To disable the sensor error correction, set Pts to 0. Changing Pts will not destroy the Mea and Sca values, although they may be hidden.

The ***Lock command*** is given in Mekuwin by clicking the L button next to the value. On the integral display (RMC685), browse to the item (Mea1 or Mea2) but do not start editing – nothing should be blinking. Now press the joystick one second, and a pop-up menu appears. Select Lock.

Potentiometer teaching

This transmitter can read a potentiometer by measuring its resistance. What the transmitter sees is ohms. To convert the ohms to a meaningful value, e.g. position or angle, use the two-point scaling.

First set Pts to 2. Move the potentiometer to the first teaching position. Give a Lock command (see Sensor error correction) for the Mea1 setting to copy the current ohms reading to Mea1. Enter the corresponding engineering value in Sca1. Repeat the operation on Mea2/Sca2 pair.

DIGITAL INPUT

The digital (switch) input can be used alongside the ELo program. The current switch state is available in the DigiIn register. The input is read every 50 ms. There is no additional debounce logic.

The input is of NPN type. The terminal B1 is pulled to 3.3 V internally by a 10 kΩ resistor. When the terminals B1 and B2 are shorted externally, the input is considered to be in its active state, and a value of 1 is placed in the DigiIn register. When the terminal B1 is floating or connected to a voltage higher than 2.5 V, the DigiIn register will contain 0.

There are no configuration settings related to the digital input.

ANALOG OUTPUT

The analog output settings are in the Output submenu of the configuration menu.

Connections

An active mA signal is obtained from terminal A1, and the current must return to A3.

A voltage output is available between terminals A2 (positive) and A3.

The mA and V outputs can't be used simultaneously.

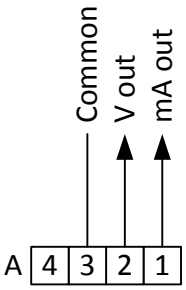


Figure 6

Source

The analog output usually follows the scaled and filtered input reading, which is obtained by selecting In in the ***Src*** setting. It is possible to select any other register. The analog output can be taken from the Table output, from a Setpoint register, from an ELo program controlled register, or controlled with the serial bus.

Range and scaling

The output has two kinds of scaling: fixed range and free. Fixed scaling is used to easily generate a standard signal, while the free scaling can produce exotic scalings.

If the output signal desired is any of 0-20 mA, 4-20 mA, or 0-10 V, select the corresponding range in the ***Range*** setting. Enter the low end scaling in the ***Lo*** setting and the high end scaling in the ***Hi*** setting. When the input value obtained from the register selected with Src corresponds to the Lo setting, the output will be at the low end of the range, e.g. 4 mA if using the 4-20 mA range.

For any other output range, select a free range mA or V in the Range setting. Then use the Rdg and Out settings to scale. When the input value corresponds to Rdg1, the physical output will be Out1 mA or V. When the input corresponds to Rdg2, the output will be Out2. Linear interpolation and extrapolation is used.

An example of free scaling: Generate 1-5 V signal when the input is 0-100 °C. Select V range, and enter Rdg1=0, Out1=1 (volts), Rdg2=100, Out2=5.

Limit and break

The ***Limit*** setting defines if the analog output is allowed to go beyond the range. If enabled, the output is limited within the range selected with the Range setting, or between the Out1 and Out2 when using the free scaling. In addition, on the 4-20mA range the output is limited between 3.8 and 20.5 mA (NAMUR NE 43).

The ***Break*** setting defines how the analog output behaves in a fault condition, e.g. when the analog output is configured to follow the input and the sensor is found to be faulty.

Min	The analog output will go to the lowest signal possible – 0 mA or 0 V, even if Limit is enabled. On the fixed 4-20mA range the output will go to 3.5 mA.
Lo	The analog output will go to the low end of the range, e.g. 4 mA when using the 4-20 mA range, or to the Out1 value when using the free scaling.
Hi	The analog output will go to the high end of the range, e.g. 20 mA.
Max	The analog output will go to approx. 22.5 mA or 11 V, even if Limit is enabled.

USER INTERFACE

The term user interface refers to the LCD display, the joystick, and the two LED indicators in the face of the RMC685 transmitter. The user interface can be used to watch the desired values, to configure the transmitter, and to assist in troubleshooting.

The joystick operations are symbolized with ◀▲▼▶ and ✱ in this chapter. The triangles should be self-explanatory. Symbol ✱ means pressing the joystick inwards.

The user interface has various states:

State	Description	Explained at
Normal	The contents of the display can be customized, but most typically it indicates the current input reading and analog output value. Also allows changing the setpoint values.	Normal state and the screens, page 20.
Quick config	Configuring the transmitter with a minimal set of options.	Quick configuration, page 4.
Configuration	Viewing and editing all the configuration settings to alter the operation of the transmitter.	Configuration state, page 17.
Monitor	Viewing the values of the registers, used mostly in troubleshooting.	Monitoring and simulating, page 19.
Diagnostics	Shows the internal or external faults detected.	Self-diagnostics, page 20.
Calibration	For factory calibration.	Not in this manual.

Configuration state

The configuration state is used to view and alter the configuration settings concerning the various functions of this transmitter. An alternative way to configure is to use a programming cable or a serial bus along with computer software Mekuwin, as explained on page 8.

Entering

To enter the configuration state, press the joystick inwards (symbol ✱ used in this manual) one second. In the menu, select ▲▼ Conf and press ✱. If a configuration password has been set, it must be entered now. The password consists of six joystick movements ◀▲▼▶. If an incorrect password is entered, the display will shortly indicate “Wrong” and return to the normal state.

Navigating

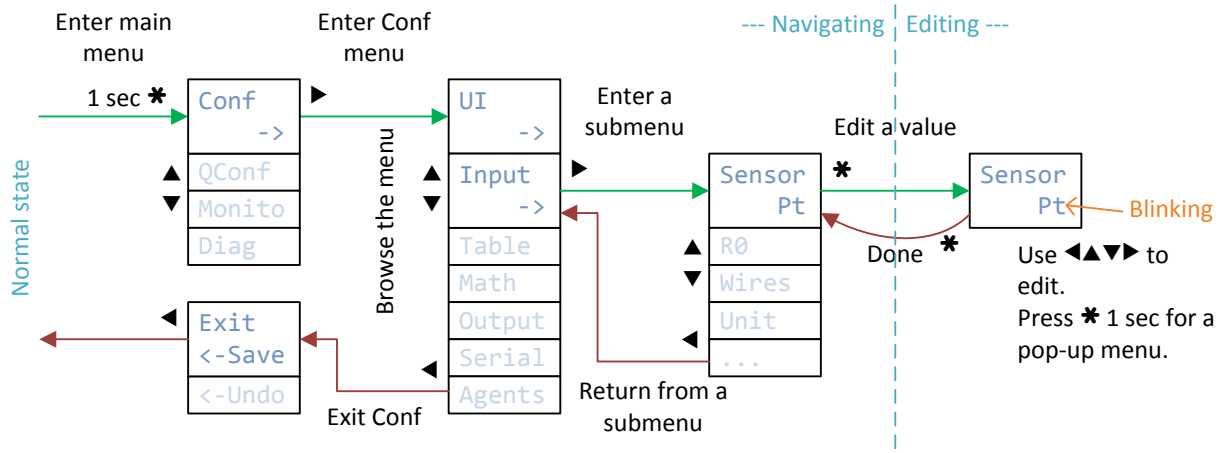


Figure 7

The configuration menu is organized hierarchically for clarity. The first level contains submenus for each functional block of the transmitter: The user interface, input, table, ELo program, analog output, serial bus, and agents.

Browse the submenus using the up/down movement ▲▼ of the joystick. Enter the desired menu by pressing the joystick inwards ✱ or to the right ▶. In the submenu, again browse the contents with ▲▼. The upper row indicates the name of the submenu or the setting, while the lower row shows the current setting value, or "->" when the item is a submenu.

To start editing a setting value, press the joystick inwards ✱ or to the right ▶. (A text "Agent" will be shown briefly if the setting is controlled by an agent.) The whole value or one digit of it starts blinking. Different types of values are edited in different ways, as detailed below. To stop editing, press the joystick inwards ✱.

Press ◀ to exit to the previous menu level.

Editing floating point values

Floating point values have a decimal point. The input and output scalings use floating point numbers, as an example. One digit is blinking when editing. The blinking digit can be changed – or in other words the cursor moved – with ◀▶. The digit can be increased with ▲ and decreased with ▼. Actually the whole number is increased and decreased instead of a single digit, e.g. if the first number to the left from the decimal point is blinking, the number is increased by one every time ▲ is pressed. The number will go 1, 2, 3 … 9, 10, 11, etc.

To change the sign of the value, there are two ways:

1. Press the joystick inwards one second and select Negate from the pop-up menu.
2. If all digits on the right side of the cursor are zeros, the value can be decremented below zero or incremented above zero. E.g. initial value 2.00, pressing ▼ several times will yield 1.00, 0.00, -1.00, -2.00.

To reset the value (to 0): Press the joystick inwards one second and select Zero from the pop-up menu.

In some cases, it is possible to "push" the cursor against the right edge of the display making the display to scroll and reveal more decimals. Since some digits will temporarily roll out of the display on the left, a symbol "_" will appear.

Editing integer values

The method to edit integers (e.g. the Dec and serial address settings) is quite similar to editing floating point numbers. When the cursor (blinking) is at the last digit, the value can be increased and decreased by 1 with joystick up/down movements. When the cursor is at the second last digit, the value will be increased or decreased by 10, etc.

To quickly reset the value to zero, press the joystick inwards one second and select Zero from the pop-up menu.

Editing options

Options, e.g. the sensor selection, are edited simply with joystick up/down movements.

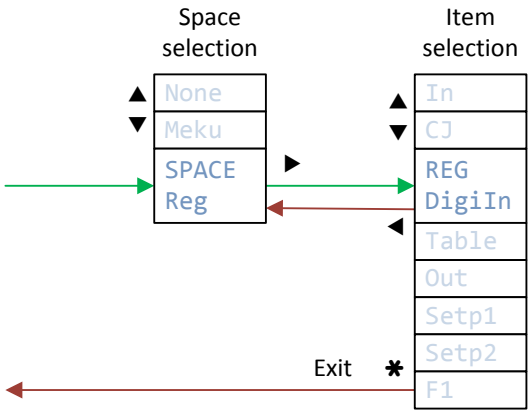
Register and Meku references

The various Src settings are used to select a register as a source for a certain function like the analog output. The agents also use Meku references, i.e. a reference to a setting in the configuration menu. In most cases it is also possible to set the reference to None.

The editor has two states for editing the references: the space selection state and the actual reference selection state. The space selection is used to switch between None, Meku, and Register spaces. In most cases it is not possible to choose between all the three spaces.

When the space has been selected, press ▶ to enter the actual reference selection state.

A register reference is selected using ▲▼ on the one-dimensional register list. Finally exit by pressing the joystick inwards ✱. Editing a Meku reference requires also ◀▶ movements due to the hierarchical nature of the configuration menu tree.



Entering a password

To set a configuration password, browse to UI/Password/Conf item and press ✱. The cursor starts to blink. Now press the joystick six times to any direction ◀▲▼▶ but not ✱. To disable the password, press ✱ twice.

Editing text

It is recommended to edit the text settings using a computer and Mekuwin software.

The blinking cursor can be moved with ◀▶. The text will scroll when trying to advance the cursor outside the display. The character at the cursor can be changed with ▲▼.

The upper row has two numbers. The leftmost indicates the current cursor location in the text, 1 meaning the first character. The other indicates the ISO 8859-1 code of the character in hexadecimal format. A dot appears after the code when the character is lowercase.

By pressing one second ✱ a pop-up menu appears. *N* will change the current character to N, which is in the middle of the alphabets. *Linefeed* adds a line feed (new line) character. *End* terminates the text at the cursor location, discarding all the characters to the right of the cursor.

Exiting

When exiting the first menu level, the transmitter will provide a menu containing options Save and Undo. Select Save and press ◀ to keep all the configuration changes done and to save them to the EEPROM, or alternatively select Undo and press ◀ to discard all the changes made in this configuration session. The user interface returns to the normal state.

Monitoring and simulating

The monitor state is used to view the values of the registers. It is a great aid in troubleshooting.

The monitor state is entered by pressing the joystick inwards one second in the normal state and by selecting Monitor in the menu. The monitor is a one-dimensional list of the registers. It can be browsed with ▲▼. The upper row of the display tells the item name. The value is visible on the lower row.

The In register can be "locked" to a simulation state in order to test the operation of the other functions without a need for different physical input values. Browse to the In item in the monitor menu. Press one second ✱ and select Lock from the pop-up menu. The register is now locked and the value can be edited. The other functions configured to follow the In register act like the simulated value was a real input reading. To return to the normal operation, do the same, but select Free in the pop-up menu. The simulation state is also exit at power-off.

To exit the monitor state, press ◀.

Self-diagnostics

If the transmitter detects a problem in itself or in the external connections, it will activate an item in the diagnostics table. The Pwr/Alm LED will blink at about 4 Hz. When the LED blinks this way, head to the diagnostics menu: In the normal state, press ✱ one second. Select Diag and enter with ✱. There may be more than one message – use ▲▼ to check. Finally exit with ◀.

The diagnostics messages are explained on page 7.

LEDs

The user interface has two red LEDs. The Pwr/Alm led will normally shine steadily, but if the transmitter detects an internal or external fault, the LED will blink, as explained in the section Self-diagnostics, page 7.

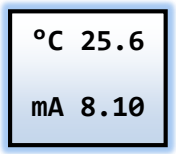
The Serial LED is active when this transmitter has received a valid message from the RS-485 bus. The Serial LED blinks very fast in the bootloader mode (that allows upgrading the firmware on the field).

Normal state and the screens

The transmitter is in the normal state after power-up. The contents of the display depend solely on the configuration settings made in the UI/Screens submenu of the configuration menu. Up to four sets of display contents can be configured. These sets are called screens.

Default screen

By factory default, and always when changing the Sensor and OutRng settings in the Quick configuration menu, the first screen is configured to have the input reading on the upper row and the output reading on the lower row, associated with a measurement unit in certain cases.



Switching among the screens

When more than one screen is configured, the user can switch the screen displayed by using the up and down movement of the joystick. Moving down will advance to a screen with a larger number.

It is also possible to configure the transmitter to intermittently change the screen displayed using the ***UI/Screens/Scan*** setting. The possible values are:

Manual	The screen displayed is not automatically changed. The screen can be changed with up/down movement of the joystick. After power-up, screen 1 is displayed.
Spring	The user can select the screen displayed with the joystick, but when not touching the joystick for five seconds, the screen 1 is displayed. This is intended for cases where the screen 1 contains the items to be viewed normally, but the user can in some situations need to access the other screens.
1 s	The transmitter will go through all the screens with one second interval. It is possible to temporarily to stop in a certain screen by pressing the joystick shortly up or down.
2 s	Like 1 s, but slower.

Defining a screen

The number of screens used is defined with the ***Screens/Count*** setting. Each screen has a submenu of its own. The first screen is configured at submenu ***Screens/1***.

Each screen submenu contains further two submenus, ***Upper*** for configuring the upper row of the display, and ***Lower*** for the lower row.

Each row submenu contains three settings: Src, Dec, and Text. ***Src*** is used to specify which register value should be shown on this row. Most typical register to show on the upper row of the first screen is the In register – the latest input reading. If no register value is to be shown, select None in the Src setting.

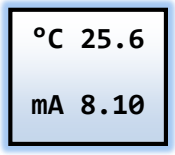
The number of digits to show after the decimal point is adjusted with the ***Dec*** setting. If requesting more decimals than can fit in the screen, the decimals are automatically dropped whenever possible.

With a negative Dec value, the corresponding amount of last digits is rounded to zero, e.g. if Dec is -2, the reading is rounded to nearest 100.

A fixed text can be displayed on the row by entering it in the ***Text*** setting. If the Src setting is not None, the text is displayed before the register value, which is convenient for adding a tag or a measurement unit. If the Src setting is None, the text can occupy the whole row. Trailing spaces can be added to the text for adjusting the appearance. In addition to the standard ASCII characters, the following characters may be used: Å Ä Ö Ø Ü ° µ. It is however not possible to show all the characters properly on the 11-segment display.

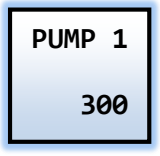
Most typical usage is to define the screen 1 upper row to display the input reading (In register) and the lower row the output reading (Out register) possibly associated with a unit text:

UI/Screens/1/Upper/Src	In
UI/Screens/1/Upper/Dec	1
UI/Screens/1/Upper/Text	°C
UI/Screens/1/Lower/Src	Out
UI/Screens/1/Lower/Dec	2
UI/Screens/1/Lower/Text	mA



An alternative use is to define only the input reading on the lower row and a fixed “tag” text on the upper row. The other screens can then be configured to show the analog output and other significant values:

UI/Screens/1/Upper/Src	None
UI/Screens/1/Upper/Dec	N/A
UI/Screens/1/Upper/Text	Pump 1
UI/Screens/1/Lower/Src	In
UI/Screens/1/Lower/Dec	0
UI/Screens/1/Lower/Text	



Setpoints

A setpoint refers to a register the value of which can be easily adjusted by the operator in the normal state. It is not associated with alarm levels. The setpoint value can be used in various ways:

- To change the operation of the ELo program. The ELo program can read a setpoint register and use it as any other register.
- To manually control the analog output. If the Src setting of the analog output is set to Setp1 or Setp2, the analog output is defined by the setpoint value.
- To quickly access some numerical configuration setting without entering the configuration state. See Setpoint controlled setting on page 34.

The setpoint value is a floating point number. The minimum and maximum allowed value for the first setpoint can be defined in the UI/Setpoints/1/Min and Max settings.

The setpoint value is stored in an EEPROM memory and is thus retained over power-off situations.

To show a setpoint in a certain screen, just set the other of the Src settings to Setp1 or Setp2. When the screen containing a setpoint register is viewed in the Normal state, the user can press the joystick: The setpoint value starts to blink and it can be increased and decreased with joystick up/down movements. An acceleration algorithm has been implemented. To finish adjusting the setpoint, press the joystick again.

It is not allowed to have a setpoint on both rows of one screen. The transmitter can't know which one of them should be adjusted when the joystick is pressed.

An example of configuring the screen 2 to have a setpoint:

UI/Screens/2/Upper/Src	None
UI/Screens/2/Upper/Dec	N/A
UI/Screens/2/Upper/Text	Level
UI/Screens/2/Lower/Src	Setp1
UI/Screens/2/Lower/Dec	1
UI/Screens/2/Lower/Text	m
UI/Setpoints/1/Min	0
UI/Setpoints/1/Max	100

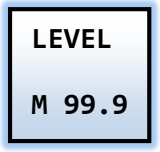


TABLE FUNCTION

The table function is most typically used to linearize a non-linear input signal or to remove the input signal errors in multiple points. It can be used for other purposes too, e.g. to generate a non-linear output signal or in co-operation with the ELo program.

The table function settings are collected in the Table submenu of the configuration menu.

The linearizer is configured to follow one register with the ***Src*** setting. Any register can be selected. To linearize the input, select the In register.

The table consists of 2 to 10 points that map the input value to an output value. The number of points is defined with the ***Pts*** setting.

Linear interpolation is used between the points. Outside the points, linear extrapolation is used obtained from the two nearest points. Each point has an ***X*** value and a ***Y*** value. X is the input, and Y is the corresponding output. The operation is equivalent to the Pts/Mea/Sca scaling of the input, but allows more than two points.

When using more than two points, the X values of the points must be in ascending order: X1 is the smallest X value. This limitation does not apply to the Y values.

The table output is available in the Table register. The display can be configured to show the table output by changing the appropriate Src setting value to Table in the UI submenu of the configuration menu. Likewise, the analog output can be configured to follow the table output by changing its Src setting to Table.

A five-point example set-up is in Figure 8. When the input value is 90, the output will be 54.

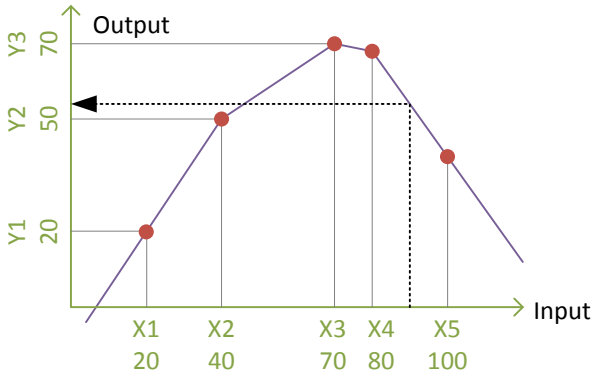


Figure 8

Teaching

In addition to manually entering the X and Y values, the X values can be taught. It means that the current input value is copied to one of the X values. Then the corresponding Y value can be set, and the table function output will be the Y value entered always with this input.

An example: A three point teaching of a potentiometer. Set Pts to value of 3. Turn the potentiometer to the first position (that gives the smallest ohms). Give a Lock command for the X1 value, and the device will copy the current input value to it. Enter the engineering value (e.g. potentiometer angle) in Y1. Move the potentiometer to the middle teaching position, and repeat the operations using X2 and Y2. Repeat once more for the third teaching position.

The ***Lock command*** is given in Mekuwin by clicking the L button next to the value. On the integral display, browse to the item (X1-X10) but do not start editing – nothing should be blinking. Now press the joystick one second, and a pop-up menu appears. Select Lock.

ELO PROGRAM

ELo is a simple scripting language developed by Nokeval. Small programs can be used to extend the capabilities offered by the other functions. ELo can do basic calculation, conditional execution and timing. The program can handle both floating point values and binary values. The ELo program is interpreted, not compiled.

The program can be entered with Mekuwin software. It is entered in the ***Program*** item in the Math submenu of the configuration menu. The maximum length is 320 characters, each line feed counted as one character.

Registers

ELo program can read all the registers (see Registers on page 7), but typically writes the F1 to F12 registers only. It is possible to write some other registers too (e.g. a setpoint register). The F registers are initialized to 0 at power-up.

There are two ways to reference a register. The most readable method is to use the name of the register directly, e.g. In or F1. The case of the register name is important: “in” or “f1” do not do the job. Since the program is interpreted, this method limits the execution speed of the program, which in most cases is not a matter.

The more effective way is to write @ and the number of the register, e.g. @1 is the In register and @8 is F1.

A special register Intv or @0 contains the time interval between the beginning of the previous and the current execution of the ELo program in seconds. It can be used to count time. Writing F1+=Intv will make F1 to count seconds (with a resolution of dozens of milliseconds).

It is important to remember that a register owned by another module (e.g. the In register) may be updated during the ELo execution. Reading the register value may yield a different value than at the previous time. However, the F registers are double-buffered: When the ELo program writes to them, the values are stored in a temporary buffer, and when the program execution ends, these values are copied to the public F registers available to the other modules.

Not-a-number

Not-a-number or NaN is a special value for a floating point register to indicate an erroneous value. The ELo program can set any F register to NaN, and test any register against NaN.

The NaN value is used by the other modules of this transmitter too: the In register will contain NaN in case of sensor fault. If the analog output is configured to follow a register that has a NaN value, the output will indicate fault as defined by its Break setting.

Any calculation involving a NaN will yield NaN, e.g. NaN/2 will result in NaN.

Trigger

The ***Trigger*** setting defines when the program execution is launched. One register can be selected as the trigger. Whenever the value of that register is updated, the program will execute. If the program is mainly processing the input readings, a natural choice as a trigger is the In register.

If the trigger register is not updated for a long time, the program will be executed approximately 1 second after the previous execution finished.

It can't be guaranteed that the program execution is finished before the trigger is tripped again. In this case the program will re-execute almost immediately when the previous execution is finished.

If the Trigger setting is set to None, the program is re-executed approximately 200 milliseconds after the end of the previous execution.

Limitation

The program execution is limited to 200 operations (one line counted as one operation) in order to prevent total lock-up in a case of e.g. an eternal loop.

Program structure

The program consists of lines. Every line has one simple command. The command can change a register or cause a conditional or unconditional jump inside the program.

Math commands

Each row can contain only one command, i.e. it is not allowed to write dest=src1+src2+src3.

dest=src	Copies a value from src to dest. Src can be a register reference or a constant. E.g. F1=3.14 will place a value 3.14 in register F1.
dest=NaN	Puts a not-a-number value to dest.
dest=src1+src2	Sums src1 and src2 value and places the result in dest register. E.g. F1=In+10 will add register In contents to a value of 10 and place the result in register F1.
dest=src1-src2	Subtracts.
dest=src1*src2	Multiplies.
dest=src1/src2	Divides.
dest=src1src2**	Src1 raised to power of src2.
dest=src10.5**	Square root.
dest=src1&src2	Bitwise AND. If applied to a floating-point value, it is first converted to an 8-bit integer.
dest=src1|src2	Bitwise OR.
dest=src1^src2	Exclusive OR.
dest+=src	Sums src and dest and places the result in dest. Exactly same as writing dest=dest+src.
destX=src	The same for other operators, e.g. F1*=10.

Jumps and conditional jumps

?lines	Jumps given number of lines forward (+) or backward (-). E.g. ?-2 will execute the command that is two lines above this line. ?2 will skip the next line. ?0 creates an eternal loop. Jumping outside of the program is allowed: ?99 will exit the program.
x==y?lines	Jumps given number of lines if x equals y. E.g. F2==0?3
x!=y?lines	Jumps if x is not equal to y.
x<y?lines	Jumps if x is less than y.
x<=y?lines	Jumps if x is less or equal to y.
x>=y?lines	Jumps if x is greater or equal to y.
x>y?lines	Jumps if x is greater than y.
x==NaN?lines	Jumps if x is not-a-number
x!=NaN?lines	Jumps if x is not not-a-number

References

1.23	Decimal constant. Allowed characters plus, minus, point, digits 0…9.
F1	Register F1.
@1	Register 1.
@F1	Register defined by register F1 contents – indirect reference.

Errors

Whenever an error occurs in the execution, the Math error diagnostic message will be activated. The error number and error line can be checked in the configuration menu items Math/Error and Math/ErrLine correspondingly. The error numbers are:

0	No errors.
1	Too long operand or operator.
2	Unknown operator.
3	Execution exceeds 200 operations.
4	Failed to write a register.
5	Illegal register reference. Please remember that the references are case-sensitive.

Examples

Peak hold

Remembers the highest value of the In register and provides the result in F1 register. The digital input can be used to reset the hold.

```
F2=In          Take a working copy in F2 because the In register can change anytime
F1>=F2?2       If current hold (F1) is greater than current input (F2), skip the next command
F1=F2          Copy the current input to current hold – we have a new peak
DigiIn==0?2    If digital input is not active, skip the next line
F1=F2          Digital input is active – reset the hold by copying the current input to it
```

Totalizer

Increases one register at the rate defined by the current input. This example does not have a reset.

```
F2=In*Intv     Current input multiplied by the time interval from the previous execution
F1+=F2         Increase the totalizer register
```

Slow scanning display

The standard options in the user interface settings allow “scanning” the display, i.e. automatically changing the current screen every 1 or 2 seconds. This example makes a 4 second scanning.

```
F1+=Intv       Increase F1 by the time interval from previous execution
F1<=4?6        Still less than 4 seconds – skip all the next rows
F1=0           Reset the timing
Screen==4?3    If in the last screen, jump to the code that will handle that situation
Screen+=1      Advance to the next screen
?2             Jump over the next line
Screen=1       Go to the first screen
```

Polynomial

Calculates y = 30 x^3 – 20 x^2 + 10 x – 5. Input: In register copied to F2. Output: F1.

```
F2=In          Take a working copy, because In register can change anytime
F1=30          Highest coefficient
F1*=F2         Multiplied by x
F1+=-20        Second coefficient
F1*=F2         Etc
F1+=10
F1*=F2
F1+=-5
```

SERIAL BUS RS-485

The transmitter can be connected to an RS-485 bus. The bus allows several slaves and one master. This is a slave device.

Connecting

The RS-485 bus is connected to connector B. The bus consists of a twisted full-duplex (bidirectional) data pair and a common wire. The cable should be shielded, the shield earthed at one point. The nominal impedance should be approx. 100-120 Ω.

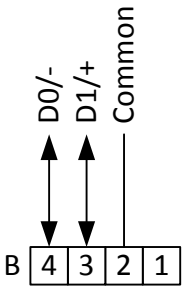


Figure 9

The polarity of the data pair is important. Modbus specifications call the positive idling line D1, but it is also commonly known on names +, B, and A. Correspondingly the negative idling line is -, A, or B. The wires are never crossed.

If the bus is long (say more than 100 m), it is recommended to terminate the first and last device on the bus. Some devices provide internal means for it, but this transmitter does not. An external 100…120 ohm resistor can be connected between the data wires.

The bus needs one device that gives a small voltage (0.2 V or more) between the data wires when no device is transmitting on the bus. This is called polarizing, biasing, or fail-safing. The master device is usually the natural choice for this. Using a multimeter to measure the equal value and polarity of the idle voltage at each bus device is a good way to check the integrity of the wiring.

Settings

The serial bus settings are collected in the Serial submenu in the configuration menu.

This device supports two protocols: Nokeval SCL and Modbus RTU. The selection is made with the ***Protocol*** setting. More about the protocols in chapters SCL protocol, page 28, and Modbus protocol, page 29.

Since several devices can be attached to a bus, each device needs to be configured to a different ***Address***. Legal addresses with Nokeval SCL are 0…123. Modbus allows addresses 1…247.

This device supports common ***baud*** rates 300…115200 bits/s.

Nokeval SCL uses always 8N1 ***parity***. Modbus RTU allows using 8E1 (recommended), 8O1, 8N2 and 8N1.

With Modbus, the registers can be read as 16-bit integers or 32-bit floating point numbers. This transmitter performs all internal calculations with 32-bit floating point numbers. When they are converted to a 16-bit integer, they can be scaled with the ***Dec*** setting to better utilize the limited resolution of the integer. If Dec=0, the value is just rounded to an integer. If Dec=1, the value is multiplied by 10 before conversion. With Dec=2, the value is multiplied by 100, etc. If measuring a temperature that never exceeds 327 °C, it is advisable to set Dec=2 to get maximum resolution, because the signed 16-bit integer can have a value of -32767 to +32767. Integer value of -32768 (8000 in hexadecimal) is used to indicate invalid value.

The Conf setting defines if the configuration settings are allowed to be changed by the Modbus Write Holding register command. It also controls resetting the device with the Nopsa command.

SCL PROTOCOL

A full specification of the Nokeval SCL protocol can be downloaded from Nokeval WWW site. Shortly, the command frame consists of address byte (bus address+128), human-readable command, ETX character (ASCII 3), and XOR checksum of all bytes excluding the address byte. A normal response consists of ACK (ASCII 6), human-readable response, ETX, and XOR checksum of all the bytes including ACK. An error response is similar but ACK is replaced by NAK (ASCII 21). The parity is 8N1.

In this device, SCL protocol allows querying the register values, including the latest input reading, configuring with the Mekuwin program, and many operations with the Nopsa language, see page 33.

Command set

TYPE ?

Returns the model name and the firmware version of this device space-separated: RMC685 V1.2. The RTC685 will return RMC685 as its model name.

SN ?

Returns the serial number, e.g. A123456.

MEA CH <ch> ?

Returns the value of register ch. E.g. MEA CH 1 ?. The registers are numbered as in table on page 7. Every register can be fetched, including the F and the Keys registers.

The response may contain digits 0-9, minus sign, and a decimal point. The scientific representation 1.00E-3 is not used. In case of a fault (NaN), the response consists of dashes only -----.

MEA SCAN <first> <last>

Returns the values of registers first to last separated with a space. E.g. MEA SCAN 1 3 will return the values of registers 1, 2, and 3, i.e. In, CJ, and DigiIn. An example response: 25.6667 29.1240 0.

If requesting more registers than the serial buffer can hold, less registers are returned.

DI CH 1 ?

Returns the state of the digital input: 0 or 1 (pulled low).

OUT CH <ch> <value>

Writes a value to one of the Ser registers. OUT CH 1 100 will write a value of 100 to Ser1. An analog output can be configured to follow a Ser register, or the value may be used in the ELo program. The Ser registers are reset at power-up.

OUT SCAN <first> <last> <value1> <value2>...

Writes a value to several Ser registers. E.g. OUT SCAN 1 2 100 200.

N <hexadecimal data>

Encapsulating a Nopsa command in SCL. The Nopsa command is converted to hexadecimal characters without spaces. E.g. querying the serial number: N 0102. The device responds with hexadecimal characters carrying a Nopsa response. See page 33.

MN <hexadecimal data>

A legacy command for encapsulating Meku configuration commands in SCL protocol. Using Nopsa is recommended.

MODBUS PROTOCOL

The maximum frame length is 100 bytes.

Supported commands

3 Read Holding Registers

Allows reading the configuration settings. Also allows reading the Input registers. The register usage is specified in a table below.

4 Read Input Registers

Allows reading the input reading and all other variable value registers.

6 Write Single Register

Changing the configuration settings and the Ser registers. When changing the configuration settings, the settings are automatically saved to EEPROM. The Ser registers are never saved in EEPROM, but they are zeroed at power-up.

If changing the Serial settings, the changes will not affect until the transmitter is powered down. This is to prevent breaking the connection while making the changes.

It is not possible to change the configuration if the Conf setting is disabled in the Serial settings.

16 Write Multiple registers

Like Write Single Register but allows writing several registers at once.

17 Report Slave ID

Returns information about this device. The run indicator is always 0xFF. The free bytes contain a space-separated string consisting of device type, firmware version, and serial number, e.g. RMC685 V1.2 A012345. The model RTC685 will return RMC685 as its model.

43 14 Read Device Identification

Returns information of this device according to Modbus specifications. Supports only the Basic Device Information and the stream access.

109 Meku

A legacy command for Meku configuration language used by the Mekuwin software.

110 Nopsa

Encapsulating Nopsa commands in Modbus. The PDU consists of Modbus function code 110, a length byte, and the Nopsa command. The length byte indicates the Nopsa command length. The response format is the same. The supported Nopsa commands are listed on page 33.

Data types

Name	Size (regs)	Description
BOOL	1	Off/on value. 0=False, 1=True.
BYTE	1	One byte value. Only the lower (rightmost) byte of the Modbus register is used.
WORD	1	16-bit value.
ENUM	1	Option list setting. The options are listed in tables.
CODE	1	Password 12 bits. 0=not used.
FLOAT	2	32-bit floating point number IEEE 754. Least significant word first (LSWF, little-

		endian).
STRINGZ	Any	String. If the string occupies less space than reserved, it will be terminated by a zero byte.
REF	2	Reference to a register or to Meku. The first word defines the space: 0x0000=None, 0xFF02=Meku, 0xFF03=Register. In a Meku reference, the second word defines the Meku item number. In a Register reference, the second word defines the register (see page 7), indexed from 0.

Within one Modbus register, the data is represented the most significant byte first (MSBF) as required by the Modbus standard.

Holding registers

The table shows register addresses (address transferred in the PDU + 1). To get a Modicon style address, add 40000 to the table address.

Address	Name	Type	Values
1…2	Ser1	FLOAT	
3…4	Ser2	FLOAT	
1001	Ser1	WORD	
1002	Ser2	WORD	
2001	Conf\UI\Screens\Count	BYTE	Unsigned 1…4
2002…	Conf\UI\Screens\Scan	ENUM	See table E1
2003…2004	Conf\UI\Screens\1\Upper\Src	REF	
2005	Conf\UI\Screens\1\Upper\Dec	BYTE	Signed -4…5
2006…2009	Conf\UI\Screens\1\Upper\Text	STRINGZ	Len=8
2010…2011	Conf\UI\Screens\1\Lower\Src	REF	
2012	Conf\UI\Screens\1\Lower\Dec	BYTE	Signed -4…5
2013…2016	Conf\UI\Screens\1\Lower\Text	STRINGZ	Len=8
2017…2018	Conf\UI\Screens\2\Upper\Src	REF	
2055…2058	Conf\UI\Screens\4\Lower\Text	STRINGZ	Len=8
2059…2060	Conf\UI\Setpoints\1\Min	FLOAT	Signed
2061…2062	Conf\UI\Setpoints\1\Max	FLOAT	Signed
2063…2064	Conf\UI\Setpoints\2\Min	FLOAT	Signed
2065…2066	Conf\UI\Setpoints\2\Max	FLOAT	Signed
2067	Conf\UI\Passwords\Conf	CODE	
2068	Conf\UI\Passwords\Oper	CODE	
2069	Conf\Input\Sensor	ENUM	See table E2
2070…2071	Conf\Input\R0	FLOAT	Unsigned
2072	Conf\Input\Wires	ENUM	See table E3
2073	Conf\Input\Unit	ENUM	See table E4
2074	Conf\Input\Supply	BOOL	
2075	Conf\Input\Pullup	BOOL	
2076	Conf\Input\Speed	ENUM	See table E5
2077	Conf\Input\MovAvg	BYTE	Unsigned 1…20
2078…2079	Conf\Input\Lopass	FLOAT	Unsigned
2080	Conf\Input\Pts	BYTE	Unsigned 0…2
2081…2082	Conf\Input\Mea1	FLOAT	Signed
2083…2084	Conf\Input\Sca1	FLOAT	Signed
2085…2086	Conf\Input\Mea2	FLOAT	Signed
2087…2088	Conf\Input\Sca2	FLOAT	Signed
2089…2090	Conf\Input\Lo	FLOAT	Signed
2091…2092	Conf\Input\Hi	FLOAT	Signed
2093…2094	Conf\Table\Src	REF	
2095	Conf\Table\Pts	BYTE	Unsigned 0…10
2096…2097	Conf\Table\X1	FLOAT	Signed

2098…2099	Conf\Table\Y1	FLOAT	Signed
2100…2101	Conf\Table\X2	FLOAT	Signed
2134…2135	Conf\Table\Y10	FLOAT	Signed
2136…2295	Conf\Math\Program	STRINGZ	Len=320
2296…2297	Conf\Math\Trigger	REF	
2298	Conf\Math\Error	BYTE	Unsigned 0…255
2299	Conf\Math\ErrLine	BYTE	Unsigned 0…255
2300…2301	Conf\Output\Src	REF	
2302	Conf\Output\Range	ENUM	See table E6
2303…2304	Conf\Output\Lo	FLOAT	Signed
2305…2306	Conf\Output\Hi	FLOAT	Signed
2307…2308	Conf\Output\Rdg1	FLOAT	Signed
2309…2310	Conf\Output\Out1	FLOAT	Signed
2311…2312	Conf\Output\Rdg2	FLOAT	Signed
2313…2314	Conf\Output\Out2	FLOAT	Signed
2315	Conf\Output\Limit	BOOL	
2316	Conf\Output\Break	ENUM	See table E7
2317	Conf\Serial\Protocol	ENUM	See table E8
2318	Conf\Serial\Address	BYTE	Unsigned 0…255
2319	Conf\Serial\Baud	ENUM	See table E9
2320	Conf\Serial\Parity	ENUM	See table E10
2321	Conf\Serial\Dec	BYTE	Signed 0…3
2322	Conf\Serial\Conf	BOOL	
2323…2324	Conf\Agents\1\Src	REF	
2325…2326	Conf\Agents\1\Dest	REF	
2327…2328	Conf\Agents\2\Src	REF	
2329…2330	Conf\Agents\2\Dest	REF	
5001…5043	Input registers 1…43 shadowed		
6001…6023	Input registers 1001…1023 shadowed	WORD	

Input registers

The table shows register addresses (address transferred in the PDU + 1). To get a Modicon style address, add 30000 to the table address.

Address	Name	Type	Values	Integer address
1…2	In	FLOAT	Signed	1001
3…4	CJ	FLOAT	Signed	1002
5	DigiIn	BOOL	0…1	1003
6…7	Table	FLOAT	Signed	1004
8…9	Out	FLOAT	Signed	1005
10…11	Setp1	FLOAT	Signed	1006
12…13	Setp2	FLOAT	Signed	1007
14…15	F1	FLOAT	Signed	1008
16…17	F2	FLOAT	Signed	1009
36…37	F12	FLOAT	Signed	1019
38…39	Ser1	FLOAT	Signed	1020
40…41	Ser2	FLOAT	Signed	1021
42	Screen	BYTE	Unsigned 1…4	1022
43	Keys	BYTE	Bits 0…31	1023

Registers 1001…1023 contain the registers as 16-bit integers, multiplied by a coefficient before conversion. The coefficient is 10^{Dec}, where Dec is the setting Serial/Dec.

Enum values

Table E1

Value	Scan
0	Manual
1	Spring
2	1s
3	2s

Table E2

Value	Sensor
0	Off
1	9mV
2	70mV
3	290mV
4	1100mV
5	±1100mV
6	11V
7	0-10V
8	0.18mA
9	1.5mA
10	24mA
11	0-20mA
12	4-20mA
13	75ohm
14	600ohm
15	3000ohm
16	10000ohm
17	Pt
18	Ni
19	Cu
20	KTY83
21	NTCLE3977
22	TcB
23	TcC
24	TcD
25	TcE
26	TcG
27	TcJ
28	TcK
29	TcL
30	TcN
31	TcR
32	TcS
33	TcT

Table E3

Value	Wires
0	2
1	3
2	4

Table E4

Value	Unit
0	°C
1	°F
2	K

Table E5

Value	Speed
0	Slow
1	Normal
2	Brisk
3	Fast
4	Super

Table E6

Value	Range
0	0-20mA
1	4-20mA
2	mA
3	0-10V
4	V

Table E7

Value	Break
0	Min
1	Lo
2	Hi
3	Max

Table E8

Value	Protocol
0	SCL
1	Modbus

Table E9

Value	Baud
0	300
1	600
2	1200
3	2400
4	4800
5	9600
6	19200
7	38400
8	57600
9	115200
10	230400

Table E10

Value	Parity
0	(7E1)
1	8N1
2	8E1
3	8O1
4	8N2

NOPSA LANGUAGE

Nopsa is a protocol-independent machine-to-machine language specified by Nokeval. The commands are described in detail in a separate Nopsa specification.

Command number	Name	Remarks
1/0	Type	
1/1	Version	
1/2	Serial number	
1/3	Description	
1/4	Serial buffer size	
1/16	Reset	Works only if Conf/Serial/Conf is set to Yes.
1/32	Meku	
1/33	Go to bootloader	Works only from the POL port. The bootloader supports a different set of Nopsa commands (8/1, 8/2, 8/4, 8/5), allowing firmware field update.
2/0	Get value	Reads the registers, indexed from 0.
2/1	Get info	
2/2	Put value	Writes the Ser registers (Index 0=Ser1)
2/3	Put info	
4/0	Ring buffer info	See Ring buffer.
4/1	Move to oldest	
4/2	Move to newest	
4/3	Read by index	
4/4	Read next	
4/5	Re-read	
7/32	Keyboard test	For production testing
7/33	Display test	For production testing. Mode: 0=upper row, 1=lower, 2=indicators. Data: ISO 8859-1 string; char 255 will show all the segments.

RING BUFFER

The ring buffer enables reading the input readings sample by sample instead of querying the latest value with arbitrary interval. This is a good way to read when targeting for a maximum update rate. The Conf/Input/Speed setting defines the rate at which new samples are put in the ring buffer. It is not possible to use the ring buffer to store other registers than In.

The ring buffer size is 40 entries.

Since this device does not have a real-time clock, the timestamp is given relative to the moment of processing the read command, expressed in milliseconds. The id field is always 0, referring to the first register, “In”. Data is always stored as a 32-bit floating point number.

AGENTS

The configuration menu contains dozens of settings that define the behavior of this device. They are usually changed very seldom, mainly when setting up a new device. On the other hand, the registers contain continuously varying values, e.g. the input reading, the ELo program variables, and the table function output.

An agent is a bridge between these. Once configured, it reads one register and copies its value to a certain configuration setting. This device has two agents. Each of them has two configuration settings. ***Src*** defines the register to read, and ***Dest*** is used to select the configuration setting to write to. To disable an agent, set either or both of the settings to None.

ELo controlled setting

The ELo program can't directly access a configuration setting. With an agent, the ELo program can alter a setting. An example: The user wants to enable or disable the lowpass filter with an external switch. The desired lowpass time constant is 2 seconds. Write an ELo program:

```
F1=DigiIn*2          Since DigiIn is either 0 or 1, the result is 0 or 2.
```

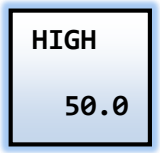
Then configure an agent to copy F1 to the lowpass filter setting:

```
Agents/1/Src         Reg/F1
Agents/1/Dest        Meku/Conf/Input/Lopass
```

Setpoint controlled setting

If a certain configuration setting is often needed to be changed, a quick access for it can be arranged using an agent and a setpoint. An example of quick access to the analog output high end scaling. Set up a setpoint in UI/Setpoints/1 menu under the configuration menu. Define suitable minimum and maximum values e.g. 50 to 500. Call the setpoint to a suitable screen by setting one of its Src settings to Setp1. An example:

```
UI/Screens/2/Upper/Src     None
UI/Screens/2/Upper/Text    High
UI/Screens/2/Lower/Src     Reg/Setp1
UI/Screens/2/Lower/Dec     1
UI/Screens/2/Lower/Text
```



Set up the agent to copy Reg/Setp1 to Meku/Conf/Output/Hi.

In the normal state, select screen 2 and press *. The lower row starts to blink, and the value can be adjusted. More about the setpoints on page 22.

Serial controlled setting

There are two registers that can be controlled by the serial bus: Ser1 and Ser2. Just like in the previous examples, an agent can be configured to copy a value from Ser1 or Ser2 to a configuration setting. The Ser registers will be reset to zero at power-off.

Limitations

The agent works most fluently on floating point numbers. It is also possible to write to an integer type configuration setting e.g. Screens/1/Upper/Dec or Input/MovAvg. Negative values may not work properly.

Further it is also possible to write to an options type setting like Serial/Baud, but the value must be an index to the option beginning from 0. E.g. to select 9600 baud, write a value of 5.

SPECIFICATIONS

Environmental

Operating temp	-20…+60 °C (the LCD will be readable at 0…50 °C)
Storage temp	-20…+60 °C
Humidity	10…90 %Rh non-condensing
Altitude	Max 2000 m
Protection	IP 20
Pollution degree	2
Dimensions	23×110×117 WHD
Weight	130 g

Power supply

Voltage	24 V DC ±15%
Typical current	25 mA without mA output nor transmitter supply
Max current	120 mA

Galvanic isolation

Description	These groups are isolated from each other: 1) Power supply. 2) Input. 3) Analog output, RS-485, digital input, and programming connector share a common ground.
Isolation voltage	50 V AC or 120 V DC continuous, 500 V transient

Input

General

Update rates	1.9, 7.8, 15.6, and 50 Hz. Specifications valid for 1.9 and 7.8 Hz rates.
Filters	1 to 20 samples moving average and 0 to 60 seconds 1st degree lowpass
Overvoltage protect	±18 V DC
Cable length	< 30 m
Power-up time	1 s to first reading
Warm-up time	10 min to full accuracy

Pt100 (α=0.00385)

Connection methods	2, 3, or 4 wires
Range	-200…+700 °C
Wire resistance	Max 10 Ω
Accuracy 3-wire	0.05% rdg + 0.2 °C at 25 °C env
Accuracy 4-wire	0.05% rdg + 0.1 °C at 25 °C env
Thermal drift	0.01 °C/°C ref 25 °C env
Noise (unfiltered)	3-wire: typ 0.02 °C rms, 4-wire: typ 0.01 °C rms
Sensor current	approx. 250 µA DC continuous

Pt1000

Range	-200…+700 °C
Accuracy	0.1% rdg + 0.2 °C

Ni100

Range	-60…+180 °C
Accuracy	0.05% rdg + 0.2 °C

Cu10

Range	-200…+260 °C
Accuracy	1 °C

KTY83

Range	-55…+175 °C

NTCLE3977 Vishay NTCLE β=3977

Range	-40…+150 °C

Ohm (resistance, potentiometer)

Ranges	0-75, 0-600, 0-3000, and 0-10000 Ω
Accuracy	0.05% rdg + 0.5 Ω

Thermocouples

Type	Range °C	Lin.error °C
B	400…1700	0.3
C	0…2300	0.5
D	0…2300	1
E	-100…900	0.2
G	1000…2300	2
J	-160…950	1
K	-150…1370	0.5
L	-150…900	0.5
N	0…1300	0.1

R	0…1700	0.5
S	0…1700	0.5
T	-200…400	1

Thermal drift	< 0.02 °C/°C ref 25 °C env
Accuracy	0.05% rdg + 0.6 °C + lin.error + thermal drift
Noise	typ 0.03 °C rms

Millivolts

Ranges	±9, ±70, -70…290, ±1100 mV
Accuracy	0.05% rdg + 0.01 mV at 25 °C
Offset thermal drift	< 200 nV/°C
Gain thermal drift	< 50 ppm/°C
Noise	9 mV range: typ 500 nV rms
Input impedance	> 10 MΩ | 250 nF

Volts

Range	-11…11 V
Accuracy	0.05% rdg + 0.001 V at 25 °C
Gain thermal drift	< 50 ppm/°C ref 25 °C
Noise	typ 200 µV rms
Input impedance	> 10 MΩ | 150 nF

Milliamperes

Ranges	±0.18, ±1.5, and ±24 mA
Accuracy	0.05% rdg + 0.004 mA at 25 °C
Gain thermal drift	< 50 ppm/°C ref 25 °C
Noise	typ 300 nA rms
Load	50…80 Ω
Transmitter supply	15 V ±10%, max 50 mA

Digital input

Input type	For NPN output or switch (internal pull-up 10 kΩ to 3.3 V)
Active state	-3 … 0.7 V
Passive state	2.6 … 5.5 V
Cable length	< 3 m

Table function

Number of points	2 to 10
Interpolation	Linear

Analog output

General

Response speed	0.5 s fully settled

mA signal

Standard ranges	0-20 mA, 4-20 mA
Custom ranges	Any within 0…22 mA
Accuracy	0.008 mA at 25 °C
Gain thermal drift	< 50 ppm/°C
Sensor break indication	0 mA, 3.5mA, or 22.5 mA
Max load	600 Ω

V signal

Standard range	0-10 V
Custom ranges	Any within 0…10 V
Accuracy	0.005 V at 25 °C
Gain thermal drift	< 50 ppm/°C
Sensor break indication	0 or 11 V
Max load	5 kΩ

RS-485 bus

Protocols	Nokeval SCL and Modbus RTU
Baud rates	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bits/s
Modbus parity	8E1, 8O1, 8N2, 8N1
Min response time	3.5 characters or 1.7 ms whichever greater
Typ response time	Querying one reading: the same as Min response time.
Max response time	15 ms when reading input register 1 or 1001. Much longer when operating with holding registers.

Regulations

EMC immunity	EN 61326
EMC emissions	EN 61326 class A