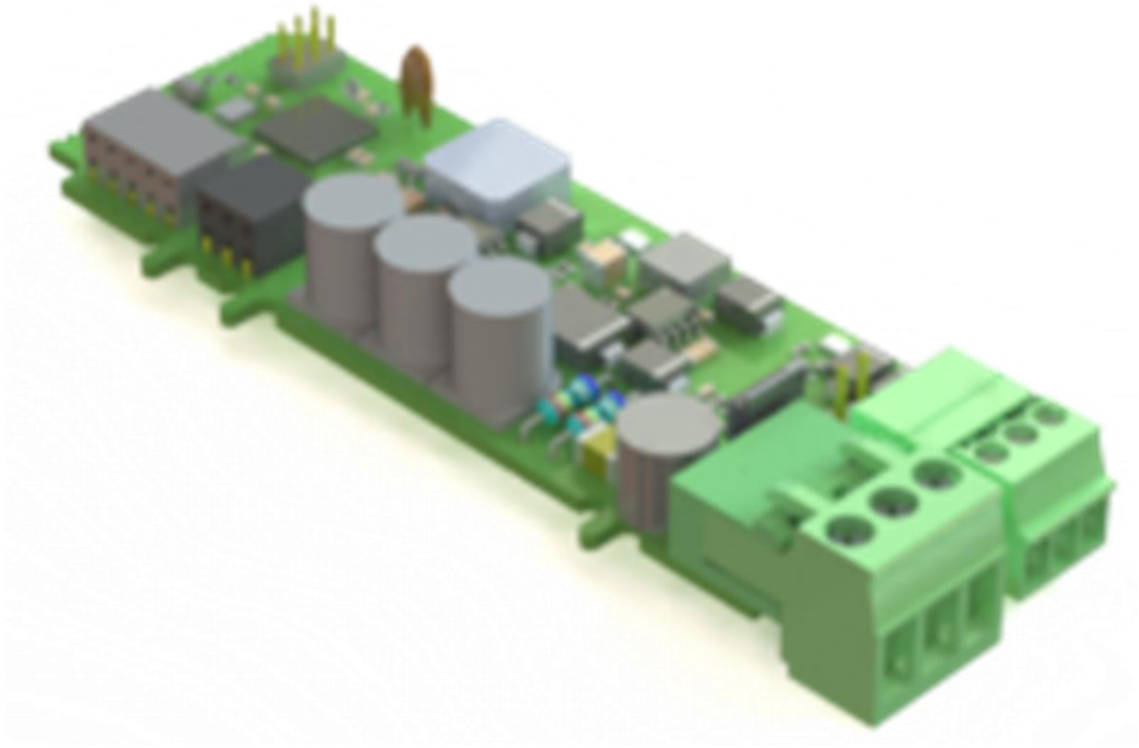


PM10POW24SA card

Manual

Document ID 6505 V9 15.4.2015

Table of contents

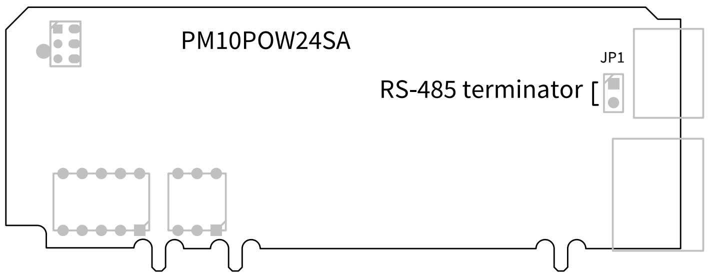
Introduction

PM10POW24SA is a 24 V input power supply card for a PM10 panel meter. It has a slave RS-485 port and one digital I/O line. PM10POW24A is the same card without the serial port and the I/O line.

This manual covers the PM10POW24SA card only. The rest of the device is explained in the device manual (e.g. PM10A Manual).

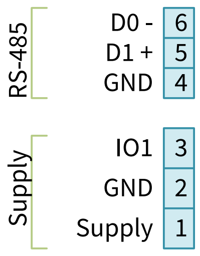
PM10REL2A uses the FreeRTOS real-time operating system V8.0.1. The FreeRTOS source code is available from Nokeval support on request.

Jumpers



The only jumper is for enabling the RS-485 terminator. It should be shorted when this card is the last device on a long bus (dozens of meters). Alternatively, an external terminator (e.g. 100-120 Ω resistor) may be used on the screw terminals.

Connecting

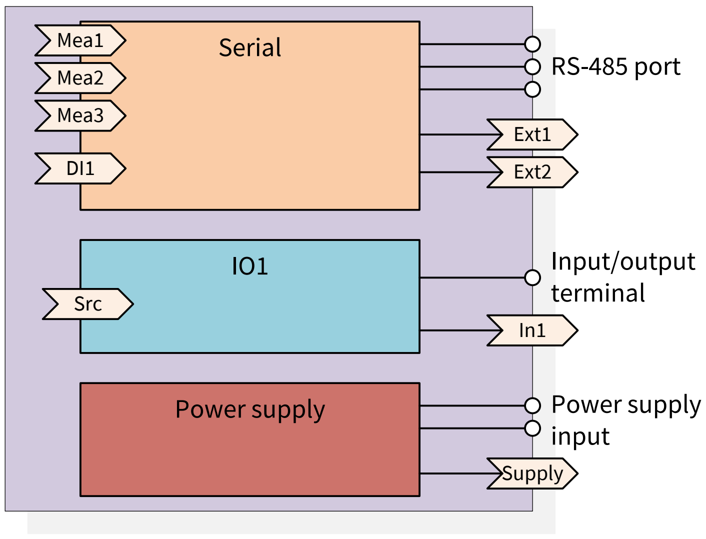


The 24 V ±15% supply is connected in terminals 1 and 2. The card has a 1 AT fuse. An external fuse is not required, but if one is used, it should have at least the same rating as the internal fuse.

The terminal 3 is an input/output, sharing the ground with the supply.

A three-wire RS-485 port is available at the terminals 4-6. Its ground is internally interconnected to the power supply ground.

The blocks and the registers



Serial

The RS-485 port can be used to read measurement and other readings out of this device, and to write values to the device. The supported protocols are Nokeval SCL slave, Modbus RTU slave, and a simple Ascii auto-transmission.

This block provides ten Ext registers, holding values that can be externally controlled.

IO1

This card has one input/output line (terminal 3). It can be configured to several modes:

- AI: Analog input – measures the input voltage coarsely 0 to 33 VDC. The reading is published in the In1 register in volts.
- DI: Digital input – acts as a PNP input. The state, either 0 or 1 is published in the In1 register.
- DO: Digital output – acts as a PNP output.

Power supply

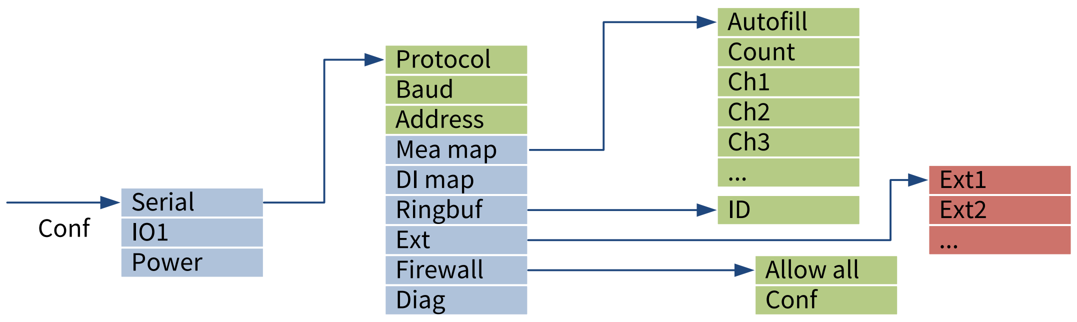
The power supply accepts a 24 VDC voltage and regulates internal 3.3 V and 15 V supply voltages for all the cards of the device. In addition, it measures and monitors the 24 V supply, which benefits:

- The measured voltage is available to all the cards in the Supply register, allowing e.g. to monitor a battery on a battery powered system.
- When the supply voltage is lost or drops below the minimum allowed voltage this card alerts all the other cards about that. The cards can then save their totalizers and other data to the EEPROM memory in order to retain it at a power-off.

Serial port with SCL protocol

SCL is a proprietary serial bus protocol developed by Nokeval. Its specification can be downloaded from the Nokeval web site. This card can act as a slave only.

Configuration menu overview



Setting port parameters

Navigate to the Serial submenu of the configuration menu of this card and set up the port settings:

Protocol	Select SCL.
Baud	Select any baud rate. Factory setting 115200.
Address	Select any address between 0…123. Factory setting 0. The device will always respond at address 126 too.

Querying the type and other info

This card supports the following information commands:

TYPE ?	The card returns “PM10POW24SA V0.3” without the quotes (the firmware version may vary).
SN ?	The card returns its serial number, e.g. “P012345”. The serial number of the whole device is not accessible.

Reading measurement readings out

The serial block sees all the registers of all the cards in the device. It can’t know which of the dozens of registers should be available for the serial bus, and how they should be fitted to the 1-dimensional channel numbering of the SCL protocol. The desired registers must be manually selected.

1. Navigate to the Serial menu and from there to the Mea map submenu.
2. Decide the number of channels you need and enter it in the Count setting.
3. For each Ch item, select one of the registers on this device. Almost any of the registers are available.

The following SCL commands are then available:

MEA CH 1 ?	Returns the value of the register selected in the Ch1 item in the menu. The other channels work the same way.
MEA SCAN 1 4	Returns the values of the registers configured in Ch1 to Ch4 items in the menu. The amount of channels to read at once is limited by the serial buffer length, which is 150 characters. The values are formatted as below, and separated by one space. E.g. 23.456 -22.888 45.000 99999.

The register values are formatted depending on their original data type (floating point is the most common, used by analog inputs etc):

Data type	Format	Example
Boolean (on/off)	0 or 1	0
Integer	Digits 0-9	1023
Floating point number	Minus, digits 0-9, decimal point.	-23.456
String (text)	As is; spaces are replaced with "_" and special characters with "?".	Test_1
Struct (special data)	Converted to hexadecimal	01FF00
Error (fault)	Five dashes (minus signs)	-----

Reading "digital" data out

The MEA command is intended for numerical (mainly floating point) values. For on/off, or boolean, or "digital" values, SCL provides the DI command. It can be used to read out the states of the front panel buttons, alarms, digital inputs, and many other on/off registers.

Select the wanted registers in the Serial / DI Map menu similarly to the MEA registers above.

The following SCL commands are then available:

DI CH 1 ?	Returns the value of the register selected in the DI1 item in the menu, either 0 or 1. The other channels work the same way. An error value will be expressed as 0.
DI SCAN 1 4	Returns the values of the registers configured in DI1 to DI4 items in the menu. The values are 0's and 1's separated with one space.

Writing numerical data to the device

The SCL protocol provides the OUT command for writing numerical (floating-point) data and DO command for writing digital data. This card supports them both. The values written are stored in Ext registers, available to all the cards. There are no configuration settings involved, but the values can be viewed in the Ext submenu. To write numerical values, use the OUT commands:

OUT CH n value	Writes a value to Ext register n. The value can contain a minus sign, digits 0-9, and a decimal point.	OUT CH 1 56.7
OUT SCAN first last value value…	Writes several values to Ext registers first…last. Separate the values by one (or more) space.	OUT SCAN 2 4 20 -30 44.444

If an Ext register is not written for 15 seconds, the value is considered outdated and is replaced by an error value. Consequently each Ext register should be written more often than every 15 seconds.

Writing “digital” data to the device

Use the DO command for writing digital data. The values written are stored in Ext registers, the same registers as the values written with the OUT command. If using both OUT and DO commands, coordinate the usage of the Ext registers.

DO CH n value	Writes a value to Ext register n. The value should be 0 or 1.	DO CH 5 1
DO SCAN first last value value…	Writes several values to Ext registers first…last. Separate the values by one space.	DO SCAN 5 8 1 0 1 0

Sending low-level commands to the device

It is possible to send Nokeval Nopsa commands to this card. Convert the command to hexadecimal and prefix with “N” and a space.

To access the other cards, use the Nopsa Route command (3/0). To allow this, check that the Serial menu and its Firewall submenu has Allow all setting set to Yes (doesn’t affect in firmware V0.3).

Examples:

- Read the type of this card: N 0100
- Read the type of A slot card: N 0300010100 (0300=route, 01=slot A, 0100=query type).

Using as a serial-controlled display

The Ext registers can hold string data. The SCL command DISP will set any text in the Ext1 register. Configure the display to show the Ext1 register, possibly without a tag. Then send a SCL command e.g.:

DISP Test!

The maximum length is 16 characters. A too long message will be truncated. If the message is longer than the display, the display may scroll (depending on the display type and its settings).

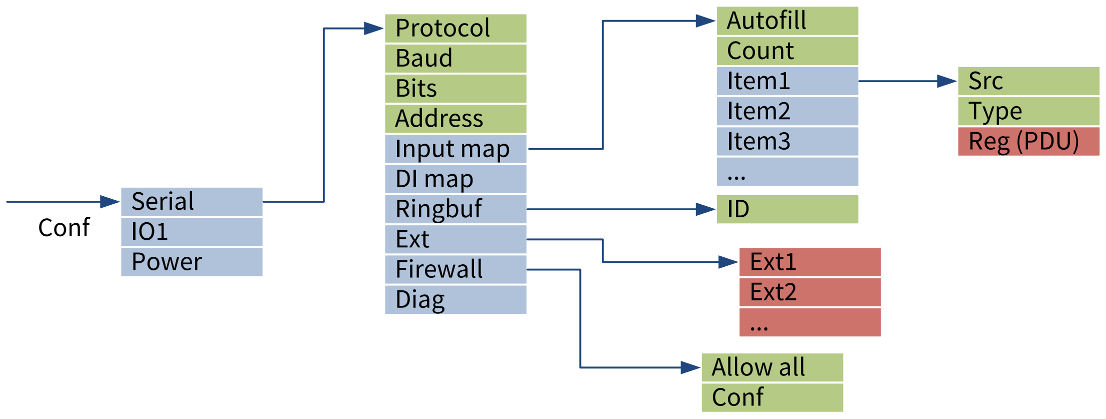
The message should be updated (re-sent) regularly (<15 sec), otherwise the display will show an error.

It is possible to read the states of the front panel keys, which are available as registers on the display card. Configure them in the DI1…DI5 settings and use DI SCAN 1 5 command. Alternatively configure the numerical Keys register to Mea map -> Ch1 submenu and read it with MEA CH 1 ?.

Serial port with Modbus RTU

This card is a Modbus RTU slave implementing Modbus.org MODBUS Application Protocol Specification V1.1b in applicable parts, and the Modbus over Serial Line V1.02 specification.

Configuration menu overview



Setting port parameters

Navigate to the Serial submenu of the configuration menu of this card and set up the port settings:

Protocol	Select Modbus.
Baud	Select any baud rate. Factory setting 115200.
Bits	Select the parity and number of stop bits. Modbus.org recommends 8E1.
Address	Select any address 1…247. Note that 0 will not work!

Querying the type and other info

This card supports the following information commands:

Report slave ID (17)	The card returns a response containing: • ID = 0 • Running = 0xFF • Additional data = e.g. “PM10POW24SA V0.3 P012345” without the quotes. V0.3 is the firmware version and P012345 is the serial number of this card.
Read device identification (43/14)	The card returns the Basic Device Identification: Manufacturer name (Nokeval), ProductCode (PM10POW24SA), and firmware version (e.g. V0.3).

Reading measurement readings out

The two uses of the word *register* must not be confused:

- Each PM10 card has a set of registers, providing measurement readings and other live values to the other cards.
- Modbus has Input registers and Holding registers.

The serial block sees all the registers of all the cards in the device. It can't know which of the dozens of registers should be available for the serial bus and in which format. The desired registers and the data types must be manually selected. As a benefit, all the desired data is tightly packed in consequent input registers and can be read in one transaction in most cases.

1. Navigate to the Serial menu and from there to the Input map submenu.
2. Decide the number of values (card registers) you need and enter it in the Count setting.
3. In each Item submenu, select one of the card registers on this device. E.g. the In1 register of an analog input card in slot A.
4. Further in each Item submenu, select the data representation in the Modbus registers using the Type setting. See below.

The card register selected in Input map item 1 will appear in the Modbus input register 0 in the data type configured. It may occupy several input registers, depending on the size of the selected data type.

The card register selected in item 2 will appear in the next free Modbus input registers, and so on.

An example:

- Item1: Src=Reg -> Slot A -> In1, Type=Float.
- Item2: Src=Reg -> Slot A -> In2, Type=Float.
- Item3: Src=Reg -> Master -> Page, Type=Uint0Dec.
- The first item, In1, will occupy Modbus input registers 0 and 1 as the Type is Float, which consumes two registers.
- The second item will occupy Modbus input registers 2 and 3.
- The third item will occupy register 4.

The types and their Modbus register consumptions are:

Type	Modbus reg count	Description
Sint3Dec	1	Signed 16-bit integer. The original value is multiplied by 1000 before converting to an integer, efficiently having “three decimals”. The original value can be -32.767 to 32.767. An error yields -32768 (0x8000).
Sint2Dec…Sint0Dec	1	The same as Sint3Dec but multiplied by 100, 10, or 1.
Sint-1Dec…Sint-2Dec	1	The original value is divided by 10 or 100 before converting to a 16-bit signed integer.
Uint3Dec…Uint-2Dec	1	Same as Sint types, but converted to an unsigned integer 0…65535. An error yields 65535.
Uint32bit	2	Unsigned 32 bit integer, e.g. a pulse count. Least significant word first.
Float	2	32-bit floating point number IEEE-754, least significant word first. An error yields Quiet NaN.
Raw1, Raw2	1	The original data as is, with no type conversion. Raw1 uses the lower byte only, Raw2 two bytes.
Raw4…12	2…6	The original data spanned over several Modbus registers, two bytes per register. Can be used for string and special data. The

		unused bytes are zeroed.

Use the standard Read Input Registers command (4) to read the data. The maximum number of data in one transaction is limited by the serial buffer size, which is 150 bytes.

The response time is heavily depending on the type conversions used. If the mapping is done using the same type as the original data has, there are no type conversions, resulting in a fast response.

The Input registers are shadowed in the Holding register space with an offset of 5000 for systems that can't access the Input registers.

Writing numerical data

The Ext registers of this card can be written with Modbus commands. The other cards can then read the Ext registers, e.g. for displaying the value. The writing can be done with Write Single Register command (6) or Write Multiple Registers command (16). They can be read back with command 3.

The Ext registers can be accessed in several data types:

Type	Ext1 Modbus reg	Ext2 Modbus reg	Ext3 Modbus reg…
Float (IEEE 754 LSBF)	0	2	4
Uint	1000	1001	1002

If an Ext register has not been updated for 15 seconds, its value will be replaced by an error value.

Sending low-level commands to the device

It is possible to send Nokeval Nopsa commands to this card. Use Modbus command 0x6E ("n"), place the length of the Nopsa command, and the Nopsa command itself.

To access the other cards, use the Nopsa Route command (3/0). To allow this, make sure the Serial -> Firewall -> Allow all setting is set to Yes (doesn't affect in firmware V0.3).

Examples:

- Read the type of this card (excluding the address and CRC bytes): 6E 02 01 00
- Read the type of A slot card: 6E 05 03 00 01 01 00 (6E=Nopsa, 05=length, 03 00=route, 01=slot A, 01 00=query type).

Serial port with Ascii transmission

In the Ascii mode, the card will automatically transmit selected values on the RS-485 bus every second. The card doesn't accept any commands in this mode. To use:

1. In the Serial menu, select Protocol = Ascii and some baud rate.
2. Under Mea map, select the desired registers to be output, similarly to "Reading measurement readings out" on page 5.
3. If you want "digital" outputs, configure DI Map.

The card will output messages consisting of the Mea map items, followed by the DI Map items. The format is same as with SCL MEA and DI commands – the values are human-readable and separated by a space. After the data comes CR+LF.

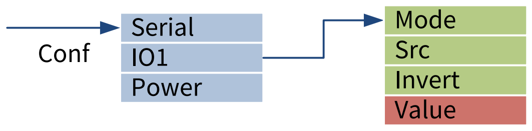
E.g.

```
12.3456 -33.2211 999999. 1 0 1 0<CR><LF>
12.3433 -33.1201 999999. 1 0 1 1<CR><LF>
…
```

Input/output line

This card has one terminal that can be an analog input, digital PNP input, or digital PNP output.

Configuration menu overview



Using as an analog input

In the configuration menu, set IO1 -> Mode = AI.

The terminal voltage is measured and published in the In1 register as a floating point number. The value is also seen in the Value item of the configuration menu.

Using as a digital input

In the configuration menu, under IO1, set Mode = DI. If the signal is active low, set Invert = Yes.

The state of the input is published in the In1 register as a boolean value. The value is also seen in the Value item of the configuration menu.

Electrically a high state is a voltage higher than 1.9 V, and a low state lower than 1.6 V. The card has a 36 kΩ pull-down resistance.

Using as a digital output

In the configuration menu, under IO1, set Mode = DO. To invert the operation, set Invert = Yes.

Configure the register, that should control the output in Src. When the selected register has a positive value, the output will go high (unless Invert = Yes). When the selected register indicates an error, the output behaves like with a value 0.

Electrically the output is pulled to the 24 V supply voltage (terminal 1 of this card). There is no active pull-down, only a 36 kΩ pull-down resistance.

(The digital input and its In1 register are working also in DO mode.)

Maintenance

This card doesn’t need maintenance.

Troubleshooting

No serial communications.

1. **Check if the serial bus LED D8 is blinking when the master sends a command.**
2. **Measure the idle voltage between the wires at every device of the bus. The voltage and its polarity should be approximately the same (otherwise a wiring fault), and D1 should be +0.2…5.5 V referenced to D0 (otherwise the bus master is not providing the biasing).**

Specifications

Environmental

Storage temperature	-40…+70 °C
Operating temperature	-30…+70 °C
Weight	28 g (P10POW24SA)

Power supply

Voltage range	20.4…27.6 V DC
Max power	15 W
Pre-fuse	1 AT or heavier, not required
Polarity protection	Yes, series diode
Voltage measurement	Resolution 50 mV, accuracy 200 mV @ 25 °C

Analog input

Measurement range	0…33 V typ.
Resolution	33 mV
Accuracy 25 °C	2% rdg + 200 mV
Impedance	36 kΩ
Update rate	Approx. 15 Hz
Cable length	Max 30 m indoors (not surge protected)

Digital PNP input

Low state voltage	-1…+1.6 V typ
High state voltage	1.9…33 V typ
Impedance	36 kΩ to ground
Update rate	Approx. 15 Hz

Digital PNP output

Passive state	36 kΩ to ground
Active state	Pulled to the supply voltage (24 V)
Max load	200 mA
Overcurrent protect	Internal 1 A fuse common with the other supply circuits

Serial port RS-485

Unit load	1/8
Protocols	Nokeval SCL slave, Modbus RTU slave, Ascii auto-transmission
Baud rates	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400 bits/s
Parities (Modbus only)	8N1, 8E1, 8O1, 8N2

Warnings

The device must not be disposed with household waste. Observe local regulations concerning electronic waste recycling.

Manufacturer

Nokeval Oy
Rounionkatu 107
FI-37150 Nokia
Finland

Tel +358 3 342 4800 (Mo-Fr 8:30-16:00 EET)
WWW http://www.nokeval.com/
Email sales@nokeval.com,
support@nokeval.com