User manual

2015-05-11
Firmware V4.0
PCB V1.2

7470
Serial to analog converter

Nokeval

INTRODUCTION

7470 is a serial-bus controlled analog output unit. It provides four mA or V outputs that can be controlled via an RS-232 or RS-485 bus. As a slave, it supports Nokeval SCL, Modbus RTU, and Ascii protocols. Alternatively, it can function as a SCL Master querying other devices.

Analog output ranges include 0-20mA, 4-20mA, 0-5V, and 0-10V.

7470 needs a 24 VDC supply voltage. The analog outputs are not galvanically isolated from the power supply or each other. They share the common negative wire. However the serial bus is galvanically isolated.

For using 7470 with Vaisala weather stations, a separate application note is available.

How to use this manual

The chapter Installing shows how to connect this device and set the jumpers. It also tells how to access the configuration settings. The settings itself are described in chapter Configuration menu. Each of the available serial modes are described in the chapters of their own.

Table of contents

Manufacturer

Nokeval Oy
Rounionkatu 107
FIN-37150 Nokia
Finland

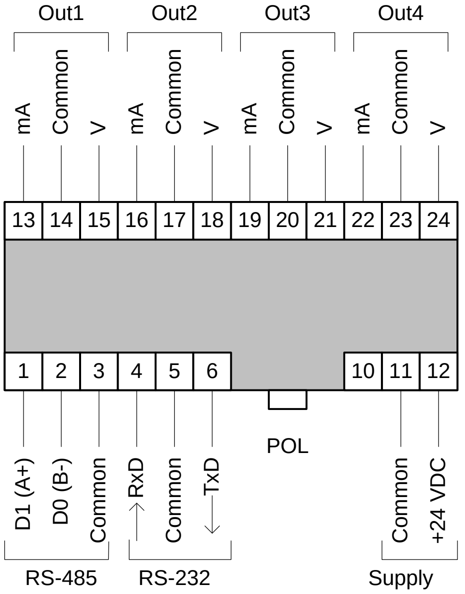
Tel +358 3 3424800
Fax +358 3 3422066
www.nokeval.com

INSTALLING

Mounting

This transmitter is intended to be mounted on a 35 mm DIN rail. The position is free.

Connections



Analog outputs

The upper row provides 4 analog outputs. The mA and V outputs on a single channel can't be used simultaneously. The analog output commons are internally interconnected with each other and the power supply common.

Serial buses

7470 provides RS-485 and RS-232 buses. There is no need to change anything to select which bus to use, but because both buses share a connector block to prevent attempts of simultaneous use, the connector block is meant to be moved to the RS-bus that is used.

RS-485-bus should be used with ground wire (common) between the devices, whenever possible. If not possible JP4 should be set to the 2-wire configuration. See the next JP4 table and 2-wire section. The data pair D1/D0 should be a twisted pair.

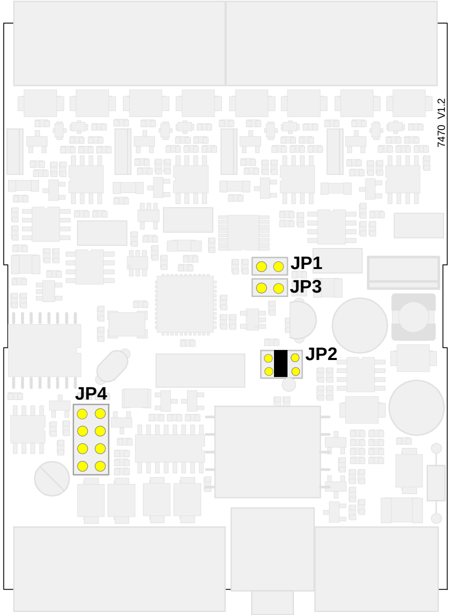
Supply voltage

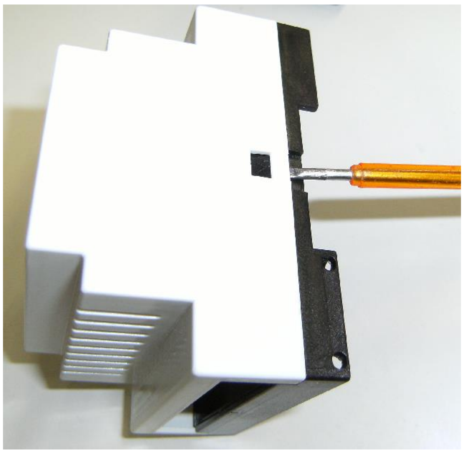
The supply voltage is connected according to the picture. There is a polarity protector diode.

Detachable connectors

The green connector blocks are detachable to make connecting easier. They accept up to 2.5mm^2 wires.

Jumpers





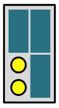
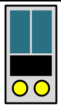
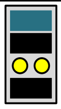
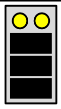
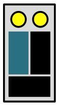
To set up the jumpers, the enclosure has to be opened.
Detach the device from the DIN rail, and insert a flat bladed screwdriver between the black and grey parts of the enclosure and lever carefully.

The jumper JP1 forces the RS-232/485 port to use SCL protocol, 9600 baud, address 0. This is to enable configuring the device without using a programming cable. When the jumper is removed, this port uses the settings defined in the configuration menu.

The jumper JP2 is for programming the microcontroller at factory. The middle pins must be shorted with a jumper.

The jumper JP3 is not used yet.

The jumper field JP4 selects RS-485 bus options, see the table below. The blue-grey jumpers are not required for operation, but indicate a place for spare jumpers. This table applies only to the PCB version V1.2.

JP4	Configuration	Usage
	Nothing	The middle devices on the bus, or when the bus is short. Factory setting.
	Terminated	The first and the last device of the bus.
	Polarized	The middle devices on the bus when there is no other device to give the polarization (fail-safe, biasing, idle voltage). Used typically, when 7470 acts as a bus master (Ser/Mode=SCL Master) and is not the first nor the last device.
	Terminated and polarized	The first and the last device on the bus when there is no other device to give the polarization (fail-safe, biasing, idle voltage). Used typically, when 7470 acts as the bus master (Ser/Mode=SCL Master) and is the first or the last device.
	2-wire	When RS-485 common (ground) wire is not available.

Configuring

The 7470 can be configured with a PC or with a hand-held programmer.

PC configuration with a programming cable

Two types of programming cable are available:

- **POL-RS232** for RS-232 port of the PC
- **DCS772** for USB port of the PC

The software used is Mekuwin, available for free.

The cable is attached in the 3.5mm jack, marked as POL. 7470 has to be powered up.

In Mekuwin, select the adequate serial port, SCL protocol, 9600 baud, and address 0 or 126. Connect.

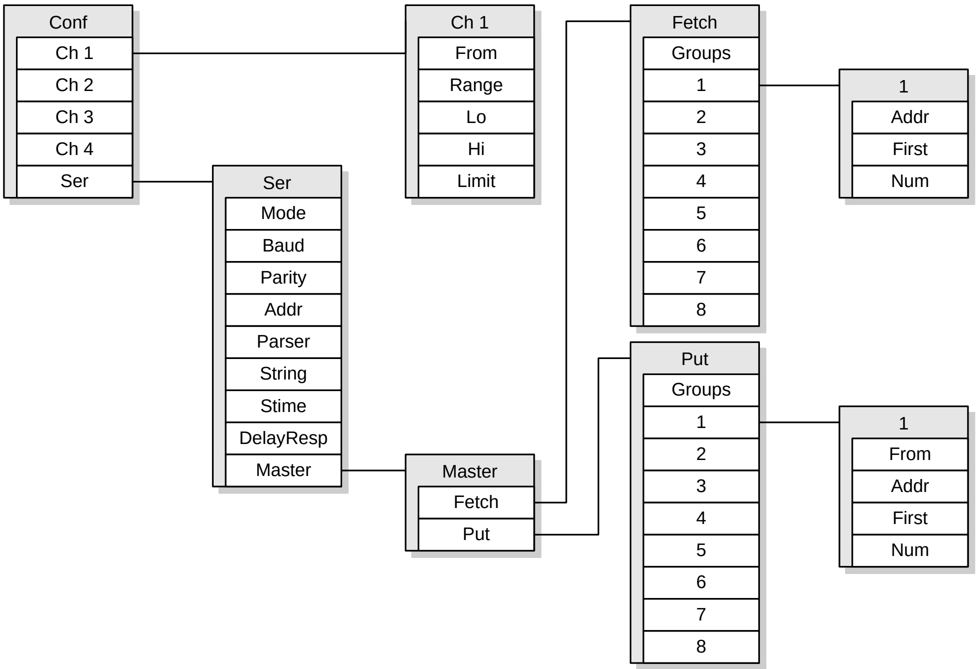
The contents of the configuration menu are explained in chapter Configuration menu. For details of Mekuwin itself, refer to the Mekuwin manual.

PC configuration via serial bus

If a programming cable is not available, 7470 can be configured via RS-232/485 port.

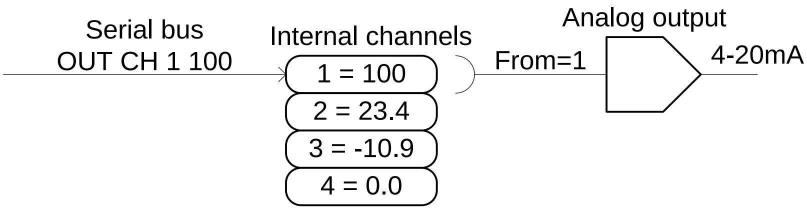
Configuring is possible only when 7470 uses SCL or Modbus protocols. If 7470 is configured for any other protocol, or the serial settings are not known, it is possible to use jumper JP1 to force the RS-232/485 port to use SCL protocol, 9600 baud, and address 0 or 126.

CONFIGURATION MENU



The menu is arranged hierarchically. The first level contains four submenus, one for each analog output, and a serial communications submenu. Some of the settings are hidden when they are not used.

Ch submenus



Each analog output is configured independently.

From

7470 contains 32 internal channels storing the variable values received via the serial bus. The channels are numbered 1…32. The From setting is used to select, which internal channel is used to control each analog output.

Most often the values are sent to internal channels 1…4, and the first analog output channel follows the first internal channel etc. However, sometimes it may be useful to be able to have several analog outputs following the same internal channel.

It is advisable to set From = 0 on unused channels. Otherwise the LED will blink an error code on these channels. This is possible beginning from firmware V4.0.

Range

The analog output range selection. Options are:

- **0-20mA**
- **4-20mA**
- **0-5V**

- **0-10V**

Lo, Hi

Analog output scaling. When the channel value controlling the analog output corresponds to the Lo setting, the output will give its low end signal (e.g. 4 mA if the range is 4-20mA). Likewise, the value of Hi will give the high end signal.

Limit

Limits the analog output within the nominal range. If enabled, the analog output will not exceed its end points, e.g. the 4-20mA output will not go below 4 mA nor above 20 mA even if requested to.

If the safety timer expires, the output will go to 0 mA or 0 V despite of this setting.

Ser submenu

The Ser menu settings affect the RS-232/485 port only. The POL port uses always SCL protocol, 9600 baud.

Mode

Serial bus protocol and mode.

- **SCL** (slave): Nokeval SCL protocol. 7470 may be controlled using SCL commands, like OUT CH and OUT SCAN. See chapter SCL protocol.
- **Modbus**: Modbus RTU protocol. 7470 may be controlled using "write holding register" and "write multiple registers" functions. All the configuration settings are also accessible. See chapter Modbus protocol.
- **Ascii**: 7470 accepts simple messages with no proprietary protocol. The messages have to end with CR, LF, or both. See chapter Ascii mode.
- **SCL Master**: 7470 acts as a serial bus master, sending commands to the other devices on the bus. Only one device should be a master on a single bus. In this mode, 7470 can read serial bus transmitters and Nokeval RTR970PRO radio receiver without a PC or another master. See chapter SCL Master.
- **Metso HART**: Hart-style protocol. See chapter Metso HART.

Baud

Baud rate 300…115200. All the devices on the bus must have the same baud rate. Factory default is 9600.

Parity

Applicable for Modbus and Ascii only. Options:

- **8N1** (with Ascii, accepts 7E1 and 7O1 too)
- **8N2**
- **8E1** (recommended for Modbus)
- **8O1**

Addr

Serial bus address of this device in SCL (slave), Modbus, and Metso HART modes.

- In SCL mode, legal values are 0…123. In addition to the selected address, 7470 will always respond in address 126. When the POL plug is inserted in the front panel, 7470 will respond in any address.
- In Modbus mode, legal values are 1…247.
- In HART mode, legal values are 0…127.

Parser

Parser selection in Ascii mode. Options are:

- **Classic**: Non-configurable parser, equivalent to 7470 firmware versions 1.x.
- **Custom**: Configurable parser.

See chapter Ascii mode.

String

Parser control string in Ascii mode, when Custom parser is selected. See chapter Ascii mode.

Stime

Safety timer. Defines a safety time in seconds. If some of the internal channels (see Ch\From on page 6) is not refreshed within this time, the safety timer will expire on that channel. An analog output configured to follow that channel will then go low (0 V or mA). This works in every mode.

The time can be selected from 1 to 60 seconds in 1 second steps. To disable the safety timers, set value of 0 here.

DelayResp

When some other device sends a SCL command to the 7470, it will process the command and send a response as soon as possible, but not earlier than 3.5 characters time from the end of the command. At 9600 baud, this minimum time corresponds to 3.6 ms.

If DelayResp is engaged, the minimum interval between the command and response is increased to 25 ms.

In SCL Master mode, setting DelayResp on makes the 7470 to keep a 5 second pause between each full cycle of sending master commands. This is useful with radio modems to not block the whole radio band. If DelayResp is off, 7470 will send commands without pauses.

Master

The submenu used to configure how this device acts as a SCL Master. See chapter SCL Master.

TROUBLESHOOTING

Diagnostic messages

7470 can detect some abnormalities. The messages can be viewed with Mekuwin, or observed on the red LED inside the enclosure.

Message	Blinks	Explanation
Stime exceeded	2	Safety timer has expired. At least one output has been dropped to 0 mA or 0 V because its safety timer has been expired due to not being updated within the time defined in Stime setting.
No serial	3	No proper packets are received from the RS-232/485 port within the last 30 seconds.
Serial error	4	Invalid data has been received from the RS-232/485 port: wrong protocol, baud rate, parity, incorrect checksum, for example.

LED

The red LED inside the enclosure blinks status information. To see the LED, the enclosure has to be opened, as guided in chapter Jumpers.

If everything is well, the LED blinks shortly every 2 seconds.

When there are active diagnostic messages, the LED blinks a series of as many blinks as stated in the table above; e.g. 2 blinks and a pause tells that the safety timer has expired. If there are several messages active, they will be blinked one after another.

Monitor

With the Monitor function of Mekuwin, some of the internal values of 7470 can be observed.

- **Ch1...Ch32** indicate the channel values. They are updated by sending data on the RS-232/485 bus. These indicate the latest received value. An expired safety timer does not affect these.
- **Out1...Out4** indicate the electrical analog output value in mA or V. An expired safety timer drops the analog outputs to zero, so these will go to zero too.
- **RecvString** is the latest received packet. It is useful for troubleshooting and for investigating the structure of the message.

Simulation

The Ch1...Ch32 items in Mekuwin monitor can be manually edited for testing purposes. The consequences are the same as when sending packets on the serial bus. The safety timer is applied for the manual input too.

Electrical

If the LED is alive, the supply voltage is present, but it might be worth checking its correct value, since a too low supply voltage can cause various problems.

If RS-485 is used, use a multimeter to measure the voltage between D1 and D0 lines. It should be positive, between 0.2 and 5 volts. If it is zero, the bus may be short-circuited, or it is lacking a polarization (also known as fail-safe, bias, or idle voltage), which is usually provided by the bus master. If the voltage is negative, the bus wires may be crossed.

Measure the D1 to D0 voltage at the other end of the bus. It should match.

With RS-232, it is advisable to measure the Tx and Rx voltage against the common. Both of them should idle -3 to -15V.

SCL PROTOCOL

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

This unit understands the following SCL commands:

TYPE ?

Returns the model name and software version ”7470 V4.0” without the quotation marks.

SN ?

Returns the serial number, e.g. ”A012345”.

OUT CH 1 100.0

Sends a value of 100 to the internal channel 1. Channels 1…9 are writable this way, although normally up to four of them are used since 7470 has four analog outputs. About internal channels, see the From setting on page 6.

7470 will return an empty response (ACK, ETX, BCC).

Acceptable characters are 0…9, minus, decimal point, and leading or trailing spaces. A deliberately invalid value (NaN) can be expressed as ----- (several minus characters) to steer the analog output to 0 V or 0 mA to indicate something is wrong.

OUT SCAN 1 4 10 20 30 40

Sends several values to the internal channels 1 to 4 (give the first and the last). Maximum of 8 values can be sent with one command; channels 1…9 can be accessed this way. The values are separated with one or more spaces.

MN xxxxx

Commands used by the Mekuwin configuration software.

N xxxxx

Nopsa commands encapsulated in SCL. See chapter Nopsa.

C xxxxx

Sending a command to an external device. Available only at POL port. The string after the command C is sent to the RS-232/485 port terminated with CRLF. The port has to be configured for Ascii protocol. See also command G.

G

Requesting Ascii packets received from the RS-232/485 port. Returns the packets from a string buffer. When all is read out, returns an empty response. This command is available at the POL port only.

PORTID

Returns 1 if the command is given in RS-232/485 port, and 2 if commanded from the POL port. For factory testing.

MODBUS PROTOCOL

Commands

- 3 Read Holding Registers: read current configuration
- 6 Write Single Register: change the configuration and control the outputs
- 16 Write Multiple registers: change several registers.
- 17 Report Slave ID: device type query.
- 109 Meku: Mekuwin uses this.
- 110 Nopsa: Nopsa commands.

Maximum Modbus frame is 150 bytes.

Command 17 returns 0x11 <bytecount> 0x00 0xFF, and then ”7470 V4.0 A012345”, for example.

Data types

- BOOL: Off/On setting. 0=off, 1=on on the lower byte.
- BYTE: Single byte value. Only the lower (least significant) byte used.
- WORD: 16-bit value. Most significant byte first.
- ENUM: List of options.
- FLOAT: 32-bit floating point IEEE 754. Least significant word first. Inside the word, most significant byte first.

Holding registers

The analog outputs can be controlled via registers 0..7 in IEEE-754 format, or alternatively via registers 1000..1003 in regular 16-bit signed integer format. These correspond to the internal registers 1…4, see page 6.

The configuration settings are accessible in register 2000 onwards. When the configuration is changed, the changes are automatically stored in the EEPROM. If the serial configuration is changed via Modbus, the new settings do not affect until powered down. Do not write excessively many settings in one transaction.

Register, FW3.x	Register, FW4.x	Name	Type	Values
0..1	0..1	Ch1	FLOAT	Signed
2..3	2..3	Ch2	FLOAT	Signed
4..5	4..5	Ch3	FLOAT	Signed
6..7	6..7	Ch4	FLOAT	Signed
1000	1000	Ch1	WORD	Signed
1001	1001	Ch2	WORD	Signed
1002	1002	Ch3	WORD	Signed
1003	1003	Ch4	WORD	Signed
2000	2000	Conf\Ch 1\From	BYTE	Unsigned 0...32
2001	2001	Conf\Ch 1\Range	ENUM	See table E1
2002..2003	2002..2003	Conf\Ch 1\Lo	FLOAT	Signed
2004..2005	2004..2005	Conf\Ch 1\Hi	FLOAT	Signed
2006	2006	Conf\Ch 1\Limit	BOOL	
2007	2007	Conf\Ch 2\From	BYTE	Unsigned 0...32
2008	2008	Conf\Ch 2\Range	ENUM	See table E1

2009..2010	2009..2010	Conf\Ch 2\Lo	FLOAT	Signed
2011..2012	2011..2012	Conf\Ch 2\Hi	FLOAT	Signed
2013	2013	Conf\Ch 2\Limit	BOOL	
2014	2014	Conf\Ch 3\From	BYTE	Unsigned 0...32
2015	2015	Conf\Ch 3\Range	ENUM	See table E1
2016..2017	2016..2017	Conf\Ch 3\Lo	FLOAT	Signed
2018..2019	2018..2019	Conf\Ch 3\Hi	FLOAT	Signed
2020	2020	Conf\Ch 3\Limit	BOOL	
2021	2021	Conf\Ch 4\From	BYTE	Unsigned 0...32
2022	2022	Conf\Ch 4\Range	ENUM	See table E1
2023..2024	2023..2024	Conf\Ch 4\Lo	FLOAT	Signed
2025..2026	2025..2026	Conf\Ch 4\Hi	FLOAT	Signed
2027	2027	Conf\Ch 4\Limit	BOOL	
2028	2028	Conf\Ser\Mode	ENUM	See table E2
2029	2029	Conf\Ser\Baud	ENUM	See table E3
2030	2030	Conf\Ser\Parity	ENUM	See table E4
2031	2031	Conf\Ser\Addr	BYTE	Unsigned 0...127
2032	2032	Conf\Ser\Parser	ENUM	See table E5
2033..2048	2033..2064	Conf\Ser\String	STRINGZ	Len=32 or 64
2049	2065	Conf\Ser\Stime	BYTE	Unsigned 0...60
2050	2066	Conf\Ser\DelayResp	BOOL	
2051	2067	Conf\Ser\Master\Fetch\Groups	BYTE	Unsigned 0...8
2052	2068	Conf\Ser\Master\Fetch\1\Addr	BYTE	Unsigned 0...255
2053	2069	Conf\Ser\Master\Fetch\1\First	BYTE	Unsigned 0...255
2054	2070	Conf\Ser\Master\Fetch\1\Num	BYTE	Unsigned 0...8
2055	2071	Conf\Ser\Master\Fetch\2\Addr	BYTE	Unsigned 0...255
2056	2072	Conf\Ser\Master\Fetch\2\First	BYTE	Unsigned 0...255
2057	2073	Conf\Ser\Master\Fetch\2\Num	BYTE	Unsigned 0...8
2058	2074	Conf\Ser\Master\Fetch\3\Addr Each group increases register address by 3.	BYTE	Unsigned 0...255
2076	2092	Conf\Ser\Master\Put\Groups	BYTE	Unsigned 0...8
2077	2093	Conf\Ser\Master\Put\1\From	BYTE	Unsigned 1...32
2078	2094	Conf\Ser\Master\Put\1\Addr	BYTE	Unsigned 0...255
2079	2095	Conf\Ser\Master\Put\1\First	BYTE	Unsigned 0...255
2080	2096	Conf\Ser\Master\Put\1\Num	BYTE	Unsigned 0...8
2081	2097	Conf\Ser\Master\Put\2\From Each group increases register address by 4.	BYTE	Unsigned 1...32
2108	2124	Conf\Ser\Master\Put\8\Num	BYTE	Unsigned 0...8

Table E1

Value	Range
0	0-5V
1	0-10V
2	0-20mA
3	4-20mA

Table E2

Value	Mode
0	SCL
1	Modbus
2	Ascii
3	SCL Master
4	Metso Hart

Table E3

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 E4

Value	Parity
0	8E1
1	8O1
2	8N2
3	8N1

Table E5

Value	Parser
0	Classic
1	Custom

ASCII MODE

Some weighing and weather transmitters are able to output their readings in "Ascii", which means human-readable data terminated by a CR or LF or both. 7470 can receive and interpret most of this kind of messages. The transmitter has to be configured to send its readings automatically – 7470 does not send query commands. 7470 will not respond to the Ascii packets.

Several 7470's can be parallel-connected to a RS-485 bus to receive the Ascii messages. Each one can be configured to handle a different part of the message. The Ascii mode does not use any addressing nor checksum. However some addressing may be realized with the custom parser, see below.

Ascii packets can be sent with any terminal software, e.g. PuTTY.

Prior to firmware V3.4, 7470 expects 8 data bits and none parity (8N1), but it discards the most significant bit. So it accepts 7E1 and 7O1 too but does not check the parity.

Beginning from firmware V3.4, 7470 can be configured for parities 8O1, 8E1, and 8N2 at Conf/Ser/Parity. Selecting 8N1 accepts 8N1, 7E1, and 7O1 by discarding the most significant bit.

There are two parsers (algorithms that split the Ascii message to fields and tries to find numerical values there): "classic" and "custom".

The classic parser does not allow any configuration, but it is simple to use. It can be used when the message contains values separated by spaces, commas, semicolons, or tabulators. When more control is needed, the custom parser gives more freedom.

Classic parser

If the message is simple like this:

```
100.0,200.0,300.0,400.0<cr><lf>
```

7470 is able to handle it with the classic parser. The fields may be separated with one comma, one semicolon, one tabulator, or one or more spaces.

If there is non-numerical characters within the field, 7470 will ignore them until a numerical character (0...9, minus, decimal point) is encountered. It will then read in the figure until a field separator or any other non-numerical character is encountered. So, 7470 is able to read this message:

```
A=100.0, B=200.0, C=300kg, D=400m2, E=0
```

Reading of 100.0 is read to the internal channel 1, 200.0 to channel 2, etc. m2 is ignored because one numeric portion has been already found in that comma-separated field.

The maximum length of a message is 150 characters. Up to 32 fields may be read in to the internal registers. The analog outputs can be programmed to follow any of these internal registers.

Custom parser

If the message is not field-separated (delimited) by a comma, space, semicolon, or tabulator, the custom parser has to be used. Also, if the transmitter sends several different types of packets and only one of them is to be picked by a certain 7470, the custom parser is needed.

The custom parser is configured with one control string that contains instructions to 7470 how to handle the message. The control string consists of the following parts:

Part	Name	Description
*	Replace string	Any number of characters in the message will be ignored until a character following the * is encountered in the message. E.g. *+ will ignore all characters in the message until a + is found. The + will be ignored too.
?	Replace character	One character will be ignored in the message. Several ?'s may be used

Part	Name	Description
		contiguously to ignore more characters.
%1	Pick	Picks a value to the internal channel 1 from the message. Will pick until a character following the %1 is encountered in the message or the message ends. E.g. %1, will pick characters until a comma is found in the message. The comma will be ignored then. The channel number can be 1…32.
%FS=;	Set field separator	Defines a field separator ";". Once defined, 7470 will split the message to fields at every separator and apply each control string row to each field.
any other	Match	If the control string contains other characters, the input string must have equivalent characters, or the parser will exit. This feature can be used to pick a certain type of packet, if the transmitter sends several different packets. When the parser exits, all the fields handled before exiting will be still used.

Some examples will clarify things up.

Simple message

Consider a message like:

```
10,20,30,40
```

To read this in, enter a control string:

```
%1,%2,%3,%4
```

(A comma-separated message could be handled by the classic parser too.)

Custom delimiter

If the message is separated (delimited) with e.g. slashes like this:

```
10/20/30/40<cr>
```

The control string that can interpret this to the four first channels:

```
%1/%2/%3/%4
```

If the delimiter is * or ? or %, it can't be mentioned directly in the control string, because those characters have a special meaning to the parser. They must be escaped by preceding them with a %. A message separated by *'s would be handled by a control string like this:

```
%1%*%2%*%3%*%4%
```

Ignoring fields

If we have a message like:

```
10,20,30,40,50,60,70,80
```

And want to read in the third and fifth field only, enter the following control string:

```
*,*,%1,*,%2
```

Looking for a pattern

A message containing something interesting in the middle, e.g. a net weight:

```
G:2334.4;N:1999.9;T:0334.5
```

The following control string will search the net weight identifier "N:" and pick the value:

```
*N:%1
```

Field separators

If the message is arranged to fields separated by some character, it is advisable to let the 7470 to handle the message field by field. This is not possible before firmware version 4.0. The field separator is defined by writing e.g. %FS=, in the control string, followed by a newline. Then give the patterns on the next line(s) in the control string. If we have a Vaisala style message:

```
0R1,Dn=236D,Dm=283D,Dx=031D,Sn=0.0M,Sm=1.0M,Sx=
2.2M
```

We can pick Dm and Sm conveniently (note that the control string has multiple rows):

```
%FS=,
Dm=%1
Sm=%2
```

7470 splits the message at each comma. Then it applies each control string row to each comma-split field.

Different messages

Beginning from firmware version 4.0, 7470 can accept several different kinds of messages and pick different values from each. For example, NMEA-0183 messages:

```
$GPGGA,092750.000,5321.6802,N,00630.3372,W,1,8,
1.03,61.7,M,55.2,M,,*76
```

```
$GPGSA,A,3,10,07,05,02,29,04,08,13,,,,,1.72,1.0
3,1.38*0A
```

```
$GPGSV,3,1,11,10,63,137,17,07,61,098,15,05,59,2
90,20,08,54,157,30*70
```

Want to pick the third and the fifth field of the $GPGGA message (latitude and longitude) plus the fourth field of $GPGSV (number of satellites). Two rows of control string will do it:

```
$GPGGA,*,%1,*,%2,*
@GPGSV,*,*,%3
```

SCL MASTER

When there is no PC or any other device available that could "master" the bus (i.e. command the bus devices), one 7470 can be configured to act as a master. It can read values from other devices on the bus and redirect them to other devices. The four analog outputs of the mastering 7470 are available too.

There must always be exactly one master on the bus. If there is no master at all, all the devices only listen. If there are two or more masters, they will collide.

The master does not have an address of its own. Only the slaves use an address. Every slave must be configured to a different address.

Fetching

Fetching means querying values from the other devices on the bus. These devices are most often transmitters measuring temperature, voltage, current etc. Also Nokeval RTR970PRO radio receiver can be read. 7470 uses SCL command MEA CH or MEA SCAN for this.

The fetch operations are configured in the menu Ser\Master\Fetch. There may be up to eight fetch groups. One group corresponds to one serial command sent to another bus device and it may fetch up to eight values.

The first setting, Groups, is used to select how many fetch groups are needed.

Within each group, there are three settings. Addr defines the serial bus address of the other device. First tells from which channel to read on the other device, and Num tells how many channels to read.

The values fetched with group 1 are stored in the internal channels 1 onwards. The values fetched with group 2 are stored in the next "free" internal channels etc. E.g. if group 1 fetches 4 channels and group 2 fetches another 4, the values will be placed in internal channels 1…4 and 5…8, respectively.

The fetches may be configured freely. It is allowed to do several fetches from the same bus device (e.g. to fetch more than 8 values from a 16-channel transmitter).

Every time a fetch is done successfully, the safety timers of the updated internal channels are reset. If some channel is not updated for a time specified in Stime setting in the Ser submenu, that channel will be considered expired. If an analog output is following that channel, it will be pulled to 0 mA or 0 V.

Putting

7470 is able to send or put the fetched values to other devices on the bus. E.g. one 7470 can read 16 channels from a temperature transmitter and then deploy the values to other 7470's to have more than 4 analog outputs. 7470 uses SCL command OUT CH or OUT SCAN for this.

Up to eight put groups may be defined. Each group corresponds to one serial command sending up to eight values to another device on the bus.

The first setting, Groups, is used to define how many put groups are needed.

Within each group, there are four settings. Addr defines the bus address of the other device. First tells the first channel on the other device that the values are sent to. Num tells how many values to send.

From defines the internal channel of this 7470 that provides the first value to be sent. For example, if configured as follows:

- From = 5
- Addr = 2
- First = 1
- Num = 4

The mastering 7470 will send values from its internal channels 5…8 to the channels 1…4 of the device on bus address 2.

If some internal channel has expired, dashes ----- will be sent to indicate fault.

METSO HART

7470 can be controlled using the Hart protocol on RS-485 bus. Supported command set is minimalistic: commands 0 and 206. As always with Hart, odd parity 8O1 is used.

7470 accepts only the long address, not the ”polling address”. Manufacturer ID is 47d or 2Fh (Valmet/Metso). Device type byte is 01h. The two most significant bytes of the address are zero, and the least significant is selectable on the configuration menu Ser\Addr, selectable 0..127d.

This device needs 2 preambles and sends 5.

If no command is received within 15 seconds, then the serial port will be reset. This is repeated every 15 seconds until a command is received.

Command 0: Read Unique Identifier

This command is useful to see the device exists on the bus.

Command 206: Controlling the outputs

The analog outputs are controlled with this command. After the command byte 206d or CEh comes one byte indicating the number of data bytes. It must be 16d. After that comes 32-bit floating point numbers in IEEE754 format the most significant or exponent byte first. The value of these numbers has to be between 0 and 100 inclusive in order to be accepted – if they are not, they are discarded and the safety timer is not reset.

An example of the whole frame (in hexadecimal):

FF	FF	82	AF	01	00	00	00	CE	10	40	00	00	00	40	C0	00	00	41	40	00	00	41	C0	00	00	B2
Pream		D	Mf	T	ID			C	Bc	Out1 = 2				Out2 = 4				Out3 = 8				Out4 = 16				LP

- Mf = six lowest bits of manufacturer id, and most significant bit set for primary master
- T = device type, always 01h
- ID = 000000h + the address selected in the configuration
- C = command
- Bc = byte count
- LP = longitudinal parity

The device will respond:

5xFF	86	AF	01	00	00	00	CE	02	00	00	E4
Pream	D	Mf	T	ID			C	Bc	St1	St2	LP

Possible errors are:

- Longitudinal parity error, status bytes 88 00 (in hexadecimal)
- Command not implemented 40 00
- Incorrect byte count 05 00
- Buffer overflow 82 00

If the analog outputs are requested to go outside their range, they will go as far they can (Limit=off) or to the end of the range (Limit=on). No error is given.

NOPSA

Nopsa is a simple language intended for data interchange between devices. The data is transferred in binary format, making it ideal for machine-to-machine communications. 7470 allows Nopsa commands to be sent over SCL and Modbus protocols.

Nopsa is specified in a separate document, available on request.

This device supports the following Nopsa commands:

- 1/0: Type query, returns ”7470”.
- 1/1: Software version query, returns ”V4.0”.
- 1/2: Serial number query, returns ”A012345”.
- 1/3: Description query, returns ”Analog output unit”.
- 1/32: Meku configuration commands over Nopsa.
- 1/36: Fetching a delayed response. Used with the Route commands.
- 2/2: Control the analog outputs. Floating point format only supported.
- 2/3: Information about the outputs. The outputs are named Out1...Out4.
- 3/0: Routing commands. Allows routing SCL-Nopsa commands from the POL port to the RS-232/485 bus devices, mainly for configuring them through the 7470. Available only with the SCL Master protocol. The route address defines the target device address on the bus. 7470 returns always a delayed response (PROCESSING), and the final response is then polled with Nopsa command 1/36.
- 3/1: Information about the routing capability of this device.

SPECIFICATIONS

Outputs

Channels	4 outputs
Isolation	Non-isolated from each other and power. Common negative wire.

mA output

Ranges	0-20mA and 4-20mA
Accuracy at 25°C	0.1% of range
Operating range	0..22mA typ
Maximum load	600 ohms

V output:

Ranges	0-5V and 0-10V
Accuracy at 25°C	0.1% of range
Operating range	0..11.1V typ
Maximum load	5 kilo-ohms
Short circuit protect	yes (~15mA)

Response time

Output reaction time	0-40ms
Output timing	4.7ms +reaction 0-40ms (67% of end value) 12.5ms +reaction 0-40ms (93% of end value) 150ms +reaction 0-40ms (100% of end value)

Serial ports

Buses	RS-232, RS-485
Protocol	Nokeval SCL, Ascii, Modbus RTU
Baud rates	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 baud
Response time	Min 3.5 characters Max 10ms or 3.5 characters
Termination	Externally with a resistor or internally with a jumper
Isolation	Functional isolation 50 VAC/120 VDC

General

Temp stablility	50ppm/°C of range
DA conversion	12 bits (resolution 4096)
Supply voltage	18…28VDC
Curr consumption	30mA…120mA
Oper temperature	0…55 °C
Protection	IP20
Mounting	35 mm DIN rail
Weight	125g
EMC	EN 61326-1:2013

Dimensions