



POWERED BY
NOKEVAL

Flex-Sky-Link-RS485

Manual

Contents

Introduction	2
Installation	3
Maintenance	4
Firmware settings	5
Operation	9
SCL protocol.....	10
Modbus protocol.....	11
Nopsa commands	16
Specifications.....	21
Warnings	22
Manufacturer	22

Introduction

Flex-Sky-Link-RS485 is a data receiver for Nokeval Ovasky 433 MHz wireless data transmission technology. It receives and buffers the data packets that Ovasky transmitters have sent.

Flex-Sky-Link-RS485 is housed in a watertight (IP 66) and impact resistant plastic enclosure. The joint between the two modules is sealed with two rubber O-rings. The receiver requires 8..30 VDC power supply and is connected to a computer or PLC via RS-485.

Nokeval SCL or Modbus RTU protocol is used for data transmission between the receiver and the computer. Flex-Sky-Link-RS485 has one diagnostic LED.

Installation

Mounting

The best installation environment for a radio receiver is above a big grounded horizontal metal surface surrounded by as few as possible vertical metal surfaces. Antenna is to be installed perpendicular to the metal surface. The best range for the transmitter is achieved when there is a line-of-sight from the receiver to the transmitter. Walls and objects between the transmitters attenuate the signal and thereby shorten the range. On the other hand, metal surfaces cause reflections which might lengthen the range.

Enclosure

Flex-Sky-Link-RS485 consists of a radio transceiver module and a processor/serial module. Before use, please make sure that the modules are pushed together all the way so that the locking latches snap into position.

Opening the case

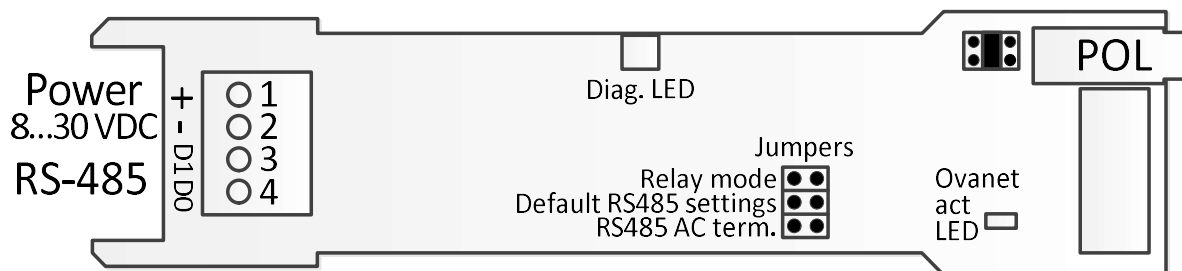
Modules can be detached from each other by pressing the locking latches on the sides of the enclosure and at the same time pulling the modules apart.

For receiver operation the serial module must be opened for RS485 and power cable installation. Serial module can be opened by gently pushing with the tip of a flat-blade screwdriver into the slot that holds the cover plate.



When the cover plate is removed, the circuit board can be pulled out of the enclosure.

Connections



Serial module has a 4-pin terminal block for power supply and RS-485 connection. Serial module features also a round 3.5 mm POL programming connector jack found on many other Nokeval products as well.

Power supply

The supply voltage range is 8...30 VDC and is connected to terminals 1 (+) and 2 (-). Power demand is 50 mA. Flex-Sky-Link-RS485 is protected against wrong polarity of the supply voltage.

RS-485 serial bus

RS-485 is used when the device operates as a receiver. RS-485 interface can easily be added to a computer by using Nokeval DCS770 or DC771B USB - RS-485 converter or RCS770 USB/RS-232-RS-485 converter. If you use DC771B converter, no external power supply needed (DC771B's power supply should be set to 10V). RS-485 bus is connected to terminals 3 (D1) and 4 (D0). The supply voltage's negative terminal 2 (-) can also be used as the ground for RS-485.

The RS-485 bus consists of a bidirectional half-duplex twisted 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 .

The length of an RS-485 bus can be up to 1 km and it can be connected to a maximum of 32 devices, more can be connected via bus repeaters. If the bus is long (say more than 100 m), it is recommended to terminate the first and last device on the bus (set the "RS485 AC term" jumper to "on" position).

The polarity of the data pair is important. Modbus specifications call the positive idling line D1, but it is also commonly known as +, B, and A. Correspondingly the negative idling line is -, A, or B.

The bus needs one device that gives a small voltage between the data wires when no device is transmitting on the bus. This is called biasing or fail-safing. The master device is usually the natural choice for biasing.

POL connector

The POL connector is convenient for configuring when the device operates in relay mode. This connector can be used when the modules are disconnected from each other. Note that when the POL plug is connected the RS-485 bus is disabled.

Two types of programming cables are available:

- POL-RS232 for the RS-232 port of the computer.
- DCS772 for the USB port of the computer (recommended).

Jumpers

Flex-Sky-Link-RS485 has three jumpers in the serial module: "RS485 AC term" RS-485 termination jumper and default serial settings jumper.

If the device is the last device on the RS485 bus and the bus is long (say more than 100 m), it is recommended to set the "RS485 AC term" jumper to "on" position. When this jumper is set, AC termination is used which means that 1 nF capacitor and 110 ohm resistor are connected in series between the bus wires.

Maintenance

Cleaning

The plastic parts can be cleaned with a soft cloth and soap water. The cloth must be damp but not wet. Cleaning with isopropyl alcohol is also allowed.

Firmware settings

The Flex-Sky-Link-RS485 can store the data packets in two ways: Internal buffer and channels.

All the received data is stored into an internal buffer. Data packets from the buffer can be read by Nopsa protocol (encapsulated to Modbus RTU or Nokeval SCL).

Data is also stored to those channels which have been set to the corresponding Ovanet address and channel. Channels can be read by Modbus RTU or Nokeval SCL protocol.

Mekuwin

MekuWin can be used to change settings on various Nokeval products. It has a unique feature: it loads the structure and the contents of the configuration menu from the target device, so the same MekuWin version can be used with past and forthcoming products. There is no need to update this software every time a new product or product version is released. You can download Mekuwin from Nokevals web site for free. Mekuwin has its own instruction manual.

It is also possible to change the configuration settings remotely by writing to the appropriate Modbus Holding registers.

With a POL programming cable

The communications parameters for POL are always:

Protocol	SCL
Baud	9600
Parity	8N1
Address	0

Over the RS-485 serial bus

The Mekuwin communications parameters must match the selections made in the device (Serial menu).

Resetting RS-485 serial communication settings

In case RS-485 serial settings are for some reason not known, they can be temporarily set to defaults by the “default RS485 settings” jumper (shown in the figure on page 3) when the device is powered up.

Default communication parameter settings:

Protocol	Modbus RTU
Baud	9600
Parity	8E1
Address	1

Configuration settings

The configuration settings are arranged as a hierarchical tree: The Conf menu has submenus, and these contain settings and possibly more submenus, etc.

Serial submenu

Serial communication settings.

Protocol

- SCL. Nokeval SCL protocol.
- Modbus RTU. Modbus RTU protocol. Default.

Baud rate

Baud rate selection: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400.
Default 9600.

Bits

8N1, 8N2, 8E1, 8O1

Note: SCL protocol uses always 8N1 (this item does not appear when the protocol is SCL).
Modbus RTU uses commonly 8E1.

Address

Serial communications address selection. Since several devices can be attached to the same bus, each device needs to be configured to a different Address. Valid SCL-addresses are 0...123. Valid Modbus RTU-addresses are 1...247. Default address is 1.

Ovasky submenu

Note, especially if you use POL cable: Non-volatile Ovasky settings are stored in the radio module.

When power is on and modules are connected to each other, processor module reads Ovasky settings to its temporary memory (with Meku also use). If the modules are separated from each other when Ovasky parameters are set, those settings will be lost if the power is switched off from processor module before the modules are combined back together.

Network

0..255. The network address that must be the same for all devices on the same network.

Effort

1..7, Custom.

Small value: Less power consumption, faster data rate, shorter radio range.

Large value: More power consumption, slower data rate, longer radio range.

Custom: Expert settings for frequency, bandwidth and spreading factor.

Power

0..10 dBm. Output power range .

Channel

1..7. Disabled if effort "custom" is selected.

Enable Acknowledgement

Reserved for future use.

Auth. key

Key for Message Integrity Check and data encryption.

Enable beacon

Enable beacon for coverage mapping.

Relay RX margin

Received signal margin can be stored in the buffer as its own channel. The reading should be a positive value. If the reading is close to zero, the strength of the received signal is at the reception limits.

RX margin channel

Select channel for the received signal margin.

Channels submenu

Up to 100 channels can be set up in the receiver. These channels can be set up to contain measuring data received from the Ovasky transmitters. The values of these channels can then be queried over the serial interface. This menu contains settings for configuring these channels.

Timeout

If the measurement is older than the value entered here (in minutes), then the measurement is considered obsolete, and the result is read NaN (not a number). For example, if Timeout = 10 min then channel value is set to NaN when more than 10 minutes have passed since last reception.

Ch count

0..100. Number of channels.

Ch1 ... Ch100

Submenus for channel-specific settings.

Value

Defines where the contents (reading) of the channel comes from:

- **Meas. value:** Measurement result from the Ovasky transmitter.
- **Age:** The time in seconds from the last measurement.
- **Batt level:** Battery level (%) in the Ovasky transmitter.
- **Ext power:** 1 if the Ovasky transmitter is currently gets power from the external power source. 0 if not.

In the Ovasky network, data of “Batt level” and “Ext power” are in separate data packet than measurement data and therefore arrives at the receiver at different time and less frequently than “Meas value”.

Radio ID

Node address of the Ovasky transmitter (1...16777213).

Channel.

Measuring channel of the Ovazone transmitter (1...255).

Factor.

The measurement result can be multiplied or divided by decades. This can be helpful, for example when measurement value is very large or small. This makes it possible to change numerical values to fit in Modbus integer register.

Reading

Channel reading.

Monitor menu

Monitor menu contains some information that can be useful during installation and which may facilitate the monitoring of system's operating condition.

Uptime

Time the device has been running.

Rssi

Current Rssi value when no packets are received.

RX duty

Received data packets duty cycle.

TX duty

Transmitted data packets duty cycle.

Unid duty cycle

Received (and rejected) unidentified data packets duty cycle.

Switches

Displays the jumper settings (except no “RS485 AC term”). 0 if no jumpers. 2 or 3 if default serial settings jumper is in place.

Operation

Flex-Sky-Link-RS485 has one diagnostic LED that shines through the case of the device.

If data has been received during the last 60 seconds, LED gives one quick flash every two seconds. Otherwise LED gives two quick flash every two seconds. If an error occurs, the LED lights up continuously and winks two times every two seconds

Receiver operation

Serial commands

Following protocols can be used to read data from the device:

- Nokeval SCL protocol (page 10).
 - Reading with SCL commands.
 - Reading with Nopsa commands over SCL protocol (page 16).
- Modbus RTU protocol (page 11).
 - Reading Modbus registers.
 - Reading with Nopsa commands over Modbus protocol (page 16).

SCL protocol

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

Nokeval SCL always uses 8N1 parity.

SCL Commands

This device supports the following SCL commands:

TYPE ?

Returns the model name and software version of the device.

SN ?

Returns the device's serial number, for example "A123456".

MEA CH <ch> ?

Returns the last received value from the channel <ch>. The response may contain digits 0-9, minus sign, and a decimal point. The scientific representation 1.00E-3 is not used. If the channels result is NaN (Not A Number), device returns -----.

MEA SCAN <first> <last>

Returns the last received values of channels <first> to <last> separated with a space. E.g. MEA SCAN 1 3 will return the values of channels 1, 2, and 3. An example response: 25.6 29.1 0. If the channels result is NaN (Not A Number), device returns -----

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 section "Nopsa commands".

MN <hexadecimal data>

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

Modbus protocol

Supported Modbus RTU commands:

- 3 Read Holding Registers: Read settings.
- 4 Read Input Registers: Read result values.
- 6 Write Single Register: Change settings.
- 16 Write Multiple registers: Change multiple settings at once.
- 17 Report Slave ID: Device type information.
- 109 Meku: This is used by Mekuwin configuration software.
- 110 Nopsa: This is used to transport Nopsa protocol on Modbus.

Command 17 returns 0x11 <byte count> 0x00 0xFF, followed by for example “FT20 V1.0 A123456”

Maximum Modbus packet length is 240 bytes. This affects the maximum possible register count that can be accessed simultaneously with commands 3, 4 and 16.

When settings are changed, the device will save the settings instantly into the configuration EEPROM memory. If serial settings are changed, new settings will take effect only after cycling the device power, it works this way so that all serial settings can be done at once without breaking the serial connection.

Data types:

- BOOL: On/off value. 0=off, 1=on, in lower (right hand side) byte.
- BYTE: 8-bit value. Only lower (right hand side) byte used.
- WORD: 16-bit value.
- ENUM: List of alternatives.
- FLOAT: 32-bit float IEEE 754. Least significant word first, inside word most significant byte first.
- STRINGZ: Zero terminated string. In one Modbus register data is presented as most significant byte first.

Input registers

0..1	Ch1\Reading	FLOAT (LSW, MSB)	Signed
2..3	Ch2\Reading	FLOAT (LSW, MSB)	Signed
...	...		
198..199	Ch100\Reading	FLOAT (LSW, MSB)	Signed
200..201	Ch1\Reading	FLOAT (MSW, MSB)	Signed
202..203	Ch2\Reading	FLOAT (MSW, MSB)	Signed
...	...		
398..399	Ch100\Reading	FLOAT (MSW, MSB)	Signed
400..401	Ch1\Reading	FLOAT (LSW, LSB)	Signed
402..403	Ch2\Reading	FLOAT (LSW, LSB)	Signed
...	...		
598..599	Ch100\Reading	FLOAT (LSW, LSB)	Signed
600..601	Ch1\Reading	FLOAT (MSW, LSB)	Signed
602..603	Ch2\Reading	FLOAT (MSW, LSB)	Signed
...	...		
798..799	Ch100\Reading	FLOAT (MSW, LSB)	Signed
1000	Ch1\Reading	WORD	Signed
1001	Ch2\Reading	WORD	Signed
...	...		
1099	Ch100\Reading	WORD	Signed
1200..1201	Ch1\Reading	LONG (LSW)	Signed
1202..1203	Ch2\Reading	LONG (LSW)	Signed
...	...		
1398..1399	Ch100\Reading	LONG (LSW)	Signed
1400..1401	Ch1\Reading	LONG (MSW)	Signed
1402..1403	Ch2\Reading	LONG (MSW)	Signed
...	...		
1598..1599	Ch100\Reading	LONG (MSW)	Signed
2000	Ch1\NW1 datatype	ENUM	See Table D1
2001	Ch1\Ext power	BOOL	
2002	Ch1\Battery level	BYTE	0..100 [%]
2003	Ch1\RX margin	WORD	dB*10, Signed
2004	Ch1\Utility flags	WORD	See Table D2
2005	Ch1\Data flags	WORD	See Table D2
2010	Ch2\ NW1 datatype	ENUM	See Table D1
...	...		
2995	Ch100\Data flags	WORD	See Table D2
3000..3001	Ch1\Node address	LONG (LSW)	Unsigned
3002	Ch1\Channel	WORD	Unsigned
3003	Ch1\Device type	ENUM	See Table D3

Table D1

Value	Type
0	Fault
1	Bit
2	8-bit unsigned integer
3	8-bit signed integer
4	16-bit unsigned integer
5	16-bit signed integer
6	32-bit unsigned integer
7	32-bit signed integer
8	64-bit unsigned integer
9	32-bit floating point
10	64-bit floating point
11	String
12	Struct
13	Data is not yet received

Table D2

Bits	Bits
0..14	Age counter [sec]
15	Data changed

Table D3

Value	Type
258	Core-Node-...
259	Beat-...
260	Scout
261	Flex-T
262	Flex-T-RH
263	Flex-ES -CS
265	Flex-Router

3004	Ch1\Firmware ver.	WORD	
3005	Ch1\Cal date	WORD	
3006..3009	Ch1\Serial	STRINGZ	
3010..3011	Ch2\Node address	LONG (LSW)	Unsigned
...	...		
3996..3999	Ch100\Serial	STRINGZ	

In the Ovanet network, utilitydata (Ext power, battery level, device type, firmware ver. and cal date) are in separate data packet than the measuring data and therefore arrives at the receiver at different time and less frequently.

Measured values are available in 4 different word/byte order formats in registers below 1000.

All floats are 32-bit floating point numbers according to IEEE 754.

- In registers 0...199: Least significant word first, inside word most significant byte first.
- In registers 200...399: Most significant word first, inside word most significant byte first.
- In registers 400...599: Least significant word first, inside word least significant byte first.
- In registers 600...799: Most significant word first, inside word least significant byte first.
- In registers 1000...1099: A 16-bit signed integer. Channel setting "factor" can be used to define a fixed point. For example factor 10 changes the original measurement value from 15.2 to register value 152.
- In registers 1200...1399: A 32-bit signed integer. Least significant word first. Channel setting "factor" can be used to define a fixed point. For example factor 10 changes the original measurement value from 15.2 to register value 152.
- In registers 1400...1599: A 32-bit signed integer. Most significant word first. Channel setting "factor" can be used to define a fixed point. For example factor 10 changes the original measurement value from 15.2 to register value 152.

Note! In case the reading is too old (older than the timeout parameter configured in the menu specifies) or there is no reading for a channel then float value is Quiet NaN (0x7FC00000), word value is 0x7FFF and long value is 0x7FFFFFFF.

Holding registers

Address	Name	Type	Values
0..1	Ch1\Reading	FLOAT (LSW, MSB)	Signed
2..3	Ch2\Reading	FLOAT (LSW, MSB)	Signed
...	...		
198..199	Ch100\Reading	FLOAT (LSW, MSB)	Signed
200..201	Ch1\Reading	FLOAT (MSW, MSB)	Signed
202..203	Ch2\Reading	FLOAT (MSW, MSB)	Signed
...	...		
398..399	Ch100\Reading	FLOAT (MSW, MSB)	Signed
400..401	Ch1\Reading	FLOAT (LSW, LSB)	Signed
402..403	Ch2\Reading	FLOAT (LSW, LSB)	Signed
...	...		
598..599	Ch100\Reading	FLOAT (LSW, LSB)	Signed
600..601	Ch1\Reading	FLOAT (MSW, LSB)	Signed
602..603	Ch2\Reading	FLOAT (MSW, LSB)	Signed
...	...		
798..799	Ch100\Reading	FLOAT (MSW, LSB)	Signed
1000	Ch1\Reading	WORD	Signed
1001	Ch2\Reading	WORD	Signed
...	...		
1099	Ch100\Reading	WORD	Signed
1200..1201	Ch1\Reading	LONG (LSW)	Signed
1202..1203	Ch2\Reading	LONG (LSW)	Signed
...	...		
1398..1399	Ch100\Reading	LONG (LSW)	Signed
1400..1401	Ch1\Reading	LONG (MSW)	Signed
1402..1403	Ch2\Reading	LONG (MSW)	Signed
...	...		
1598..1599	Ch100\Reading	LONG (MSW)	Signed
2000	Serial\Protocol	ENUM	See table E1
2001	Serial\Baud rate	ENUM	See table E2
2002	Serial\Bits	ENUM	See table E3
2003	Serial\Address	BYTE	Unsigned 0...247
2021	Ovasky\Enable beacon	BOOL	
2030	Ovasky\Network	BYTE	Unsigned 0...255
2031...2032	Ovasky\Frequency	FLOAT	MHz

2033	Ovasky\SF	BYTE	7..12
2034	Ovasky\BW	BYTE	0..9
2035	Ovasky\Key	STRINGZ	max 16 char.
2090	Conf\Channels\Ch count	BYTE	Unsigned 1...100
2091	Conf\Channels\Timeout[s]	WORD	Unsigned 0... 65535
2100	Conf\Channels\CH 1\Value	ENUM	See table E4
2101...2102	Conf\Channels\CH 1\Node address	LONG	Unsigned 1...16777213
2103	Conf\Channels\CH 1\Channel	BYTE	Unsigned 1...255
2104	Conf\Channels\CH 1\Factor	ENUM	See table E5
2110	Conf\Channels\CH 2\Value	ENUM	See table E4
2111...2112	Conf\Channels\CH 2\Node address	LONG	Unsigned 1...16777213
2113	Conf\Channels\CH 2\Channel	BYTE	Unsigned 1...255
2114	Conf\Channels\CH 2\Factor	ENUM	See table E5
...	...		
2990	Conf\Channels\CH 100\Value	ENUM	See table E4
2991...2992	Conf\Channels\CH 100\Node address	LONG	Unsigned 1...16777213
2993	Conf\Channels\CH 100\Channel	BYTE	Unsigned 1...255
2994	Conf\Channels\CH 100\Factor	ENUM	See table E5

Enum values

Table E1

Value	Protocol
0	SCL
1	ModbusRTU

Table E2

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

Table E3

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

Table E4

Value	Value
0	Meas. value
1	Age
2	Batt level
3	Ext power

Table E5

Value	Factor
0	0.000001
1	0.00001
2	0.0001
3	0.001
4	0.01
5	0.1
6	1
7	10
8	100
9	1000
10	10000
11	100000
12	1000000

Nopsa commands

Nopsa is a command language which enables measurement data and configuration data transfer. Nopsa can be used to transfer data between devices or from host to device. Nopsa needs some transfer layer protocol, which takes care of addresses, transfer error management and packet length. This device supports Nopsa commands over either Nokeval SCL or Modbus RTU protocols.

Supported Nopsa commands

- 1/0 (Type) Read device type
- 1/1 (Version) Read device version
- 1/2 (Serial number) Read serial number of the device
- 1/3 (Description) Read short description of the device
- 1/4 (Command set) Read command set number for the device
- 1/5 (Serial buffer size) Read serial buffer size
- 1/7 (Radio ID) Read radio ID
- 1/16 (Reset) Reset device
- 1/32 (Meku) Pass Meku configuration commands to device
- 2/0 (Out value request) Read channel reading
- 2/1 (Out resource request) Read channel metadata (name, data type)
- 4/4 (Read next from buffer) Read data entry from buffer and move read position to next
- 4/5 (Reread last) Returns last read operation contents

Transport protocol SCL

When Nopsa packets are transported on SCL data is converted to hexadecimal notation (0-9 and A-F). One Nopsa byte will become 2 bytes. No spaces between characters. Packet starts with SCL command N and a space.

ID 'N' ' ' Nopsa-packet in hexadecimal ETX BCC

Response is transferred also same way in hexadecimal, but N command is not appended.

ACK Nopsa-response in hexadecimal ETX BCC

Transport protocol Modbus RTU

Command function 110 (0x6E) is reserved for Nopsa commands in Modbus free command area. After function code there is one byte which informs Nopsa packet length.

ID 0x6E Length Nopsa-packet CRC

Response is in same format.

0x6E Length Nopsa-packet CRC

Nopsa response

Each response contains first status byte.

Bit	Description
.7	Internal error. Device has detected some internal malfunction. In example flash memory don't respond. More detailed error information need to be request by Meku Diag.
.6	External error. Device has detected some external error. More detailed error information need to be requested by Meku Diag.
.2-.0	Command progress: * 0 = OK * 1 = Command is not supported * 2 = Parameter error * 3 = Device is unable to process the command at the moment (busy) * 4 = Command is legal, but some error caused it to fail

If response is not OK, then the response data is not response for the command. Command specific data begins immediately after status byte.

Nopsa command group 1 – Basic commands

	Command	Response
1/0 (Type)	0x01 0x00	status string
string: Device type as string -> FT20		
	Command	Response
1/1 (Version)	0x01 0x01	status string
string: Device version as string -> V1.0		
	Command	Response
1/2 (Serial number)	0x01 0x02	status string
string: Device serial number as string -> A123456		
	Command	Response
1/3 (Description)	0x01 0x03	status string
string: Device description as string -> "Wireless data receiver and repeater"		
	Command	Response
1/4 (Command set)	0x01 0x04	status set*4 (*4 means 4 bytes)
Set: Informs which Nopsa command set device implements. Command sets are described in another document.		
	Command	Response
1/5 (Serial buffer size)	0x01 0x05	status size
size: Informs serial buffer size of the device		
	Command	Response
1/16 (Reset)	0x01 0x10	No response
Device resets immediately after command and don't response for it.		
	Command	Response
1/32 (Meku)	0x01 0x20 Meku command	status Meku response
Command used by Mekuwin configuration software		

Nopsa command group 2 – Data commands

	Command	Response
2/0 (value request)	0x02 0x00 number	status type data*4
number: Channel number 0..99, type: 4 (FLOAT), data: float IEEE754		
	Command	Response
2/1 (resource request)	0x02 0x01 number	status types flags name*n
number: Channel number 0..99, types: 4 (FLOAT), flags:0, name: Ch1..Ch100		

Nopsa command group 4 – Logger commands

Real-time data buffer commands

	Command	Response
4/4 (Read next)	0x04 0x04	status index*2 lap counter timestamp*4 id*2 type data*n
Index, lap counter and id are always zero. In case of no new data return only status byte.		
	Command	Response
4/5 (Reread last)	0x04 0x05	As in command 4/4

Return data which was read last. This has its uses when serial communication error happens.

	Command	Response
4/6 (Erase buffer)	0x04 0x06	status
Clears the buffer		

Data structure

Data structure in buffer is following

Data type STRUCT (32)		
Struct type	1 byte	2: Ovanet struct
Subtype	1 byte	0
Source node address	4 bytes	1 ... 16 777 213
Source endpoint	1 byte	0 ... 239: NW1 data 240 ... 255: Diagnostics data
Destination endpoint	1 byte	0 ... 239: NW1 data 240 ... 255: Diagnostics data
Data		NW1 or diagnostics data.

Example

A simple example of one-Ovasky transmitter and Flex-Sky-Link-RS485.

Ovasky transmitter	
Network	160
Effort	4
Channel	7
Auth key	Empty
Node address (not settable)	10020
Measuring channel	1

Flex-Sky-Link-RS485	
Conf/Ovasky/Network	160
Conf/Ovasky/Effort	4
Conf/Ovasky/Channel	7
Conf/Ovasky/Auth key	Empty
Conf/Channels/Timeout	300
Conf/Channels/Ch count	1
Conf/Channels/Ch 1/Value	Meas. value
Conf/Channels/Ch 1/Node address	10020
Conf/Channels/Ch 1/Channel	1
Conf/Channels/Ch 1/Factor	1

When the Flex-Sky-Link-RS485 receives a data packet from the transmitter 10020, which includes measuring value of the channel 1, it stores the value to the channel 1.

The value from channel 1 can be read by SCL protocol:

Command: ID 'MEA CH 1 ? ' ETX BCC

Response: ACK '25.5' ETX BCC

Or by Modbus protocol in which case the value can be read, for example, from the input registers 0-3.

Specifications

Environment

Storage temperature	-40...+70 °C
Operation temperature	-30...+60 °C
Operation humidity	0...100 %RH
Protection class	IP 65
Enclosure material	Plastic (PC+ABS)
Compatibility	Ovasky series devices and Ovaport service

Dimensions

Length	379.6 mm, inc. antenna
Width	60.2 mm
Height	32.5 mm
Weight	123 g

Radio

Antenna connector	50 Ω BNC female connector
Antenna	Whip antenna
Frequency	433 MHz
Transmitting power	max 10 dBm E.R.P.
Open space range	about 2 km
Indoor range	20 to 60 m typically

RS-485 serial connection

Connector	4-pin terminal block combined with power supply, terminal 3 D1, terminal 4 D0. Maximum cable length is 1000 m
Protocol	Nokeval SCL, Modbus RTU, Nopsa
Baud rates	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400 bits/s
Modbus parity	8N1, 8N2, 8E1, 8O1

Power supply

Connector	4-pin terminal block combined with RS 485, terminal 1 +, terminal 2 -.
Voltage	8...30 VDC
Current requirement	50 mA

LEDs

Internal LED

Settings

Connection	RS-485 or POL
Protocol	Nokeval Meku
Software	Mekuwin for Windows

Warnings



The device and its battery must not be disposed of in household waste. Observe local regulations concerning the disposal of electrical waste.

Manufacturer

Nokeval Oy
Rounionkatu 107
FI-37150 Nokia
Finland

Phone +358 3 342 4800 (Mon-Fri 8:30-16:00 EET)

WWW <http://www.nokeval.com/>

<http://www.ova.fi/en>

Email ova@nokeval.com,

support@nokeval.com

