Embedded Systems SIA, VAT No LV40003411103
47. Katolu str., Riga, LV 1003, LATVIA
Phone: +371 67648888, e-mail: sales@openrb.com

embedded
systems

COMPANY WITH
MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO 9001:2015

KNX
MEMBER

can
protocol creator

CANx / LoRa DALI gateway

ENG - Data sheet
Issue date 26.11.2021

Application

DALI (Digitally Addressable Lighting Interface) CANx gateway is a device specially designed for management and control of dimmable lights via CAN FT bus and wirelessly over LoRa 433. In a typical application, a DALI-bus consists of one gateway (master), and multiple slaves. In DALI-bus segment a master can control up to 64 individually addressable slaves who are also called (digital addressable) ballasts. The DALI standard enables compiling these slaves into: 16 light scenes (incl. dimming values and transitional periods) and 16 lighting groups (multiple assignments of the devices are possible).
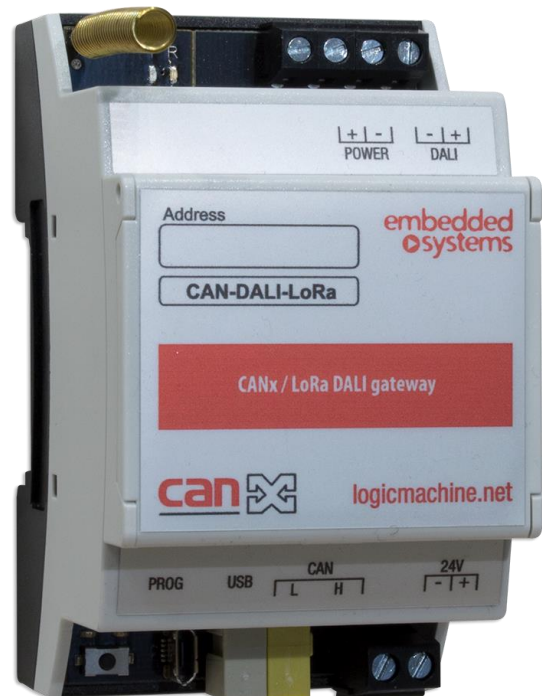
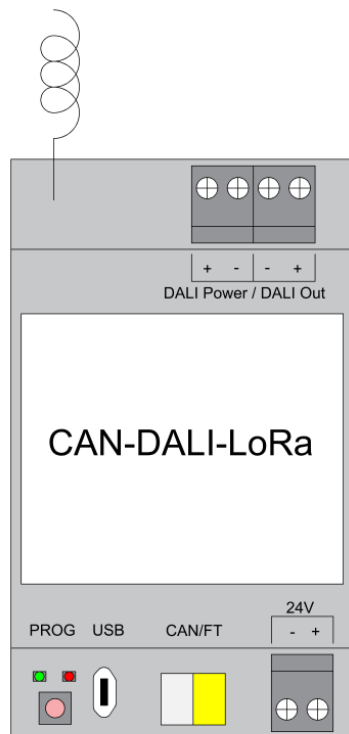| Types of product	
CAN-DALI-LoRa	CANx / LoRa DALI gateway
CAN-DALI-M3	CANx DALI gateway
Technical data	
Power supply for gateway	12-32V DC
Power consumption (at 24 V)	5.5 mA (stand-by), 20 mA (full load with LoRa)
Power supply for the DALI bus	16-18V DC
DC overvoltage protection:	50 V
Wrong wiring polarity protection	Yes
Interfaces and operating elements	
DALI output	1
Maximum count of ballasts per one CANx DALI gateway	64
USB	1 microUSB for upgrade firmware flashing
CAN FT	1
LED	1 – CPU load, 1 - Error
Programming/reset button	1
LoRa specification	
Power on transmitter	1.6-50 mW (software adjustable)
Frequency range	433-434,750 MHz
Channel bandwidth	125 / 250 / 500 kHz
Carrier frequency step	125 kHz
Spreading factor	7-12
Clamps and enclosure	
CAN FT Terminal	0.8mm2
DALI output	5 mm2
Power supply	5 mm2
Color	Gray
Dimensions	70(W)x100(H)x68(L) mm
Protection	IP20 according to EN 60529
Usage temperature	-5C ... +55C
Storage temperature	-20C ... +70C
Net weight:	86 g
Gross weight	97 g
Standards and norms compliance	

![Caution icon] **Caution**

Security advice

The installation and assembly of electrical equipment may only be performed by skilled electrician. The devices must not be used in any relation with equipment that supports, directly or indirectly, human health or life or with application that can result danger of people, animals or real value

Mounting advice

The devices are supplied in operational status. The cables connections included can be clamped to the housing if required.
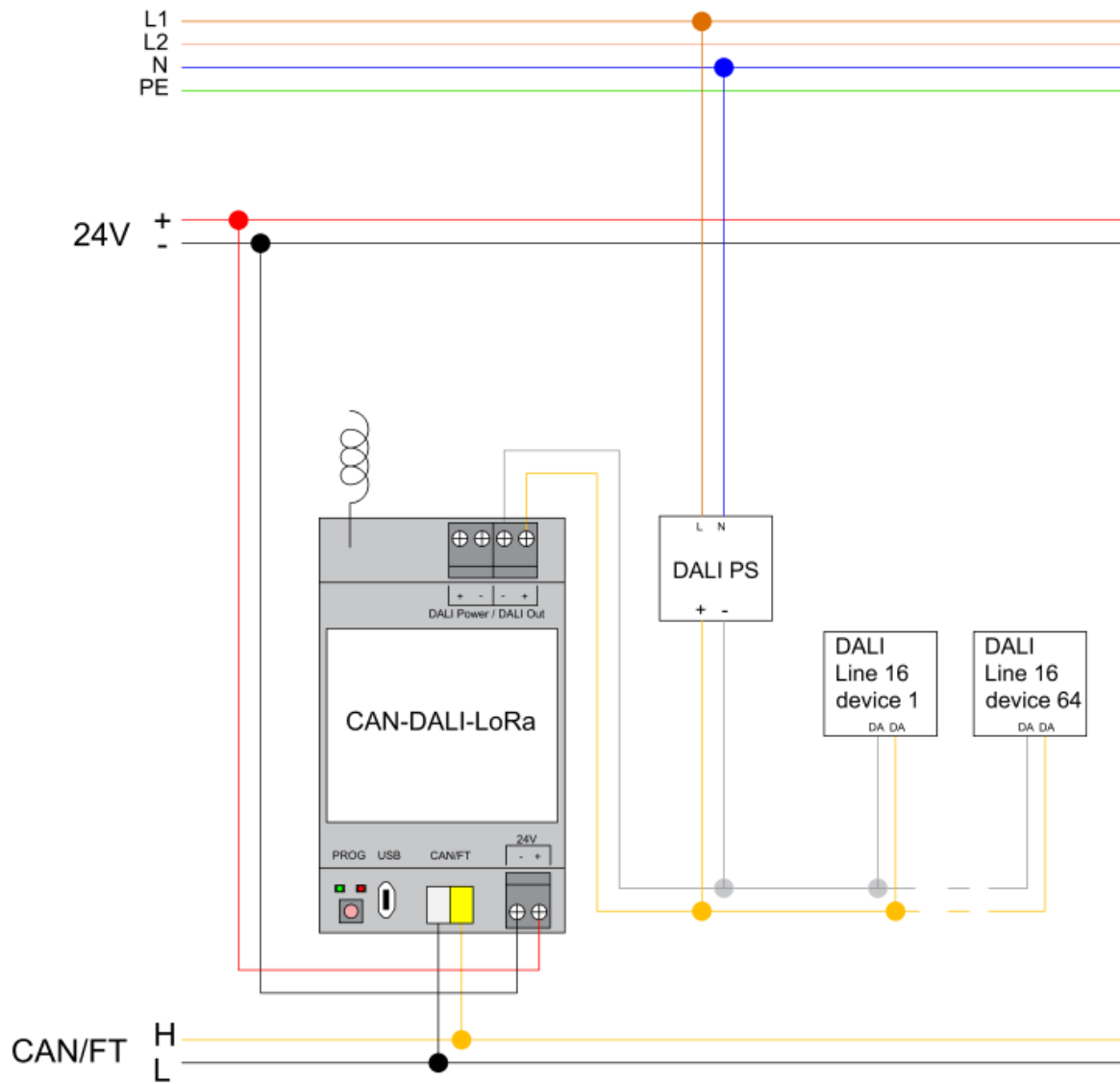
Electrical connection

The devices are constructed for the operation of protective low voltage (SELV). Grounding of device not needed. When switching the power supply on or off, power surges must be avoided.
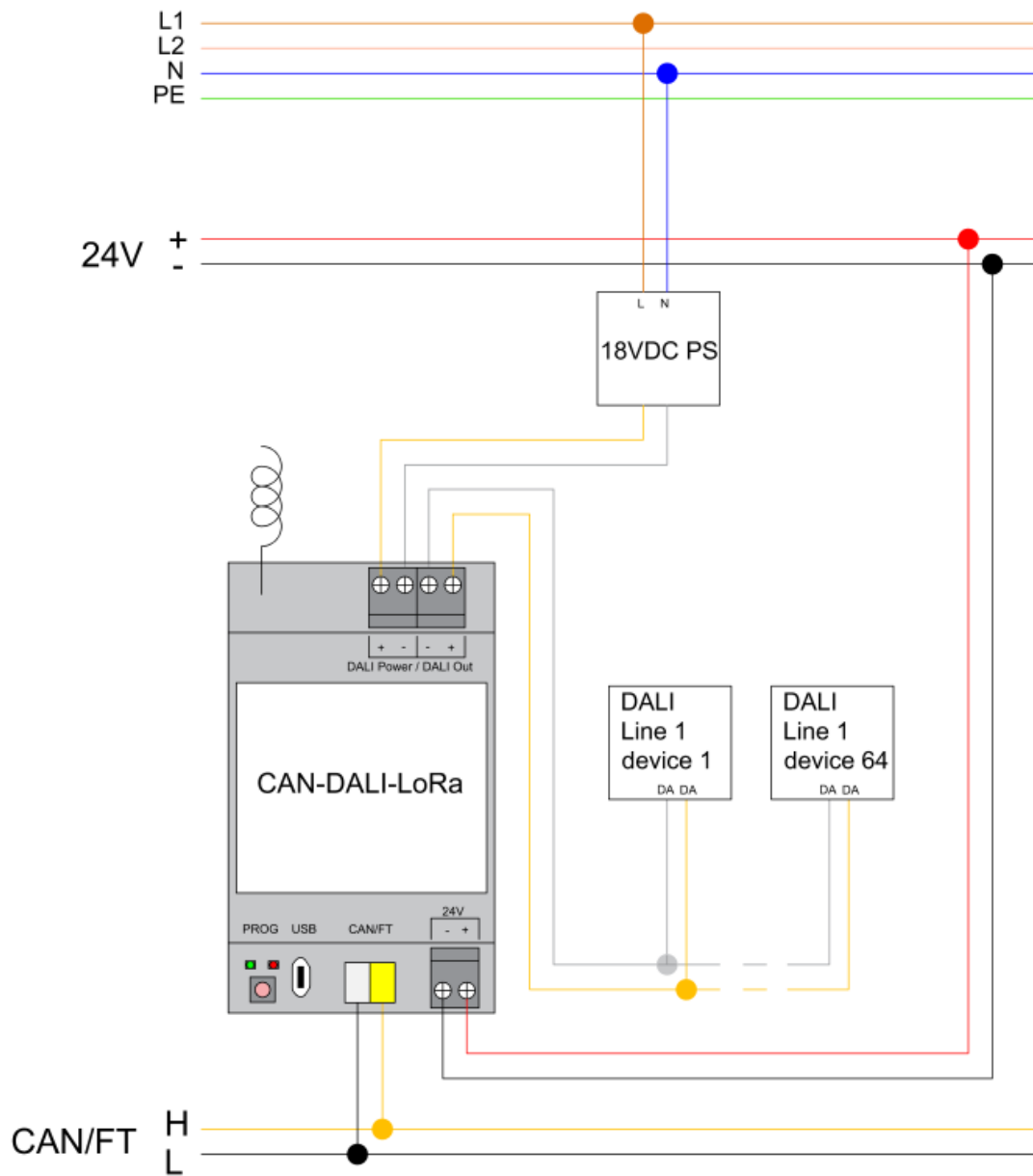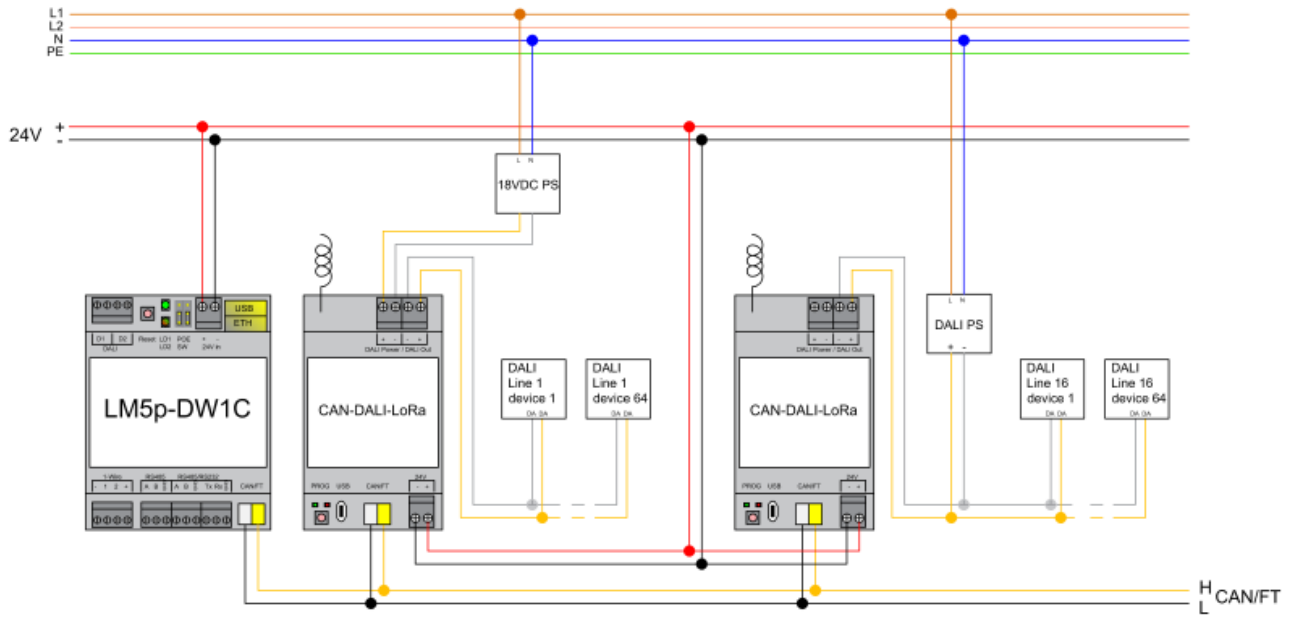
Connection diagram

External DALI power supply

Internal DALI power supply

The diagram shows:
- Power rails labeled L1, L2, N, PE
- 24V supply with + and − terminals
- 18VDC PS power supply block with L and N inputs
- CAN-DALI-LoRa device with:
  - DALI Power / DALI Out terminals (+ - - +)
  - PROG, USB, CAN/FT, and 24V (- +) connections
- DALI Line 1 device 1 (DA DA) and DALI Line 1 device 64 (DA DA)
- CAN/FT bus with H and L lines

CAN FT connection
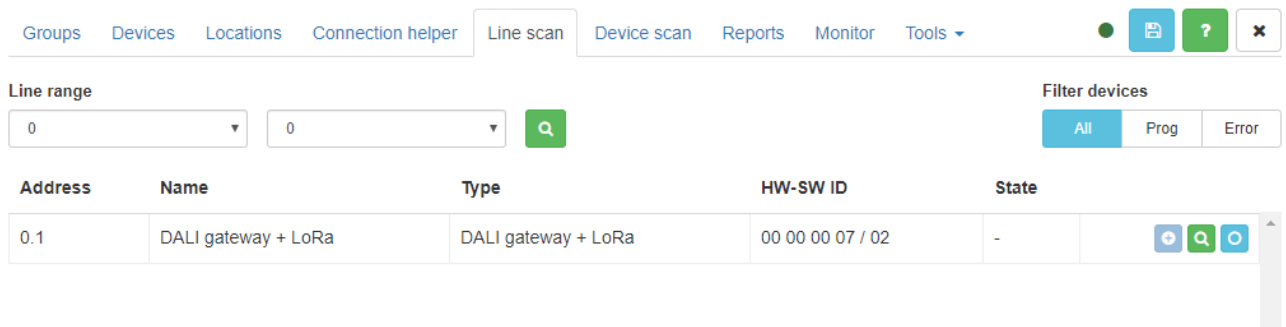
Software configuration

Two apps are mandatory for the CANx DALI Gateway:
- CANx app
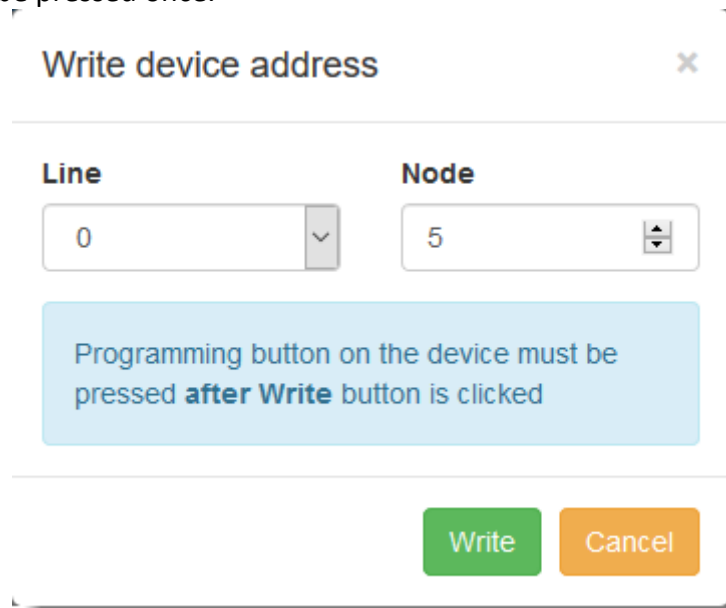- CANx DALI gateway

1.   CANx app

As first step CANx DALI gateway must be configured in CANx app
- Open the app and scan the line under *Line scan* tab. The gateway will be found as 0.1 default address.
-

	Groups	Devices	Locations	Connection helper	Line scan	Device scan	Reports	Monitor	Tools ▾				

Line range   **Filter devices**

0 ▾	0 ▾	🔍			All	Prog	Error

Address	Name	Type	HW-SW ID	State	
0.1	DALI gateway + LoRa	DALI gateway + LoRa	00 00 00 07 / 02	-	⊕ 🔍 ○

- Readdress the gateway to desire number by selecting *Tools -> Write device address*. Set the node number and press *Write*. After this procedure programming button on the DALI gateway has to be pressed once.

Write device address   ✕

Line   **Node**

| 0 ∨ | | 5 ⬍ |

> Programming button on the device must be pressed **after Write** button is clicked

Write   Cancel

- Add device to Project by pressing ⊕

- Go to Devices tab and select Configure ⚙

- Set the **optional** association to DALI power supply short circiue status.

DALI gateway + LoRa (0.5)　　　　　Device location ➕ Add ｜ - No location - ▾ ｜ ✕

| All | Enabled | Disabled |

DALI

LoRa general

LoRa messages

LoRa - CANx filtering

LoRa security

DALI power supply short circuit status (read-only) ✔

DALI power supply short circuit status (read-only)　　　　　**Flags**

| Enabled ▾ |　　　| F | T | R | W |

Group addresses ➕ Add *1 bit (boolean)*

✕ 0/0/5 DALI gateway + LoRa - DALI power supply short circuit status (read-only)

🔍

Tags

🔍 No tags set

⬇ Save and write to device　　　Save　　　Cancel

- Once done press　⬇ Save and write to device

2. CANx DALI gateway app

- Open the CANx DALI gateway app and the configured DALI gateway will be available to select.

| Gateways | Monitor |　　　　　　　　　　　　　　　✖ |

| DALI gateway (0.5) | **Short address** | **Name** | **Type** | 🔍 Scan |
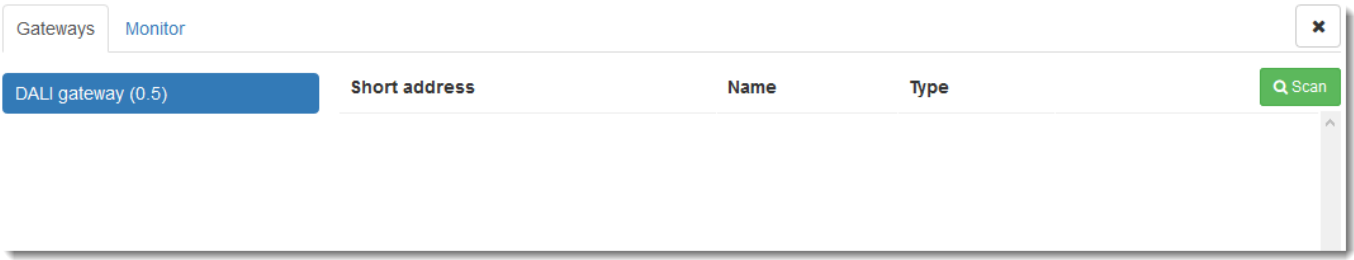
- Press **Scan** to search for connected slaves. Four options are possible

Select scan mode   ✕

Full scan, set short addresses

Full scan, clear device mapping

Full scan, keep device mapping

Partial scan, only add new devices

- Select desire option and scan will proceed. Found slaves will be available as list.

Gateways	Monitor				✕
DALI gateway (0.5)	**Short address**	**Name**	**Type**		Q Scan
	0		arc		⚙ 🗄
	1		arc		⚙ 🗄
	2		arc		⚙ 🗄

- Select [⚙] to assign group adresses to slave.
- Give a name to the slave, Select or create new group adress for the **On/Off** and **Arc value.** On preset leve is the value which will be sent to the salve on **On** command. 254 is equal to 100%**.** Mapped group addresses are available both in CANx app objects list as well as in LogicMachine objects list.
- Press [🗄] to manually control the slave arc level.

Set value of device 0   ✕

Arc value (brightness)   ☐ Mask (no change)

| 0 | ▲▼ |

Save   Cancel

Monitor

Monitor tool is used to monitor and send all commands on the DALI network.

- To send a command select DALI gateway line and a node. Select command, address type, address and respective value.

All the commands correspond to DALI specifications.

LoRa General settings

Frequency – define the frequency LoRa will operate in. Frequency should be equal on transmitter and receiver(-s).

TX power – output power of LoRa transceiver

Bandwidth – define the bandwidth of the channel. The lower the bandwidth – the lower the data rate / longer the distance. Bandwith should be equal on transmitter and receiver(-s).

Spreading factor - The basic principle of spread spectrum is that each bit of information is encoded as multiple chirps. Within the given bandwidth the relationship between the bit and chirp rate for LoRa modulation may differ between spreading factor (SF) 7 to 12. Spreading factor should be equal on transmitter and receiver(-s).

LoRa Messages

ACK mode – message acknowledgement mode
       ACK disabled - no ACK will be done (faster and less reliable communication)
       ACK enabled - each message will be acknowledged (slower, more reliable)
       ACK gateway mode – the node will retransmit ACK to the next node

ACK mode   Filter mode   Statistics ⊘

ACK mode

ACK disabled (faster, less reliable)	▾

ACK disabled (faster, less reliable)
ACK enabled (slower, more reliable)
ACK gateway mode (slower, more reliable)

Filter mode – define either to pass messages with F (Filter) flag enabled in object settings

Flags

F T R W

ACK mode | Filter mode | Statistics ⊘

Filter mode

No filtering ▼

No filtering
Pass messages without filter flag
Pass messages with filter flag

Statistics – receive statistic information to group address – source address / RSSI signal level / TX power

ACK mode | Filter mode | Statistics ⊘

Statistics

Enabled (Source, RSSI, TX power) ▼

Flags

F **T** R W

Group addresses ⊕ Add *4 byte LoRa status*

✕ 0/0/3 R6 (6 Relay outputs + LoRa) - Statistics

🔍

Tags

🔍 No tags set

Groups	Devices	Locations	Connection helper	Line scan	Device scan	Reports	Monitor	Tools ▾					●	💾	?	✕

Name or address	**Datatype**	**Tags**	All tags Any tag	**Location**	Exact Incl. sub	**Properties**				
	- All datatypes - ▼			- All locations - ▼		E R P	🔍	✕		

Address	**Name**	**Datatype**	**Tags**	**Value**	**Properties**		⬇ Import KNX project	⊕ Add
0/0/1	UIO8 (8 Universal IO ports + LoRa) - Statistics	4.5. 4 byte LoRa status		0.4 / -15 dB / 17 dBm	E R P		☰ ▤ 🗋 ✎ ✕	
0/0/2	UIO8 (8 Universal IO ports + LoRa) - Input 1	0.1. 1 bit (boolean)		0	E R P		☰ ▤ 🗋 ✎ ✕	
0/0/3	R6 (6 Relay outputs + LoRa) - Statistics	4.5. 4 byte LoRa status		0.2 / -15 dB / 17 dBm	E R P		☰ ▤ 🗋 ✎ ✕	

<u>LoRa Security</u> – define security key 1 or/and key 2 in HEX form. Up to 8 HEX characters are supported for each of the keys. Encryption keys must be equal for all LoRa devices on the same line

<u>Notification LEDs</u>

- During transmission you can see two LEDs on LoRa device

	Sending LoRa telegram
	Receiving LoRa telegram

- In case statistics is enabled on receiver device and CAN FT line is disconnected from it, both LEDs will light up (receiving telegram from sender, sending telegram with statistics).
- In case ACK is enabled, both orange and blue LEDs will light up.

<u>DALI control commands from scripts</u>

canxdali = require('applibs.canxdali')

canxdali.sendcmds(req)
Sends single or multiple DALI commands to the given gateway.
Returns number of bytes sent or nil plus error message.
This is completely asynchronous function, it adds commands
to gateway queue without waiting for returned results.

req table:
lineid - gateway line ID (number, required)
nodeid - gateway node ID (number, required)

command table:
cmd - command name (string, required)
value - command value (number, required for commands with a value)
address - DALI address (string or number, required)
addrtype - address type (string, required if address is a number)

address format:
  address can be a string with following format:
  s0..s63 - short address, from 0 to 63
  g0..g15 - group, from 0 to 15
  b - broadcast

  if address is a number then *addrtype* is required, it can be either:
  short
  group
  broadcast

Examples:

Send arc with value 0 to DALI short address 15 using gateway 0.1:

```
canxdali = require('applibs.canxdali')

  canxdali.sendcmds({
    lineid = 0,
    nodeid = 1,
    cmd = 'arc',
    address = 's15',
    value = 0,
  })
```
```
 Send multiple arc commands using gateway 1.42:
  canxdali = require('applibs.canxdali')

  canxdali.sendcmds({
    lineid = 1,
    nodeid = 42,
    cmds = {
      { cmd = 'arc', address = 's0', value = 50 },
      { cmd = 'arc', address = 's4', value = 10 },
    }
  })
```

canxdali.syncsendcmds(req)
  Similar to canxdali.sendcmds but waits for each command to complete.
  On success returns Lua table with each command result, nil plus error
  message otherwise.

canxdali.sendqueries(req)
  Similar to canxdali.syncsendcmds but checks for each command result,
  returns a table of values only for query type commands when all
  commands were successful. Useful for querying DALI device statuses.

canxdali.sethandler(type, fn)

Sets a callback to execute on a specific event.

Callback is executed for each command inside data frame separately.

type - event type (string, required):

  bus - all commands coming from bus side

  busdata - only "bus data" type commands (from other master devices)

  all - all commands coming to/from bus

fn - function to execute, or nil to remove callback (function or nil, required)

canxdali.step()

Waits for a frame or timeout, whichever happens first.

Returns frame or nil plus error message on timeout.

Frame can contain multiple commands when sent to bus.

Example (resident script):

```
if not canxdali then
  function callback(frame)
    log(frame)
  end

  canxdali = require('applibs.canxdali')
  canxdali.sethandler('bus', callback)
end

canxdali.step()
```