Nordic Automation Systems

LoRaWAN™ LUMINAIRE CONTROLLER IP68 UL2014

LoRaWAN™ Luminaire Controller is a remote controlling device for LED and HID luminaries using 0...10V analog or DALI control signal.

Control commands from/to dedicated application are to be sent/received using LoRaWAN™ network.

FW version >= 0.6.29

OVERVIEW

Universal

Supports LED and HID luminaires.

APPLICATIONS

Luminaires control

Supports customisable dimming profiles together with various configurable inputs (light intensity, movement etc.).

Array control

Allows to control an array of 32 luminaires over the DALI interface

Auxiliary light control (optional)

Relay for auxiliary light control (decorative lighting etc.).

Power monitoring

Monitor used power and luminaire burn time

FEATURES

- DALI interface
- Relay control
- Luminaire intensity control
- Thermistor input
- Digital input (e.g. movement sensor)
- Analog input (e.g. reflected light level)
- Sensor power supply
- IP68
- Surge protection
- Power metering

SPECIFICATIONS

Length:	134.0 mm
Width:	54.0 mm
Depth:	33.0 mm
Weight:	ca 260 g
Operating temperature:	-30°C ... +60°C
Power consumption:	230V -15% .. +10% <1V A
Communication range:	up to 15 km* LoRaWAN™
Tx power:	up to +20 dBm
Rx Sensitivity:	-140 dBm
MAC Layer:	LoRaWAN™
Physical Layer:	LoRa®
Body material:	ABS
IP Rating:	IP68
Communication:	LoRaWAN™

* Communication range is dependent on the location of the sensor and nearest base station.

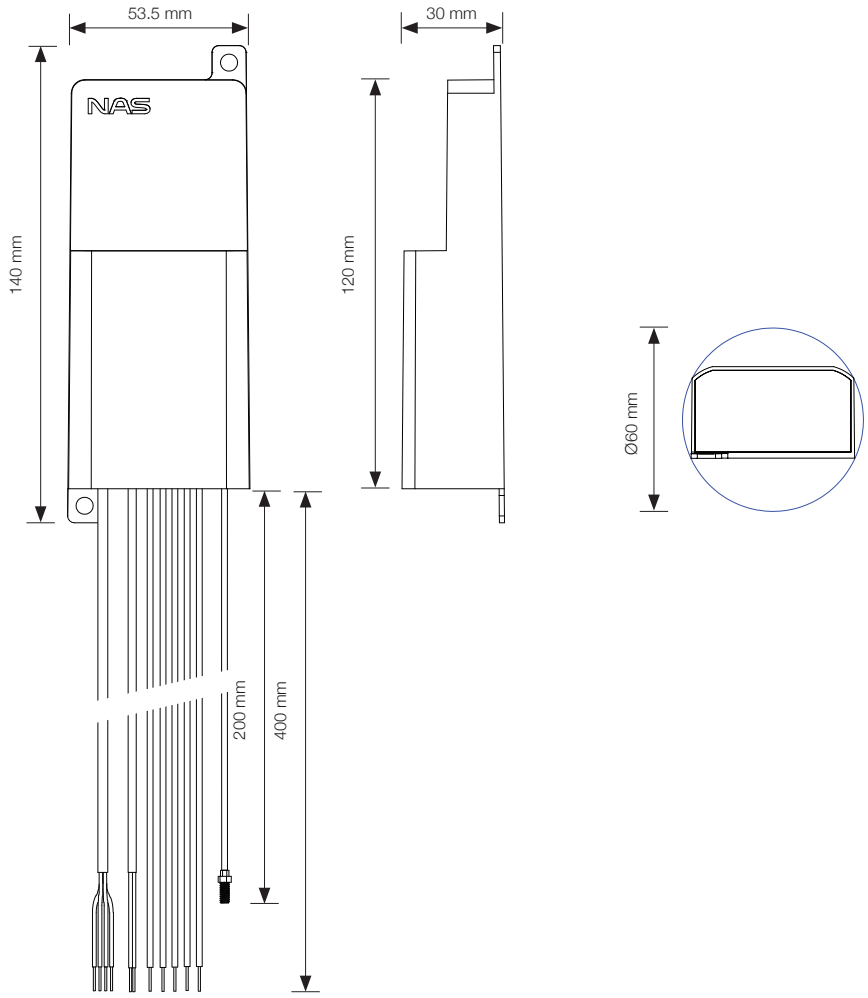
INTERFACE SPECIFICATIONS

Relay max current:	6A
Surge protection:	IEC 61000-4-5 Class 4 (4kV)
Intensity control output voltage:	0V .. 10V
DALI Interface max number of slaves on (int. power supply):	32
Thermistor input:	12V tolerant
Digital input:	12V tolerant
Analog input:	12V tolerant*
AUX relay max current:	2A (optional)
External Sensor Power supply:	Output voltage: 12V Max output current: 50mA

* Not used together with optional AUX relay.

DIMENSIONS / PACKAGING

Dimensions



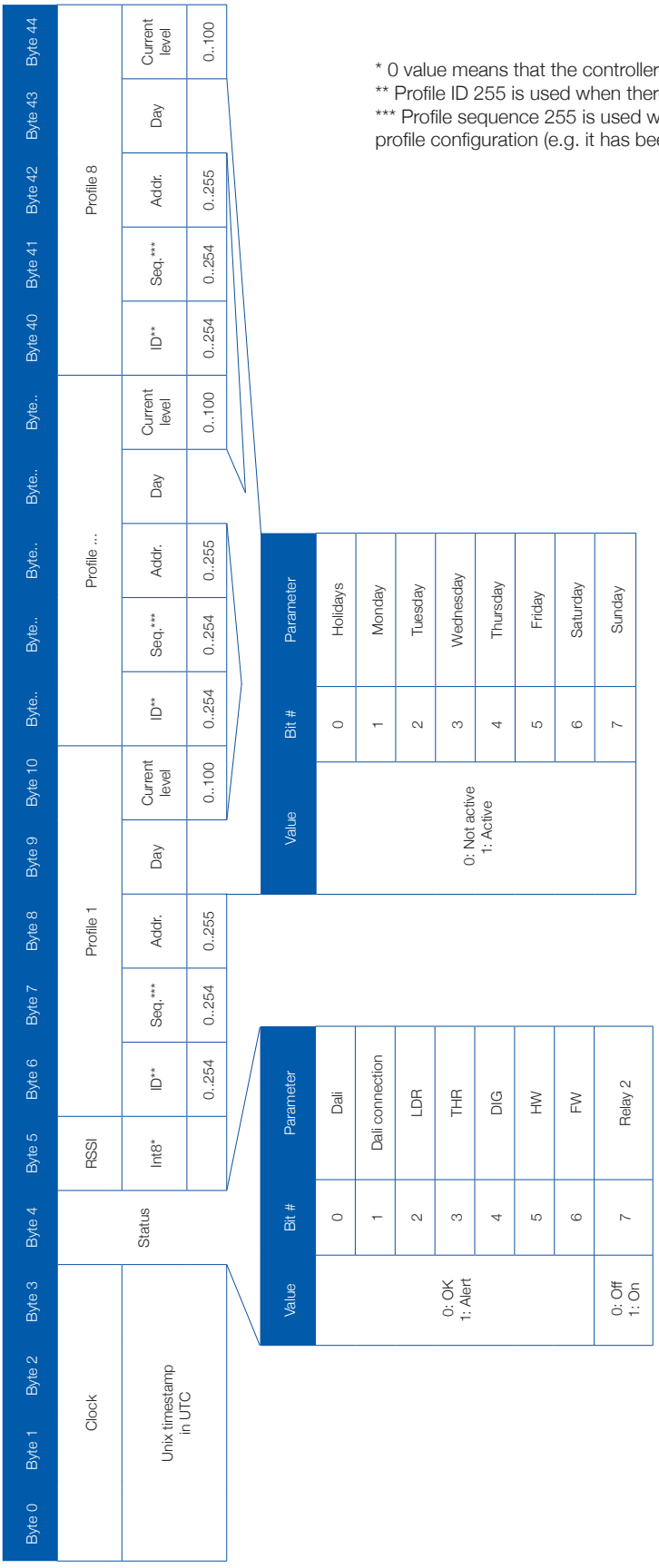
ACCESSORIES

PORT LIST

fPort	Function	Transmission	Page
24	Status	↑↑	6
25	Usage	↑↑	10
50	Configuration	↑↓	13
51	Update mode	↓↓	34
52	Multicast	↓↓	35
60	Command	↑↓	37
99	Boot/Debug	↑↑	49

For firmware version >= 0.6.29

fPort 24 Status Message



* 0 value means that the controller has not received any packages
** Profile ID 255 is used when there is no active provile and the controller is manually controlled.
*** Profile sequence 255 is used when the current actual light level is not according to the profile configuration (e.g. it has been manually overwritten).

Message sample

Message in base64

```
KOKbWQGOBBb+HjICKv7hMg==
```

Message decoded to HEX

```
28E29B59|01|8E|0416FE1E32|022AFEE132
```

Device clock `28E29B59` HEX message flip for MSB

```
599BE228
```

HEX message converted to decimal (epoch)

```
1503388200 (seconds)
```

```
Epoch time converted to date
```

```
22 August 2017 07:50:00 (UTC)
```

Status HEX message

```
01
```

Status HEX message converted to binary

```
0000 0001
```

Binary converted to statuses (LSB)

```
1 : DALI error - alert
0 : DALI Connection error - ok
0 : LDR error - ok
0 : THR error - ok
0 : DIG error - ok
0 : HW error - ok
0 : FW error - ok
0 : Relay 2 - off
```

Sensor RSSI HEX message

```
8E
```

HEX message converted to signed decimal

```
-114 (dBm)
```

Profile 1 HEX message

```
0416FE1E32
```

Profile ID in HEX

```
04
```

Profile ID converted to decimal

```
4
```

Profile sequence in HEX

```
16
```

Profile sequence converted to decimal

```
22
```

Profile address in HEX

```
FE
```

Profile HEX address converted to DALI address

```
Broadcast
```

Profile day setting in HEX

```
1E
```

Profile current light dim level converted to binary

```
00011110
```

Binary converted to statuses (LSB)

```
0 : Holidays - not active
1 : Monday - active
1 : Tuesday - active
1 : Wednesday - active
1 : Thursday - active
0 : Friday - not active
0 : Saturday - not active
0 : Sunday - not active
```

Profile current light dim level in HEX

```
32
```

Profile current light dim level converted to decimal

```
50%
```

Profile 2 HEX message

```
022AFEE132
```

Profile ID in HEX

```
02
```

Profile ID converted to decimal

```
2
```

Profile sequence in HEX

```
2A
```

Profile sequence converted to decimal

```
42
```

Profile address in HEX

```
FE
```

Profile HEX address converted to DALI address

```
Broadcast
```

Profile day setting in HEX

```
E1
```

Profile current light dim level converted to binary

```
11100001
```

Binary converted to statuses (LSB)

```
1 : Holidays - active
0 : Monday - not active
0 : Tuesday - not active
0 : Wednesday - not active
0 : Thursday - not active
1 : Friday - active
1 : Saturday - active
1 : Sunday - active
```

Profile current light dim level in HEX

```
32
```

Profile current light dim level converted to decimal

```
50%
```

fPort 25 Usage Message*

Byte 0	Byte 1	Byte ..	Byte n
Addr	Reported fields	Payload	
uint_8 0..255**			

Bit #	Parameter	Type and unit
0	Active energy total	uint32, Wh
1	Active energy instant	uint16, W
2	Load side energy total	uint32, Wh
3	Load side energy instant	uint16, W
4	Power factor instant	uint8 ***
5	System voltage	uint8, V
6	RFU	
7	RFU	

* If driver supports. Also drivers need to have short addresses assigned.
** 0xFF values measured by the controller. If driver value is available, it is preferred over internal measurement.
*** Needs to be dived by 100.

Usage parameters are reported after driver address in order of the reported fields mapping. If more than one driver are connected, then next drivers address is reported after the last parameter of the first driver and followed by the parameters in the same logic.

Message sample

Message in base64

```
BAMAAAAAAAAGAxUUAAAAAA==
```

Message decoded to HEX

```
04|03|00000000|0000|06|03|15140000|0000
```

Address 1

DALI address in HEX

```
04
```

Profile HEX address converted to DALI address

```
Single address 2
```

Reported fields in HEX

```
03
```

Reported fields converted to binary

```
00000011
```

Binary converted to statuses (LSB)

```
1 : Active energy total - sent
1 : Active energy instant - sent
0 : Load side energy total - not sent
0 : Load side energy instant - not sent
0 : Power factor instant - not sent
0 : System voltage - not sent
0 : RFU
0 : RFU
```

Active energy total `00000000` HEX message flip for MSB

```
00000000
```

HEX message converted to decimal

```
0 (Wh)
```

Current consumption `0000` HEX message flip for MSB

```
0000
```

HEX message converted to decimal

```
0 (W)
```

Address 2

DALI address in HEX

```
06
```

Profile HEX address converted to DALI address

```
Single address 3
```

Reported fields in HEX

```
03
```

Reported fields converted to binary

```
00000011
```

Binary converted to statuses (LSB)

```
1 : Active energy total - sent
1 : Active energy instant - sent
0 : Load side energy total - not sent
0 : Load side energy instant - not sent
0 : Power factor instant - not sent
0 : System voltage - not sent
0 : RFU
0 : RFU
```

Active energy total `15140000` HEX message flip for MSB

```
00001415
```

HEX message converted to decimal

```
5141 (Wh)
```

Current consumption `0000` HEX message flip for MSB

```
0000
```

HEX message converted to decimal

```
0 (W)
```

fPort 50 Configuration Message

Byte0	Byte ...	Byte n
Header	Payload	

Different headers with their respective payloads are described below

LDR Configuration

Byte 0	Byte 1	Byte 2	Byte 3
Header	Upper threshold (switch off / clear alert)	Lower threshold (switch on / create alert)	Functions
01	uint_8	uint_8	

Bit #	Function	Value
0	Switch lights on	0: disable 1: enable
1	Switch lights off	
2	RFU	
3	RFU	
4	RFU	
5	RFU	
6	RFU	
7	RFU	

Message sample

Message goal: Configure upper threshold to 160, lower threshold to 48 and enable both switching on and off of lights.

Header

Select Header HEX code

```
01
```

Upper threshold

Convert threshold value `160` to HEX

```
A0
```

Lower threshold

Convert lower threshold value `48` to HEX

```
30
```

Light switching

Function selection

```
1 : Switch lights on - enable
1 : Switch lights off - enable
0 : RFU
0 : RFU
0 : RFU
0 : RFU
0 : RFU
0 : RFU
```

Selection converted to binary

```
00000011
```

Selection converted to HEX

```
03
```

Compile message for sending (HEX)

```
01A03003
```

Control value in base64 to control after sending

```
AaAwAw==
```

THR Configuration

Byte 0	Byte 1	Byte 2	Byte 3
Header	Upper threshold (switch off / create alert)	Lower threshold (switch on / clear alert)	Functions
02	uint_8	uint_8	

Bit #	Function	Value
0	Switch lights off	0: disable 1: enable
1	RFU	
2	RFU	
3	RFU	
4	RFU	
5	RFU	
6	RFU	
7	RFU	

Message sample

Message goal: Configure upper threshold to 200, lower threshold to 120 and enable switching off of lights.

Header

Select Header HEX code

```
02
```

Upper threshold

Convert threshold value `200` to HEX

```
C8
```

Lower threshold

Convert lower threshold value `120` to HEX

```
78
```

Light switching

Function selection

```
1 : Switch lights off - enable
0 : RFU
0 : RFU
0 : RFU
0 : RFU
0 : RFU
0 : RFU
0 : RFU
```

Selection converted to binary

```
00000001
```

Selection converted to HEX

```
01
```

Compile message for sending (HEX)

```
02C87801
```

Control value in base64 to control after sending

```
Ash4AQ==
```

DIG Configuration

Byte 0	Byte 1	Byte 2	Byte 3
Header	Switch on time - s		Functions
03	uint_16		

Bit #	Function	Value
0	Switch lights on	0: disable 1: enable
1	RFU	
2	RFU	
3	RFU	
4	RFU	
5	RFU	
6	RFU	
7	RFU	

Message sample

Message goal: Configure switch time to 5 minutes and enable switching on of lights.

Header

Select Header HEX code

```
03
```

Upper threshold

Convert switch time value `300` to HEX

```
12C
```

Flip HEX value for LSB

```
2C01
```

Light switching

Function selection

```
1 : Switch lights on - enable
0 : RFU
0 : RFU
0 : RFU
0 : RFU
0 : RFU
0 : RFU
0 : RFU
```

Selection converted to binary

```
00000001
```

Selection converted to HEX

```
01
```

Compile message for sending (HEX)

```
032C0101
```

Control value in base64 to control after sending

```
AywBAQ==
```

Relay 2 Configuration

Byte 0	Byte 1	Byte 2	Byte 4	Byte 5	Byte 6	Byte *	Byte *	Byte 35	Byte 36
Header	Profile ID	Profile seq.	Day	Switching step 1		Switching step ..		Switching step 10	
05	0..254	0..254		Step time*	Switch val.	Step time*	Switch val.	Step time*	Switch val.
				0..143	00 - off FF - on	0..143	00 - off FF - on	0..143	00 - off FF - on

Value	Bit #	Parameter
0: not selected 1: Selected	0	Holidays
	1	Monday
	2	Tuesday
	3	Wednesday
	4	Thursday
	5	Friday
	6	Saturday
	7	Sunday

* Step time is 10 min increment from 00:00:00. Look at the profile step time chart for more details.

Message sample

Message goal: Apply profile 22 sequence 3 to external relay to switch on lights at 20:00 and off at 03:30 on Fridays, Saturdays, Sundays and holidays.

Header

Select Header HEX code

```
05
```

Profile ID

Convert profile ID `22` to HEX

```
16
```

Sequence

Convert profile sequence `3` to HEX

```
03
```

Day

Day selection

```
1 : Holidays - active
0 : Monday - not active
0 : Tuesday - not active
0 : Wednesday - not active
0 : Thursday - not active
1 : Friday - active
1 : Saturday - active
1 : Sunday - active
```

Selection converted to binary

```
11100001
```

Selection converted to HEX

```
E1
```

Switch step 1

Choose desired time

```
20:00
```

Convert time to offset

```
120 (see step time conversion map)
```

Convert offset to HEX

```
78
```

Choose desired switch

```
on
```

Convert switch HEX

```
FF
```

Switch step 2

Choose desired time

```
03:30
```

Convert time to offset

```
21 (see step time conversion map)
```

Convert offset to HEX

```
15
```

Choose desired switch

```
off
```

Convert switch HEX

```
00
```

Compile message for sending (HEX)

```
05|16|03|E1|78FF|1500
```

Control value in base64 to control after sending

```
BRYD4Xj/FQA=
```

Sunrise/Sunset offset configuration*

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Header	Sunrise offset - min	Sunset offset - min	Latitude		Longitude	
06	int_8	int_8	int_16**		int_16**	

*0x06FFFFFFFFFFFF will disable the sunrise/sunset calendar
**2 decimals times 100. Example : 59.425284 > 59.43 > 5943

Message sample

Message goal: Set lights to turn off 30 minutes before sunrise and turn on 30 minutes after sunrise. Set calendar to Viimsi, Estonia.

Header

Select Header HEX code

```
06
```

Sunrise offset

Convert offset `-30` to HEX

```
E2
```

Sunset offset

Convert offset `30` to HEX

```
1E
```

Latitude

Convert coordinates `65.500226` to 2 decimals

```
65.50
```

Multiply with 100 to get rid of the decimal places

```
6550
```

Convert to HEX

```
1996
```

Flip HEX value to LSB

```
9619
```

Longitude

Convert coordinates `24.833547` to 2 decimals

```
24.83
```

Multiply with 100 to get rid of the decimal places

```
2483
```

Convert to HEX

```
9B3
```

Flip HEX value to LSB

```
B309
```

Compile message for sending (HEX)

```
06E21E9619B309
```

Control value in base64 to control after sending

```
BuIelhmzCQ==
```

Status reporting configuration

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Header	Status interval* - s			
07	uint_32			

* Minimum value - 600

Message sample

Message goal: Set status reporting to 1 hour

Header

Select Header HEX code

```
07
```

Reporting interval

Convert interval `3600` to HEX

```
E10
```

Flip HEX value to LSB

```
100E0000
```

Compile message for sending (HEX)

```
07100E0000
```

Control value in base64 to control after sending

```
BxAOAAA=
```

Profile configuration

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte *	Byte *	Byte 35	Byte 36
Header	Profile ID	Profile seq.	Addr.	Day	Dimming step 1		Dimming step ..		Dimming step 10	
08	0..254	0..254	0..255		Step time*	Dimming val.	Step time*	Dimming val.	Step time*	Dimming val.
					0..143	0..100	0..143	0..100	0..143	0..100

Value	Bit #	Parameter
0: not selected 1: Selected	0	Holidays
	1	Monday
	2	Tuesday
	3	Wednesday
	4	Thursday
	5	Friday
	6	Saturday
	7	Sunday

* Step time is 10 min increment from 00:00:00. Look at the profile step time chart for more details.

Message sample

Message goal: Apply profile 22 sequence 3 to Broadcast addrress to dim the lights (01:00 to 30%, 06:00 to 80%, 10:00 to 30%, 17:00 to 80%) on Mondays, Tuesdays, Wednesdays and Thursdays.

Header

Select Header HEX code

```
08
```

Profile ID

Convert profile ID `22` to HEX

```
16
```

Sequence

Convert profile sequence `3` to HEX

```
03
```

Address

Convert `Broadcast` adress to HEX

```
FE
```

Day

Day selection

```
0 : Holidays - not active
1 : Monday - active
1 : Tuesday - active
1 : Wednesday - active
1 : Thursday - active
0 : Friday - not active
0 : Saturday - not active
0 : Sunday - not active
```

Selection converted to binary

```
00011110
```

Selection converted to HEX

```
1E
```

Dimming step 1

Choose desired time

```
01:00
```

Convert time to offset

```
6 (see step time conversion map)
```

Convert offset to HEX

```
06
```

Choose desired dimming level

```
30%
```

Convert level to HEX

```
1E
```

Dimming step 2

Choose desired time

```
06:00
```

Convert time to offset

```
36 (see step time conversion map)
```

Convert offset to HEX

```
24
```

Choose desired dimming level

```
80%
```

Convert level to HEX

```
50
```

Dimming step 3

Choose desired time

```
10:00
```

Convert time to offset

```
60 (see step time conversion map)
```

Convert offset to HEX

```
3C
```

Choose desired dimming level

```
30%
```

Convert level to HEX

```
1E
```

Dimming step 4

Choose desired time

```
17:00
```

Convert time to offset

```
102 (see step time conversion map)
```

Convert offset to HEX

```
66
```

Choose desired dimming level

```
80%
```

Convert level to HEX

```
50
```

Compile message for sending (HEX)

```
08|16|03|FE|1E|061E|2450|3C1E|6650
```

Control value in base64 to control after sending

```
CBYD/h4GHiRQPB5mUA==
```

Time settings

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Header	Device clock (Unix timestamp in UTC)			
09	uint32			

Message sample

Message goal: Set device clock to 22 August 2017 11:50:00

Header

Select Header HEX code

```
09
```

Device clock

Choose desired time

```
22 August 2017 11:50:00 (UTC)
```

Convert to epoch

```
1503402600
```

Covert to hex

```
599C1A68
```

Flip HEX value for LSB

```
681A9C59
```

Compile message for sending (HEX)

```
09681A9C59
```

Control value in base64 to control after sending

```
CWganFk=
```

Default settings

Byte 0	Byte 1	Byte 2
Header	Default light 0..100	Alert enable
0A	uint8	

Bit #	Function	Value
0	LDR alarm enable	0: off (default) 1: on
1	THR alarm enable	
2	DIG alarm enable	
3	DALI alarm enable	
4	RFU	
5	RFU	
6	RFU	
7	RFU	

Message sample

Message goal: Set default light to 0% and enable DALI alerts

Header

Select Header HEX code

```
0A
```

Default light

Convert default light dim level `0%` from decimal to HEX

```
00
```

Alerts

Alert selection

```
0 : LDR alarm - disable
0 : THR alarm - disable
0 : DIG alarm - disable
1 : DALI alarm - enable
0 : RFU
0 : RFU
0 : RFU
0 : RFU
```

Selection converted to binary

```
00001000
```

Selection converted to HEX

```
08
```

Compile message for sending (HEX)

```
0A0008
```

Control value in base64 to control after sending

```
CgAI
```

Usage reporting configuration

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Header	Usage interval* - s				System voltage**
0B	uint_32				uint_8

* Minimum value - 600
** Default 230

Message sample

Message goal: Set usage reporting to 1 hour and system voltage to 240V.

Header

Select Header HEX code

```
0B
```

Reporting interval

Convert interval `3600` to HEX

```
E10
```

Flip HEX value to LSB

```
100E0000
```

System voltage

Convert voltage `240` to HEX

```
F0
```

Compile message for sending (HEX)

```
0B100E0000F0
```

Control value in base64 to control after sending

```
CxAOAADw
```

Holiday configuration

Byte 0	Byte 1	Byte 2	Byte ..	Byte ..	Byte 49	byte 50
Header	Holiday 1		Holiday ..		Holiday 25	
0C	0..355		0..355		0..355	

Holiday are marked with days from January 1. Maximum of 25 holidays is supported. Sending in configuration for new holidays will replace old config

Message sample

Message goal: Configure January 1st, Febraury 24th, May 1st, June 23rd, August 20th and December 23rd as holidays.

Header

Select Header HEX code

```
0C
```

Holiday 1

Choose desired day

```
January 1st
```

Convert to day number in year

```
1
```

Convert value to HEX

```
01
```

Flip HEX value for LSB

```
0100
```

Holiday 2

Choose desired day

```
February 24th
```

Convert to day number in year

```
55
```

Convert value to HEX

```
37
```

Flip HEX value for LSB

```
3700
```

Holiday 3

Choose desired day

```
May 1st
```

Convert to day number in year

```
121
```

Convert value to HEX

```
79
```

Flip HEX value for LSB

```
7900
```

Holiday 4

Choose desired day

```
June 23rd
```

Convert to day number in year

```
174
```

Convert value to HEX

```
AE
```

Flip HEX value for LSB

```
AE00
```

Holiday 5

iothubdev.nasys.

Choose desired day

```
August 20th
```

Convert to day number in year

```
234
```

Convert value to HEX

```
E2
```

Flip HEX value for LSB

```
E200
```

Holiday 6

Choose desired day

```
December 23rd
```

Convert to day number in year

```
357
```

Convert value to HEX

```
165
```

Flip HEX value for LSB

```
6501
```

Compile message for sending (HEX)

```
0C|0100|3700|7900|AE00|E200|6501
```

Control value in base64 to control after sending

```
DAEANwB5AK4A4gBlAQ==
```

Clear Configuration

Byte 0	Byte 1	Byte 2	Comment
Header	Function		
FF	01	N/A	Clear LDR settings
	02	N/A	Clear THR settings
	03	N/A	Clear DIG settings
	04	Addr. 0..255	Clear profile from addr.
	05	N/A	RFU

Message sample

Message goal: Clear configuration from DALI broadcast address

Header

Select Header HEX code

```
FF
```

Function

Select clear profile function

```
04
```

Destination address

Convert DALI `Broadcast` address to HEX

```
FE
```

Compile message for sending (HEX)

```
FF04FE
```

Control value in base64 to control after sending

```
/wT+
```

fPort 51 OTA Activation Message

Byte 0	Comment
Header	
FF	Activates OTA for 2 minutes

Message sample

Message goal: Enable OTA mode

Header

Select Header HEX code

```
FF
```

Compile message for sending (HEX)

```
FF
```

Control value in base64 to control after sending

```
/w==
```

fPort 52 Configure Multicast*

* The Multicast key provisioning is not yet accoriding to LoRa Alliances “LoRaWAN Remote Multicast Setup Specification v1.0.0”

Set Multicast

Byte 0	Byte 1..Byte 4	Byte 5..Byte 20	Byte 21..Byte 36
Header	DevAddr	NwkSKey	AppSKey
0x01..0x03 Multicast group number*	HEX	HEX	HEX

* Upto 3 different sets of Multicast keys can be configured. When the same set number is used, then the old keys are overwritten.

Message sample

Message goal: Set Multicast group 1 keys to DevAddr: 82840c70,
NwkSKey: 82840c7056429b143d21974557f93a53, AppSKey: 82840c70c08494b931fe2fa6f8835c6a

Header

Select Header HEX code

```
0x01
```

DevAddr

DevAddr in HEX

```
82840c70
```

NwkSKey

NwkSKey in HEX

```
82840C7056429B143D21974557F93A53
```

AppSKey

AppSKey in HEX

```
82840C70C08494B931FE2FA6F8835C6A
```

Compile message for sending (HEX)

```
01|82840C7|082840C7056429B143D21974557F93A53|82840C70C08494B931FE2F
A6F8835C6A
```

Control value in base64 to control after sending

```
AYKEDHCChAxwVkKbFD0hl0VX+TpTgoQMcMCElLkx/i+m+INcag==
```

Clear Multicast Configuration

Byte 0	Byte 1
Header	Mulitcast group
0xFF	0x01..0x03 Clear group number 0xFF - Cleaar all groups

Message sample

Message goal: Clear Multicast group number 2

Header

Select Header HEX code

```
0xFF
```

Grouo number

Group number in HEX

```
0x02
```

Compile message for sending (HEX)

```
FF02
```

Control value in base64 to control after sending

```
/wI=
```

fPort 60 Command Message

Byte0	Byte ...	Byte n
Header	Payload	

Different headers with their respective payloads are described below

Query detailed DALI status

Byte 0	Byte 1
Header	Address
00	0..254*

* Value 254(0xFE) Queries all the automatically detected ballasts and sends the statuses as an array (Maximum of 25 items).

Message sample

Message goal: Get status of all connected DALI devices

Header

Select Header HEX code

```
00
```

Destination address

Convert DALI `Broadcast` address to HEX

```
FE
```

Compile message for sending (HEX)

```
00FE
```

Control value in base64 to control after sending

```
AP4=
```

Answer for detailed DALI status query (uplink only)

Byte 0	Byte 1		Byte 2	Byte *		Byte *	Byte 49		Byte 50
Header	1st device			.. device			25th device		
	Addr.		Status	Addr.		Status	Addr.		Status
00	0..255		0..255	0..255		0..255	0..255		0..255

Message sample

Message in base64

```
AAIEBgIMAg==
```

Message decoded to hex

```
00|02|04|06|02|0C|02
```

Header `00` decoded

```
Answer for detailed DALI status request
```

1st Device status

Profile address in HEX

```
02
```

Profile HEX address converted to DALI address

```
Single device 1
```

DALI status in HEX

```
04
```

HEX value converted to status

```
Ballast is off
```

2nd Device status

Profile address in HEX

```
06
```

Profile HEX address converted to DALI address

```
Single device 3
```

DALI status in HEX

```
02
```

HEX value converted to status

```
Lamp is burned out
```

3rd Device status

Profile address in HEX

0C

Profile HEX address converted to DALI address

Single device 6

DALI status in HEX

02

HEX value converted to status

Lamp is burned out

Set Dimming level

Byte 0	Byte 1	Byte 2	Byte *	Byte *	Byte 9	Byte 10
Header	Addr.	Dim level	Addr.	Dim level	Addr.	Dim level
01	0..255	0..100	0..255	0..100	0..255	0..100

Message sample

Message goal: Set all luminaires to 100%

Header

Select Header HEX code

```
01
```

Destination address

Convert DALI `Broadcast` address to HEX

```
FE
```

Dimming level

Convert light dimming level `100%` from decimal to HEX

```
64
```

Compile message for sending (HEX)

```
01FE64
```

Control value in base64 to control after sending

```
Af5k
```

Switch relay 2

Byte 0	Byte 1
Header	Function
02	00 - Off FF - On

Message sample

Message goal: Switch external relay on.

Header

Select Header HEX code

```
02
```

Switch

Select switching direction

```
On
```

Conver value to HEX

```
FF
```

Compile message for sending (HEX)

```
02FF
```

Control value in base64 to control after sending

```
Av8=
```

Custom DALI request

Byte 0	Byte 1	Byte 2	Byte *	Byte *	Byte 9	Byte 10
Header	Addr.	Query	Addr.	Query	Addr.	Query
03	0..255	0..255	0..255	0..255	0..255	0..255

Message sample

Message goal: Ask Max level, Min Level, Power on level, System failure level and Fade time/rate from device 36

Header

Select Header HEX code

```
03
```

1st Query

Destination address

Convert DALI `Single device 36` address to HEX

```
48
```

Convert Max level query code `161` to HEX

```
A1
```

2nd Query

Destination address

Convert DALI `Single device 36` address to HEX

```
48
```

Convert Min level query code `162` to HEX

```
A2
```

3rd Query

Destination address

Convert DALI `Single device 36` address to HEX

```
48
```

Convert Power on level query code `163` to HEX

```
A3
```

4th Query

Destination address

Convert DALI `Single device 36` address to HEX

```
48
```

Convert System failure level query code `164` to HEX

```
A4
```

5th Query

Destination address

Convert DALI `Single device 36` address to HEX

```
48
```

Convert Fade time/rate query code `165` to HEX

```
A5
```

Compile message for sending (HEX)

```
03|48A1|48A2|48A3|48A4|48A5
```

Control value in base64 to control after sending

```
A0ihSKJIo0ikSKU=
```

Answer for custom DALI request

Byte 0	Byte 1	Byte 2	Byte 3	Byte *	Byte *	Byte *	Byte 47	Byte 48	Byte 49
Header	Addr.	Query	Answer	Addr.	Query	Answer	Addr.	Query	Answer
03	0..255	0..255	0..255	0..255	0..255	0..255	0..255	0..255	0..255

Message sample

Message in base64

```
BEih/kiiqEij/kik/kilBw==
```

Message decoded to hex

```
04|48A1FE|48A2A8|48A3FE|48A4FE|48A507
```

Header `04` decoded

```
Answer for custom DALI request
```

1st Answer

Profile address in HEX

48

Profile HEX address converted to DALI address

Single device 36

Query in HEX

A1

HEX value converted to decimal

161

Decimal value translated to query

Max level

Answer in HEX

FE

HEX value converted to decimal

254

2nd Answer

Profile address in HEX

48

Profile HEX address converted to DALI address

Single device 36

Query in HEX

A2

HEX value converted to decimal

162

Decimal value translated to query

Min level

Answer in HEX

A8

HEX value converted to decimal

168

3rd Answer

Profile address in HEX

48

Profile HEX address converted to DALI address

Single device 36

Query in HEX

A3

HEX value converted to decimal

163

Decimal value translated to query

Power on level

Answer in HEX

FE

HEX value converted to decimal

254

4th Answer

Profile address in HEX

48

Profile HEX address converted to DALI address

Single device 36

Query in HEX

A4

HEX value converted to decimal

164

Decimal value translated to query

Failure level

Answer in HEX

FE

HEX value converted to decimal

254

5th Answer

Profile address in HEX

```
48
```

Profile HEX address converted to DALI address

Single device 36

Query in HEX

```
A5
```

HEX value converted to decimal

165

Decimal value translated to query

Fade time/rate

Answer in HEX

```
07
```

HEX value converted to status

<0.7s / 45 steps/s

Custom DALI command

Byte 0	Byte 1	Byte ..	Byte 50
Header	Data	Data	Data
04	0..255	0..255	0..255

Message sample

Message goal: Set Single device 1 minimum light level to 127.

Header

Select Header HEX code

```
04
```

DALI command

DALI command in HEX

```
027F0321032B
```

Compile message for sending (HEX)

```
04027F0321032B
```

Control value in base64 to control after sending

```
BAJ/AyEDKw==
```

Request input statuses

Byte 0	Byte 1
Header	Request
06	FF - All inputs

Message sample

Message goal: Request all input values

Header

Select Header HEX code

```
06
```

Request

All values

```
FF
```

Compile message for sending (HEX)

```
06FF
```

Control value in base64 to control after sending

```
Bv8=
```

Answer for custom DALI request

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Header	DIG	Value	LDR	Value	THR	Value
03	01	00 - low 01 - high FF - n/a	02	0..254 FF - n/a	03	0..254 FF - n/a

Message sample

Message in base64

```
BgEAAuQD/w==
```

Message decoded to hex

```
06010002E403FF
```

Header `06` decoded

```
Answer for input value request
```

Digital input `01` value in HEX

```
00
```

HEX value tranlated to status

```
signal low
```

LDR input `02` value in HEX

```
E4
```

HEX value converted to decimal

```
234
```

THR input `03` value in HEX

```
FF
```

HEX value translated to status

```
n/a
```

fPort 99 Boot/Debug Message

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 12	Byte 12
Header (HEX)	Payload												
0x00 Boot	Serial (HEX)					Firmware (HEX)			Clock - Unix timestamp UTC (uint)			HW (HEX)	Opt. (HEX)
0x01 Shutdown													
0x10 Error code	Error code (uint)												

Error codes

Code	Comment
02	Multiple unconfigured drivers detected

Message sample

Message in base64

```
AP8AEEcABAXDFEFZBQA=
```

Message decoded to hex

```
00FF001047000405C3144159 0500
```

Header `00` decoded

```
Boot message
```

Device serial `FF001047` HEX message flip for MSB

```
471000FF
```

Firmware version

Major version in HEX

```
00
```

HEX value converted to decimal

```
0
```

Minor version in HEX

```
04
```

HEX value converted to decimal

```
4
```

Patch version in HEX

```
05
```

HEX value converted to decimal

```
5
```

Device clock `C3144159` HEX message flip for MSB

```
594114C3
```

HEX message converted to decimal (epoch)

```
1497437379 (seconds)
```

```
Epoch time converted to date
```

```
14 June 2017 10:49:39 (UTC)
```

Hardware configuration

Hardware setup in HEX

```
05
```

HEX value converted to setup

```
DALI & 0..10V & NC Realy
```

Optional features in HEX

```
00
```

HEX value converted to optional features

```
None
```

Cells marked with X contain address bits

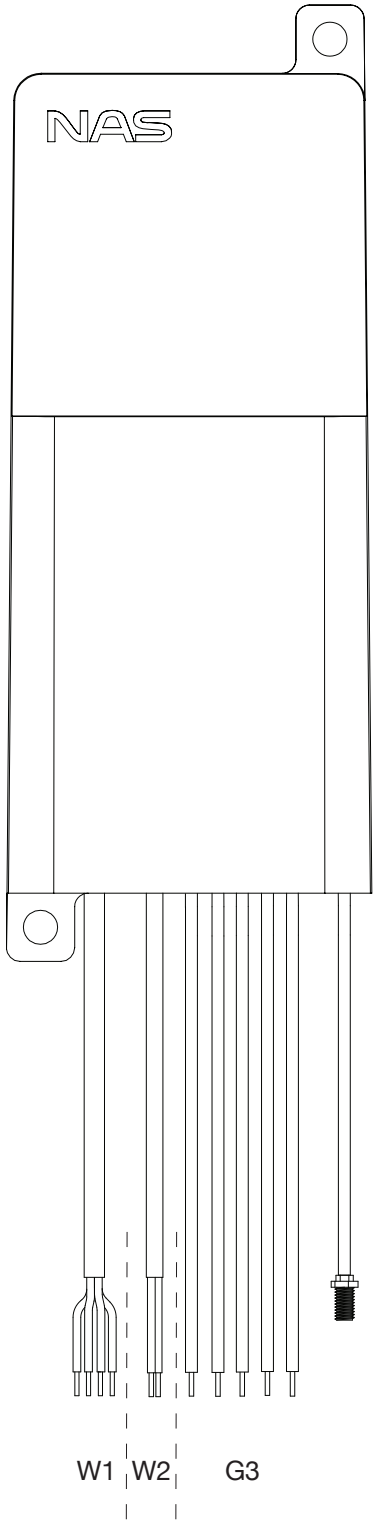
DALI Address / Destination mapping

Bit #	Single address	Group address	Broadcast
Addr.	0-63	0-15	n/a
0	0	0	0
1	x	x	1
2	x	x	1
3	x	x	1
4	x	x	1
5	x	0	1
6	x	0	1
7	0	1	1

PROFILE STEP TIME CHART

0 - 00:00
1 - 00:10
2 - 00:20
3 - 00:30
4 - 00:40
5 - 00:50
6 - 01:00
7 - 01:10
8 - 01:20
9 - 01:30
10 - 01:40
11 - 01:50
12 - 02:00
13 - 02:10
14 - 02:20
15 - 02:30
16 - 02:40
17 - 02:50
18 - 03:00
19 - 03:10
20 - 03:20
21 - 03:30
22 - 03:40
23 - 03:50
24 - 04:00
25 - 04:10
26 - 04:20
27 - 04:30
28 - 04:40
29 - 04:50
30 - 05:00
31 - 05:10
32 - 05:20
33 - 05:30
34 - 05:40
35 - 05:50
36 - 06:00
37 - 06:10
38 - 06:20
39 - 06:30
40 - 06:40
41 - 06:50
42 - 07:00
43 - 07:10
44 - 07:20
45 - 07:30
46 - 07:40
47 - 07:50
48 - 08:00
49 - 08:10
50 - 08:20
51 - 08:30
52 - 08:40
53 - 08:50
54 - 09:00
55 - 09:10
56 - 09:20
57 - 09:30
58 - 09:40
59 - 09:50
60 - 10:00
61 - 10:10
62 - 10:20
63 - 10:30
64 - 10:40
65 - 10:50
66 - 11:00
67 - 11:10
68 - 11:20
69 - 11:30
70 - 11:40
71 - 11:50
72 - 12:00
73 - 12:10
74 - 12:20
75 - 12:30
76 - 12:40
77 - 12:50
78 - 13:00
79 - 13:10
80 - 13:20
81 - 13:30
82 - 13:40
83 - 13:50
84 - 14:00
85 - 14:10
86 - 14:20
87 - 14:30
88 - 14:40
89 - 14:50
90 - 15:00
91 - 15:10
92 - 15:20
93 - 15:30
94 - 15:40
95 - 15:50
96 - 16:00
97 - 16:10
98 - 16:20
99 - 16:30
100 - 16:40
101 - 16:50
102 - 17:00
103 - 17:10
104 - 17:20
105 - 17:30
106 - 17:40
107 - 17:50
108 - 18:00
109 - 18:10
110 - 18:20
111 - 18:30
112 - 18:40
113 - 18:50
114 - 19:00
115 - 19:10
116 - 19:20
117 - 19:30
118 - 19:40
119 - 19:50
120 - 20:00
121 - 20:10
122 - 20:20
123 - 20:30
124 - 20:40
125 - 20:50
126 - 21:00
127 - 21:10
128 - 21:20
129 - 21:30
130 - 21:40
131 - 21:50
132 - 22:00
133 - 22:10
134 - 22:20
135 - 22:30
136 - 22:40
137 - 22:50
138 - 23:00
139 - 23:10
140 - 23:20
141 - 23:30
142 - 23:40
143 - 23:50

WIRING

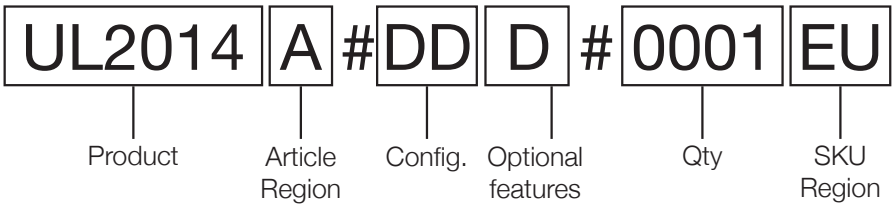


Amount of wires in each section is dependant of the ordered configuration

	Description
W1 - Wire 1:	
Brown	Normally closed or open AC 230V relay output to driver
Blue	230V AC neutral output to driver
Grey	DALI- control signal to driver
Black	DALI+ control signal to driver
W2 - Wire 3:	
Brown	AC 230V input
Blue	230V AC neutral input
G3 - Group 3	
Purple	0..10V+ control signal to driver*
White	0..10V- control signal to driver*
Red	+12V output for sensor powering
Pink	THR in
Yellow	LDR in
Green	DIG in
Red/Grey	Ground for DIG sensor

* Manufactured after 05.04.2019

ORDERING INFORMATION



Product/SKU	Package qty	IP Rating
UL2014x#xx#0001xx	1	IP68

Article region	SKU region	Frequency
A	EU	868 MHz
B	AU	922 MHz
C	US	915 MHz
D	AS	923 MHz
F	KR	922 MHz
I	IN	866 MHz

Model	Description	Optional features (Choose only 1 from each column)			
UL2014	**DD** - DALI only	**T** - THR	**D** - DIG	**L** - LDR **O** - Relay 2	**X*** - custom request
	DC - DALI & NC relay				
	AC - 0..10V & NC relay				
	UC - DALI & 0..10V & NC relay				

* Custom request must be verified with sales before ordering. On order X must be accompanied with the description of agreed functionality.

CONTACT INFORMATION

Nordic Automation Systems AS

www.nasys.no

info@nasys.no

REVISION HISTORY

1.0 - First version

1.1 - Added configuration details.

1.2 - Added configuration details.

1.3 - Updated communication protocol

1.4 - Wiring diagram and ordering information

1.5 - fw 0.5.0 support, power metering

2.0 - Split from ULw datasheet

2.1 - Updated to UL2014

2.2 - Analog wiring change. New housing

2.3 - New format for usage message. Calendar user configurable. Possible to request input statuses.

2.4 - Multicast configuration added

All content contained herein is subject to change without notice. Nordic Automation Systems reserves the right to change or modify the content at any time.

Save trees
Print only if necessary