

# Remoticom ZSC160

## Instruction Manual



The information contained in this document is subject to change with or without notice. For this reason, always make sure you are using the latest version available at [support.remoticom.com](https://support.remoticom.com). If you have any additional questions, please contact your point of contact.

# Table of Contents

Table of Contents.....	2
1. Revision history .....	3
2. General overview .....	4
3. Device specifications .....	5
a. Communication .....	5
b. DALI Interface .....	5
c. Power supply .....	5
d. GPS.....	6
e. Tilt .....	6
f. LEDs .....	6
g. Casing .....	6
h. External influences.....	6
i. Certification .....	6
j. Firmware.....	6
4. Installation .....	7
5. Software.....	8
6. Operation modes.....	9
a. ASTRO mode .....	9
b. MAINS mode.....	9
c. STATIC mode.....	10
7. Default configuration values.....	11
8. How to encode downlink data .....	12
a. General commands.....	12
b. ZSC configuration .....	12
c. Week schedule steps .....	14
d. Exception step configuration .....	15
e. Fragmentation session .....	16
9. How to decode uplink data .....	20
a. Sensor info .....	20
b. Dali light state.....	21
c. Alarms .....	23
d. ZSC configuration .....	24
e. Session response .....	26

## 1. Revision history

Version	Description	Editor	Date
0.1	Initial draft	J. Mariën	2023-02-13
1.0	Initial release	P. Poncelet	2023-03-31
1.1	Correction example payload	J. Mariën	2023-09-18
1.2	Changes firmware version 0.4.3	J. Mariën	2023-10-05
1.3	Correction example schedule	J. Mariën	2023-11-23
1.4	Changes firmware version 1.0.1	J. Mariën	2024-03-18

## 2. General overview

The ZSC160 is a LoRaWAN Connected Smart Light Controller which can be mounted on a streetlight that supports Zhaga book 18. The ZSC160 uses LoRa technology to communicate with a backend application through its dedicated API, which is extensively described in this document.

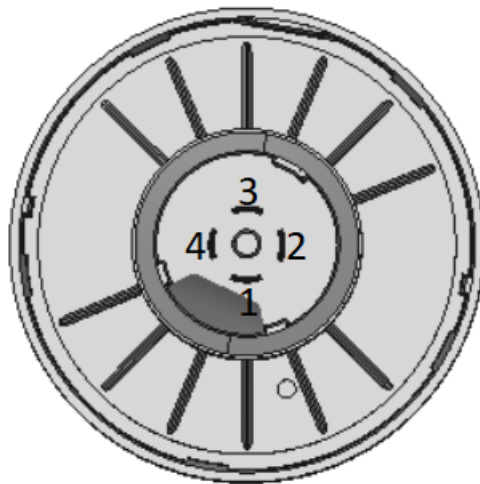
### 3. Device specifications

#### a. Communication

- The ZSC160 is designed according to the LoRaWAN 1.0.4 specification.
- The device supports LoRaWAN regions EU868, US915. The LoRaWAN region is commissioned during production and cannot be changed.
- The device uses LoRaWAN OTA activation. During the join procedure the sensor starts with a random delay and uses an incremental back-off sequence.
- The device enables ADR to let the server optimize to the best data rate.
- The ZSC160 is a Class C LoRaWAN device.
- The device sends regular DeviceTimeReq MAC messages after join to synchronise the GPS time from the server in case the GPS module has no fix.
- The device can re-join when it detects the server connection is lost.
- The sensor may send confirmed messages to determine if it is still connected to a LoRaWAN Network Server. After a new join request the server should assume the sensor has lost any or all settings.

#### b. DALI Interface

- The device is connected by Zhaga Book 18 connector.
- The pinout for the Zhaga connector is 1=24V, 2=Dali-, 3=Dali+. 4 is not used.



- Compatible with other DALI type B devices.
- The device does not have an integrated DALI bus power supply.

#### c. Power supply

- The device is powered over the Zhaga connector.
- Operating range 24V  $\pm$ 2.4V.
- Max current consumption on the Dali bus = 2mA (DALI type A device)

#### d. GPS

- The GPS location has a precision of  $\pm 10$  meter.
- When not obstructed by e.g., trees and buildings the device will have a GPS fix in less than 10 minutes.

#### e. Tilt

- After 120 minutes of operation the current tilt value is saved as its default tilt.
- After 180 minutes the device can send tilt alarms to the portal.
- A tilt alarm is sent to the portal if the angle differs 15 degrees from the calibrated angle for more than 15 minutes.
- When a tilt alarm is sent, the current tilt value will be saved as the new default tilt.
- The default values of these parameters can be configured via a LoRaWAN Downlink message.

#### f. LEDs

- The device contains one green LED, one blue LED, one red LED.
- During OTAA join procedure the red LED will blink.
- During LoRaWAN TX the green LED will blink.
- The blue LED is currently not used.
- After 1 hour of operation the LEDs are disabled.

#### g. Casing

- Rated IP65.
- Diameter of 80mm.
- Height 40mm.
- The material of the casing is polycarbonate.

#### h. External influences

- Operating Temperature range is  $-20^{\circ}\text{C}$  to  $60^{\circ}\text{C}$ .
- Operating Air humidity range (RH) is 0 to 100%.
- The device is compatible with IEC 61000-4-2 (ESD): Air  $\pm 15\text{kV}$ , Contact  $\pm 8\text{kV}$ .
- Electrical insulation class III.

#### i. Certification

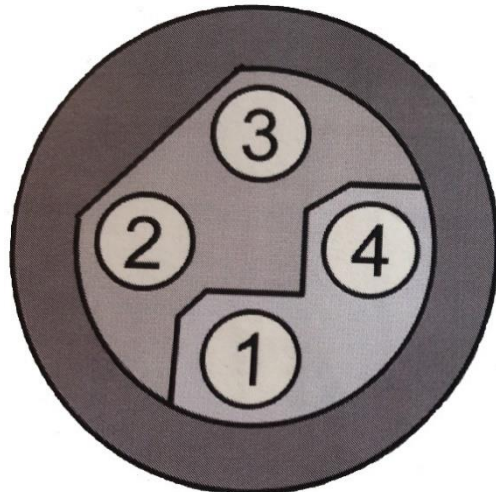
- EU DoC (CE) certification.
- RoHs compliant.

#### j. Firmware

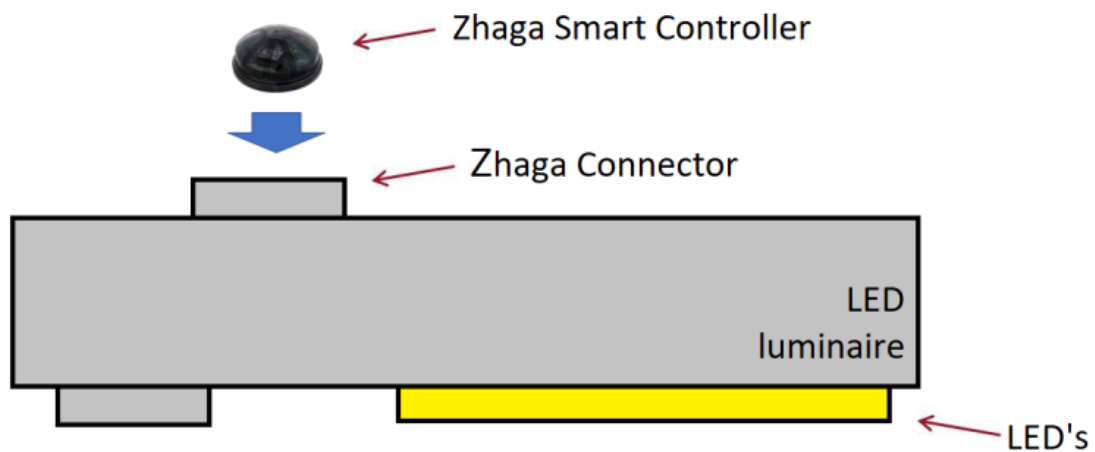
- LoRaWAN firmware over the air (FOTA) is not supported.

## 4. Installation

1. Make sure that the Zhaga connector (Zhaga book 18) is connected correctly to the luminaire:
  - a. pin 1 = Aux24v m
  - b. pin 2 = Dali -
  - c. pin 3 = Dali +



2. Connect the Zhaga Smart Controller (ZSC 160) to the Zhaga connector on top of the luminaire, by inserting the ZSC and turning it slightly clockwise.



## 5. Software

With every sensor, or batch of sensors, a csv file will be provided with the keys needed to communicate with the device. If you did not receive such a file, please contact your Remoticom point of contact.

Each device has a unique and fixed DevEUI and a fixed crypto-random AppKey. All devices have the same JoinEUI. The IEEE\_OPTION\_OUI is used as a prefix for the DevEUI and JoinEUI. During production the sensor generates the DevEUI and AppKey. These keys are not readable from any debug port.

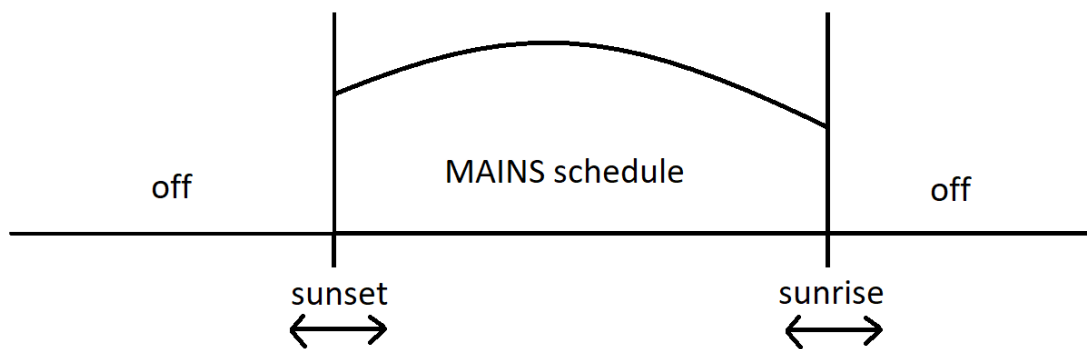


## 6. Operation modes

Regardless of the mode that the device is in, on startup the Dali level will be set to 254 (max value). A reboot of the device will also always retain the latest configuration. There are 3 possible operation modes:

### a. ASTRO mode

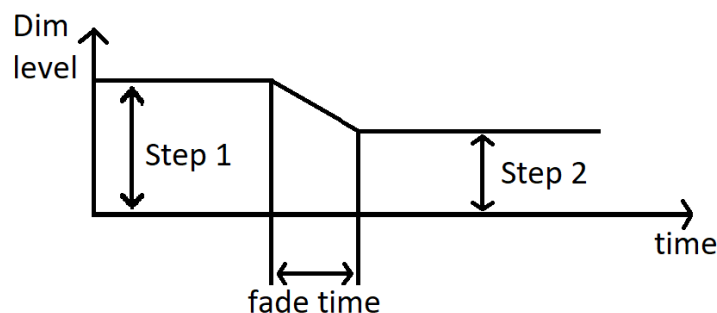
In ASTRO mode the device will calculate a sunrise and sunset using GPS position and date. It will use the OFFICIAL sunset/sunrise which is 0 degrees below the horizon. The light will be dimmed according to its (MAINS) planning during the night.



⇒ Sunset and sunrise times can be altered by enabling daylight saving time and sunset/sunrise offsets

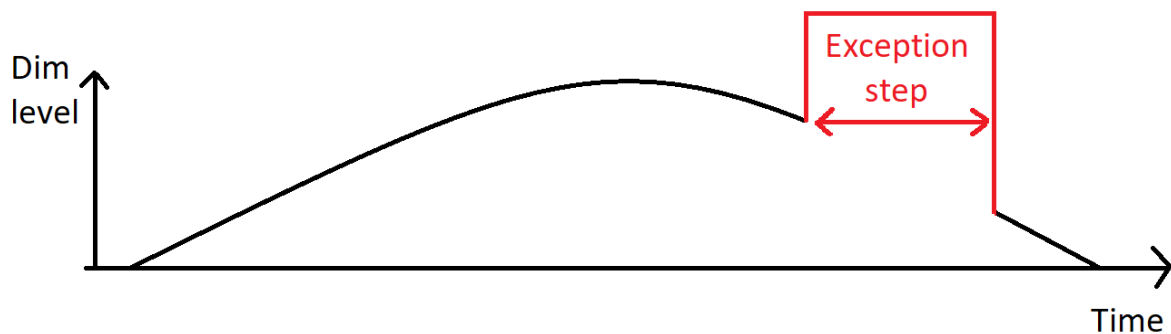
### b. MAINS mode

In MAINS mode the light will be dimmed according to its dim planning. The time it takes to transition between 2 dim steps is called the fading time.



A dim planning is defined as a weekly repeating schedule and a non-repeating exception schedule. Each can have a maximum of 50 steps.

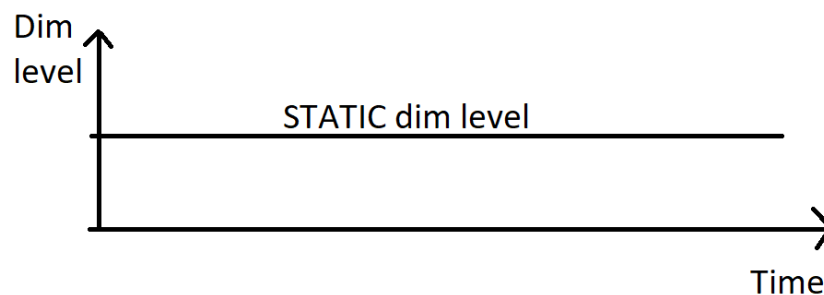
- One step in the week-schedule consists of one dim step and a relative offset of the time since the last Monday **UTC midnight** (resolution in minutes). A step is active until the next step in the planning is reached. One step in the exception-schedule consists of one dim step, an absolute UNIX timestamp (in seconds) and a duration (in seconds) for this exception step to be active. The maximum duration of a single step is 18 hours. An exception step never repeats. When an exception step is active it overrides any week schedule.



The week schedule can be compensated with one hour for daylight saving time. The server sends a daylight-saving time start and end UNIX timestamp so that the sensor knows when daylight saving time is active.

### c. STATIC mode

In STATIC mode the light will be dimmed according to its set STATIC level (0% - 100%).



## 7. Default configuration values

Here you can find an overview of the default configuration values:

Variable	Value
Tilt angle	15 °
Tilt delay	180 minutes
Tilt duration	15 minutes
LoRa heartbeat interval	180 minutes
DALI report interval	60 minutes
Sunrise offset	0
Operating mode	STATIC
Static DIM level	0

## 8. How to encode downlink data

### a. General commands

- Port: 1
- Data:
  - Uint8\_t command:
    - 0x01: reboot the sensor
    - 0x02: request a LoRa info uplink
    - 0x03: request a Dali info uplink
    - 0x04: request a ZSC configuration uplink
    - 0x05: request a GPS alarm uplink
    - 0x07: calibrate tilt sensor
  - Example:  
Send **01** to reboot the sensor.

### b. ZSC configuration

- Port: 2
- Data:

The data starts with a one-byte length field indicating the total length (without the length field). Next is a sequence of tag and data fields. The tag indicates the format of the data.

Len	Tag	Data	Tag	data
-----	-----	------	-----	------

Supported tags:

- 0x00: ZSC\_CFG\_MODE  
data:
  - 0x01: MAINS
  - 0x02 : ASTRO
  - 0x04 : STATIC
- 0x01: ZSC\_CFG\_DALI  
data:
  - uint8\_t Dali level between 0 and 254
  - uint8\_t fade time ( $\leq 15$ ). The actual fade time can be calculated as  $2^{\frac{\text{fade time}}{2}}$  in seconds, fade time set to 15 for example equates to roughly 90.5 seconds.
- 0x02: ZSC\_CFG\_SUN  
data:
  - int8\_t sunrise offset in minutes
  - int8\_t sunset offset in minutes

- 0x03: ZSC\_CFG\_DST  
data:
  - uint32\_t daylight saving time start (UNIX time in seconds)
  - uint32\_t daylight saving time stop (UNIX time in seconds)
- 0x04: ZSC\_CFG\_TILT  
data:
  - uint8\_t angle in degrees
  - uint8\_t delay in minutes
  - uint8\_t duration in minutes
- 0x05: ZSC\_CFG\_INT  
Data:
  - uint16\_t heartbeat interval in minutes (max 1 day = 60\*24)
  - uint16\_t Dali interval in minutes (max 1 day)
- Example:  
**05 00 04 01 3C 0F:** Set the mode to static, the Dali level to 60 and the fade time to 15 minutes.

### c. Week schedule steps

Send a week schedule configuration. You can send a maximum of 50 bytes, otherwise you will have to make use of the fragmentation protocol. The amount of week steps you can add to the planning is limited to 50 steps.

Before the starting point of the first step in the week schedule, the dim level of the last step will be used. Keep this in mind when making a dim schedule for your devices.

- Port: 3
- Data:

The data starts with 2 a bytes “unique” schedule id and a one-byte “Nsteps” field indicating the number of steps.

The nsteps can be zero to clear the schedule.

Id	Nsteps	Step 1	Step 2	Step 3
----	--------	--------	--------	--------

Each step contains the following data:

- uint16\_t time (relative offset of the time in minutes since the last Monday UTC midnight.)
- uint8\_t value: dim level between 0 and 254  
This DIM level translates into light output according to the following formula:

$$\text{Light output}(\text{DIM level}) = 10^{\frac{\text{DIM level}-1}{253/3}-1} \%$$

- Example:

**07 5A 02 70 08 E5 40 0B 00:**

Configure 2 steps in the week schedule: dim level 229 (≈50% light output using the formula above) starting 36 hours after 00:00 UTC Monday, dim level 0 starting 48 hours after 00:00 UTC Monday. Effectively have the dim level be at 50% from Tuesday noon to midnight and 0% at all other times.

## d. Exception step configuration

Send an exception schedule. You can send a maximum of 50 bytes, otherwise you will have to make use of the fragmentation protocol. The amount of exception steps you can add to the exception planning is limited to 50 steps.

Each exception step contains the start time and duration of the exception step. When an exception step is active, it will override the regular week planning for the duration of the step. Since the start time is an absolute value/timestamp, this cannot be set as a recurrent event.

- Port: 4
- Data:

The data starts with a 2 bytes “unique” schedule id and a one-byte “Nsteps” field indicating the number of steps. The number of steps can be zero to clear the schedule.

Id	Nsteps	Step 1	Step 2	Step 3
----	--------	--------	--------	--------

Each step contains the following data:

- uint32\_t time (UNIX time in seconds)
- uint8\_t value: dim level between 0 and 254  
This DIM level translates into light output according to the following formula:

$$\text{Light output}(\text{DIM level}) = 10^{\frac{\text{DIM level}-1}{253/3}}\%$$

- uint16\_t duration in seconds
- Example:  
**00 5A 01 00 CF EA 63 00 84 A8:** Add the exception step that from 14/02/2023 12:00 to 14/02/2023 23:59 the dim level is set to 0.

## e. Fragmentation session

A dedicated protocol that can send multiple lora downlink messages to avoid the limit of 50 bytes. This is based on the lora fragmentation protocol with some changes to simplify it.

It is used to configure the week planning, the exception planning and the ZSC configuration.

Upon completion of a fragmentation session, an uplink is sent in response. See section 9.e for more information about this uplink.

### Set up a fragmentation session:

Setup a new fragmentation session with its corresponding settings.

Session should be set to a random byte (or counter value) for every new session.

- NbFrag indicates the number of downlink fragments that will be sent.
- FragSize is the size of the data in every fragment. Best to set to 48 bytes max.
- Padding is the remaining data for the last fragment ( $\text{NbFrag} \times \text{FragSize} - \#data$ ).
- id is unused (set to 1).

It is advised to send this DL as confirmed.

- Port: 100
- Data:
  - uint8\_t command: 0x02
  - uint8\_t Session
  - uint8\_t Number of fragments
  - uint8\_t Fragment size
  - uint8\_t Padding
  - uint8\_t id

Cmd	Session	NbFrag	FragSize	Padding	id
-----	---------	--------	----------	---------	----

### Send a downlink fragment:

Send NbFrag downlinks of size FragSize, with the last downlink having a padding of size Padding. The Session indicates the session id, counter indicates the current downlink fragment counter. The data has length FragSize.

- Port: 100
- Data:
  - uint8\_t command: 0x08
  - uint8\_t session id
  - uint8\_t counter
  - data: length determined by size of FragSize

Cmd	Session	Counter	Data
-----	---------	---------	------



- Fragmentation data:

The data is a byte string consisting of tags and data. This byte string is divided into chunks of the size determined in the fragmentation setup. The decoding of the data happens once all the data has been received by the sensor.

Supported tags:

- 0x02: LORA\_CFG\_ZSC

Data:

- This data has the same format as the ZSC configuration message. See section 7.b.

- 0x03: LORA\_CFG\_WEEK

Data:

- This data has the same format as the ZSC week schedule message. See section 7.c.

- 0x04: LORA\_CFG\_EXC

Data:

- This data has the same format as the ZSC exception schedule message. See section 7.e.

Tag	Data	Tag	data
-----	------	-----	------

- Example of data sent over fragmentation:

To demonstrate how the fragmentation protocol works, a complete example has been worked out here for the year 2023. We will be sending the ZSC config, week planning and exception planning in a single fragmentation session.

#### 1. ZSC config

- mode = mains  
tag: 00, data: 01 [total: 2 bytes]
- dali level = 20, fade time = 15 ( $\pm 1.5$  minutes)  
tag: 01, data: 14 0F [total: 3 bytes]
- dst start = 26/03/2023 02:00 (01:00 in UTC, 1679792400 in UNIX)  
dst stop = 29/10/2023 03:00 (02:00 in UTC, 1698544800 in UNIX)  
tag: 03, data: 10 99 1F 64 A0 BC 3D 65 [total: 9 bytes]
- heartbeat interval = 1 hour, dali interval = 6 hour  
tag: 05, data: 3C 00 68 01 [total: 5 bytes]

length = 19 bytes

The complete message, including the fragmentation tag and data length would look like: **02 13 00 01 01 14 0F 03 10 99 1F 64 A0 BC 3D 65 05 3C 00 68 01** [21 bytes]

## 2. Week planning

Because the ZSC uses UTC time and for this example we are located in Belgium (UTC+1 or UTC+2 depending on DST) we need to subtract one hour from these times.

eg: 04:00 -> 06:00 needs to be sent as starting from 03:00.

Monday:

00:00 -> 04:00 = 30 (start time: 00 00 dim level: 1E)  
04:00 -> 06:00 = 60 (start time: B4 00 dim level: 3C)  
06:00 -> 08:00 = 100 (start time: 2C 01 dim level: 64)  
08:00 -> 18:00 = 0 (start time: A4 01 dim level: 00)  
18:00 -> 20:00 = 100 (start time: FC 03 dim level: 64)  
20:00 -> 22:00 = 60 (start time: 74 04 dim level: 3C)  
22:00 -> 00:00 = 30 (start time: EC 04 dim level: 1E)

The date for Monday alone would look like this: **07 00 00 1E B4 00 3C 2C 01 64 A4 01 00 FC 03 64 74 04 3C EC 04 1E** [22 bytes]

For this example we will be using the same planning for all days, so in total we will have  $7 + 6 \times 6 = 43$  week steps. We only have 6 steps for the other days of the week because the dim level will already be set to 30 from 00:00 to 04:00.

The complete message, including the fragmentation tag and uid would look like: **03 07 5A 2B 00 00 1E B4 00 3C 2C 01 64 A4 01 00 FC 03 64 74 04 3C EC 04 1E 54 06 3C CC 06 64 44 07 00 9C 09 64 14 0A 3C 8C 0A 1E F4 0B 3C 6C 0C 64 E4 0C 00 3C 0F 64 B4 0F 3C 2C 10 1E 94 11 3C 0C 12 64 84 12 00 DC 14 64 54 15 3C CC 15 1E 34 17 3C AC 17 64 24 18 00 7C 1A 64 F4 1A 3C 6C 1B 1E D4 1C 3C 4C 1D 64 C4 1D 00 1C 20 64 94 20 3C 0C 21 1E 74 22 3C EC 22 64 64 23 00 BC 25 64 34 26 3C AC 26 1E** [133 bytes]

## 3. exception planning

For the exception planning we decided that we want to have the lights on dim level 100 throughout the night on all public holidays.

- 01/01/2023 New Year's Day: start = 1672527600, duration = 24300 seconds
- 10/04/2023 Easter Monday: start = 1681151340, duration = 37800 seconds
- 01/05/2023 Labour Day: start = 1682967780, duration = 33300 seconds
- 18/05/2023 Ascension Day: start = 1684438140, duration = 30600 seconds
- 29/05/2023 White Monday: start = 1685389380, duration = 27000 seconds
- 21/07/2023 National Day: start = 1689968640, duration = 29700 seconds
- 15/08/2023 Assumption Day: start = 1692126240, duration = 34200 seconds
- 01/11/2023 All Saints' Day: start = 1698855480, duration = 52200 seconds
- 11/11/2023 Armistice Day: start = 1699718520, duration = 54000 seconds
- 25/12/2023 Christmas Day: start = 1703518860, duration = 57600 seconds

This is a total of 10 exception steps.

The complete message, including the fragmentation tag and uid would look like: **04 00 5A 0A F0 BE B0 63 64 95 01 6C 55 34 64 64 76 02 E4 0C 50 64 64 2B 02 7C 7C 66 64 64 FE 01 44 00 75 64 64 C2 01 00 E0 BA 64 64 EF 01 20 CC DB 64 64 3A 02 38 7A 42 65 64 66 03 78 A5 4F 65 64 84 03 8C A2 89 65 64 C0 03** [74 bytes]

#### 4. Fragmentation session

First we set up a fragmentation session with the following parameters:

- Session id = 199
- Number of fragments = 5
- Fragment size = 48
- Padding = 12
- payload: **02 C7 05 30 0C 01** [6 bytes]

Then we send the 5 fragments as such:

- fragment 1:  
payload: **08 67 01 02 13 00 01 01 14 0F 03 10 99 1F 64 A0 BC 3D 65 05 3C 00 68 01 03 07 5A 2B 00 00 1E B4 00 3C 2C 01 64 A4 01 00 FC 03 64 74 04 3C EC 04 1E 54 06** [51 bytes]
- fragment 2:  
payload: **08 40 02 3C CC 06 64 44 07 00 9C 09 64 14 0A 3C 8C 0A 1E F4 0B 3C 6C 0C 64 E4 0C 00 3C 0F 64 B4 0F 3C 2C 10 1E 94 11 3C 0C 12 64 84 12 00 DC 14 64 54 15** [51 bytes]
- fragment 3:  
payload: **08 40 03 3C CC 15 1E 34 17 3C AC 17 64 24 18 00 7C 1A 64 F4 1A 3C 6C 1B 1E D4 1C 3C 4C 1D 64 C4 1D 00 1C 20 64 94 20 3C 0C 21 1E 74 22 3C EC 22 64 64 23** [51 bytes]
- fragment 4:  
payload: **08 40 04 00 BC 25 64 34 26 3C AC 26 1E 04 00 5A 0A F0 BE B0 63 64 EC 5E 6C 55 34 64 64 A8 93 E4 0C 50 64 64 14 82 7C 7C 66 64 64 88 77 44 00 75 64 64 78** [51 bytes]
- fragment 5:  
payload: **08 40 05 69 00 E0 BA 64 64 04 74 20 CC DB 64 64 98 85 38 7A 42 65 64 E8 CB 78 A5 4F 65 64 F0 D2 8C A2 89 65 64 00 E1 00 00 00 00 00 00 00 00 00 00 00 00** [51 bytes]

You'll notice that the actual size of the fragment is 51, this is because we still have to add the command, session id and counter to each individual fragment.

## 9. How to decode uplink data

### a. Sensor info

General sensor info is sent on every heartbeat, or when a LoRa info uplink is requested (see section 7.a).

- Port: 1
- Data:
  - uint8\_t sensor type  
This value will always be 3.
  - uint8\_t hardware version
  - uint32\_t app version
  - uint8\_t lora class  
0 = class A, 1 = class B and 2 = class C
  - uint8\_t reset reason  
0 = NONE, 1 = low power reset, 2 = window watchdog reset, 3 = independent watchdog reset, 4 = software reset, 5 = external reset pin reset and 6 = brownout reset
- Example:  
**03 01 00 00 02 00 02 00** [8 bytes]  
Sensor type: 03  
hardware version: 01  
app version: 00 00 02 00 = v0.2.0.0  
lora class: 02 = class C  
reset reason: 00 = none

## b. Dali light state

The dali report is sent every dali report interval or when the dimlevel has changed (eg. due to a scheduled step).

- Port: 2
- Data:
  - uint8\_t mode  
1 = MAINS, 2 = ASTRO and 4 = STATIC
  - uint8\_t DIAG  
Diagnostic parameters included in the DIAG byte:
    - bit 0 (LSB): control gear failure
    - bit 1: lamp failure
    - bit 2: lamp on
    - bit 3: limit error
    - bit 4: fade running
    - bit 5: reset state
    - bit 6: short address
    - bit 7 (MSB): power cycle seen
  - uint8\_t DIM level (0-254)  
This DIM level translates into light output according to the following formula:
$$Light\ output(DIM\ level) = 10^{\frac{DIM\ level - 1}{253/3} - 1} \%$$
  - int8\_t temperature in °C
  - uint16\_t current in mA
  - uint16\_t voltage in dV
  - uint32\_t ontime in seconds
  - uint32\_t active\_energy  
The integral of the instantaneous power over a time interval, measure in Wh.
  - uint32\_t active\_power  
Under periodic conditions, mean value, taken over one period of the instantaneous power, measure in mW.
  - uint32\_t apparent\_energy  
The integral of apparent power over a time interval, measure in VAh.
  - uint32\_t apparent\_power  
The product of the RMS voltage between the RMS electric current, measure in mVA.
  - uint32\_t loadside\_energy  
The integral of load-side power over a time interval, measure in Wh.
  - uint32\_t loadside\_power;  
The input power minus the sum of power used for the DALI bus power supply (if present) and the power used for the AUX power supply (if present), measure in mW.

- Example:

**01 4A FF 2A 00 00 53 00 2F 79 00 00 A5 08 00 00 89 0E 00 00 28 17 00 00 99  
1A 00 00 55 06 00 00 FD 0C 00 00** [36 bytes]

Mode: 01 = mains mode

DIAG = 01001010

Dim level = 100

Temperature = 42 °C

Current = 0 mA

Voltage = 83 dV

ontime: 2F 79 00 00 = 31023s

active\_energy: A5 08 00 00 = 2213 Wh

active\_power: 89 0E 00 00 = 3721 mW

apparent\_energy: 28 17 00 00 = 5928 VAh

apparent\_power: 99 1A 00 00 = 6809 mVA

loadside\_energy: 55 06 00 00 = 1621 Wh

loadside\_power: FD 0C 00 00 = 3325 mW

## c. Alarms

Are sent when an event occurs. There are two possible alarms: GPS alarm and tilt alarm. One, or both, alarm(s) can be sent in a single message.

- Port: 3

- Data:

Tilt alarm:

- uint8\_t id: 01 for tilt alarm
- int8\_t angle  
This is the angle between the current and calibrated vectors.
- int16\_t curr[3]  
x, y and z values of the current vector
- int16\_t cal[3]  
x, y and z values of the calibrated vector

GPS alarm:

- uint8\_t Id: 02 for GPS alarm
- int32\_t latitude
- int32\_t longitude  
latitude and longitude are converted to "degrees decimal minutes"  
notation multiplied by 10000.  
e.g.: 50 degrees 50.9529 minutes is sent as 50509529

- Example tilt alarm:

**01 1E 80 00 14 FE 88 03 F6 FF 15 00 1C 04** [14 bytes]

Id: 01 = tilt alarm

angle: 1E = 30

curr:

- 80 00 = 128
- 14 FE = -492
- 88 03 = 904

cal:

- F6 FF = -10
- 15 00 = 21
- 1C 04 = 1052

- Example GPS alarm:

**02 D0 B5 02 03 57 AA 43 00** [9 bytes]

Id: 02 = GPS alarm

Latitude: D0 B5 02 03 = 50509264

➔ 50 deg 50.9264 minutes or 50.848773 deg

Longitude: 57 AA 43 00 = 4434519

➔ 4 deg 43.4519 minutes or 4.724198 deg

#### d. ZSC configuration

This is sent when a ZSC configuration uplink is requested (see also section 8.a).

- Port: 4
- Data:
  - uint8\_t mode
    - 0x01: MAINS
    - 0x02 : ASTRO
    - 0x04 : STATIC
  - uint8\_t fade time

The actual fade time can be calculated as  $2^{\frac{\text{fade time}}{2}}$  in seconds, fade time set to 15 for example equates to roughly 90.5 seconds.
  - uint8\_t amount of week steps
  - uint8\_t amount of exception steps
  - int8\_t sunrise offset in minutes
  - int8\_t sunset offset in minutes
  - uint32\_t daylight saving time start (UNIX time in seconds)
  - uint32\_t daylight saving time end (UNIX time in seconds)
  - uint8\_t tilt angle in degrees
  - uint8\_t tilt delay in minutes
  - uint8\_t tilt duration in minutes
  - uint16\_t week schedule uid
  - uint16\_t exception schedule uid
  - uint16\_t lora heartbeat interval in minutes
  - uint16\_t dali report interval in minutes
  - int16\_t tilt\_cal\_vec[3]  
x, y, and z values of the calibrated tilt vector
  - uint8\_t reserved



- Example:

**01 05 0F 0A 00 00 10 99 1F 64 A0 BC 3D 65 05 05 02 1C 58 00 5A B4 00 3C 00  
F6 FF 15 00 1C 04 00** [32 bytes]

mode: 01 = MAINS mode

fade time = 5.7 seconds

week steps = 15

exception steps = 10

sunrise offset = 0 minutes

sunset offset = 0 minutes

daylight saving time start = 2023-03-26T01:00:00.000Z

daylight saving time stop = 2023-10-29T02:00:00.000Z

tilt angle = 5 degrees

tilt delay = 5 minutes

tilt duration = 2 minutes

week schedule uid = 22556

exception schedule uid = 23040

lora heartbeat interval = 180 minutes

dali report interval = 60 minutes

cal:

- F6 FF = -10 (x value)
- 15 00 = 21 (y value)
- 1C 04 = 1052 (z value)

## e. Session response

When the fragmentation session is completed, the sensor will send an uplink data with the result.

- Port: 100
- Data:
  - The Session and id are the ones you used when setting up the fragmentation session.
    - uint8\_t Session
    - uint8\_t Id
    - uint8\_t state
      - 0 = FRAG\_STATE\_OK, 1 = FRAG\_STATE\_SESSION\_ERROR and
      - 2 = FRAG\_STATE\_DECODE\_ERROR
    - uint16\_t payload size
- example:
  - C7 01 00 FF 00** [3 bytes]
  - Response to session 199 and id 1 with response FRAG\_STATE\_OK. The total payload size was 255.