Technical Documentation

Kuando IoT Busylight – LoRaWAN

Version 3.1.1

Contents

1. Composing the LoRa payload for kuando Busylight.

Before the Busylight can change color, an application controlling the logic needs to be integrated/developed. Please see documentation on LoRaWAN network server to learn how to connect an application.

Payload for the Lora Busylight is a 5 byte array.
Byte 0: Red Color intensity (0..255)
Byte 1: Blue Color intensity (0..255)
Byte 2: Green Color intensity (0..255)
Byte 3: On duration (0..255) (1/10 seconds)
Byte 4: Off duration (0..255) (1/10 seconds)

<u>Example for a blue static light:</u>
Byte[0]=0
Byte[1]=255
Byte[2]=0
Byte[3]=255
Byte[4]=0

Depending on the network provider, this byte array needs to be encoded in base64 or something similar.
The above byte array would result in this base64 string: AP8A/wA=. decoded string 00FF00FF00, a solid blue color with 100% brightness.

Other examples:

Solid Red	990000FF00
Solid Green	000099FF00
Solid Yellow	FF00FF6400
Solid Purple	FFFF006400
Blue Flashing	00FF000A0A

Confirmed Downlink

Please consider the use of confirmed data downlink when composing the integration. This will ensure that the command is delivered.

2. Decoding the Uplink Payload

An uplink, keep alive signal, is sent every 30 minutes from the device. It is a 24-byte array structured as follows:

Byte	*Structure*	*Content*
0-3	Signed int 32 Bit	RSSI from device perspective (dBm)
4-7	Signed int 32 Bit	SNR from device perspective (dB)
8-11	Unsigned int 32 Bit	Downlinks received, since last join
12-15	Unsigned int 32 Bit	Uploads sent, since last join
16	Byte	Last color(Red)
17	Byte	Last color(Blue)
18	byte	Last color(Green)
19	Byte	Time On (1/10 seconds)
20	Byte	Time Off (1/10 seconds)
21	Byte	SW revision
22	Byte	HW revision
23	Byte	ADR state (1=on, 0=off)

Example of a decoded uplink payload:

9DFFFFFF2200000074000000D40300000000FFFF001E0B01	2	4	6	8		
9DFFFFFF	9D	FF	FF	FF	-99	*RSSI from device perspective*
22000000	22	00	00	00	34	*SNR from device perspective*
74000000	74	00	00	00	116	*Downlinks received, since last join*
D4030000	D4	03	00	00	980	*Uploads sent, since last join*
0000FF	0	0	255		#00FF00	*Last color(R,B,G)*
FF	255				255	*Time On*
00	0				0	*Time Off*
1E	1E				30	*sw revision*
0B	0B				11	*HW revision*
01	01				1	*ADR state*

Example of a structure in a C type language:

```
[StructLayout(LayoutKind.Sequential, Pack = 1)]
struct busylightuplinkmessagestruct
{
    Int32 rssi;
    Int32 snr;
    UInt32 messages_received;
    UInt32 messages_send;
    byte lastcolor_red;
    byte lastcolor_blue;
    byte lastcolor_green;
    byte lastcolor_ontime;
    byte lastcolor_offtime;
    byte sw_rev;
    byte hw_rev;
    byte adr_state;
}
```

Please note: If you are using pure C, you need to adjust the data types according to your data type sizes, which depends of the compiler and processor properties. This specific example is C#, where the size of the members can be defined explicitly. The Structure layout annotation tells the compiler how to arrange the members to directly map the byte array to the struct.

3. ADR

ADR is by default enabled on the Busylight.
We recommend NOT to enable ADR on the server side.

As the Busylights do not need optimization of battery usage and we typically have control of gateway position in indoor environments, we recommend to set a standard RX2 DR for downlink messages.

Region	RX2 Data rate
EU	DR4 or DR5
US	DR2
AU	DR4

4. LoRaWan version and LoRa FPort

Busylight IoT is based on LoRaWAN MAC version 1.0.3 and uses FPort 15

5. Power Consumption

- Power Supply: SPD3303X
- Amp-meter: DVM345DI
- DUT: JSE #1 – device EUI: 2020204135260602
- Voltage setting: 5.000V
- Waiting for join (faint yellow): 9.4mA
- Joined (green): 20.4mA
- LED's off consumption: 16.5mA

Color vs. intensity consumption table (numbers are in mA):

Color/Power(%)	10% (26)	20% (51)	40% (102)	60% (153)	80% (204)	100% (255)
Red	20.4	24.9	33.3	41.9	50.5	59.1
Green	20.3	24.5	32.5	40.5	48.5	56.6
Blue	20.0	24.2	32.9	39.7	47.5	55.3
Yellow	24.8	33.0	48.4	62.3	75.7	87.2
Cyan	23.9	31.3	45.2	58.5	70.9	81.5
Purple	24.0	30.3	43.4	57.1	71.4	85.3
White	27.9	37.4	56.3	75.0	90.1	113.2

6. Uplink watchdog

There is a built-in watchdog function (counter) that will restart the device if a given number of uplinks are sent without having received downlink. The counter default is 192 uplinks (4 days with 30 min uplink interval).
If needed, the counter can be changed with a direct command to the device. See Chapter 7 for more info.

7. Commands

Commands to control the Busylight can be sent directly to the device.
Commands can be sent to the Busylight after the first uplink and the join is registered in the NS.
Note: If device is reset/restarted default settings will be restored.

Command	Ctrl format	Example	Function
0xAA		AA00	Restart and reset to factory settings.

0x01		0100	Enable ADR (enabled as default)
0x02		0200	Disable ADR
0x03	Number	0330	Customized number of UL count without DL before reset. Default is 192 (4 days with 30 min UL interval)
0x04	Time is in minutes	04**0F** (0F=15)	Customized count of time between uplinks in minutes. Will start when previous count runs out (example here is 15 minutes, default is 30 minutes).

8. Contact us

If you have questions or need support, please contact us here:

####