Orange IoT Starter KiT User Guide

orange™

Document Control

Title:	User Guide for IoT Starter Kit
Issue:	Version 1.0
Date:	8 August 2016
Author:	MA. Martin
Distribution:	Orange Labs Products and Services

Document Sign Off

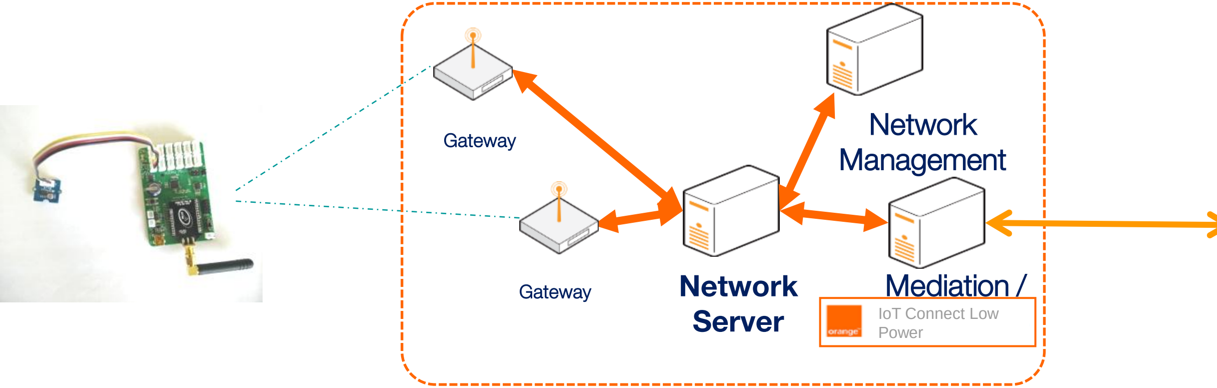
Nature of Signoff	Person	Department	Date	Role
Authors	MA.Martin	Orange Labs	08/09/2016	Project Member
Reviewers	S. Quélard	Orange Labs	16/09/2016	Project Manager

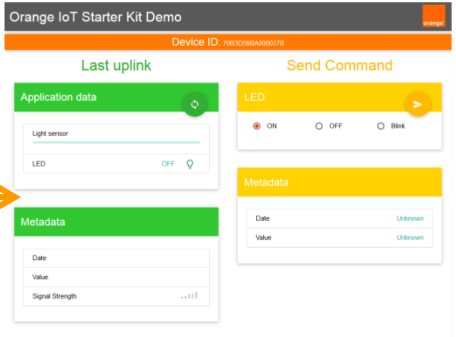
Document Changes

Date	Version	Author	Change Details
08 August 2016	Version 1.0	MA. Martin	Created document base on Stalker doc

Purpose

- This document presents the **Orange IoT Starter Kit**, that is based on an Arduino platform and integrates LoRa®* connectivity of Orange LoRa® national network.
- The kit is provided with a sample application.





Orange IoT Starter Kit Demo

Device ID: 70B3D59BA000037B

Last uplink

Application data

Light sensor

LED OFF

Metadata

Date

Value

Signal Strength

Send Command

LED

● ON ○ OFF ○ Blink

Metadata

Date Unknown

Value Unknown

sample application

*LoRa® est une marque enregistrée au nom de Semtech Corporation .

Contents

1. Get your starter kit up with the web application running
2. Starter Kit components
3. Orange IoT Starter Kit LoRa® connectivity

1. Get your starter kit up & web application running

Orange IoT Starter Kit components

The kit is composed of :

- A SODAQ Mbili board
- A radio shield integrating LoRa® * connectivity, format Xbee, provided by ATIM
- A LIPO battery 3.7 volts
- A Light Sensor with I2C cable
- One USB cable
- An antenna

For more information about each of these components, have a look at the Starter Kit Components part.

*: LoRa ® est une marque enregistrée au nom de Semtech Corporation

Sample application

Description of the sample application

- ✓ the device sends periodically -every 3 minutes- a message containing the last luminosity sensor value. The default periodicity is set to 3 minutes in order to respect the duty cycle imposed on the unlicensed bands, considering the device is in a constrained radio environment (SF 12). (this periodicity can be decreased if your device is in good radio conditions).

- ✓ The web application can display the last received luminosity sensor information.
- ✓ The web application can be used to send a message to set on/ off or in blink state the led of the Arduino board.

The sample application code can be found on
https://github.com/Orange-OpenSource/lpwa-iot-kit

The sample application is composed of

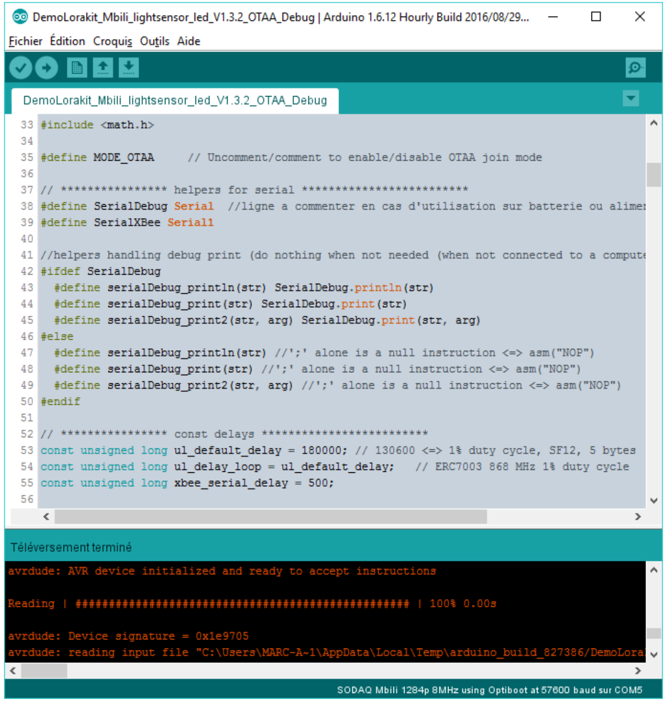
- a program to run on the Arduino :
  "*DemoLorakit_Mbili_lightsensor_led_V1.3.2-1_OTAA_debug.ino*"
- a web application program, that uses IoT Connect Low Power APIs
  "*DemoStarterkit*"

Sample Arduino program code

The sample program code is a simple Arduino code written in C language including a Setup function and a main loop.

The setup function initialized the UART used for Debug and the serial software link used to exchange data with the modem by calling two methods:

- ✓ **initXbeeDebugSerials (**19200, 19200); // set both Serial speeds of the Atmega 1284p
- ✓ **initXbeeNanoN8** (); // init xbee ATIM LoRa module and display some information


DemoLorakit_Mbili_lightsensor_led_V1.3.2_OTAA_Debug | Arduino 1.6.12 Hourly Build 2016/08/29...
Fichier Édition Croquis Outils Aide
DemoLorakit_Mbili_lightsensor_led_V1.3.2_OTAA_Debug

```
#include <math.h>

#define MODE_OTAA     // Uncomment/comment to enable/disable OTAA join mode

// *************** helpers for serial ***********************
#define SerialDebug Serial  //ligne a commenter en cas d'utilisation sur batterie ou alimen
#define SerialXBee Serial1

//helpers handling debug print (do nothing when not needed (when not connected to a compute
#ifdef SerialDebug
  #define serialDebug_println(str) SerialDebug.println(str)
  #define serialDebug_print(str) SerialDebug.print(str)
  #define serialDebug_print2(str, arg) SerialDebug.print(str, arg)
#else
  #define serialDebug_println(str) //';' alone is a null instruction <=> asm("NOP")
  #define serialDebug_print(str) //';' alone is a null instruction <=> asm("NOP")
  #define serialDebug_print2(str, arg) //';' alone is a null instruction <=> asm("NOP")
#endif

// *************** const delays ***********************
const unsigned long ul_default_delay = 180000; // 130600 <=> 1% duty cycle, SF12, 5 bytes
const unsigned long ul_delay_loop = ul_default_delay;   // ERC7003 868 MHz 1% duty cycle
const unsigned long xbee_serial_delay = 500;

```

Téléversement terminé

```
avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.00s

avrdude: Device signature = 0x1e9705
avrdude: reading input file "C:\Users\MARC-A~1\AppData\Local\Temp\arduino_build_827386/DemoLora
```

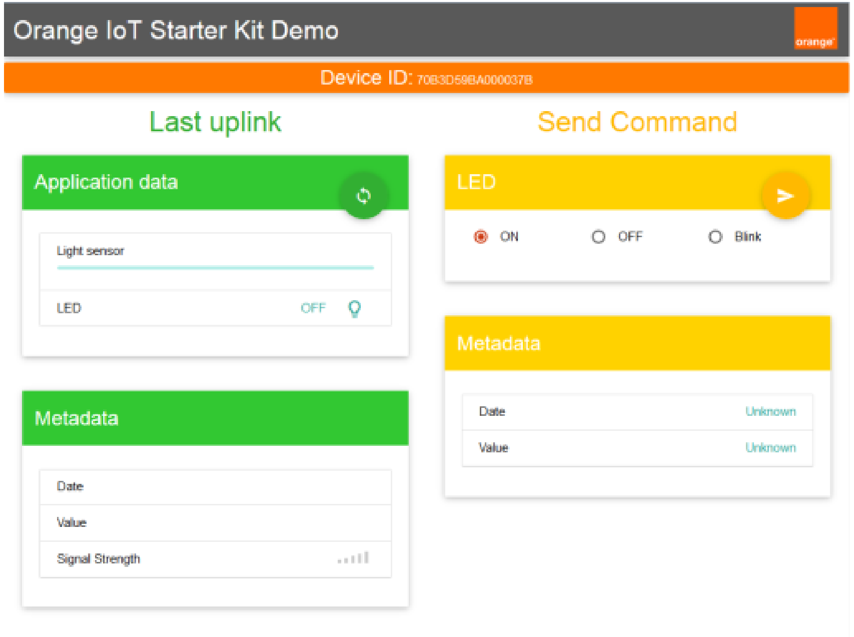
SODAQ Mbili 1284p 8MHz using Optiboot at 57600 baud sur COM5

Sample Arduino program code

The main loop sequences the exchanges with the modem on the IoT Network :

- **Reading** Light Sensor value
- **Transmitting** frame payload of 3 bytes (one byte for led state and two bytes for light Sensor Value)
- **Transmit time** 2,1 seconds (time max to send a LoRa frame).
- **Listening** and **Reading** data from ATIM modem.
- **DutyCycle tempo.** ~3 minutes (do not try to use a lower tempo, it may result a bad LoRa connection due to free band (868MHz) DutyCyle not respected).

Sample web application code



Orange IoT Starter Kit Demo

Device ID: 70B3D59BA000037B

Last uplink

Application data

Light sensor

LED OFF

Metadata

Date

Value

Signal Strength

Send Command

LED

◉ ON ○ OFF ○ Blink

Metadata

Date	Unknown
Value	Unknown

The sample web application is a simple JavaScript application, that can be launched from a PC.

It uses IoT Connect Low Power APIs, to send commands to the device and retrieve messages sent by the device.

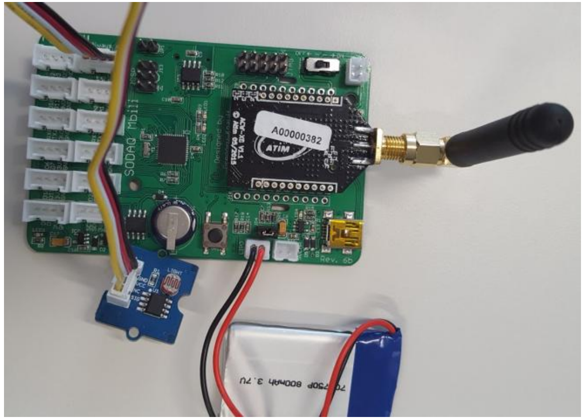
Your device is a LoRa® class A device, it is not in permanent reception. It will receive the sent command just after an emission (emission every 3 minutes).

Starter Kit set up

1 Plug the ATIM radio modem on the Mbili platform on the bee socket.

2 Plug the Groove Light sensor on the socket (A4, A5, 3v3, GND).

3 Then connect the starter kit to the USB serial of your computer. USB and battery connected will charge this last.

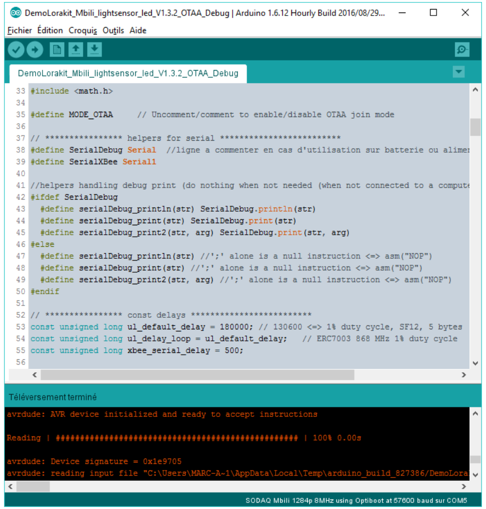


Starter Kit set up

4 Please follow instructions of SODAQ getting start with Arduino website (Arduino IDE installation is explained). (When downloading the device List, please choose the SODAQ AVR version).

5 Open in the Arduino IDE the sample Arduino program.

6 Unless already done, set the board (Tools -> board) to “SODAQ Mbili 1284p 8MHz” and the port to the right port (depends on your computer) should be “COM#”. Arduino will inform you if the wrong port is chosen, in this case, please choose another one.


DemoLorakit_Mbili_lightsensor_led_V1.3.2_OTAA_Debug | Arduino 1.6.12 Hourly Build 2016/08/29...
Fichier Édition Croquis Outils Aide
DemoLorakit_Mbili_lightsensor_led_V1.3.2_OTAA_Debug

```
#include <math.h>

#define MODE_OTAA     // Uncomment/comment to enable/disable OTAA join mode

// **************** helpers for serial ************************
#define SerialDebug Serial  //ligne a commenter en cas d'utilisation sur batterie ou alimen
#define SerialXBee Serial1

//helpers handling debug print (do nothing when not needed (when not connected to a compute
#ifdef SerialDebug
  #define serialDebug_println(str) SerialDebug.println(str)
  #define serialDebug_print(str) SerialDebug.print(str)
  #define serialDebug_print2(str, arg) SerialDebug.print(str, arg)
#else
  #define serialDebug_println(str) //';' alone is a null instruction <=> asm("NOP")
  #define serialDebug_print(str) //';' alone is a null instruction <=> asm("NOP")
  #define serialDebug_print2(str, arg) //';' alone is a null instruction <=> asm("NOP")
#endif

// **************** const delays ************************
const unsigned long ul_default_delay = 180000; // 130600 <=> 1% duty cycle, SF12, 5 bytes
const unsigned long ul_delay_loop = ul_default_delay;   // ERC7003 868 MHz 1% duty cycle
const unsigned long xbee_serial_delay = 500;

```

Téléversement terminé

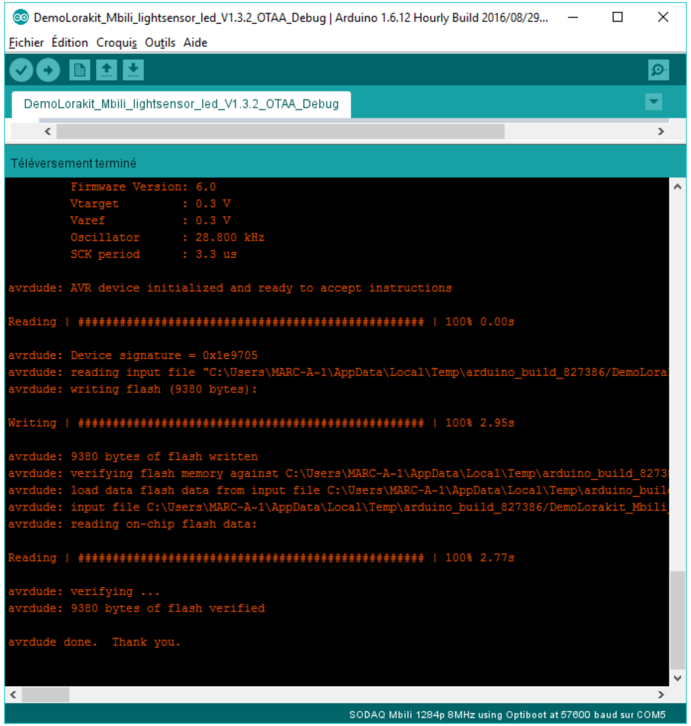
```
avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.00s

avrdude: Device signature = 0x1e9705
avrdude: reading input file "C:\Users\MARC-A~1\AppData\Local\Temp\arduino_build_827386/DemoLora
```

SODAQ Mbili 1284p 8MHz using Optiboot at 57600 baud sur COM5

Starter Kit set up

7 Upload the program ”DemoStarterKit_Mbili_LightSensor_Led_V1.3.2_OTAA_debug.ino”

If this is the first time your device is joining the network, please place it in good radio conditions (outdoor by preference).


DemoLorakit_Mbili_lightsensor_led_V1.3.2_OTAA_Debug | Arduino 1.6.12 Hourly Build 2016/08/29...
Fichier Édition Croquis Outils Aide
DemoLorakit_Mbili_lightsensor_led_V1.3.2_OTAA_Debug

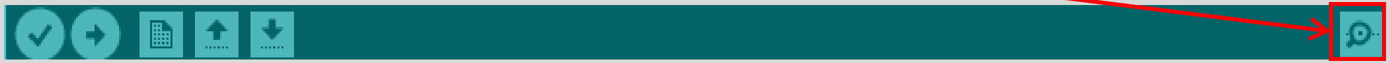
Téléversement terminé

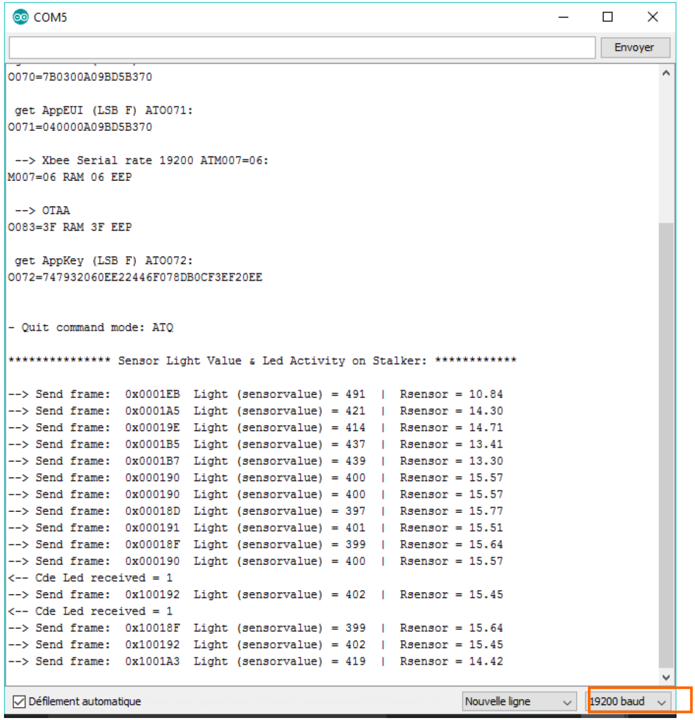
```
         Firmware Version: 6.0
         Vtarget         : 0.3 V
         Varef           : 0.3 V
         Oscillator      : 28.800 kHz
         SCK period      : 3.3 us

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.00s

avrdude: Device signature = 0x1e9705
avrdude: reading input file "C:\Users\MARC-A~1\AppData\Local\Temp\arduino_build_827386/DemoLora
avrdude: writing flash (9380 bytes):

Writing | ################################################## | 100% 2.95s

avrdude: 9380 bytes of flash written
avrdude: verifying flash memory against C:\Users\MARC-A~1\AppData\Local\Temp\arduino_build_8273
avrdude: load data flash data from input file C:\Users\MARC-A~1\AppData\Local\Temp\arduino_buil
avrdude: input file C:\Users\MARC-A~1\AppData\Local\Temp\arduino_build_827386/DemoLorakit_Mbili_
avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 2.77s

avrdude: verifying ...
avrdude: 9380 bytes of flash verified

avrdude done.  Thank you.
```

SODAQ Mbili 1284p 8MHz using Optiboot at 57600 baud sur COM5

Starter Kit debug

7 Open the Serial Monitor from the Arduino IDE





```
0070=7B0300A09BD5B370

 get AppEUI (LSB F) AT0071:
0071=040000A09BD5B370

 --> Xbee Serial rate 19200 ATM007=06:
M007=06 RAM 06 EEP

 --> OTAA
0083=3F RAM 3F EEP

 get AppKey (LSB F) AT0072:
0072=747932060EE22446F078DB0CF3EF20EE


- Quit command mode: ATQ

*************** Sensor Light Value & Led Activity on Stalker: ************

--> Send frame:  0x0001EB  Light (sensorvalue) = 491  |  Rsensor = 10.84
--> Send frame:  0x0001A5  Light (sensorvalue) = 421  |  Rsensor = 14.30
--> Send frame:  0x00019E  Light (sensorvalue) = 414  |  Rsensor = 14.71
--> Send frame:  0x0001B5  Light (sensorvalue) = 437  |  Rsensor = 13.41
--> Send frame:  0x0001B7  Light (sensorvalue) = 439  |  Rsensor = 13.30
--> Send frame:  0x000190  Light (sensorvalue) = 400  |  Rsensor = 15.57
--> Send frame:  0x000190  Light (sensorvalue) = 400  |  Rsensor = 15.57
--> Send frame:  0x00018D  Light (sensorvalue) = 397  |  Rsensor = 15.77
--> Send frame:  0x000191  Light (sensorvalue) = 401  |  Rsensor = 15.51
--> Send frame:  0x00018F  Light (sensorvalue) = 399  |  Rsensor = 15.64
--> Send frame:  0x000190  Light (sensorvalue) = 400  |  Rsensor = 15.57
<-- Cde Led received = 1
--> Send frame:  0x100192  Light (sensorvalue) = 402  |  Rsensor = 15.45
<-- Cde Led received = 1
--> Send frame:  0x10018F  Light (sensorvalue) = 399  |  Rsensor = 15.64
--> Send frame:  0x100192  Light (sensorvalue) = 402  |  Rsensor = 15.45
--> Send frame:  0x1001A3  Light (sensorvalue) = 419  |  Rsensor = 14.42
```

19200 baud

Set the baud rate to 19200

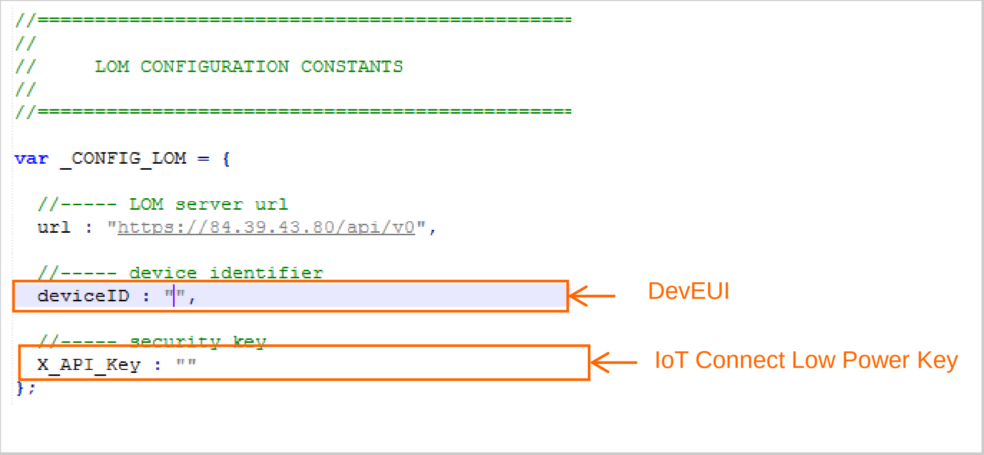
Starter Kit set up

8 Copy the web application repertory “DemoStarterKit” on your PC

9 Copy the configuration file.js that we provided you with, in the folder DemoStarterKit/demoStarterKit_files and rename it in ‘configuration.js’ (you may delete or backup the previous one)

You may check it uses your device parameters (radio and IoT Connect Low Power parameters).

- ✓ your device identifier: DevEUI
- ✓ your IoT Connect Low Power Key also called X-API-Key



```
//==============================================
//
//     LOM CONFIGURATION CONSTANTS
//
//==============================================

var _CONFIG_LOM = {

  //----- LOM server url
  url : "https://84.39.43.80/api/v0",

  //----- device identifier
  deviceID : "",

  //----- security key
  X_API_Key : ""
};
```

DevEUI

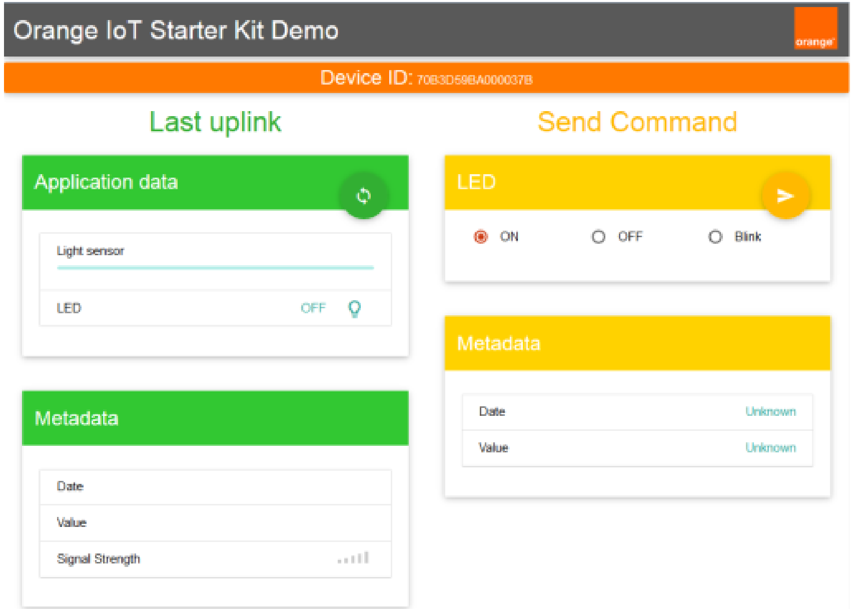
IoT Connect Low Power Key

Starter Kit set up

10 Launch the web application by double clicking “index.html” , the following window should appear in the browser:

Browsers tested and running the sample application:

navigateur	versions
Firefox	38.6.1 et +
Chrome	52.0.2743 et +
Internet Explorer	11.0.14393.0 et +


Orange IoT Starter Kit Demo

Device ID: 70B3D59BA000037B

Last uplink

Application data

Light sensor

LED OFF

Metadata

Date

Value

Signal Strength

Send Command

LED

● ON ○ OFF ○ Blink

Metadata

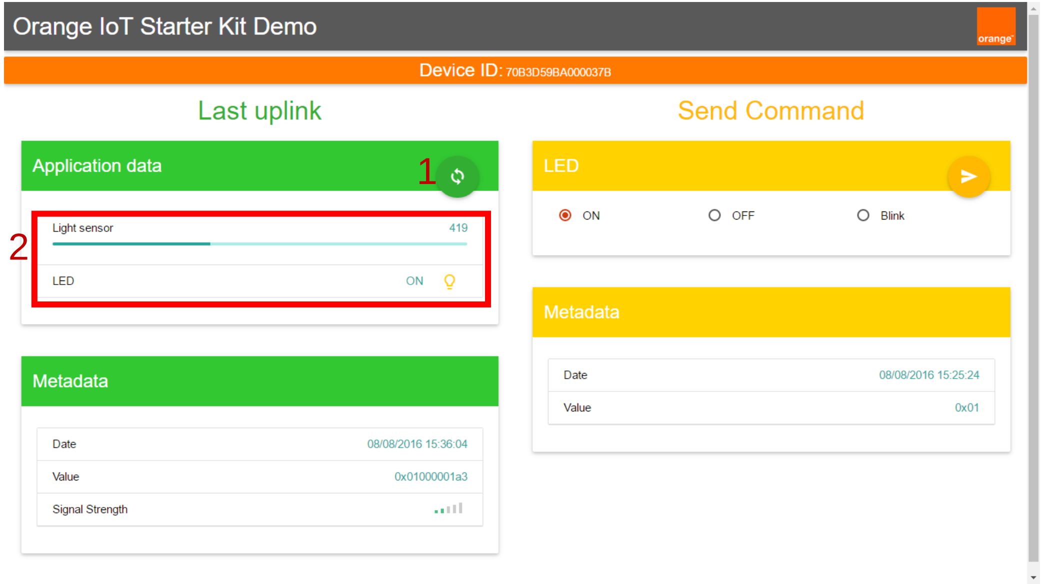
Date Unknown

Value Unknown

Starter Kit set up

11 Read the uplink data

Click on the button « refresh » you should see the number and the light sensor value should appear in the view as shown on the following figure



Orange IoT Starter Kit Demo

Device ID: 70B3D59BA000037B

Last uplink

Application data

Light sensor	419
LED	ON

Metadata

Date	08/08/2016 15:36:04
Value	0x01000001a3
Signal Strength	

Send Command

LED

● ON ○ OFF ○ Blink

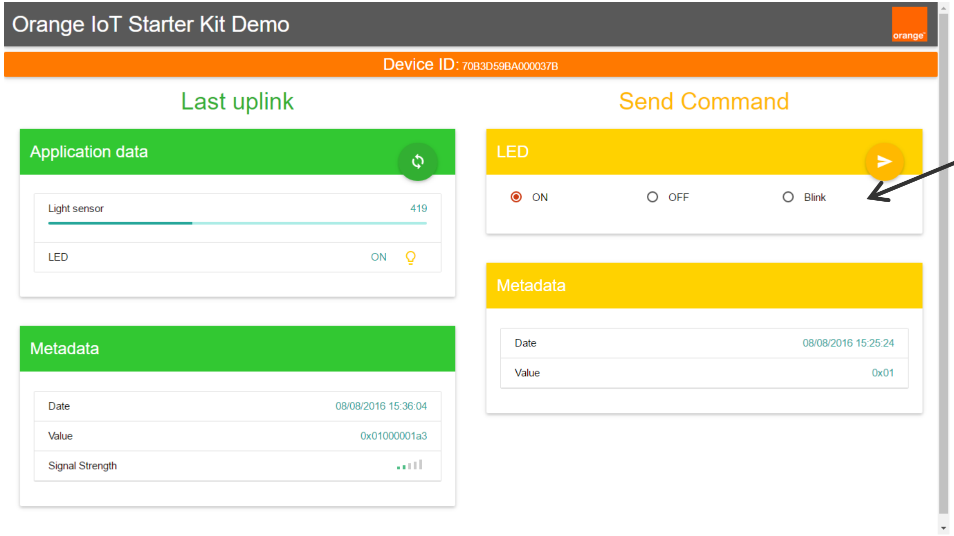
Metadata

Date	08/08/2016 15:25:24
Value	0x01

Starter Kit set up

12 Send a command from the web application

Activate either the « On », « Off » or « Blink » button on the web application then send the command and check the corresponding action of the led on the board as described.
As your device is a LoRa Class A device, it will receive the command just after its next message sending (can take up to 3 minutes, as the frequency of message emission is 3 minutes)



Orange IoT Starter Kit Demo

Device ID: 70B3D59BA000037B

Last uplink

Application data

Light sensor	419
LED	ON

Metadata

Date	08/08/2016 15:36:04
Value	0x01000001a3
Signal Strength	

Send Command

LED

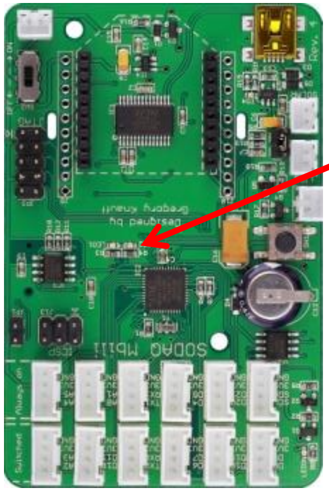
● ON ○ OFF ○ Blink

Metadata

Date	08/08/2016 15:25:24
Value	0x01

Led commands sent from the web application:

Led OFF	00
Led ON	01
Led Blink	02

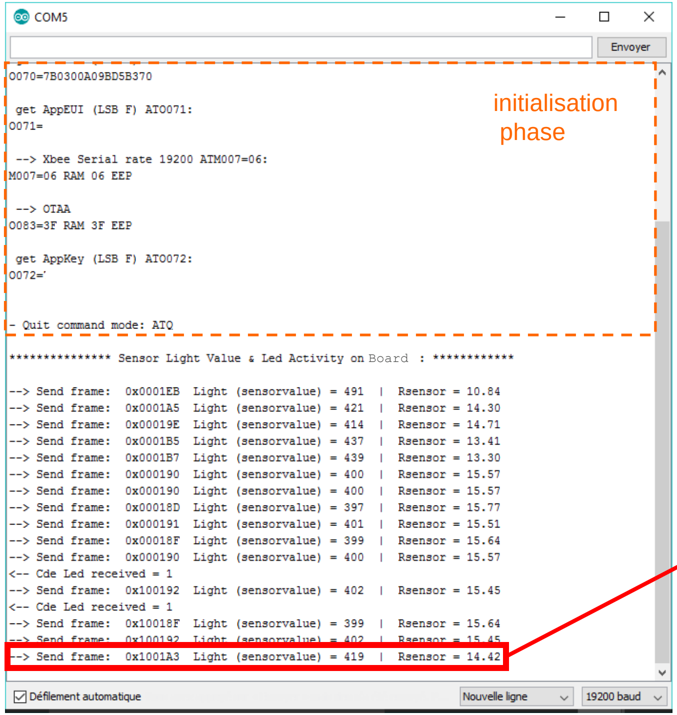


Red led (led 2)

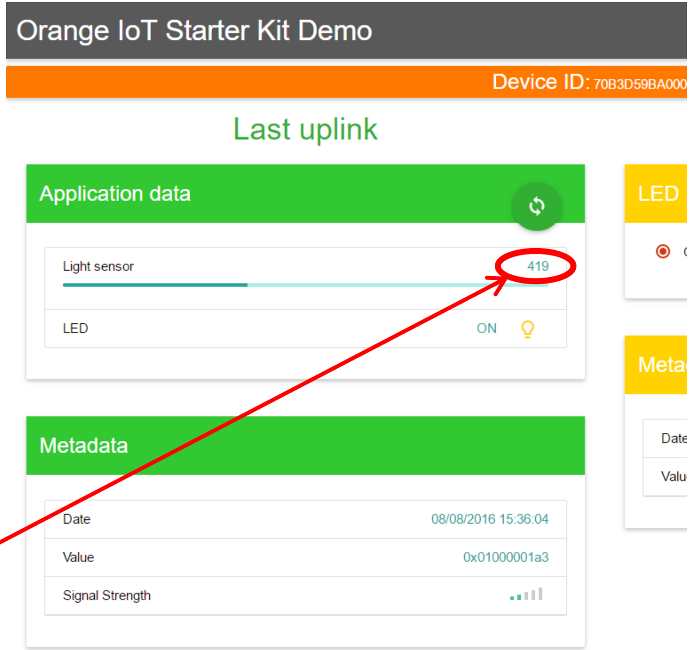
Starter Kit set up

13 You can check the received command from the web application

- ✓ open a serial monitor window
- ✓ You should be able to read the light sensor values, sent (uplink) to the application and the led commands (downlink) issued by the web application.



```
0070=7B0300A09BD5B370

 get AppEUI (LSB F) AT0071:
0071=

 --> Xbee Serial rate 19200 ATM007=06:
M007=06 RAM 06 EEP

 --> OTAA
0083=3F RAM 3F EEP

 get AppKey (LSB F) AT0072:
0072='


- Quit command mode: ATQ

*************** Sensor Light Value & Led Activity on Board  : ************

--> Send frame:  0x0001EB  Light (sensorvalue) = 491  |  Rsensor = 10.84
--> Send frame:  0x0001A5  Light (sensorvalue) = 421  |  Rsensor = 14.30
--> Send frame:  0x00019E  Light (sensorvalue) = 414  |  Rsensor = 14.71
--> Send frame:  0x0001B5  Light (sensorvalue) = 437  |  Rsensor = 13.41
--> Send frame:  0x0001B7  Light (sensorvalue) = 439  |  Rsensor = 13.30
--> Send frame:  0x000190  Light (sensorvalue) = 400  |  Rsensor = 15.57
--> Send frame:  0x000190  Light (sensorvalue) = 400  |  Rsensor = 15.57
--> Send frame:  0x00018D  Light (sensorvalue) = 397  |  Rsensor = 15.77
--> Send frame:  0x000191  Light (sensorvalue) = 401  |  Rsensor = 15.51
--> Send frame:  0x00018F  Light (sensorvalue) = 399  |  Rsensor = 15.64
--> Send frame:  0x000190  Light (sensorvalue) = 400  |  Rsensor = 15.57
<-- Cde Led received = 1
--> Send frame:  0x100192  Light (sensorvalue) = 402  |  Rsensor = 15.45
<-- Cde Led received = 1
--> Send frame:  0x10018F  Light (sensorvalue) = 399  |  Rsensor = 15.64
--> Send frame:  0x100192  Light (sensorvalue) = 402  |  Rsensor = 15.45
--> Send frame:  0x1001A3  Light (sensorvalue) = 419  |  Rsensor = 14.42
```



Orange IoT Starter Kit Demo

Device ID: 70B3D59BA000

Last uplink

Application data

Light sensor	419
LED	ON

Metadata

Date	08/08/2016 15:36:04
Value	0x01000001a3
Signal Strength	

2. Starter Kit components

Orange IoT Starter Kit components
SODAQ Mbili Board

Please refer to the SODAQ Mbili support webpage in order to have more information about its specifications and features.

Website: http://support.sodaq.com/sodaq-one/sodaq-mbili-1284p/

Orange IoT Starter Kit components LoRa® radio shield

The LoRa® radio shield provided in the starter kit a shield developed by ATIM. (http://www.atim.com/en/)

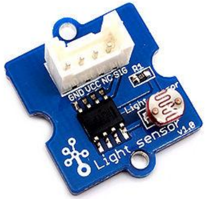
This shield integrates the ATIM Nano N8 LoRa® modem that is mounted on a Xbee form factor board of size 50x25x13 mm.

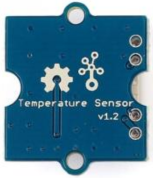
This board is powered under 3.3v and has SMA connector for the antenna.

To connect the board to an Arduino platform two Xbee 10-pin socket with 2mm pitch are used.

Orange IoT Starter Kit components
Light sensor Groove version 1.1

The Light sensor module uses the GL5528 photo-resistor to detect the light intensity of the environment. The resistance of the sensor decreases when the light intensity increases.
The light sensor uses a thermistor which returns the ambient light in the form of a resistance value, which is then used to alter the voltage signal issue from Vcc supply of the Seeeduino. The Atmega 1284p then converts this voltage value measured on an analog input pin and after converting it to digital compute the light.





3. Orange IoT Starter KiT LoRa® connectivity

LoRa® , LoRaWAN™

A LoRaWAN™ network is a Low-Power, Wide-Area Network (LPWAN), designed for long range, low power and low data rates applications.

LoRa® stands for the physical layer of the protocol. This radio technology has been developed and is owned by Semtech Corporation.
LoRaWAN™ stands for the MAC layer protocol, and is specified by the LoRa® Alliance.

A LoRaWAN™ network is a star network , in which gateways relays messages from devices to a network server.
LoRa® data rates ranges from 300 bps to ~50 kbps.
LoRaWAN™ protocol uses an adaptive data rate algorithm, that enables to optimize device consumption, and network capacity.
LoRa® technology operates on unlicensed band. (863-870 MHz in Europe). These bands are regulated.

Useful LoRa® Alliance Website documents :
LoRaWAN™ protocol specification
What is LoRaWAN™

Orange IoT Starter Kit LoRa® Set Up

The Orange IoT Starter Kit is compliant to LoRaWAN™ 1.0 specification.

The kit is configured to be a LoRa® **class A** device.
It supports OTAA activation mode only (it is provisioned on Orange national network).
The Adaptive Data Rate algorithm is activated.
The device is preconfigured to send unconfirmed UL messages.
The following LoRa® parameters are preconfigured on the device:

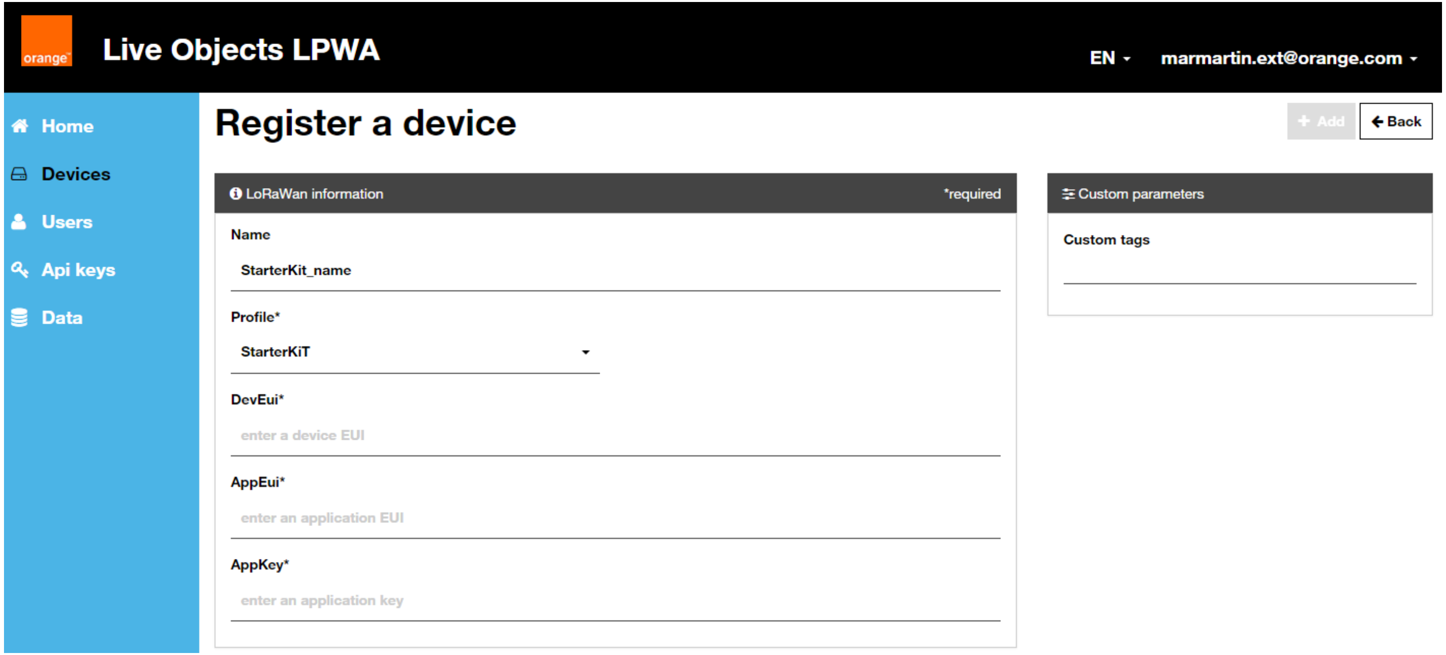
- DevEUI : Unique Device Identifier
- AppEUI : network address of the device.
- AppKey : device key (used to join the network and generate encryption/decryption keys)

The default set up of the Orange IoT Starter KiT guarantees a proper functioning of the device on the Orange network.
We highly recommend not to modify the default radio configuration of the device.

IoT Connect Low Power Provisioning

If the devices hasn't been provisioned on your IoT Connect Low Power account, you may do it manually by filling in the form as here-after:

Caution: Please make sure to use the right "Profile"



Register a device

LoRaWan information *required

Name
StarterKit_name

Profile*
StarterKiT

DevEui*
enter a device EUI

AppEui*
enter an application EUI

AppKey*
enter an application key

Custom parameters

Custom tags

IoT Connect Low Power APIs

Communication with the device is done via Orange IoT Connect Low Power APIs.

2 generic uses cases are available through the API (REST and MQTT)

- *send a command to a device*
- *retrieve data coming from a device*

More information on the APIs for Orange national network can be found on IoT Connect Low Power site in the documents:

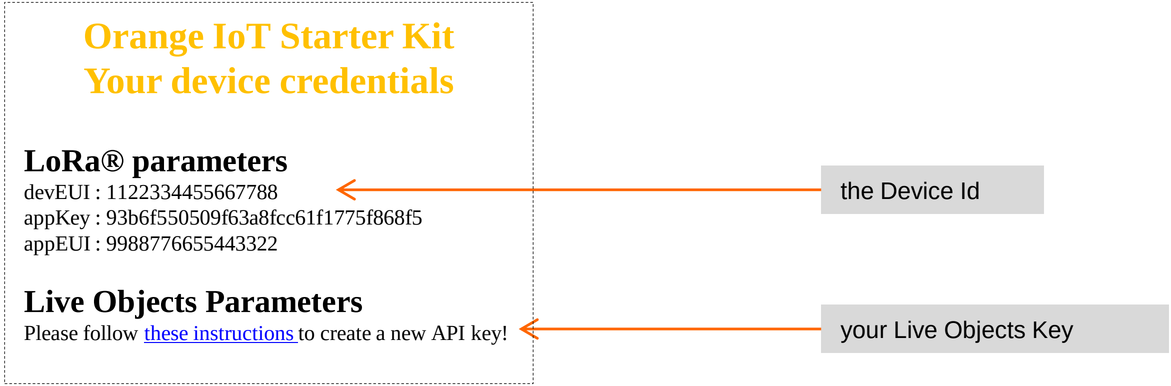
LoRa Connect – Live Objects – complete guide.pdf

Your starter kit credentials

A credential file containing LoRa parameters of your kit will be provided.

This file contains both your **IoT Connect Low Power Key** (alias Live Objects Key) and the **Device Id** required to get your device data and send commands to your device.

If you have your own Live Objects Manage account, you will have to create your own live objects key as shown in this page.



Orange IoT Starter Kit
Your device credentials

LoRa® parameters

devEUI : 1122334455667788
appKey : 93b6f550509f63a8fcc61f1775f868f5
appEUI : 9988776655443322

Live Objects Parameters

Please follow these instructions to create a new API key!

the Device Id

your Live Objects Key