



17 & 18 janvier 2017 Usine IO Paris



L'USINE DIGITALE

Orange LoRa Starter-Kit

Bienvenue au **LoRa IoT Challenge**, vous allez pouvoir prototyper vos projets avec un kit de développement qui vous sera fourni lors du Hackathon cloturant le challenge

Vos ressources

Pour le LoRa IoT Challenge, nous vous mettons à disposition toutes les ressources nécessaires à l'élaboration et au développement d'un prototype fonctionnel LoRa® sur le réseau d'Orange.

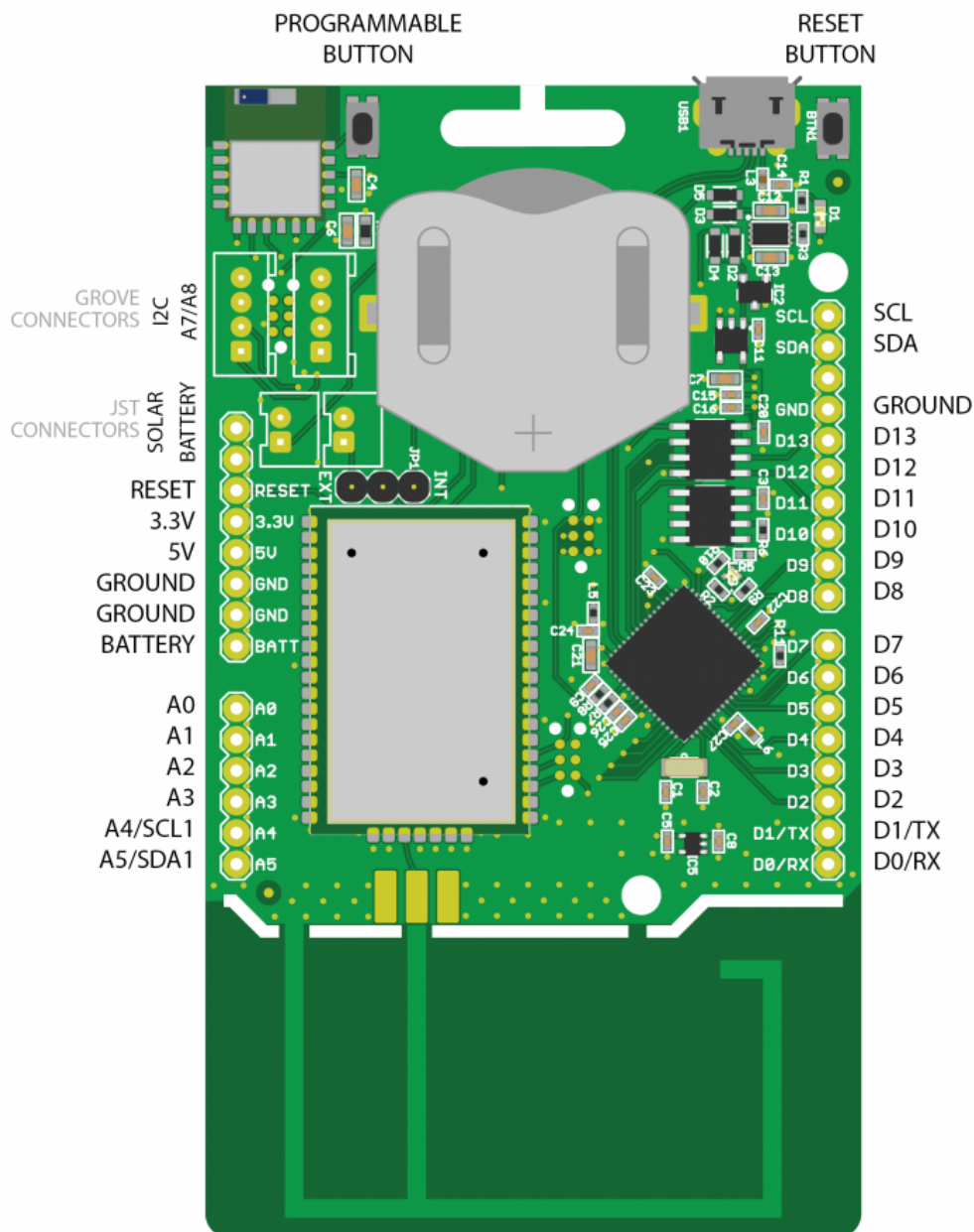
Vous aurez donc :

- Une connexion WiFi
- Un Kit de développement LoRa compatible Arduino :
 - Capteur analogique intégré (sonde de température)
 - Bouton poussoir programmable
 - Connecteur standard, compatible shield Arduino
 - Batterie
 - Antenne intégrée
 - Cables
 - Connexion au réseau LoRa d'Orange
- Accès à Live Objects, plateforme de collecte de données et de gestion de vos objets connectés
- Accès à Elasticsearch pour le traitement des données
- Accès à Kibana, pour la visualisation des données.

Le Starter KIT LoRa Orange

Voici un schema représentatif du Starter Kit LoRa d'Orange. Le starter Kit LoRa d'Orange est un Kit complet compatible Arduino, aussi, vous pourrez le programmer et l'utiliser avec l'IDE officiel d'Arduino, de façon simple et rapide, tout en gardant vos habitudes de développement. Voici le schema de notre carte.

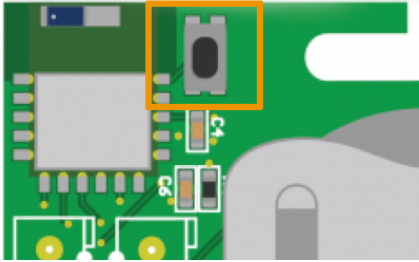
Schema de la carte



Modules

Module Bouton Programmable

Sur le Starter Kit, vous trouverez un bouton programmable, il s'agit d'un bouton poussoir que vous pouvez utiliser lors de vos développements. Ce dernier se situe en haut de notre carte, comme sur le schema ci-dessous.



Vous pourrez programmer ce bouton poussoir comme dans l'exemple ci-dessous

```
void loop() {  
  int sensorVal = digitalRead(BUTTON);  
  
  if (sensorVal == HIGH) {  
    //do some stuff  
  } else {  
    // do something else  
  }  
}
```

Module BLE

Notre Starter Kit LoRa comprend un module Bluetooth Low Energy **BLE 4.2**, pouvant être utilisé en tant qu'émetteur (beacon, eddystone) ou en tant que recepteur d'informations (scanner).

Module Batterie

Vous pouvez, si vous le désirez, brancher une batterie plus puissante que celle qui vous est fournie avec notre Starter Kit LoRa. Nous vous invitons à vous rapprocher de nos équipes afin de connaître les modalités d'implémentation et de soudure de connecteurs supplémentaires, vous permettant d'utiliser, par exemple, un panneau solaire ou une batterie supplémentaire.

Antenne

Le module comprend une antenne intégrée, que vous pouvez utiliser par défaut. Si dans votre application industrielle, vous souhaitez utiliser une antenne plus puissante pour des besoins spécifique, notre équipe vous expliquera comment souder une antenne plus puissante à notre carte.

Capteur de température

Notre Starter Kit comprend aussi un capteur de température, que vous pourrez implémenter et utiliser de façon simple dans votre code comme dans l'exemple ci-dessous

```
void loop() {  
  
float mVolts = (float)analogRead(TEMP_SENSOR) * 3300.0 / 1023.0;  
float temp = (mVolts - 500.0) / 10.0;  
  debugSerial.print(temp);  
  debugSerial.println(" C");  
  delay(1000);  
}
```

LED BLEUE

Notre plateforme comprend une LED bleue que vous pouvez utiliser comme ceci

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);    // allume la led  delay(1000);  
  
  digitalWrite(LED_BUILTIN, LOW);     // eteint la LED  
  delay(1000);  
}
```

LED RGB

Notre plateforme comprend une LED RGB que vous pouvez utiliser comme ceci

```
void RED() {
    digitalWrite(LED_RED, LOW);
    digitalWrite(LED_GREEN, HIGH);
    digitalWrite(LED_BLUE, HIGH);
}

void GREEN() {
    digitalWrite(LED_RED, HIGH);
    digitalWrite(LED_GREEN, LOW);
    digitalWrite(LED_BLUE, HIGH);
}

void BLUE() {
    digitalWrite(LED_RED, HIGH);
    digitalWrite(LED_GREEN, HIGH);
    digitalWrite(LED_BLUE, LOW);
}

void setup() {
    pinMode(LED_RED, OUTPUT);
    pinMode(LED_GREEN, OUTPUT);
    pinMode(LED_BLUE, OUTPUT);
}

void loop() {
    RED();
    delay(1000);
    GREEN();
    delay(1000);
    BLUE();
    delay(1000);
}
```

Caractéristique techniques des composants de la carte

Microcontroller	ATSAMD21G18, 32-Bit ARM Cortex M0+
Compatibility	Arduino M0 Compatible
Size	40 x 25 mm
Operating Voltage	3.3V
I/O Pins	20
Analog Output Pin	10-bit DAC
External Interrupts	Available on all pins
DC Current per I/O pin	7 mA
Flash Memory	256 KB and 4MB (external flash)
SRAM	32KB
EEPROM	Up to 16KB by emulation
Clock Speed	48 MHz
Power	5V USB power and/or 3.7 LiPo battery
Charging	Solar charge controller, up to 500mA charge current
LED	RGB LED, Blue LED
LoRa	Microchip RN2483 Module
Bluetooth	Microchip RN4871 Module
Cyptochip	ATECC508A
Temperature sensor	MCP9700AT
USB	MicroUSB Port

Téléchargez l'IDE Arduino

<https://www.arduino.cc/en/Main/Software>

Pour utiliser notre carte avec l'IDE Arduino, vous devez lui intégrer un nouveau fichier de carte personnalisée. Il s'agit de la SODAQ SAMD boards.

Allez dans Fichier -> Preferences

http://downloads.sodaq.net/packagesodaqindex.json

preferences



Settings Network

Sketchbook location:

C:\Users\janwillem\Documents\Arduino

Browse

Editor language: English (English) (requires restart of Arduino)

Editor font size: 12

Show verbose output during: ☐ compilation ☒ upload

Compiler warnings: None

☐ Display line numbers

☐ Enable Code Folding

☒ Verify code after upload

☐ Use external editor

☒ Check for updates on startup

☒ Update sketch files to new extension on save (.pde -> .ino)

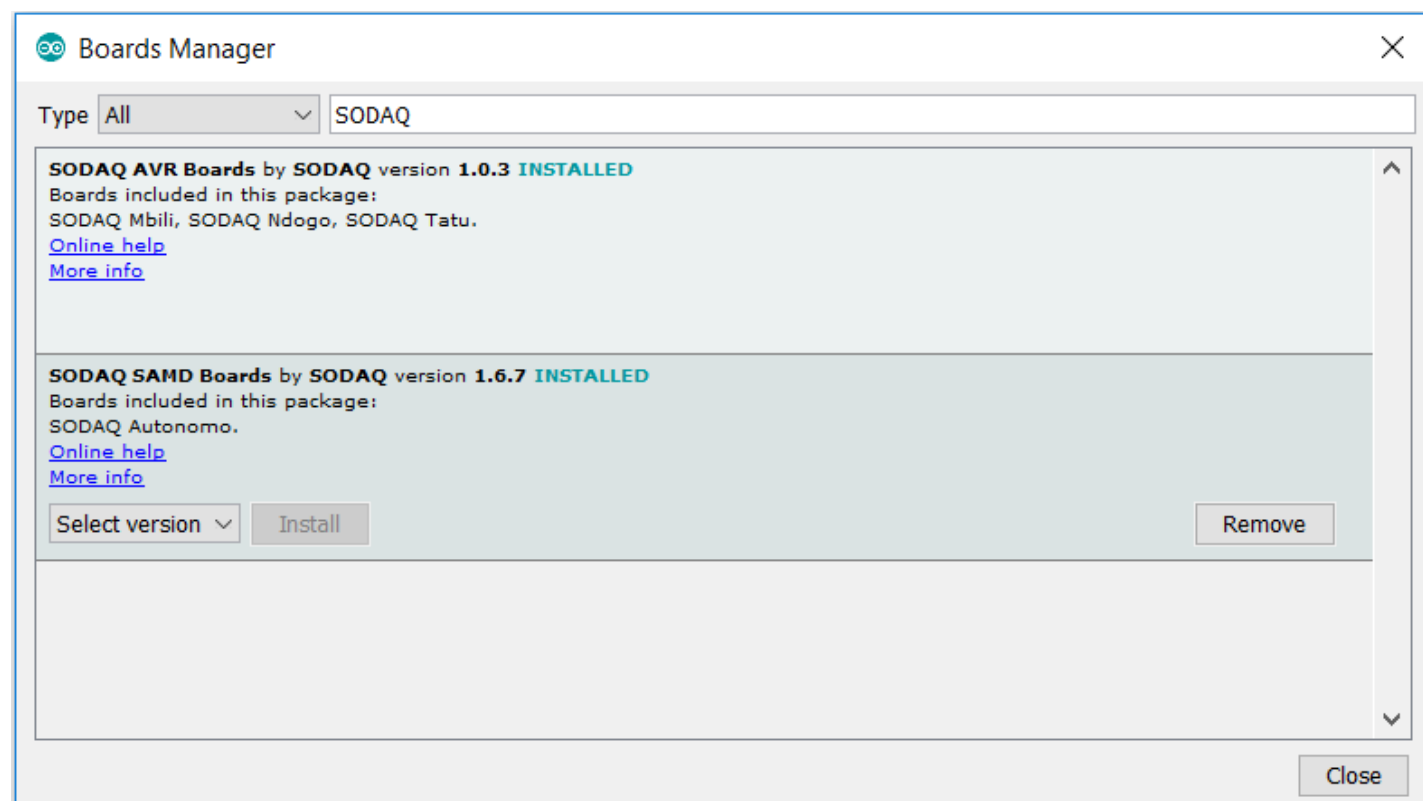
☒ Save when verifying or uploading

Additional Boards Manager URLs: http://downloads.sodaq.net/package_sodaq_index.json



Installez la dernière version des cartes SODAQ SAMD

Outils→Carte→Manager de cartes...



Pin description	Pin number	Definition
RGB Red LED		LED_RED
RGB Green LED		LED_GREEN
RGB Blue LED		LED_BLUE
Blue LED	D13	LED_BUILTIN
Bluetooth Wake		BLUETOOTH_WAKE
Bluetooth Reset		BT_RESET
Push Button		BUTTON
LoRa Reset		LORA_RESET
Temperature Sensor	A6	TEMP_SENSOR

Utilisation de LoRa

Envoi de données sur le réseau

```
LpwaOrange.flush();
LpwaOrange.addFloat(temperature);
LpwaOrange.addInt(nbrPush);

int len;
const char* frame = LpwaOrange.getFramePayload(&len);
debugFrame(frame, len);

LpwaOrange.send(LORA_PORT, (const uint8_t*)frame, len);
```

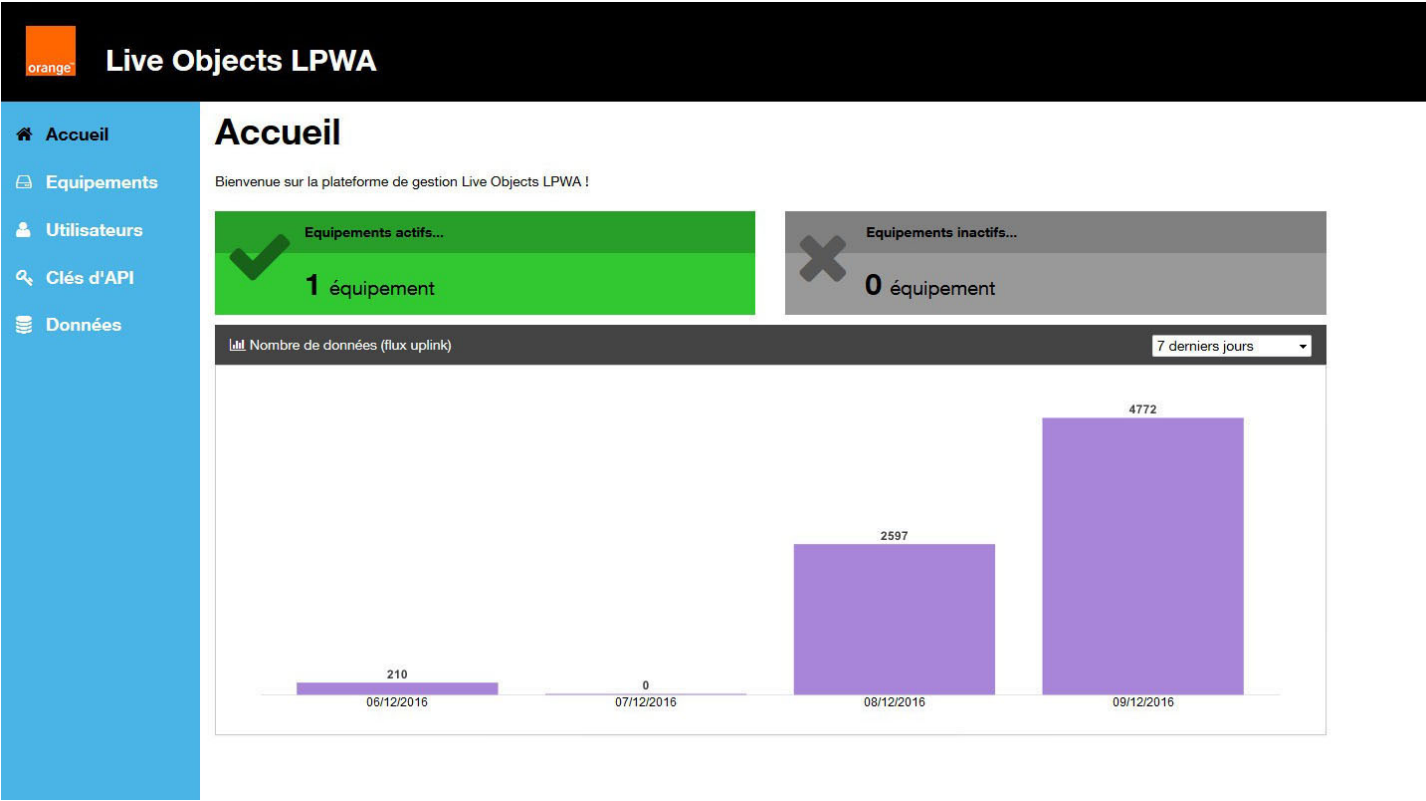
Reception de données

```
int frameReceivedSize = LpwaOrange.receive(frameReceived, 255);

debugSerial.print("received : ");
for (int i = 0; i<frameReceivedSize; i++){
debugSerial.print(frameReceived[i], HEX);
debugSerial.print(" ");
}
debugSerial.println("");
```


Live Objects

Vous aurez accès à notre plateforme pour la gestion de vos objets connectés : Live Objects



Live Objects vous permettra de gérer les données de vos objets connectés

The screenshot shows the 'Données' (Data) page of the 'Live Objects LPWA' platform. The left sidebar contains navigation links: Accueil, Equipements, Utilisateurs, Clés d'API, and Données. The main content area displays a table of data records. The table has columns: Date, Payload, Statut de décodage, DevEui, Fcnt, Port, and Sig rés. The table shows 10 records, all with a status of 'Statut de décodage' and a 'Sig rés' of '10'. An 'Exporter .csv' button is visible in the top right corner.

Date	Payload	Statut de décodage	DevEui	Fcnt	Port	Sig rés
12/12/2016 11:26:38	626970		70B3D59BA00004C6	33815	5	10
12/12/2016 11:26:28	626970		70B3D59BA00004C6	33814	5	10
12/12/2016 11:26:18	626970		70B3D59BA00004C6	33813	5	10
12/12/2016 11:26:08	626970		70B3D59BA00004C6	33812	5	10
12/12/2016 11:25:58	626970		70B3D59BA00004C6	33811	5	10
12/12/2016 11:25:08	626970		70B3D59BA00004C6	33806	5	10
12/12/2016 11:24:58	626970		70B3D59BA00004C6	33805	5	10
12/12/2016 11:24:48	626970		70B3D59BA00004C6	33804	5	10

Les données pourront être traitées par exemple avec Elastic Search et visualisée dans Kibana

