

SAE3.01

PLAN :

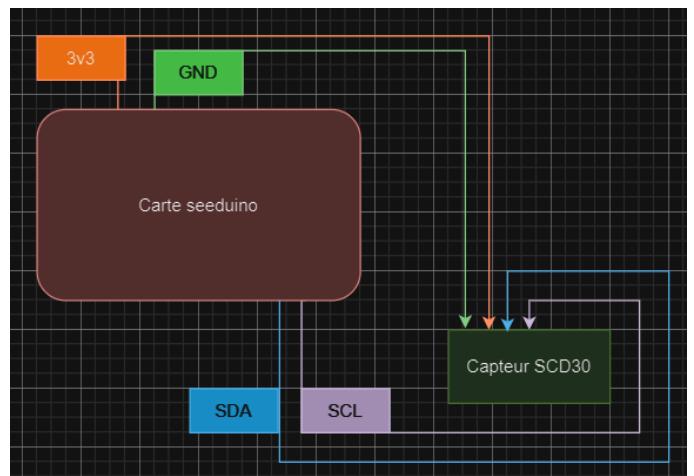
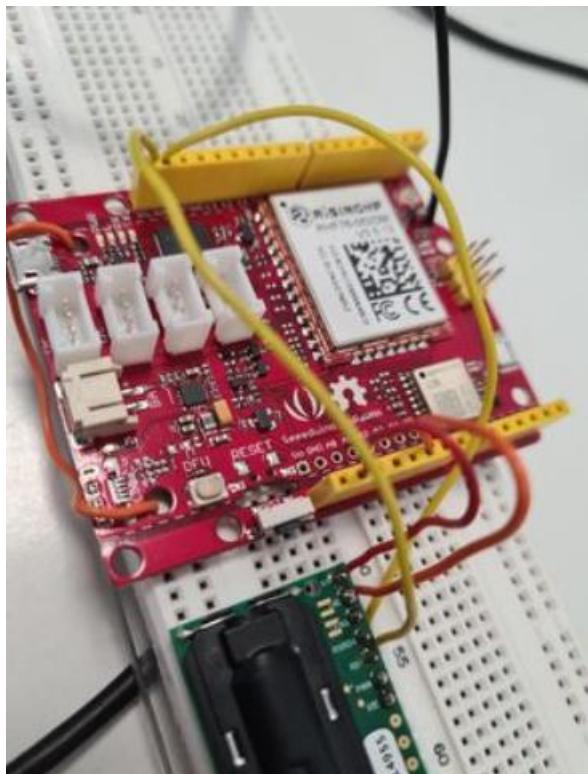
1. Récupération des données (page 1)
2. Envoie des données sur TTN (page 3)
3. Intégration sur un Dashboard (page 10)
4. Initiative
 - a. Mise en place de la passerelle (page 12)
 - b. Mise en place de Chirpstack (page 15)
 - c. Mise en place du capteur de Co2 (page 19)
 - d. Intégration (page 24)
5. BONUS (page 31)

1. RECUPERATION DES DONNEES :

Tout d'abord, pour commencer cette SAE, j'ai commencé par lire la documentation de mon capteur de Co2 et de ma carte seeduino.

(https://wiki.seeedstudio.com/Grove-CO2_Temperature_Humidity_Sensor-SCD30/)

Et lors de ma lecture, j'ai pu découvrir, comment le brancher :



On peut également trouver un code qui nous permet de récupérer les données du capteur :

```
#include "SCD30.h"
#if defined(ARDUINO_ARCH_AVR)
#pragma message("Defined architecture for ARDUINO_ARCH_AVR.")
#define SERIAL Serial
#elif defined(ARDUINO_ARCH_SAM)
#pragma message("Defined architecture for ARDUINO_ARCH_SAM.")
#define SERIAL SerialUSB
#elif defined(ARDUINO_ARCH_SAMD)
#pragma message("Defined architecture for ARDUINO_ARCH_SAMD.")
#define SERIAL SerialUSB
#elif defined(ARDUINO_ARCH_STM32F4)
#pragma message("Defined architecture for ARDUINO_ARCH_STM32F4.")
#define SERIAL SerialUSB
#else
#pragma message("Not found any architecture.")
#define SERIAL Serial
#endif

void setup()
{
    Wire.begin();
    SERIAL.begin(115200);
    SERIAL.println("SCD30 Raw Data");
    scd30.initialize();
}

void loop()
{
    float result[3] = {0};
    if(scd30.isAvailable())
    {
        scd30.getCarbonDioxideConcentration(result);
        SERIAL.print("Carbon Dioxide Concentration is: ");
        SERIAL.print(result[0]);
        SERIAL.println(" ppm");
        SERIAL.println(" ");
        SERIAL.print("Temperature = ");
        SERIAL.print(result[1]);
        SERIAL.println(" °C");
        SERIAL.println(" ");
        SERIAL.print("Humidity = ");
        SERIAL.print(result[2]);
        SERIAL.println(" %");
        SERIAL.println(" ");
        SERIAL.println(" ");
        SERIAL.println(" ");
    }
}
```

Et cela nous donne le Co2, la température et l'humidité

Carbon Dioxide Concentration is: 1200.60 ppm

Temperature = 22.37 °C

Humidity = 44.37 %

2. ENVOIE DES DONNEES SUR TTN :

Une fois que l'on a nos données du capteur, il faut s'occuper de TTN.

J'ai dû créer une application :

Applications > sae301-edouard

 **sae301-edouard**
ID: sae301-edouard

• Last activity 14 seconds ago ⓘ

General information

Application ID	sae301-edouard	
Created at	Oct 23, 2023 09:19:28	
Last updated at	Oct 23, 2023 09:19:28	

Et mon end device pour avoir les clés liées à mon applis :

 **eui-70b3d57ed0061f71**
ID: eui-70b3d57ed0061f71

↑ 36 ↓ 1 • Last activity 18 seconds ago ⓘ

[Overview](#) [Live data](#) [Messaging](#) [Location](#) [Payload formatters](#) [General settings](#)

General information

End device ID	eui-70b3d57ed0061f71	
Frequency plan	Europe 863-870 MHz (SF9 for RX2 - recommended...)	
LoRaWAN version	LoRaWAN Specification 1.0.0	
Regional Parameters version	TS001 Technical Specification 1.0.0	
Created at	Oct 23, 2023 09:51:44	

Activation information

AppEUI	8C F9 57 20 00 00 00 00	 
DevEUI	70 B3 D5 7E D0 06 1F 71	 
AppKey	  

Ensuite, lorsque l'on va sur la documentation de notre carte seeduino (https://wiki.seeedstudio.com/Seeeduino_LoRAWAN/), on peut voir un code qui permet d'envoyer un message sur TTN.

```
#include <LoRaWan.h>
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
unsigned char data[2] = {1, 2};
char buffer[256];

void setup(void)
{
    dht.begin();
    SerialUSB.begin(115200);
    lora.init();
    memset(buffer, 0, 256);
    lora.getVersion(buffer, 256, 1);
    SerialUSB.print(buffer);
    memset(buffer, 0, 256);
    lora.getId(buffer, 256, 1);
    SerialUSB.print(buffer);

    lora.setKey("2B7E151628AED2A6ABF7158809CF4F3C",
    "2B7E151628AED2A6ABF7158809CF4F3C", "2B7E151628AED2A6ABF7158809CF4F3C");

    lora.setDeciveMode(LWABP);
    lora.setDataRate(DR0, EU868);
    lora.setChannel(0, 867.7);
    lora.setChannel(1, 867.9);
    lora.setChannel(2, 868.8);
    lora.setReceiceWindowFirst(0, 867.7);
    lora.setReceiceWindowSecond(869.5, DR3);

    lora.setDutyCycle(false);
    lora.setJoinDutyCycle(false);
    lora.setPower(14);
}

void loop(void)
{
    bool result = false;
    delay(2000);
    int h = dht.readHumidity();
    int t = dht.readTemperature();
    result = lora.transferPacket("Hello World!", 10);
    data[0] = h;
    data[1] = t;
```

```

if(result)
{
    short length;
    short rssi;
    memset(buffer, 0, 256);
    length = lora.receivePacket(buffer, 256, &rssi);
    if(length)
    {
        SerialUSB.print("Length is: ");
        SerialUSB.println(length);
        SerialUSB.print("RSSI is: ");
        SerialUSB.println(rssi);
        SerialUSB.print("Data is: ");
        for(unsigned char i = 0; i < length; i++)
        {
            SerialUSB.print("0x");
            SerialUSB.print(buffer[i], HEX);
            SerialUSB.print(" ");
        }
        SerialUSB.println();
    }
}
}

```

Et avec les informations de notre device, sur notre applis TTN, il nous suffit de changer le « lora.setKey », puisque cela définit la direction du message à envoyer.

Une fois les bonne clé mises, cela se connecte et envoie le message.

```

09:13:09.906 -> +VER: 3.5.13
09:13:10.919 -> +ID: DevAddr, 26:0B:4A:63
09:13:10.919 -> +ID: DevEui, 70:B3:D5:7E:D0:06:1F:F5
09:13:10.919 -> +ID: AppEui, E7:0B:3D:57:ED:00:61:F7
09:13:11.087 -> +ID: DevEui, 70:B3:D5:7E:D0:06:1F:71
09:13:11.087 -> +ID: AppEui, 8C:F9:57:20:00:00:00:00
09:13:11.264 -> +KEY: APPKEY DB9E0937174BCA9A21AAC71083C352BB
09:13:12.260 -> +MODE: LWOTAA
09:13:13.357 -> +DR: EU868
09:13:14.409 -> +DR: DR0
09:13:14.409 -> +DR: EU868 DR0 SF12 BW125K
09:13:15.539 -> +CH: 0,868100000,DR0:DR5
09:13:16.624 -> +CH: 1,868300000,DR0:DR5
09:13:17.747 -> +CH: 2,868500000,DR0:DR5
09:13:18.858 -> +RXWIN1: 0,868100000
09:13:19.921 -> +RXWIN2: 869500000,DR3
09:13:21.041 -> +LW: DC, OFF, 0
09:13:22.144 -> +LW: JDC, OFF
09:13:23.227 -> +POWER: 14
09:13:24.363 -> +JOIN: Start
09:13:24.363 -> +JOIN: NORMAL

```

Cela veut dire qu'il se connecte.

Et après quelque modification, on peut envoyer nos données prise par notre capteur scd30 vers TTN.

Mon code :

```
#include <LoRaWan.h>
#include "SCD30.h"

// Déclaration de la sortie série en fonction de l'architecture
#if defined(ARDUINO_ARCH_AVR)
#pragma message("Defined architecture for ARDUINO_ARCH_AVR.")
#define SERIAL Serial
#elif defined(ARDUINO_ARCH_SAM)
#pragma message("Defined architecture for ARDUINO_ARCH_SAM.")
#define SERIAL SerialUSB
#elif defined(ARDUINO_ARCH_SAMD)
#pragma message("Defined architecture for ARDUINO_ARCH_SAMD.")
#define SERIAL SerialUSB
#elif defined(ARDUINO_ARCH_STM32F4)
#pragma message("Defined architecture for ARDUINO_ARCH_STM32F4.")
#define SERIAL SerialUSB
#else
#pragma message("Not found any architecture.")
#define SERIAL Serial
#endif

// Tampon pour stocker les données
char buffer[256];

// Paramètres TTN
const char *appEui = "8CF9572000000000";
const char *appKey = "DB9E0937174BCA9A21AAC71083C352BB";

void setup(void)
{
    SerialUSB.begin(115200);
    while (!SerialUSB);

    // Initialisation de la communication LoRa
    lora.init();

    // Affichage de la version du module LoRa
    memset(buffer, 0, 256);
    lora.getVersion(buffer, 256, 1);
    SerialUSB.print(buffer);

    // Affichage de l'ID du module LoRa
    memset(buffer, 0, 256);
    lora.getId(buffer, 256, 1);
    SerialUSB.print(buffer);
```

```

// Configuration des paramètres LoRa
lora.setId(NULL, "70B3D57ED0061F71", "8CF9572000000000");
lora.setKey(NULL, NULL, "DB9E0937174BCA9A21AAC71083C352BB");

lora.setDeciveMode(LWOTAA);
lora.setDataRate(DR0, EU868);

lora.setChannel(0, 868.1);
lora.setChannel(1, 868.3);
lora.setChannel(2, 868.5);

lora.setReceiceWindowFirst(0, 868.1);
lora.setReceiceWindowSecond(869.5, DR3);

lora.setDutyCycle(false);
lora.setJoinDutyCycle(false);

lora.setPower(14);

// Connexion au réseau LoRaWAN
while (!lora.setOTAAJoin(JOIN));

SerialUSB.println("SCD30 Raw Data");
scd30.initialize();
}

void loop(void)
{
    bool result = false;

    float scdresult[3] = {0};

    // Vérification de la disponibilité des données du capteur SCD30
    if (scd30.isAvailable())
    {
        // Récupération des concentrations de dioxyde de carbone du capteur SCD30
        scd30.getCarbonDioxideConcentration(scdresult);

        // Formatage des données pour la charge utile TTN
        String payload = String(scdresult[0]) + "," + String(scdresult[1]) + "," +
String(scdresult[2]);

        // Conversion de la charge utile en tableau de caractères
        char payloadCharArray[payload.length() + 1];
        payload.toCharArray(payloadCharArray, payload.length() + 1);

        // Envoi des données vers TTN
        result = lora.transferPacket(payloadCharArray, strlen(payloadCharArray));
    }
}

```

```

if (result)
{
    short length;
    short rssi;

    // Réception des données depuis TTN
    memset(buffer, 0, 256);
    length = lora.receivePacket(buffer, 256, &rssi);

    if (length)
    {
        // Affichage des informations de réception
        SerialUSB.print("Length is: ");
        SerialUSB.println(length);
        SerialUSB.print("RSSI is: ");
        SerialUSB.println(rssi);
        SerialUSB.print("Data is: ");
        for (unsigned char i = 0; i < length; i++)
        {
            SerialUSB.print("0x");
            SerialUSB.print(buffer[i], HEX);
            SerialUSB.print(" ");
        }
        SerialUSB.println();
    }

    // Affichage des données du capteur SCD30
    SERIAL.print("Carbon Dioxide Concentration is: ");
    SERIAL.print(scdresult[0]);
    SERIAL.println(" ppm");
    SERIAL.println(" ");
    SERIAL.print("Temperature = ");
    SERIAL.print(scdresult[1]);
    SERIAL.println(" °C");
    SERIAL.println(" ");
    SERIAL.print("Humidity = ");
    SERIAL.print(scdresult[2]);
    SERIAL.println(" %");
    SERIAL.println(" ");
    SERIAL.println(" ");
    SERIAL.println(" ");
}
}

```

Et donc désormais on reçoit :

↑ 09:36:57 eui-70b3d57ed0061f... Forward uplink data message DevAddr: 26 0B 17 9C <> [] 31 31 35 33 2E 39 36 2C 32 32 2E 32 36 2C 34 33 2E 35 37 ... <> [] FPort: 8

Le problème est que pour réduire la taille des paquets de données transmises, la passerelle LoRaWan convertit les données en base 64. Et TTN nous les affiche ensuite en ASCII.

Et donc on va mettre en place un décodeur pour pouvoir avoir des données humainement lisibles.

Décodeur :

```
function Decoder(bytes, port) {
    // Stockage des données décodées
    var decoded = {};
    // Chaîne de caractères pour stocker la conversion des bytes
    var result = "";

    // Parcours de tous les bytes dans le tableau 'bytes'
    for (i = 0; i < bytes.length; i++) {
        // Conversion de chaque byte en caractère et ajout à la chaîne 'result'
        result += String.fromCharCode(bytes[i]);
    }

    // Séparation de la chaîne 'result' par des virgules pour obtenir des
    // valeurs individuelles
    var values = result.split(',');

    // Analyse et assignation des valeurs aux champs correspondants de l'objet
    // 'decoded'
    decoded.CO2 = parseFloat(values[0]); // On suppose que CO2 est une valeur à
    // virgule flottante
    decoded.temperature = parseFloat(values[1]); // On suppose que la
    // température est une valeur à virgule flottante
    decoded.humidity = parseFloat(values[2]); // On suppose que l'humidité est
    // une valeur à virgule flottante

    // Renvoi de l'objet 'decoded' contenant les données décodées
    return decoded;
}
```

On va le mettre dans « Uplink » et la version javascript formater, ce qui va permettre de retourner les informations suivantes désormais :

↑ 09:40:26 eui-70b3d57ed0061f... Forward uplink data message DevAddr: 26 0B 17 9C ↳ ↲ Payload: { CO2: 1157.9, humidity: 43.88, temperature: 22.33 } 31 31 3E

3. INTEGRATION SUR UN DASHBOARD :

Une fois que je reçois mes données sur TTN, je peux passer au fait de les monter sur un Dashboard, histoire d'avoir une vision global de nos données sur un graphique.

J'ai choisis Tago.io, étant un site simple et avec une grande facilitation d'intégration et de configuration.

Sur TTN, lorsque l'on va dans integration -> Webhooks, on peut ajouter des intégrations. Et on retrouve Tago.io. Et d'ailleurs, on peut retrouver plus d'information dans la documentation de TTN.

Tout d'abord je vais créer le device sur tago,

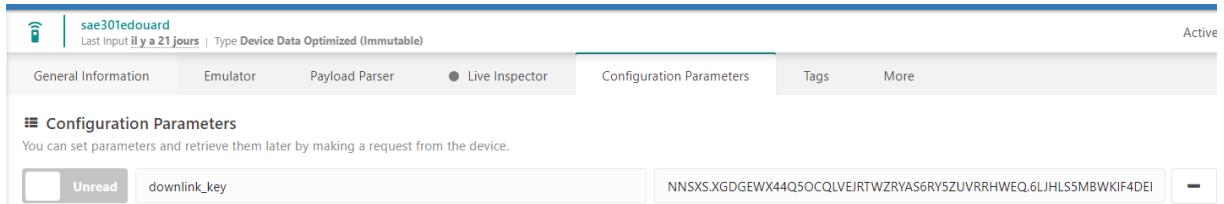
The screenshot shows the Tago.io dashboard with a blue header bar. Below it is a sidebar with icons for Home, Devices, Buckets, Files, Analysis, Actions, Explore, Access, and Users. The main area is titled 'Connector Selection' with the sub-instruction 'Browse through networks & connectors and create your device.' It includes a search bar 'search a connector for your dev...' and a list of available connectors: BeWhere, Cellio, HTTP / HTTPS, LOKA, LoRaWAN Actility, LoRaWAN CityKinect, LoRaWAN Everynet, LoRaWAN Kerlink, LoRaWAN Loriot, LoRaWAN MachineQ, LoRaWAN Orbiwise, LoRaWAN Senet, and LoRaWAN SenRa. A specific connector, 'Custom The Things Industries', is highlighted with a red box. This connector is described as 'Use this custom connector if your device connected through TTI doesn't show up in the...'.

Une fois créée, je vais devoir créer un token.

The screenshot shows the Tago.io device configuration interface for a device named 'sae301edouard'. The top bar shows the device name and a note 'Last Input il y a 21 jours | Type Device Data Optimized (Immutable)'. Below this are tabs for General Information, Emulator, Payload Parser, Live Inspector, and a partially visible tab. The 'General Information' tab is active, showing a 'Name' field containing 'sae301edouard' and a 'Connector' section with 'Custom TTI / TTN'. The 'Network' section shows 'LoRaWAN TTI/TTN v3'. At the bottom, there is a 'Token & Serial Number' section with fields for 'Token Name' (Token #2), 'Device EUI' (00-00-00-00-00-00-00-00), and a 'Generate' button. Below this are fields for 'Token #1' (70-b3-d5-7e-d0-06-1f-71) and three small icons.

Suite à ça les données seront stocké tout seul dans le bucket ayant le même nom que mon device.

Et avant d'aller y configurer, il nous faut l'API de notre device :



The screenshot shows the TTN Device Management interface for device 'sae301edouard'. The device was last input 'il y a 21 jours' (21 days ago) and is Type Device Data Optimized (Immutable). The 'Configuration Parameters' tab is selected, showing the 'downlink_key' parameter with the value 'NNSXS.XGDGEWX44Q50CQLVEJRTWZRYAS6RY5ZUVRRHWEQ.6LJHLS5MBWKIF4DEI'. The status is 'Active'.

Une fois cela fait, on doit aller y configurer sur l'intégration sur TTN :

Applications > sae301-edouard > Webhooks > Edit

Edit webhook for TagoIO

Integrate with TagoIO
[About TagoIO](#) | [Edit](#)

General settings

Webhook ID *

sae301edouard

Webhook format*

JSON

Base URL *

<https://ttn.middleware.tago.io>

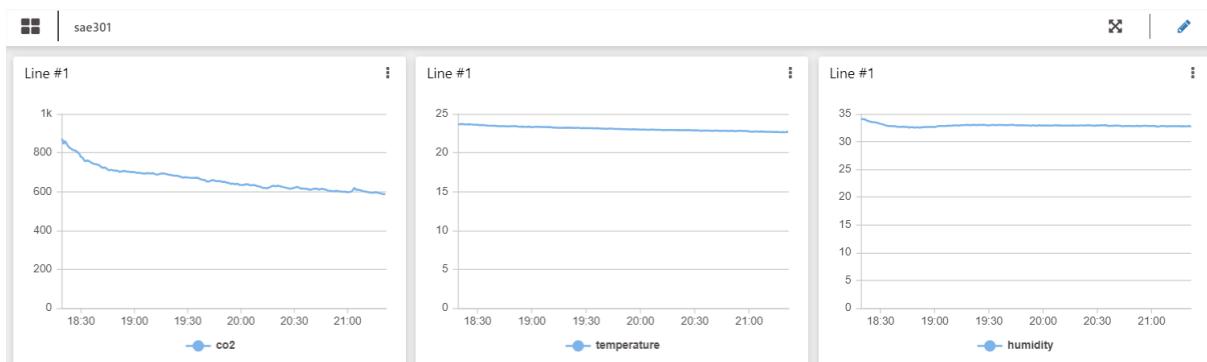
Downlink API key

NNSXS . XGDGEWX44Q50CQLVEJRTWZRYAS6RY5ZUVRRHWEQ . 6LJHLS5MBWKIF4DEI 

The API key will be provided to the endpoint using the "X-Downlink-Apikey" header

Et on renseigne l'URL de tago, puisque on est dessus.

Et une fois que l'on reçois les données, on a plus qu'a créé les Dashboards :



Associé à nos variables.

4. Initiative :

a. MISE EN PLACE DE LA PASSERELLE :

Tout d'abord, j'ai alimenté la passerelle, j'ai dû la configurer. D'après sa documentation on peut voir qu'elle a l'adresse IP « 10.130.1.1 ».

Donc en me connectant dessus par wifi,

Id wifi : dragino

Mdp wifi : dragino+dragino

et en utilisant le mot de passe de base du site (10.130.1.1) :

Username : **root**
Password : **dragino**

J'ai ainsi accès à l'interface de configuration :



Tout d'abord, puisque je suis connecté dessus, mais je n'ai pas internet.

Alors j'ai mis en place dans NETWORK -> WIFI, un WAN client rediriger vers ma LiveBox :

WiFi WAN Client Settings

Enable WiFi WAN Client

Host WiFi SSID

Livebox-5162

Passphrase

..... [Show](#)

WiFi Survey

Encryption

Choose WiFi SSID... [▼](#)

WPA2 [▼](#)

[Save&Apply](#) [Cancel](#) [Refresh](#)

Ce qui fait que la passerelle me passe aussi internet.

PS : (vu après toute cette 1^{ère} partie de l'innovation)

On peut également connecté la passerelle d'une autre manière, c'est-à-dire que d'au lieu avoir d'avoir la passerelle liée en filaire à ma LiveBox, je peux le branché, toujours en filaire, à mon 2^{ième} PC, celui-ci est en point d'accès mobile (partage de connexion), ce qu'il fait qu'il simule un routeur (ou ma LiveBox).

Point d'accès mobile (sur mon 2^{ième} PC) :

Propriétés du réseau			Modifier
Nom :	MSI	Mot de passe :	39691234
Bande :	2,4 GHz		
Appareils connectés : 1 sur 8			
Nom de l'appareil	Adresse IP	Adresse physique (MAC)	
dragino-255fe0	192.168.137.228	a8:40:41:25:f:e0	

Et en renseignant tout ça dans notre passerelle, elle peut se connecté :

WiFi WAN Client Settings

Enable WiFi WAN Client

Host WiFi SSID

MSI

Passphrase

39691234

[Hide](#)

WiFi Survey

Encryption

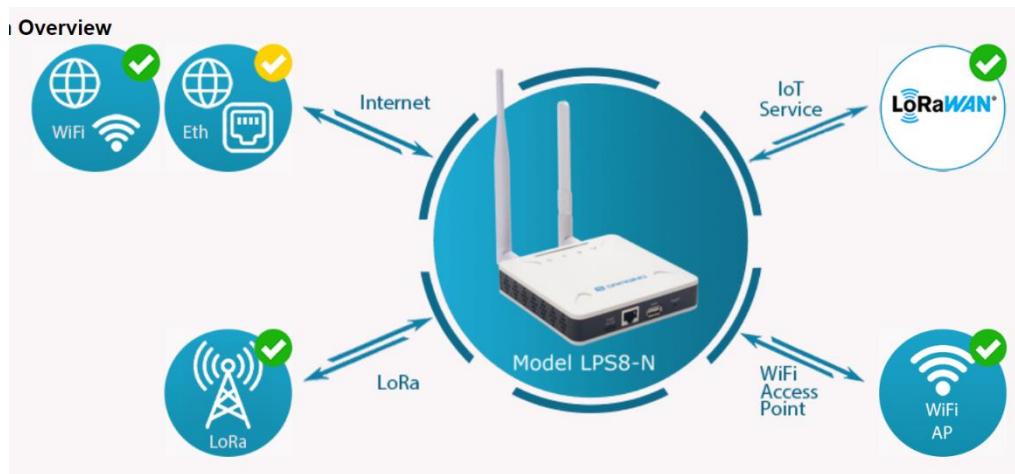
MSI (Ch: 13 Encr: WPA [▼](#)

WPA2 [▼](#)

WiFi status: OK. Click Refresh to check status.

[Save&Apply](#) [Cancel](#) [Refresh](#)

Et cela fait qu'il est connecté :



Partie sur Livebox

Lorsqu'on se connecte en ssh a la passerelle,

Mdp : dragino

On peut avoir son adresse IP sur son réseau (LAN) et l'adresse IP sur ma LiveBox (WAN).

```
Network
-----
Lan IP Address: Réseau passerelle
    inet addr:10.130.1.1 Bcast:10.130.1.255 Mask:255.255.255.0

Eth WAN IP Address: Réseau wifi personnel
    inet addr:192.168.1.24 Bcast:192.168.1.255 Mask:255.255.255.0
```

Et donc pour vérifier que cela marche on peut aller sur internet depuis la connexion avec celle-ci, ou on peut la ping depuis un autre pc sur le réseau wifi.

Puis j'ai dû rediriger la passerelle pour qu'elle renvoie des données en direction de mon pc
IP de mon pc connecté à la passerelle :

De plus, si on regarde à l'arrière de la passerelle, on peut voir l'ID unique de la passerelle : a84041ffff255fe0

Et en renseignant tout ça dans la configuration LoraWan et en choisissant réseau privée LoraWan.

LoRaWAN Configuration

General Settings

Email	<input type="text" value="dragino-255fe0@dragino.com"/>
Gateway EUI	<input type="text" value="a84041ffff255fe0"/>

Primary LoRaWAN Server

Service Provider	<input type="text" value="Custom / Private LoRaWAN"/> ▼	Server Address	<input type="text" value="10.130.1.239"/>
Uplink Port	<input type="text" value="1700"/>	Downlink Port	<input type="text" value="1700"/>

b. MISE EN PLACE DE CHIRPSTACK :

J'ai tout d'abord téléchargé docker desktop et Git



Ensuite j'ai fais la commande :

« git clone <https://github.com/brocaar/chirpstack-docker.git> »

Ce qui m'a permis d'installer mon serveur en montant les données depuis GitHub.

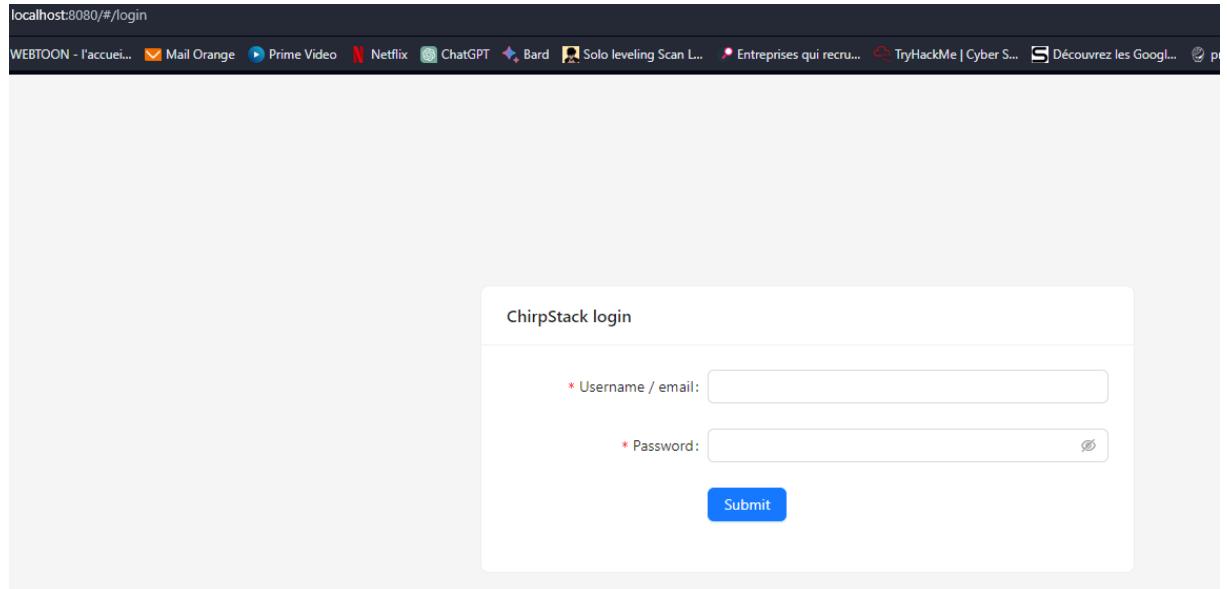
Puis en allant dans le dossier « chirpstack-docker » et en faisant :

« docker-compose-up »

Cela a fait en sorte de tout monter sur docker.

□	⌄	 chirpstack-docker		Running (7/7)	1.17%	1 day ago
□		 chirpstack-rest-api-1 03f95e6cc9a	chirpstack/chirpstack-rest-api:4	Running	0% 8090:8090	1 day ago
□		 chirpstack-1 3223d5ab0564	chirpstack/chirpstack:4	Running	0.22% 8080:8080	1 day ago
□		 chirpstack-gateway-bridge-basicstation-1 186b40d650f5	chirpstack/chirpstack-gateway-bridge:4	Running	0.2% 3001:3001	1 day ago
□		 chirpstack-gateway-bridge-1 95708891ef5c	chirpstack/chirpstack-gateway-bridge:4	Running	0.19% 1700:1700 (UDP)	1 day ago
□		 redis-1 d2833ce8fce0	redis:7-alpine	Running	0.24%	1 day ago
□		 mosquitto-1 4c248dbc4963	eclipse-mosquitto:2	Running	0.04% 1883:1883	1 day ago
□		 postgres-1 373ee65b40f2	postgres:14-alpine	Running	0.28%	1 day ago

Et on peut vérifier en allant sur le port 8080



Avec les mdp et ID par défauts (admin) j'ai pu me connecter.

Une fois la passerelle mise en place et Chirpstack qui marche, il faut les relier :

Add gateway

General Tags Metadata

* Name
Sae301-passerelle

Description

* Gateway ID (EUI64)
a84041ffff255fe0

ID passerelle

Et vu que la passerelle avait bien été paramétrée la passerelle va se connecter avec Chirpstack et je vais la voir en « Online ».

Tenants / ChirpStack / Gateways					
Gateways					
<input type="checkbox"/>	Last seen	Gateway ID	Name	Region ID	Region common-name
<input type="checkbox"/>	● Online 2023-12-08 13:36:56	a84041ffff255fe0	dragino-gw	eu868	EU868

Ensuite avant de monter les données, on va créer l'application.

D'abord on va faire le device, où il va y avoir le décodeur.

* Name
sae301-device-profile

Description

* Region
EU868 Region configuration ⓘ

(On met toujours en fréquence 868 puisqu'on est en France)

On va également rajouter le décodage :

General Join (OTAA / ABP) Class-B Class-C **Codec** Relay Tags Measurements

Payload codec ⓘ
JavaScript functions

Codec functions

```
function decodeUplink(input) { // Chirpstack v4
    return {
        data: Decode(input.fPort, input.bytes, input.variables)
    };
}

function Decode(fPort, bytes, variables) {
    var data = {"Co2": "",
               "Temperature":"",
               "Humidite":""};

    // Parcours les 6 premiers octets et les ajouter à la chaîne
    for (var i = 0; i < Math.min(6, bytes.length); i++) {
        data.Co2 += String.fromCharCode(bytes[i]);
    }
    for (var i = 7; i < Math.min(12, bytes.length); i++) {
        data.Temperature += String.fromCharCode(bytes[i]);
    }
    for (var i = 13; i < Math.min(18, bytes.length); i++) {
        data.Humidite += String.fromCharCode(bytes[i]);
    }
    return data;
}
```

Et cela va créer automatiquement des « Measurements » lors de la validation, que l'on nommera par la suite.

The screenshot shows the ChirpStack device configuration interface. The top navigation bar includes tabs for General, Join (OTAA / ABP), Class-B, Class-C, Codec, Relay, Tags, and Measurements, with the Measurements tab currently selected. Below the tabs, there is a note about aggregating decoded device measurements. A section titled "Automatically detect measurement keys" has a toggle switch that is turned on. Below this, three measurement entries are listed: Co2 (Unknown / unset), Humidite (Unknown / unset), and Temperature (Unknown / unset). Each entry consists of a name input field, a type dropdown, and a unit input field.

Une fois le device créé, on va créer l'application et l'implémenter dedans.

The screenshot shows the ChirpStack application configuration interface for the device "sae301-edouard". The top navigation bar includes tabs for Tenants, ChirpStack, Applications, and the current application (sae301-edouard). A red "Delete application" button is visible. Below the tabs, there are tabs for Devices, Multicast groups, Relays, Application configuration, and Integrations, with the Application configuration tab selected. A "Selected devices" button is also present. The main table lists the device details: Last seen (2023-12-09 12:40:49), DevEUI (1c8ba037b372e8e6), Name (sae301-device), and Device profile (sae301-device-profile).

Et pour activer le lien entre le device et l'application, on va aller dans l'application configuration et y enregistrer.

The screenshot shows the ChirpStack application configuration form. The top navigation bar includes tabs for Devices, Multicast groups, Relays, Application configuration, and Integrations, with the Application configuration tab selected. The form fields include a required "Name" field containing "sae301-edouard" and a "Description" field which is empty. A blue "Submit" button is at the bottom.

Maintenant que la passerelle est liée à Chirpstack et que Chirpstack a son application. Il nous manque plus qu'à monter les données.

c. MISE EN PLACE DU CAPTEUR DE CO2 :

J'ai réutiliser le code que j'ai fait pour monter les données sur TTN, mais j'ai des choses à modifier, et ce sont des informations de Chirpstack que je dois récupérer.

le « devEUI », le « appKEY » et le « appEUI » :

The screenshot shows the Chirpstack Device configuration interface. At the top, the path is: Tenants / Chirpstack / Applications / sae301-edouard / Devices / sae301-device. The 'Configuration' tab is selected. Under the 'Device' tab, the 'Name' field contains 'sae301-device'. The 'Description' field is empty. The 'Device EUI (EUI64)' field contains '1c8ba037b372e8e6' and the 'Join EUI (EUI64)' field contains 'e6c5301b79311336'. Below this, the 'OTAA keys' tab is selected. The 'Application key' field contains '428ce839dc992fcfaeb64d6b38fa278'. A 'Submit' button is at the bottom.

Une fois dans mon code cela donne :

```
#include <LoRaWan.h>
#include "SCD30.h"

#if defined(ARDUINO_ARCH_AVR)
#pragma message("Defined architecture for ARDUINO_ARCH_AVR.")
#define SERIAL Serial
#elif defined(ARDUINO_ARCH_SAM)
#pragma message("Defined architecture for ARDUINO_ARCH_SAM.")
#define SERIAL SerialUSB
#elif defined(ARDUINO_ARCH_SAMD)
```

```

#pragma message("Defined architecture for ARDUINO_ARCH_SAMD.")
#define SERIAL SerialUSB
#elif defined(ARDUINO_ARCH_STM32F4)
#pragma message("Defined architecture for ARDUINO_ARCH_STM32F4.")
#define SERIAL SerialUSB
#else
#pragma message("Not found any architecture.")
#define SERIAL Serial
#endif

char buffer[256];

// Chirpstack parameters
const char *appEui = "e6c5301b79311336";
const char *appKey = "428ce839dcd992fcfaeb64d6b38fa278";

void setup(void)
{
    SerialUSB.begin(115200);
    while(!SerialUSB);

    lora.init();

    memset(buffer, 0, 256);
    lora.getVersion(buffer, 256, 1);
    SerialUSB.print(buffer);

    memset(buffer, 0, 256);
    lora.getId(buffer, 256, 1);
    SerialUSB.print(buffer);           devEUI      appEUI

    lora.setId(NULL, "1c8ba037b372e8e6", "e6c5301b79311336");           appKEY
    lora.setKey(NULL, NULL, "428ce839dcd992fcfaeb64d6b38fa278");

    lora.setDeciveMode(LWOTAA);
    lora.setDataRate(DR0, EU868);           appKEY

    //Fréquence attribuées
    lora.setChannel(0, 868.1);
    lora.setChannel(1, 868.3);
    lora.setChannel(2, 868.5);

    lora.setReceiceWindowFirst(0, 868.1);
    lora.setReceiceWindowSecond(869.5, DR3);

    lora.setDutyCycle(false);
    lora.setJoinDutyCycle(false);

    lora.setPower(14);

```

```

while(!lora.setOTAAJoin(JOIN));

//initialise les données du capteur
SerialUSB.println("SCD30 Raw Data");
scd30.initialize();
}

void loop(void)
{
    bool result = false;

    float scdresult[3] = {0};

    if(scd30.isAvailable())
    {
        scd30.getCarbonDioxideConcentration(scdresult);

        // definis le payload avec les données du capteur
        // scdresult[0] = Co2
        // scdresult[1] = Temperature
        // scdresult[2] = Humidité
        String payload = String(scdresult[0]) + "," + String(scdresult[1]) +
        "," + String(scdresult[2]);

        // Convertis le payload en Char
        char payloadCharArray[payload.length() + 1];
        payload.toCharArray(payloadCharArray, payload.length() + 1);

        // Envoie des données
        result = lora.transferPacket(payloadCharArray,
        strlen(payloadCharArray));

        if(result)
        {
            short length;
            short rssi;

            memset(buffer, 0, 256);
            length = lora.receivePacket(buffer, 256, &rssi);

            if(length)
            {
                SerialUSB.print("Length is: ");
                SerialUSB.println(length);
                SerialUSB.print("RSSI is: ");
                SerialUSB.println(rssi);
                SerialUSB.print("Data is: ");
                for(unsigned char i = 0; i < length; i++)

```

```

    {
        SerialUSB.print("0x");
        SerialUSB.print(buffer[i], HEX);
        SerialUSB.print(" ");
    }
    SerialUSB.println();
}

//écrit les résultats du capteurs sur le moniteur serial
SERIAL.print("Carbon Dioxide Concentration is: ");
SERIAL.print(scdresult[0]);
SERIAL.println(" ppm");
SERIAL.println(" ");
SERIAL.print("Temperature = ");
SERIAL.print(scdresult[1]);
SERIAL.println(" °C");
SERIAL.println(" ");
SERIAL.print("Humidity = ");
SERIAL.print(scdresult[2]);
SERIAL.println(" %");
SERIAL.println(" ");
SERIAL.println(" ");
SERIAL.println(" ");
}
}

```

Ce qui fait que maintenant je reçois les données sur Chirpstack dans l'onglet « events » qui est la lecture des données en directe.

The screenshot shows the Chirpstack interface with the 'Events' tab selected. The page displays a list of received frames from a device named 'sae301-device'. Each frame entry includes the timestamp, an uplink button, frame details (DR: 5, Data: hex string, FCnt: 164, FPort: 8), and a small icon.

Timestamp	Action	Frame Details
2023-12-09 13:02:59	Q up	DR: 5 Data: 313735312e32392c32302e37302c36302e3934 FCnt: 164 FPort: 8
2023-12-09 13:02:40	Q up	DR: 5 Data: 313735382e33332c32302e37312c36312e3133 FCnt: 163 FPort: 8
2023-12-09 13:02:21	Q up	DR: 5 Data: 313736302e34312c32302e36372c36312e3237 FCnt: 162 FPort: 8
2023-12-09 13:02:02	Q up	DR: 5 Data: 313736312e37322c32302e36392c36312e3136 FCnt: 161 FPort: 8
2023-12-09 13:01:43	Q up	DR: 5 Data: 313735392e38322c32302e37312c36312e3138 FCnt: 160 FPort: 8
2023-12-09 13:01:24	Q up	DR: 5 Data: 313736302e37382c32302e37332c36312e3232 FCnt: 159 FPort: 8
2023-12-09 13:01:05	Q up	DR: 5 Data: 313736312e37342c32302e37302c36312e3238 FCnt: 158 FPort: 8

On peut donc voir les données arrivé (up), les datas, et sur quelle port (FPort).

```

Tenants / ChirpStack / Applications / sae301-edouard / Devices / sae301-device
sae301-device device eui: 1c8ba037b372e8e6

Dashboard Configuration OTAA keys Activation Queue Events LoRaWAN frames

Données affiché ↓
↓
2023-12-09 13:05:50 [up] DR: 5 Data: 313732312e39372c32302e36392c36302e3533 FCnt: 173
2023-12-09 13:05:51 [up] DR: 5 Data: 31373302e39322c32302e37332c36302e3933 FCnt: 172
2023-12-09 13:05:12 [up] DR: 5 Data: 313733322e33372c32302e37312c36312e3935 FCnt: 171

Données reçues ↑
↑

```

Alors oui, les données ne sont pas lisible, même avec le décodeur. Et en fait si,

C'est-à-dire que la passerelle LoraWan, prends les données et les convertis en base 64 pour les envoyer à mon serveur. Puis le serveur (Chirpstack), convertis les données reçu en ASCII (dans les bytes) puis en Hexadécimal pour me les afficher.

Pour avoir des données lisible humainement, mon décodeur va décoder les données reçu après avoir été convertis en ASCII.

Et donc on va avoir nos données, mais elle seront pas affiché dans « event » mais plutôt, juste en dessous de data dans l'onglet « object » :

```

Tenants / ChirpStack / Applications / sae301-edouard / Devices / sae301-device
sae301-device device eui: 1c8ba037b372e8e6

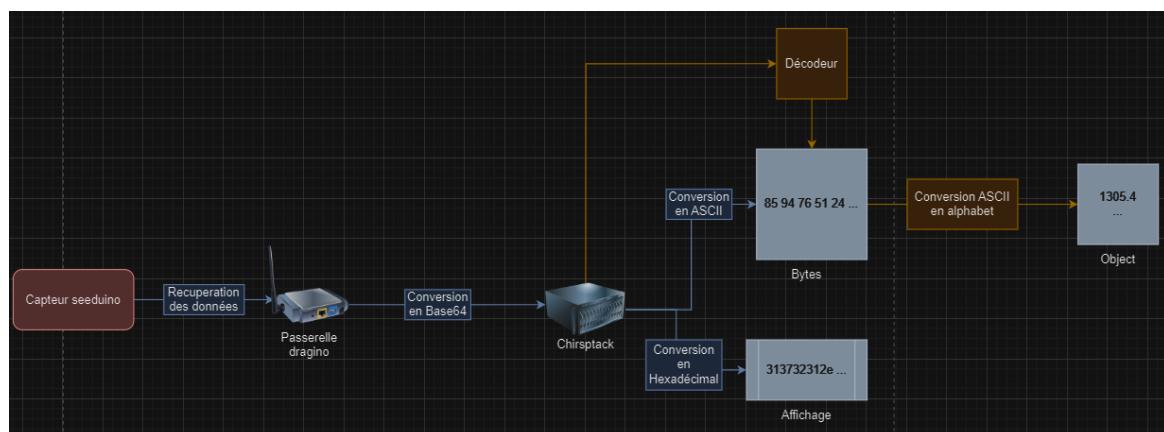
Dashboard Configuration OTAA keys Activation Queue Events LoRaWAN frames

2023-12-09 13:13:07 [up] DR: 5 Data: 313731382e37392c32302e37362c36302e3639 FCnt: 196
2023-12-09 13:12:48 [up] DR: 5 Data: 313732342e34322c32302e37342c36302e3634 FCnt: 195
2023-12-09 13:12:29 [up] DR: 5 Data: 313732392e32372c32302e37342c36302e3635 FCnt: 194
2023-12-09 13:12:10 [up] DR: 5 Data: 313733312e35342c32302e37362c36302e3739 FCnt: 193
2023-12-09 13:11:51 [up] DR: 5 Data: 313732352e30312c32302e37372c36302e3734 FCnt: 192

object: [ 3 keys
  Temperature: "60.9"
  Humidite: "60.9"
  Co2: "1730.9" ]
extinfo: [ 1 item
  0: [ 10 keys
    ... ] ]

```

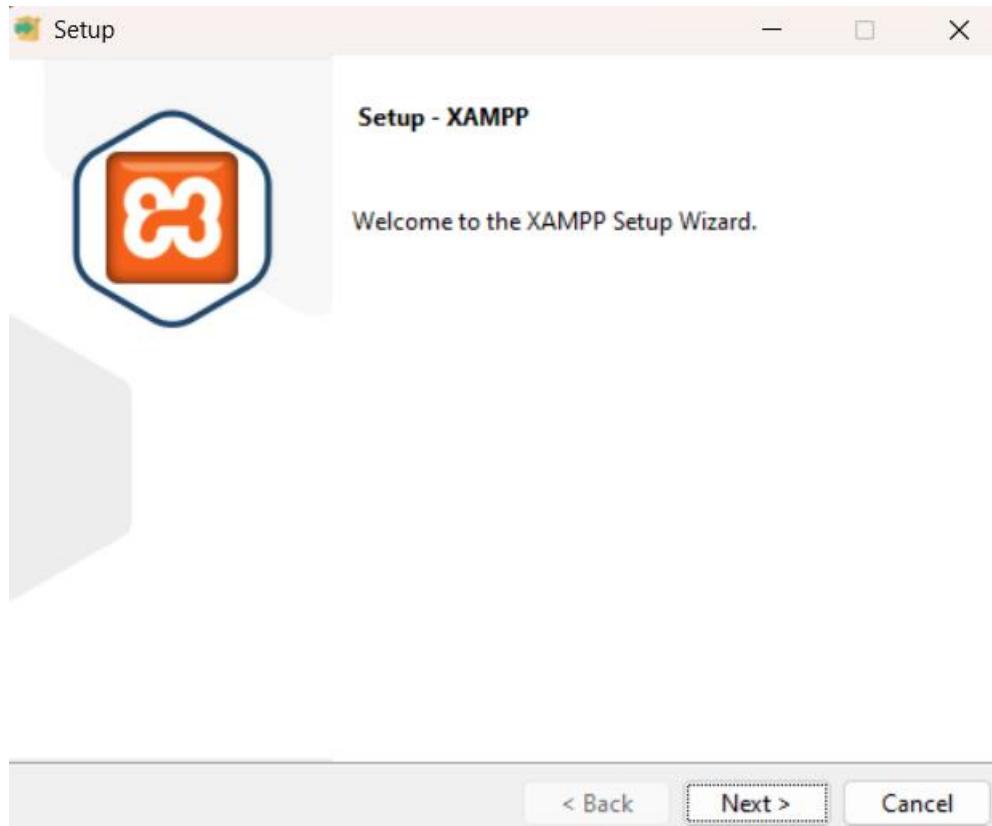
Pour résumer en image :



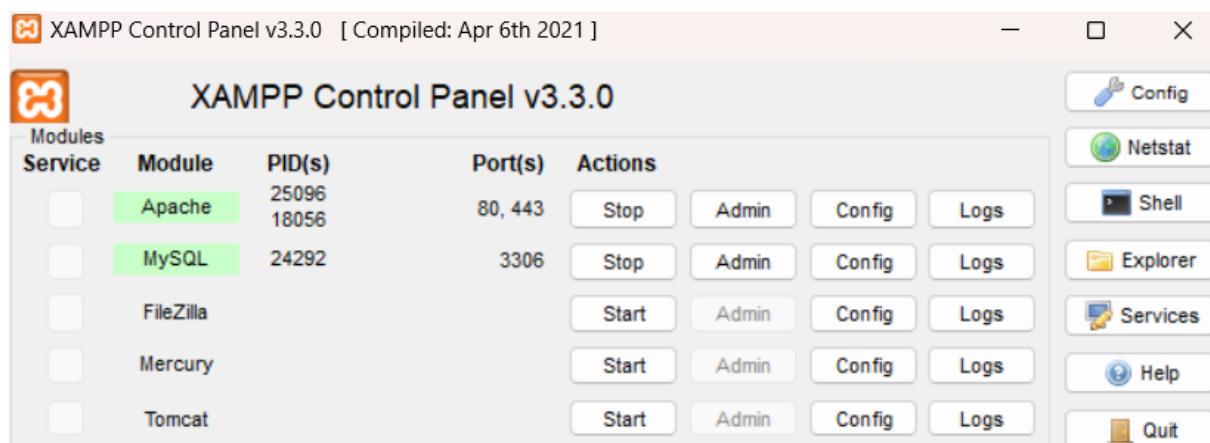
Maintenant que j'ai les données, il manque plus qu'à les monter sur un Dashboard

d. INTEGRATION :

Tout d'abord, j'ai téléchargé XAMPP pour installer et remplir le rôle du serveur apache (qui va permettre d'exécuter mon code php) et de la base de donnée SQL.



Une fois installer, j'ai lancer les module Important



MySQL pour phpMyAdmin et Apache pour le code PHP.

BASE DE DONNEE :

Une fois start, j'ai créé une base de donnée. Suite à ça, j'ai créé une table dans cette base de donnée, et enfin la structure de la table avec les différentes colonnes important pour mener à bien ce projet.

phpMyAdmin

Parcourir Structure SQL Rechercher Insérer Exporter Importer Priviléges Opérations Suivi Déclenç.

Nouvelles bases de données Récentes Préférées

Structure de table Vue relationnelle

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	ID	bigint(255)		Non	Aucun(e)			AUTO_INCREMENT	Modifier Supprimer Plus
2	Co2	float		Non	Aucun(e)				Modifier Supprimer Plus
3	Temperature	float		Non	Aucun(e)				Modifier Supprimer Plus
4	Humidité	float		Non	Aucun(e)				Modifier Supprimer Plus
5	Heure	timestamp		Non	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()		Modifier Supprimer Plus

Tout cocher Avec la sélection : Parcourir Modifier Supprimer Primaire Unique Index Spatial Texte entier

Supprimer de la liste centrale de colonnes

Imprimer Suggérer des optimisations de structure Suivre la table Déplacer des colonnes Normaliser

Ajouter 1 colonne(s) après Heure Exécuter

Index

Action	Non de l'index	Type	Unique	Comprimé	Colonne	Cardinalité	Interclassement	Null	Commentaire
Éditer Renommer Supprimer	PRIMARY	BTREE	Oui	Non	ID	297	A	Non	

Une fois la base de données configurée il faut diriger l'intégration de Chirpstack vers phpMyAdmin :

Pour information et rappeler :

80 = port apache

3306 = port de la base de donné (SQL)

Adresse IP sur le réseau dragino :

Carte réseau sans fil Wi-Fi :

```
Suffixe DNS propre à la connexion. . . . .  
Adresse IPv6 de liaison locale. . . . .  
Adresse IPv4. . . . .  
Masque de sous-réseau. . . . .  
Passerelle par défaut. . . . .
```

L'intégration sur Chirpstack :

Tenants / ChirpStack / Applications / **sae301-edouard**

sae301-edouard application id: 8fd32147-3fea-4bc0-99e1-7a6050d79b27

Devices Multicast groups Relays Application configuration **Integrations**

Update HTTP integration

* Payload encoding
JSON

* Event endpoint URL(s) ⓘ
http://10.130.1.239:80/transfert.php

Headers

+ Add header

Submit

Cela va passer par mon code (transfert.php) qui va permettre d'y insérer dans ma base de donnée.

Que l'on met dans C:\xampp\htdocs, ce sont la base des dossiers de XAMPP (et donc également de ma base de donnée).

Mon code (transfert.php) :

```
<?php

header("Content-type: application/json");
$json = file_get_contents("php://input");

$obj = json_decode($json);

// Decode la valeur de "data"
$decodedData = base64_decode($obj->data);

// Decoupe les valeurs séparées par des virgules
list($co2, $temperature, $humidity) = explode(',', $decodedData);

$server      = "localhost:3306";
$username    = "root";
$password    = "";
$DB          = "sae301";

$conn = new mysqli($server, $username, $password, $DB);
if ($conn->connect_error) {
    die("Connexion échouée: " . $conn->connect_error);
}

// Pour éviter les attaques par injection SQL
$escapedCo2 = $conn->real_escape_string($co2);
$escapedTemperature = $conn->real_escape_string($temperature);
$escapedHumidity = $conn->real_escape_string($humidity);

// Construction insertion
$insert = "INSERT INTO donnee(Co2, Temperature, Humidité) VALUES
('$escapedCo2', '$escapedTemperature', '$escapedHumidity')";

// Execute l'insertion
if ($conn->query($insert) === TRUE) {
    echo json_encode(["success" => true]);
} else {
    echo json_encode(["success" => false, "error" => $conn->error]);
}

// Ferme la connexion
$conn->close();

?>
```

Une fois le code mis et enregistrer, on peut voir que les données arrive bien sur ma base de donnée.

The screenshot shows the phpMyAdmin interface for the 'sae301' database. On the left, the database structure is visible, including the 'donnee' table which has columns for ID, Co2, Temperature, Humidite, and Heure. The main area displays a grid of data with 25 rows. A red box highlights the first few rows of the data grid, showing entries like ID 590 with Co2 1056.07 and Temperature 23.49. The data spans from 2023-12-14 00:57:23 to 2023-12-14 00:52:38.

ID	Co2	Temperature	Humidite	Heure
590	1056.07	23.49	48.95	2023-12-14 00:57:23
589	1056.94	23.48	48.94	2023-12-14 00:57:04
588	1058.67	23.44	48.94	2023-12-14 00:56:45
587	1060.54	23.48	48.98	2023-12-14 00:56:26
586	1062.03	23.49	48.95	2023-12-14 00:56:07
585	1062.05	23.47	49.02	2023-12-14 00:55:48
584	1065.11	23.47	49	2023-12-14 00:55:29
583	1065.99	23.47	49	2023-12-14 00:55:10
582	1069.54	23.49	48.96	2023-12-14 00:54:51
581	1069.07	23.49	48.95	2023-12-14 00:54:32
580	1070.93	23.49	48.99	2023-12-14 00:54:13
579	1069.48	23.48	49.06	2023-12-14 00:53:54
578	1070.61	23.49	48.97	2023-12-14 00:53:35
577	1076.26	23.49	49	2023-12-14 00:53:16
576	1071.72	23.49	48.98	2023-12-14 00:52:57
575	1072.22	23.51	48.97	2023-12-14 00:52:38
574	1070.19	23.51	48.98	2023-12-14 00:52:19

Il nous reste alors plus qu'a le monter sur Grafana.

Il faut alors installer Grafana, d'après la documentation de Grafana Labs :

<https://grafana.com/docs/grafana/latest/setup-grafana/installation/windows/>

et après avoir fait ces étapes on a :

The screenshot shows a file explorer window displaying the contents of the 'grafana-10.2.2.windows-amd64' folder. Inside, there are three files: 'grafana-cli', 'grafana-server', and 'grafana'. All three files were last modified on 20/11/2023 at 12:31. The 'grafana' file is the largest at 356434 Ko.

Nom	Modifié le	Type	Taille
grafana-cli	20/11/2023 12:31	Application	2747 Ko
grafana-server	20/11/2023 12:31	Application	2747 Ko
grafana	20/11/2023 12:31	Application	356434 Ko

Et lorsque l'on lance « grafana-server », cela lance donc le serveur. Et nous permet d'aller sur la version internet sur le port 3000 :

Welcome to Grafana

Need help? Documentation Tutorials Contact us

Basic
The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL DATA SOURCE AND DASHBOARDS
Grafana fundamentals
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

COMPLETE
Add your first data source
Create your first dashboard

Il faut ajouter une source de donnée, « data sources » et je rentre MySQL :

Type: MySQL

Settings

Alerting supported

Name MySQL

Before you can use the MySQL data source, you must configure a connection.

Connection Port mysql

Host URL * *

Authentication

Database name sae301

Username root

Additional settings

Session timezone Europe/Berlin or +02:00

Min time interval 1m

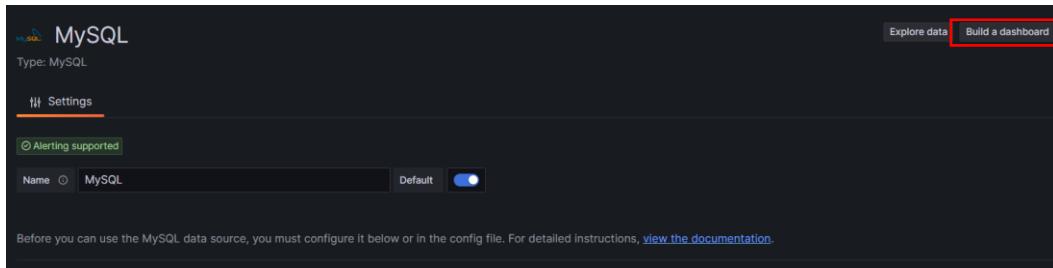
Database Connection OK

Next, you can start to visualize data by [building a dashboard](#), or by querying data in the [Explore view](#).

Suite à ce message on comprend que le lien entre PhpMyAdmin et Grafana est fait.

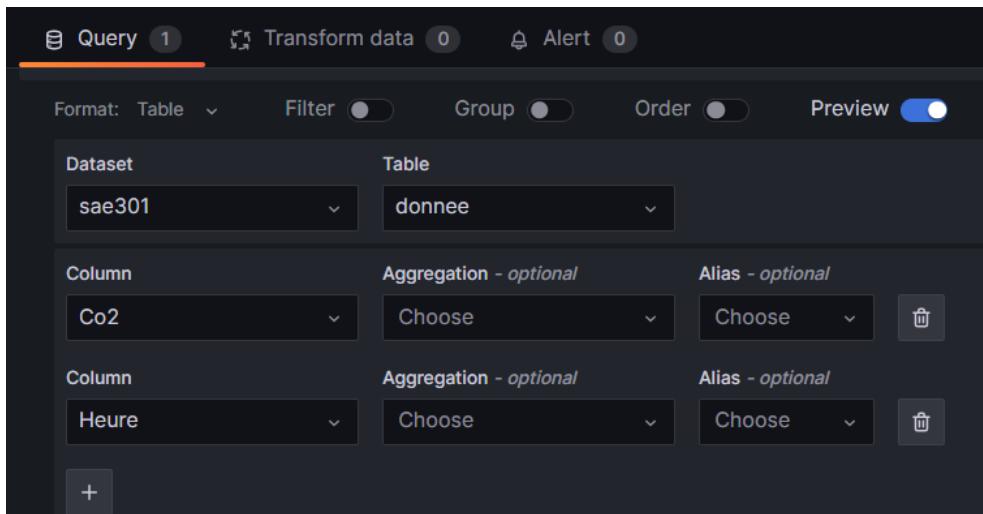
GRAFANA :

Et en haut à gauche lors de la configuration, on peut voir « build a dashboard » :



The screenshot shows the MySQL configuration screen in Grafana. At the top right, there are two buttons: 'Explore data' and 'Build a dashboard'. The 'Build a dashboard' button is highlighted with a red box. Below these buttons, there are sections for 'Alerting supported', 'Name' (set to 'MySQL'), and a toggle switch for 'Default'. A note at the bottom states: 'Before you can use the MySQL data source, you must configure it below or in the config file. For detailed instructions, [view the documentation](#)'.

Une fois dessus, il faut le paramétrer :



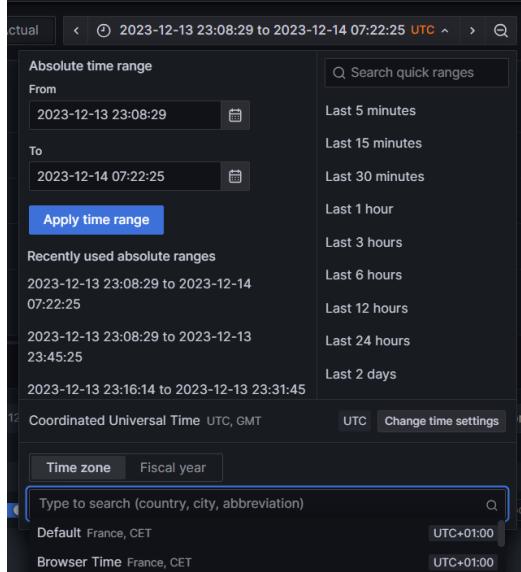
The screenshot shows the Grafana Query editor. At the top, there are tabs for 'Query' (with 1 item), 'Transform data' (with 0 items), and 'Alert' (with 0 items). Below these are buttons for 'Format: Table', 'Filter', 'Group', 'Order', and 'Preview'. The main area is divided into two sections: 'Dataset' and 'Table'. In the 'Dataset' section, 'sae301' is selected. In the 'Table' section, 'donnee' is selected. Under 'Column' for the first row, 'Co2' is selected. Under 'Column' for the second row, 'Heure' is selected. Each row has dropdowns for 'Aggregation - optional' and 'Alias - optional', and a trash icon to remove the row. A plus sign (+) button is located at the bottom left.

Cela signifie que dans la base de donnée « sae301 », dans la table « donnee » on appelle les colonnes « Co2 » et « Heure ».

Co2 puisque c'est ce qui nous intéresse

Et Heure, puisqu'il nous faut un repère de temps sur un graphique.

Avant de lancer le Dashboard, il faut pas oublier de modifier l'heure, Pour la mettre sur notre fuseau horaire :



The screenshot shows the Grafana time range selector. It includes fields for 'From' (2023-12-13 23:08:29) and 'To' (2023-12-14 07:22:25), a 'Search quick ranges' input, and a 'Apply time range' button. Below these are sections for 'Recently used absolute ranges' and 'Absolute time range'. The 'Absolute time range' section has 'From' and 'To' fields with dropdowns for selecting dates and times. To the right of these fields is a list of quick range options: 'Last 5 minutes', 'Last 15 minutes', 'Last 30 minutes', 'Last 1 hour', 'Last 3 hours', 'Last 6 hours', 'Last 12 hours', 'Last 24 hours', and 'Last 2 days'. At the bottom, there are buttons for 'Coordinated Universal Time UTC', 'UTC', 'Change time settings', and a search bar for 'Time zone' and 'Fiscal year'.

Et modifier la limite de donnée sur le graphique :

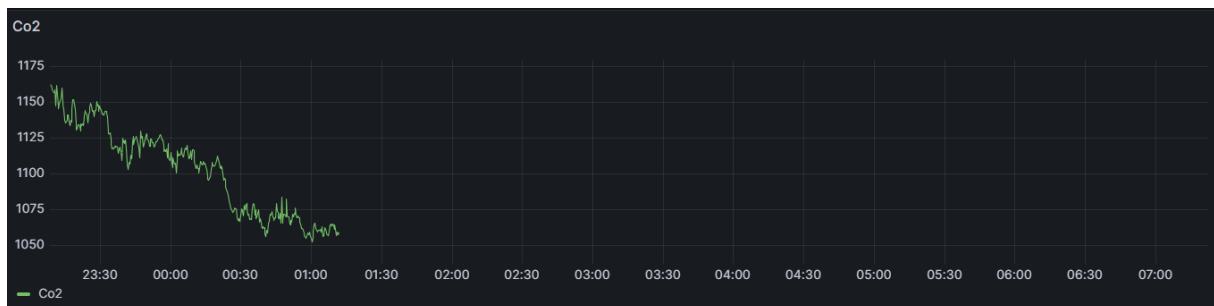
Pour ce faire il faut aller dans « code » :

The screenshot shows the Grafana Query editor interface. At the top, there are tabs for 'Query' (which is active), 'Transform data', and 'Alert'. Below that, the 'Data source' is set to 'MySQL'. The 'Query options' section shows 'MD = auto = 1215' and 'Interval = 1m'. On the right, there are buttons for 'Run query', 'Builder', and 'Code' (which is highlighted with a red box). The main area shows a dataset selection dropdown with 'sae301' and a table selection dropdown with 'donnee'. Below these are 'Format' (set to 'Table'), 'Filter', 'Group', 'Order', and 'Preview' buttons.

Et modifier la « LIMIT » :

The screenshot shows the Grafana Query editor with the 'Code' tab selected. The query in the editor is: '1 SELECT Co2, Heure FROM sae301.donnee LIMIT 5100'. The 'Run query' button is visible on the right.

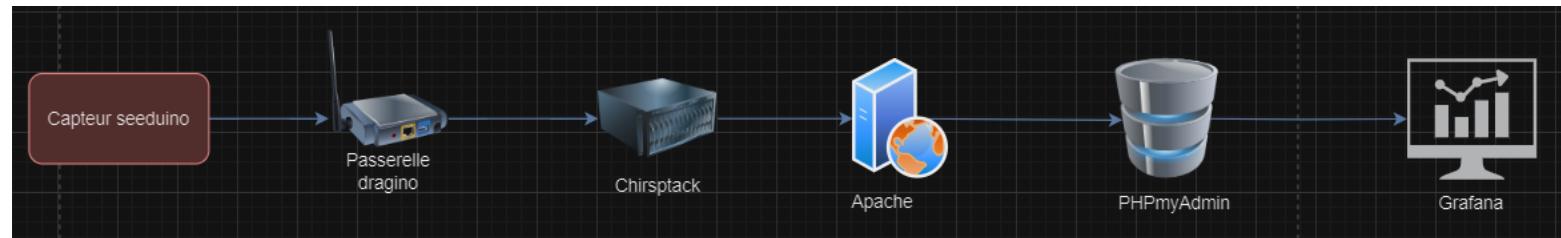
Et enfin lancer le Dashboard en cliquant sur « Run query » :



Une fois cette dernière action répéter pour les 3 valeurs, on obtient :



Pour résumer on a donc au final si on image on a :



5. BONUS

a. Intégration sur DataCake

Ps : fais avant l'integration sur grafana

Maintenant que nous recevons nos données, et qu'elles sont lisible, il faut les monter sur un Dashboard.

J'ai choisi l'intégration de type « HTTP »

Tenants / ChirpStack / Applications / **sae301-edouard**

sae301-edouard application id: 8fd32147-3fea-4bc0-99e1-7a6050d79b27

Devices Multicast groups Relays Application configuration **Integrations**

HTTP

HTTP : //

The HTTP integration forwards events to a user-configurable endpoint as POST requests.

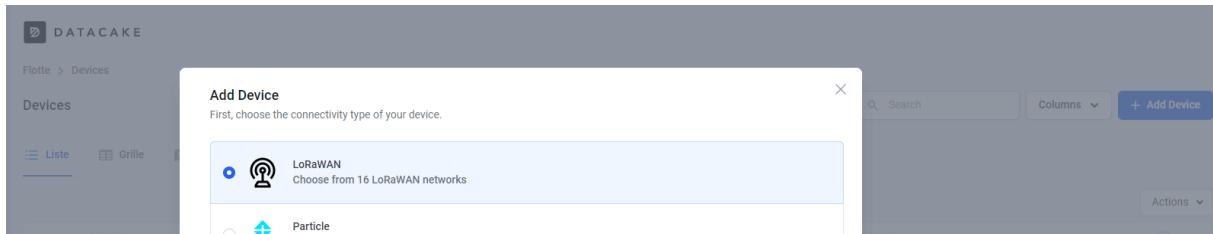
MQTT



The MQTT integration

Dashboard :

Une fois que l'on a créé un compte sur leur site, on va devoir créer un device :



On choisit notre réseau (LoRaWAN)

Ajouter un appareil LoRaWAN

STEP 1 Product STEP 2 Network Server STEP 3 Devices STEP 4 Plan

Datacake Product
You can add devices to an existing product on Datacake, create a new empty product or start with one of the templates. Products allow you to share the same configuration (fields, dashboard and more) between devices.

New Product from template
Create new product from a template

Existing Product
Add devices to an existing product

New Product
Create new empty product

New Product
If your device is not available as a template, you can start with an empty device. You will have to create the device definition (fields, dashboard) and provide the payload decoder in the device's configuration.

Product Name
Sae3.01

Retour Suivant

On veut créer un nouveau produit du nom de « Sae3.01 »

Et on va choisir de quelle serveur vont venir nos données

Ajouter un appareil LoRaWAN

X

STEP 1 Product STEP 2 Network Server STEP 3 Devices STEP 4 Plan

Serveur de réseau
Veuillez choisir le serveur réseau LoRaWAN auquel vos appareils sont connectés.

<input type="radio"/>  The Things Stack V3 TTN V3 / Things Industries	Uplinks Downlinks
<input type="radio"/>  Helium Use your own console	Uplinks Downlinks
<input type="radio"/>  Managed Helium Powered by IoT Creators	Uplinks
<input type="radio"/>  LORIOT	Uplinks Downlinks
<input type="radio"/>  ChirpStack	Uplinks Downlinks

Showing 1 to 5 of 16 results

[Previous](#) [Next](#)

[Retour](#) [Suivant](#)

Ajouter un appareil LoRaWAN

X

STEP 1
Product

STEP 2
Network Server

STEP 3
Devices

STEP 4
Plan

Add Devices

Please provide one or multiple LoRaWAN device EUIs along with the corresponding names they should have on Datacake.

Alternatively, you can choose to upload a CSV file that contains the DevEUI, device Name, location, and a set of tags. For more information on how to format the file, please refer to [our documentation](#).

Drag and drop a .csv file here or click to choose one

DEVEUI	NOM	LOCALISATION	TAGS
1C 8B A0 37 B3 72 E8 E6 8 bytes <small>The DevEUI is already in use by another device</small>	sae3.01	Localisation	Ajouter un tag

+ Ajouter un autre appareil

Retour

Suivant

Et on choisit l'applis d'où viennent les données (avec le devEUI).

Ajouter un appareil LoRaWAN

X

STEP 1
Product

STEP 2
Network Server

STEP 3
Devices

STEP 4
Plan

Free

0,00 € / month

7 days data retention
500 datapoints / day
max 5 per workspace
Cancel any time

Light

1,00 € / month

1 month data retention
1000 datapoints / day
Cancel any time

Standard

3,00 € / month

3 months data retention
2500 datapoints / day
Cancel any time

Plus

5,00 € / month

12 months data retention
7500 datapoints / day
Cancel any time

Looking for a way to consolidate your billing or a more affordable package pricing?

[Explore our package and enterprise pricing options](#)

Have a code?

Apply

Retour

Add 1 device

inconvénient de cette solution, c'est que j'ai le droit à 500 données par jour et une durée de 7 jours.

13:55
09/12/2023



Actuellement nous sommes le , ou je fais ce compte rendu, donc je ne pense pas que cela ira jusqu'à la présentation.

Une fois, mon device créé, je vais créer le membre « API » de mon serveur, pour qu'il ait les droit de transférer des données à mon Dashboard :

The screenshot shows the Chirpstack interface under 'Administration > Membres'. A member named 'chirp' is selected, with its ID 'e2bb849bad49542b0d71a24331995f90fe2ede5f' displayed below it. There are buttons for 'Copier' (Copy) and 'Cacher' (Hide).

Son ID est donc « chirp »

Et la longue liste de chiffre et de lettre est sa clé API. Soit son mot de passe.

The screenshot shows the Chirpstack interface under 'Flotte > Sae3.01'. In the 'Permissions' tab, the 'Autorisations' section lists three users: 'Edouard Bastenier', 'API chirp', and 'API test'. The 'chirp' user has checked boxes for 'Afficher', 'Metadata', 'Définition', and 'Mesures', indicating full permissions.

Il va falloir également lui autoriser les droit sur l'applications.

De retour dans l'interface de Chirpstack, on va pouvoir renseigner les informations :

The screenshot shows the Chirpstack interface under 'Tenants / ChirpStack / Applications / sae301-edouard'. In the 'Integrations' tab, there is a form titled 'Update HTTP integration' with fields for 'Payload encoding' (set to 'JSON'), 'Event endpoint URL(s)' (set to 'https://api.datacake.co/integrations/lorawan/chirpstack/'), and 'Headers' (containing 'chirp' and 'e2bb849bad49542b0d71a24331995f90fe2ede5f'). A 'Submit' button is at the bottom.

L'url, ou vont aller les données,

Et le « Headers », qui vont lui permettre de prendre le rôle du membre créé précédemment (chirp), avec sa clé API.

Ce qui va permettre de valider notre device sur DataCake :

DEVICE	PRIMAIRE	SECONDAIRE
Sae3.01	N/A	N/A

Showing 1 to 1 of 1 results

Le point vert nos permet de savoir qu'il sont connecté.

Ensute pour avoir les données, il faut encore configurer plus précisément mon device.

En allant dans configuration,

Il faut rajouter un décodeur payload, puisque Chirpstack renvoie les données en Base64 lors de sa transmission avec DataCake. Voilà donc mon décodeur ASCII sur DataCake.

Décodeur Payload

Lorsque vos appareils envoient des données, le payload est tracé

```
1 function Decoder(bytes, port) {  
2     var decoded = {};  
  
function Decoder(bytes, port) {  
    var decoded = {};  
    var result = "";  
  
    for (i = 0; i < bytes.length; i++) {  
        result += String.fromCharCode(bytes[i]);  
    }  
  
    // sépare les valeurs  
    var values = result.split(',');
```

```

// definis les differente valeurs
decoded.CO2 = parseFloat(values[0]);
decoded.temperature = parseFloat(values[1]);
decoded.humidity = parseFloat(values[2]);

return decoded;
}

```

On doit également créer les « Fields », grosso modo, ce sont les colonnes qui vont contenir les données :

Fields

Fields describe the data the device will store.

NAME	IDENTIFIER	TYPE	ROLE	CURRENT VALUE	LAST UPDATE
Temperature	TEMPERATURE	Float	N/A	20,47 °C	0 seconds ago
CO2	CO2	Float	N/A	2062,1 ppm	0 seconds ago
Humidity	HUMIDITY	Float	N/A	62,54 %	0 seconds ago

Et désormais, si on va dans l'onglet « debug », on va pouvoir voir le live data qui arrive :

The screenshot shows the Flotte > Sae3.01 device configuration page. The 'Debug' tab is active. The log table contains the following entries:

Temps	Titre	Détails
14:01:15	Recorded measurements from payload decoder	Decoder returned: {"CO2": 1915.35, "humidity": 61.89, "temperature": 20.4... Log: null Recorded measurements: {"CO2": 1915.35, "humidity": 61.89, "temperature": 20.4...}
14:01:15	Received webhook data from chirpstack	{"deduplicationId": "69a4b376-86d5-473b-8448-596b6c90f7... Log: null Recorded measurements: {"CO2": 1915.35, "humidity": 61.89, "temperature": 20.4...}
14:00:56	Recorded measurements from payload decoder	Decoder returned: {"CO2": 1943.11, "humidity": 61.9, "temperature": 20.4... Log: null Recorded measurements: {"CO2": 1943.11, "humidity": 61.9, "temperature": 20.4...}
14:00:56	Received webhook data from chirpstack	{"deduplicationId": "d935393c-7b14-4d66-bcfe-4e05cb7d03... Log: null Recorded measurements: {"CO2": 1943.11, "humidity": 61.9, "temperature": 20.4...}
14:00:37	Recorded measurements from payload decoder	Decoder returned: {"CO2": 2019.16, "humidity": 61.94, "temperature": 20.4... Log: null Recorded measurements: {"CO2": 2019.16, "humidity": 61.94, "temperature": 20.4...}
14:00:37	Received webhook data from chirpstack	{"deduplicationId": "11fb6eb5-bdb3-4f2c-9abe-204ca86fa9... Log: null Recorded measurements: {"CO2": 2019.16, "humidity": 61.94, "temperature": 20.4...}
14:00:20	Recorded measurements from payload decoder	Decoder returned: {"CO2": 2036.73, "humidity": 62.13, "temperature": 20.4... Log: null Recorded measurements: {"CO2": 2036.73, "humidity": 62.13, "temperature": 20.4...}

UJM
edleboss@orange.fr

DATA CAKE

Flotte > Sae3.01

Sae3.01

Serial Number
1C8BA037B372E8E6

Dashboard

En haut à gauche on peut voir les Dashboard, et allant dessus après les avoir créés, je peux paramétrer les données que je veux afficher,

Edit Chart Widget X

New Line Chart
4 seconds ago

Co2

Basic **Data** Appearance Axes Timeframe Reference Lines

Sae3.01 ×

Device
Sae3.01

Field Co2

Values

Absolute Change

depuis combien de temps, tous les combien de temps (ici depuis 11h, toute les minutes)



Basic Data Appearance Axes Timeframe Reference Lines

Heure Jour Semaine 2W Mois Custom

From e.g. 'now' or '7 days ago', only English dates are supported
11 hours ago

9 décembre 2023 à 03:33 UTC+1

Until e.g. 'now' or '7 days ago', only English dates are supported
now

9 décembre 2023 à 14:32 UTC+1

Resolution Minutes ▾
1

Timezone

None selected

Fonction d'agrégation

Average Minimum Maximum

Et en y modifiant, on peut désormais autoriser le lien à tout le monde qui on le lien

Nom

 Sae301

Accès

Toute personne disposant d'un lien n'a pas besoin d'un compte

Lien public

<https://app.datacake.de/dashboard/d/9d1dfc9f-cdf5-4048-ad9f-4720cb450442> 

Tabs

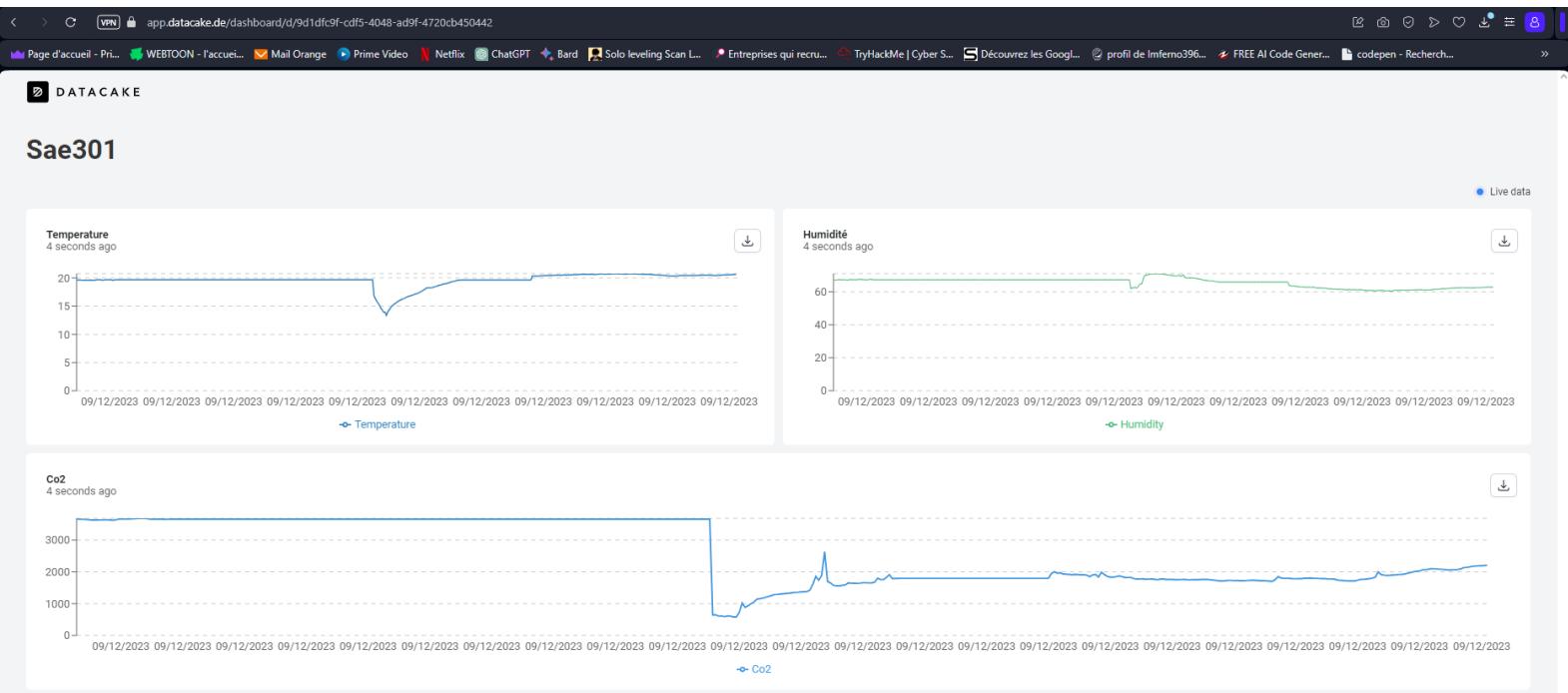
Les onglets vous permettent d'avoir plusieurs tableaux de bord secondaires à l'intérieur de ce tableau de bord.

Activer les onglets

Page 39

[Mise à jour du tableau de bord](#)

[Supprimer le tableau de bord](#)



Made with DATA CAKE

Pour voir mon branchement sur cette partie, veuillez voir la vidéo transmis avec le compte rendu.

b. Notification de mise en hors service :

Lors de la configuration de mon device sur DataCake, j'ai rajouter l'option de m'envoyer un mail, si il ne recevais plus de donnée au bout de 20 min.

General Configuration

Device name: Sae3.01

Icon:

You can override the default product icon for this device

Location description: Optional

Tags: You can use tags to group and filter your devices on dashboards and in rules

Metadata: Metadata is displayed on the device overview and can be used in dashboards

[Modifier](#)

Image: A picture showing the device
No image set [Modifier](#)

Online timeout: 20 Minutes
Sets the time that has to pass without any new data coming in for the device to be considered offline

Notify me if the device goes offline

Ce qu'il fait que je reçoit un mail, au bout de 20min sans données :

Device Offline: Sae3.01

13 décembre 2023 21:50



Datacake >
no-reply@datacake.de

À : **edleboss >**
edleboss@orange.fr

Datacake

Hi Edouard Bastenier,

the following device has not sent any data for 20 minutes and has therefore been marked as offline:

[Sae3.01](#)

c. Mise en place d'une alerte sur Tago.io

D'après les normes, la qualité de l'air d'une maison soit être inférieur à 1600 ppm.

CO ₂ [ppm]	Air Quality
2100	BAD
2000	Heavily contaminated indoor air
1900	
1800	Ventilation required
1700	
1600	
1500	MEDIOCRE
1400	Contaminated indoor air
1300	Ventilation recommended
1200	
1100	
1000	FAIR
900	
800	GOOD
700	
600	
500	EXCELLENT
400	

Source : https://www.iqhome.org/index.php?route=extension/d_blog_module/post&post_id=17

Sur Tago, j'ai pu renseigner une réglé :

The screenshot shows the Tago.io dashboard interface. At the top, there's a header with a lightning bolt icon, the name "alerte", and a status bar indicating "Type Variable | Action Send SMS | Last triggered at il y a 23 jours". To the right, there's a "Active" button with a checked checkbox. Below the header, there are tabs for "Action", "Tags", and "More".

Trigger section:

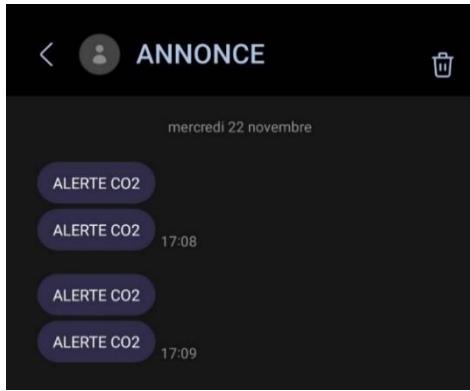
- Single device**: Watch a single device. A dropdown menu titled "Select the device" shows "sae301edouard".
- Multiple devices**: Watch all devices with matching tags.

Action section:

- Name**: alerte
- Type of action**: Send SMS
- Phone number(s)**: +33630952397
- Message**: ALERTE CO2

At the bottom left, there's a note: "Trigger Unlock (optional) Unlocks are conditions that enable the action to be triggered again." and a link "Learn more about Trigger by Variable".

Donc lorsque nos données dépasse les 1600 ppm en Co2, cela m'envoie un message par téléphone :



Malheureusement, sur la version gratuite de tago, on a le droit qu'à 4 messages par mois.

d. Dashboard DataCake, adapté pour le téléphone

J'ai également adapté le Dashboard de DataCake, qui n'est à l'origine pas responsable sur téléphone.



e. LEDs aux normes

Encore sous la base de ces normes :

CO ₂ [ppm]	Air Quality
2100	BAD
2000	Heavily contaminated indoor air
1900	Ventilation required
1800	
1700	
1600	
1500	
1400	MEDIOCRE
1300	Contaminated indoor air
1200	Ventilation recommended
1100	
1000	FAIR
900	
800	GOOD
700	
600	EXCELLENT
500	
400	

Pour représenter cela dans la réalité, j'ai rajouter 3 LED, qui simule les passage dans les zones, verte, jaune et rouge.

Pour ce faire, d'abord, il faut les brancher. Or sur les ports de ma carte si on y branche directement, on reçoit du 3v3, mais les LED peuvent recevoir 1v5. Donc si on y branche directe on y grillerai, alors il nous faut un résistance.

La Loi d'Ohm dit que :

$$R = (V_{\text{alim}} - V_{\text{LED}}) / I_{\text{LED}}$$

V_{alim} est la tension d'alimentation (3v3)

V_{LED} est la tension de la LED (1v5)

I_{LED} est le courant de la LED (0.01A)

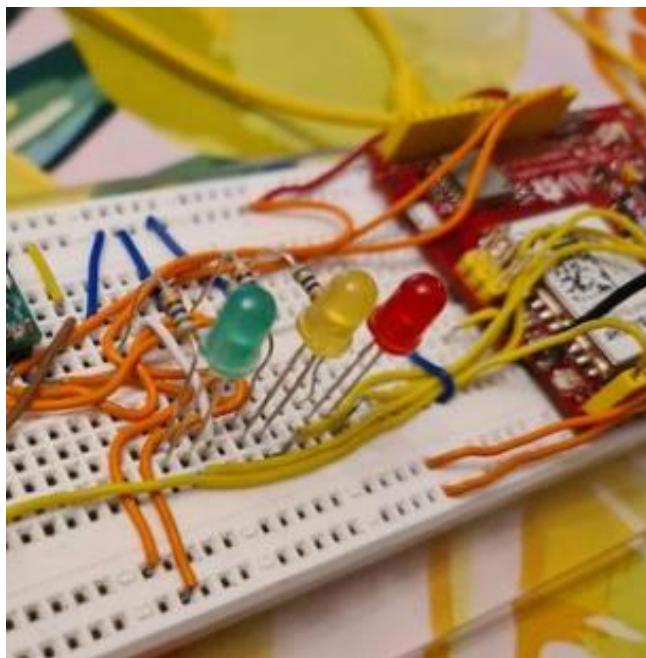
Soit :

$$3300 - 1500 = 1800 \text{ V}$$

$$R = (1800) / 0.01$$

$$R = 180 \Omega$$

Une fois la résistance acquise, il faut y branché :



Fil rouge et bleu= fil GND

Fil jaune = fil alimentation (3v3)

Pour finir, maintenant que tout est installer il faut y coder, pour y adapter au norme :

D'abord, on les initialise :

```

--  

21 // definition port LED  

22 const int pinLED1 = 11; //LED Verte  

23 const int pinLED2 = 12; //LED Rouge  

24 const int pinLED3 = 10; //LED Jaune  

25

```

Et

```

void setup(void)  

{  

    pinMode(pinLED1, OUTPUT);  

    pinMode(pinLED2, OUTPUT);  

    pinMode(pinLED3, OUTPUT);
}

```

Ensute on fait les « if » :

```

if (scdresult[0] > 1600) {  

    // Allumer la LED rouge, éteindre les autres  

    digitalWrite(pinLED1, LOW);  

    digitalWrite(pinLED2, HIGH);  

    digitalWrite(pinLED3, LOW);  

} else if (scdresult[0] >= 1000 && scdresult[0] <= 1600) {  

    // Allumer la LED jaune, éteindre les autres  

    digitalWrite(pinLED1, LOW);  

    digitalWrite(pinLED2, LOW);  

    digitalWrite(pinLED3, HIGH);  

} else {  

    // Allumer la LED verte, éteindre les autres  

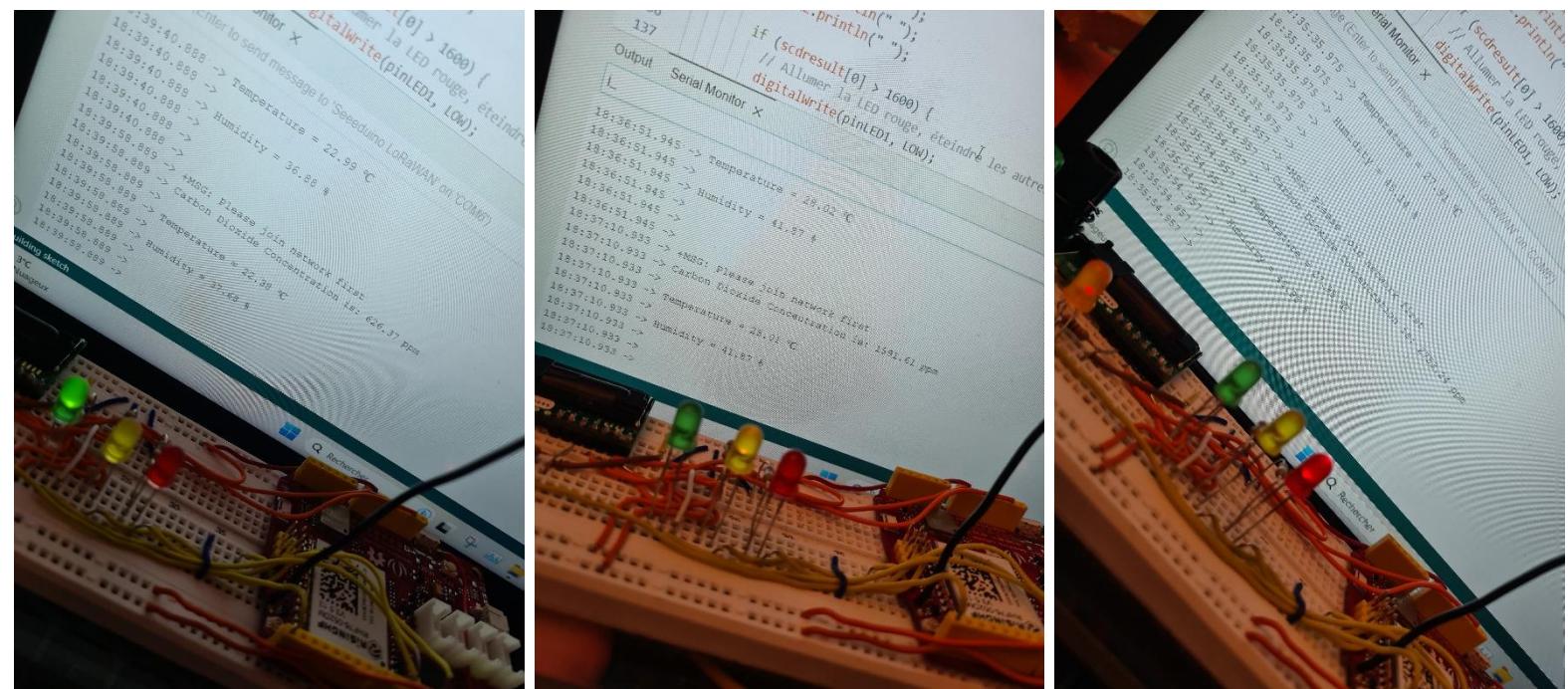
    digitalWrite(pinLED1, HIGH);  

    digitalWrite(pinLED2, LOW);  

    digitalWrite(pinLED3, LOW);
}

```

Un fois cela fait, on a bien les LED, qui s'allume en fonction du résultat de Co2 que l'on récupère par le SCD30.



f. Mise en place d'une alerte sur DataCake

J'ai mis en place une alerte du DataCake, en fonction de mes données,

The screenshot shows the DataCake administration interface under 'Règles' (Rules). A green success message at the top right says 'La règle a été mise à jour avec succès.' (The rule was updated successfully). The rule is named 'Co2'. The 'Conditions' section contains a condition: 'If Co2 is plus grand que 1600 ppm with a hysteresis of 0 ppm'. The 'Then' section specifies sending an email to 'edleboss@orange.fr (Edouard Bastenier)' with the subject 'ALERTE' and the message 'Alerte Co2, Veuillez aérer !'. There is also a 'Pour:' field with 'example@domain.com' and an 'Ajouter' button.

Ma règle dis :

Si mon Co2 est plus grand que 1600 (norme), je reçois un mail, avec le message suivant toute les 60s, comme définis.

Donc lorsque nos données dépasse les 1600 ppm en Co2, cela m'envoie un mail.

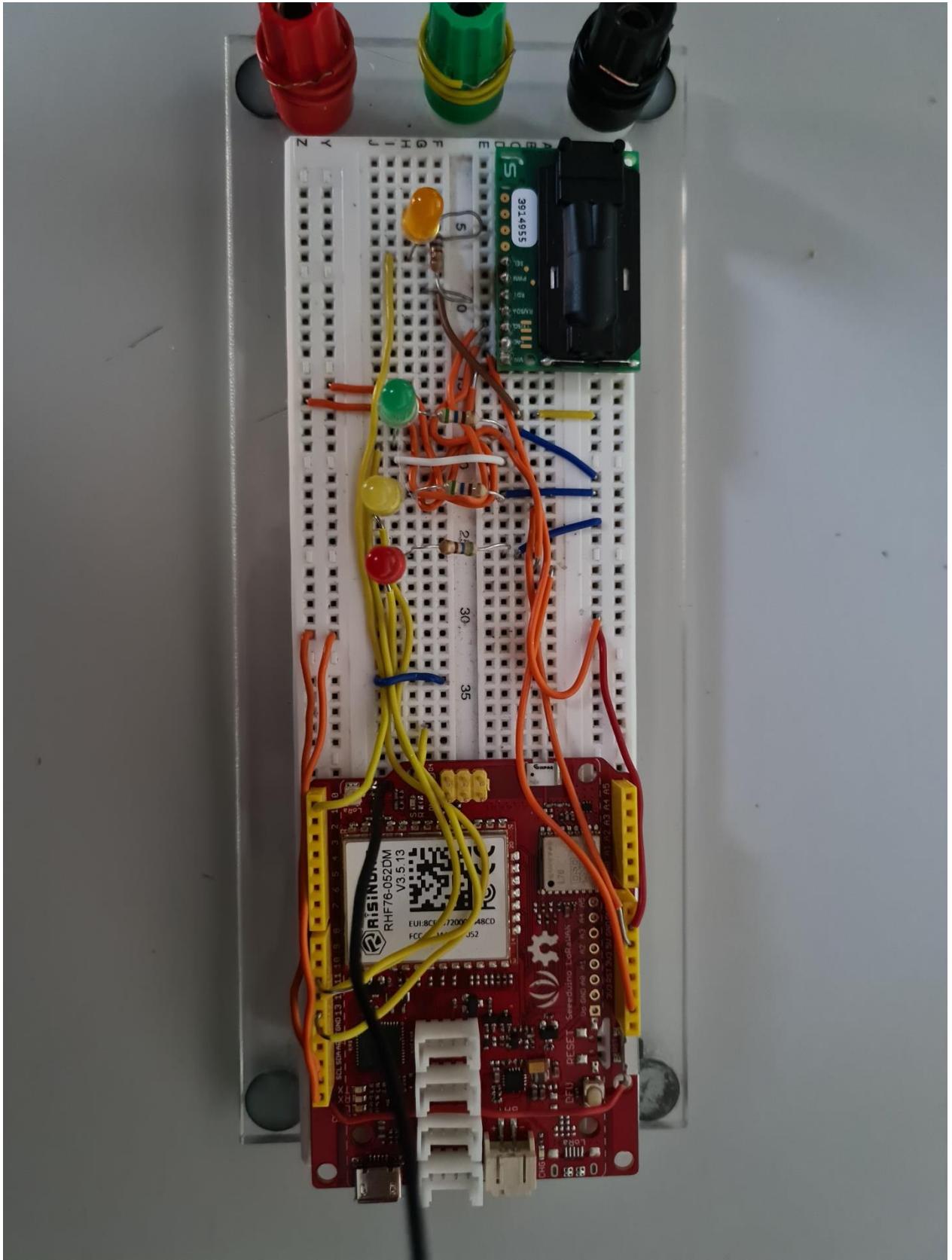
The left pane shows an email inbox with five messages from 'Datacake' with the subject 'Datacake Alert: ALERTE', all received 'Hier' (yesterday). The right pane shows a detailed view of one of these emails. The subject is 'Datacake Alert: ALERTE' and the date is '9 décembre 2023 14:52'. The message is from 'no-reply@datacake.de' to 'edleboss@orange.fr'. The body of the email contains the message 'Alerte Co2, Veuillez aérer !'.

Et on peut voir la règle :

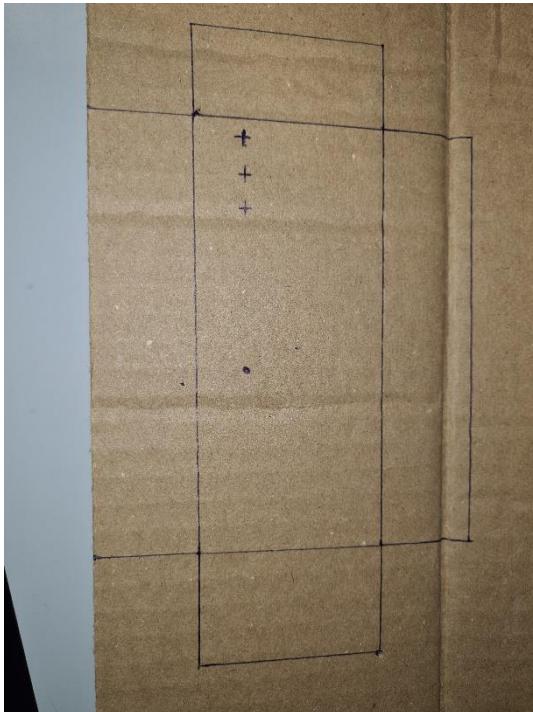
Règle	Action	En savoir plus
Co2	Envoyer un e-mail	

g. Mise en place d'un style du capteur de Co₂

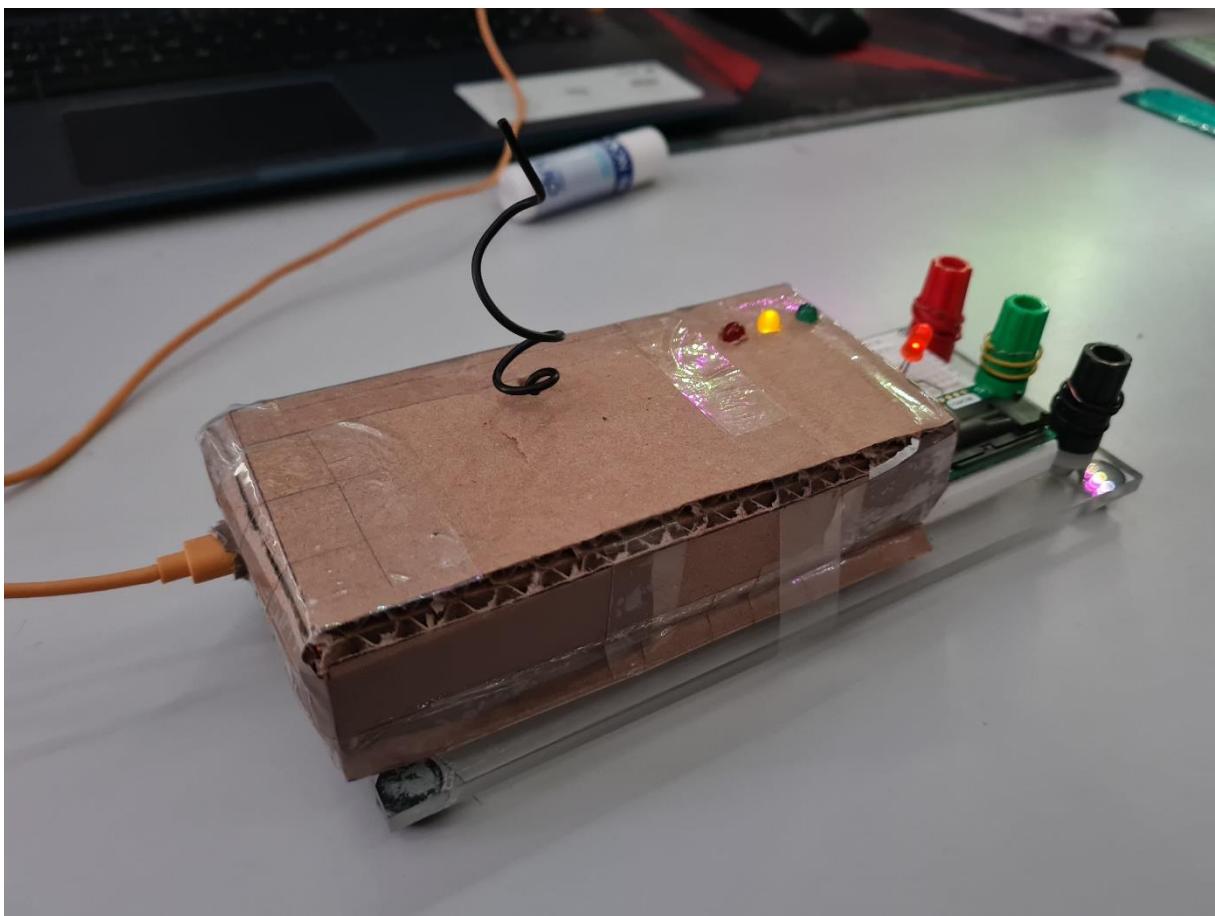
Tout d'abord, j'ai espacé les équipements et aplatis les câbles, histoire d'avoir une plaque propre.



Ensuite j'ai mesurer la plaque et fait un premier croquis pour avoir les LED qui dépasse et l'antenne :



Une fois découper, voila a quoi ressemble définitivement mon capteur de Co2 de la SAE3.01



J'ai également rajouter une LED pour voir lorsque mon capteur est alimenté.

J'ai laissé mon capteur dehors pour qu'il puissent avoir de meilleur résultat lors de son analyse de l'air

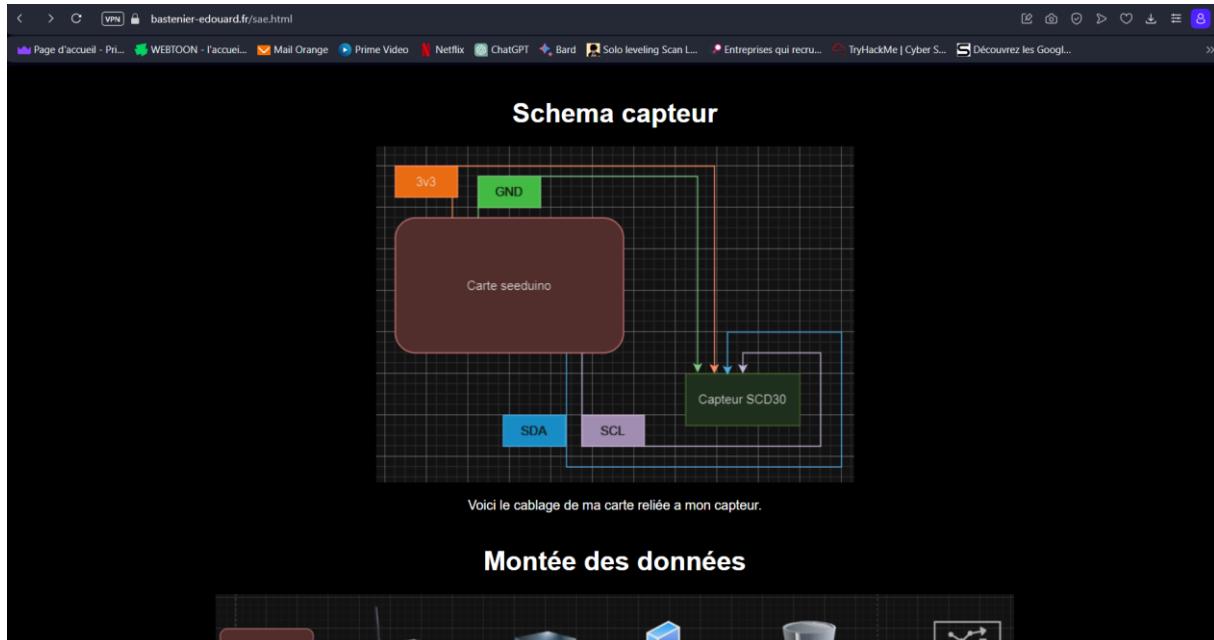
h. Site web

Pour terminer, j'ai monté mes informations sur mon site internet avec mon nom de domaine :

```
edouard@eta: /var/www/edo  X  +  ~
GNU nano 7.2                                     sae.html *
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="Css.css">
<title>SAE3.01 - Images</title>
</head>
<body>

<h1>Image 1</h1>
<br>
Voici le cablage de ma carte reliée à mon capteur.
<br><br>
<h1>Image 2</h1>
<br>
Voici comment le chemin de mon capteur a mon graphique, pour expliquer comment cela marche.
<br><br>
<h1>Image 3</h1>
<br>
Voici les graphiques sur grafana d'une analyse de la température, du Co2 et de l'humidité sur une période de 7h.
<br><br>
<h1>Image 4</h1>
<br>
Voici le design final de mon capteur de Co2
<br>
</body>
</html>
```

Ce qu'il fait que j'ai ça :



<https://bastenier-edouard.fr/sae.html>

POST-SCRIPTUM :

Voilà mon code final :

```
#include <LoRaWan.h>
#include "SCD30.h"

#if defined(ARDUINO_ARCH_AVR)
#pragma message("Defined architecture for ARDUINO_ARCH_AVR.")
#define SERIAL Serial
#elif defined(ARDUINO_ARCH_SAM)
#pragma message("Defined architecture for ARDUINO_ARCH_SAM.")
#define SERIAL SerialUSB
#elif defined(ARDUINO_ARCH_SAMD)
#pragma message("Defined architecture for ARDUINO_ARCH_SAMD.")
#define SERIAL SerialUSB
#elif defined(ARDUINO_ARCH_STM32F4)
#pragma message("Defined architecture for ARDUINO_ARCH_STM32F4.")
#define SERIAL SerialUSB
#else
#pragma message("Not found any architecture.")
#define SERIAL Serial
#endif

// definition port LED
const int pinLED1 = 11; //LED Verte
const int pinLED2 = 12; //LED Rouge
const int pinLED3 = 10; //LED Jaune

char buffer[256];

// Chirpstack parameters
const char *appEui = "e6c5301b79311336";
const char *appKey = "428ce839dc992fcfaeb64d6b38fa278";

void setup(void)
{
    pinMode(pinLED1, OUTPUT);
    pinMode(pinLED2, OUTPUT);
    pinMode(pinLED3, OUTPUT);

    SerialUSB.begin(115200);
    while(!SerialUSB);

    lora.init();
}
```

```

    memset(buffer, 0, 256);
    lora.getVersion(buffer, 256, 1);
    SerialUSB.print(buffer);

    memset(buffer, 0, 256);
    lora.getId(buffer, 256, 1);
    SerialUSB.print(buffer);

    lora.setId(NULL,"1c8ba037b372e8e6","e6c5301b79311336");
    lora.setKey(NULL,NULL, "428ce839dc992fcfaeb64d6b38fa278");

    lora.setDeciveMode(LWOTAA);
    lora.setDataRate(DR0, EU868);

    lora.setChannel(0, 868.1);
    lora.setChannel(1, 868.3);
    lora.setChannel(2, 868.5);

    lora.setReceiceWindowFirst(0, 868.1);
    lora.setReceiceWindowSecond(869.5, DR3);

    lora.setDutyCycle(false);
    lora.setJoinDutyCycle(false);

    lora.setPower(14);

    while(!lora.setOTAAJoin(JOIN));

    SerialUSB.println("SCD30 Raw Data");
    scd30.initialize();
}

void loop(void)
{
    bool result = false;

    float scdresult[3] = {0};

    if(scd30.isAvailable())
    {
        scd30.getCarbonDioxideConcentration(scdresult);

        // Format the data for TTN payload
        String payload = String(scdresult[0]) + "," + String(scdresult[1]) +
        "," + String(scdresult[2]);

        // Convert payload to char array
        char payloadCharArray[payload.length() + 1];
        payload.toCharArray(payloadCharArray, payload.length() + 1);
    }
}

```

```

    // Send the data to TTN
    result = lora.transferPacket(payloadCharArray,
strlen(payloadCharArray));

    if(result)
    {
        short length;
        short rssi;

        memset(buffer, 0, 256);
        length = lora.receivePacket(buffer, 256, &rssi);

        if(length)
        {
            SerialUSB.print("Length is: ");
            SerialUSB.println(length);
            SerialUSB.print("RSSI is: ");
            SerialUSB.println(rssi);
            SerialUSB.print("Data is: ");
            for(unsigned char i = 0; i < length; i++)
            {
                SerialUSB.print("0x");
                SerialUSB.print(buffer[i], HEX);
                SerialUSB.print(" ");
            }
            SerialUSB.println();
        }
    }

    SERIAL.print("Carbon Dioxide Concentration is: ");
    SERIAL.print(scdresult[0]);
    SERIAL.println(" ppm");
    SERIAL.println(" ");
    SERIAL.print("Temperature = ");
    SERIAL.print(scdresult[1]);
    SERIAL.println(" °C");
    SERIAL.println(" ");
    SERIAL.print("Humidity = ");
    SERIAL.print(scdresult[2]);
    SERIAL.println(" %");
    SERIAL.println(" ");
    SERIAL.println(" ");
    SERIAL.println(" ");

    if (scdresult[0] > 1600) {
        // Allumer la LED rouge, éteindre les autres
        digitalWrite(pinLED1, LOW);
        digitalWrite(pinLED2, HIGH);

```

```
    digitalWrite(pinLED3, LOW);
} else if (scdresult[0] >= 1000 && scdresult[0] <= 1600) {
    // Allumer la LED jaune, éteindre les autres
    digitalWrite(pinLED1, LOW);
    digitalWrite(pinLED2, LOW);
    digitalWrite(pinLED3, HIGH);
} else {
    // Allumer la LED verte, éteindre les autres
    digitalWrite(pinLED1, HIGH);
    digitalWrite(pinLED2, LOW);
    digitalWrite(pinLED3, LOW);
}
}
```