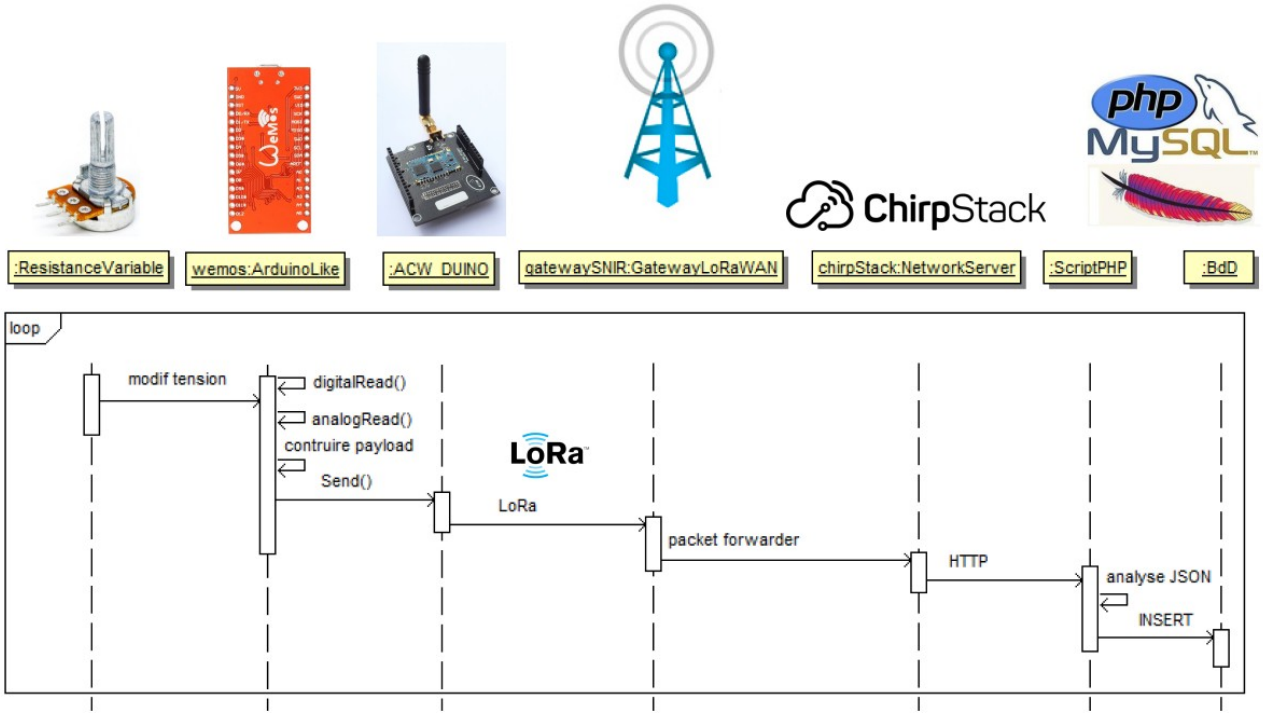
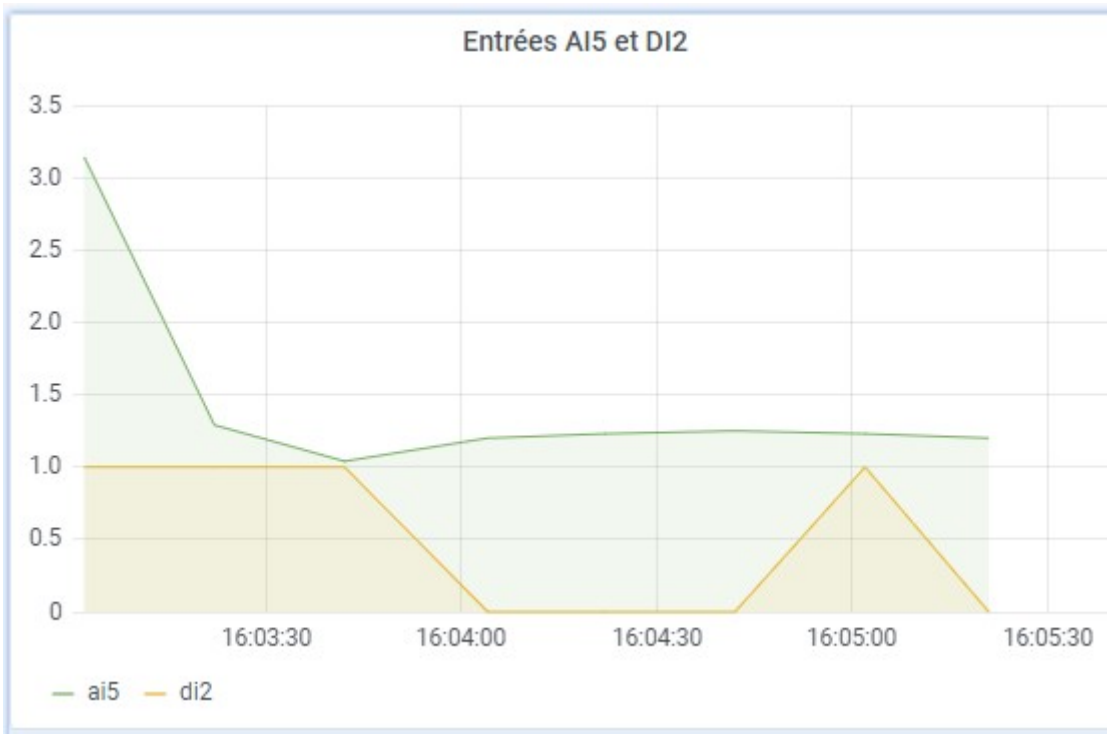
TP LoRaWAN

Le but de ce TP est d'apprendre à mettre en œuvre une chaîne complète LoRaWAN open source, du device juqu'au serveur applicatif.





A la fin de ce TP, vous devez être capables de :

- Développer un device LoRaWAN basé sur une carte de type Arduino et un shield ATIM LoRaWAN
- Enregistrer le device sur un Network Server LoRaWAN
- Configurer la liaison entre le Network Server et le serveur applicatif
- Développer la partie serveur Web permettant d'enregistrer les données (PHP + MySQL)
- Configurer l'outil Grafana permettant la visualisation des mesures enregistrées dans la BdD

Table des matières

1 Matériel et logiciels mis en œuvre

Avant de démarrer les manipulations, récupérer sur clé USB l'ensemble des ressources logicielles requises.

1.1 Potentiomètre

Il nous sert à simuler un capteur, comme le capteur d'humidité de sol.

On le choisit volontairement avec une impédance élevée pour limiter la consommation.

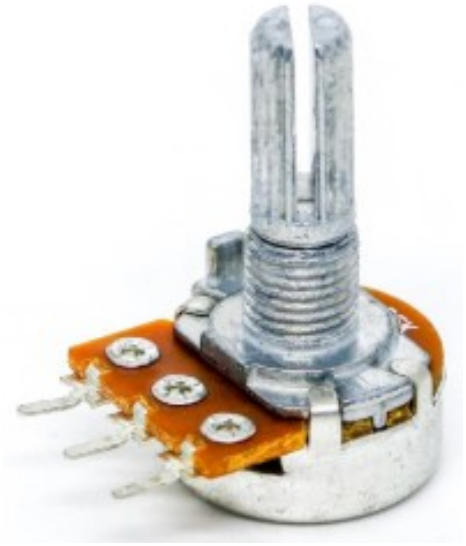


Figure 1: Potentiomètre

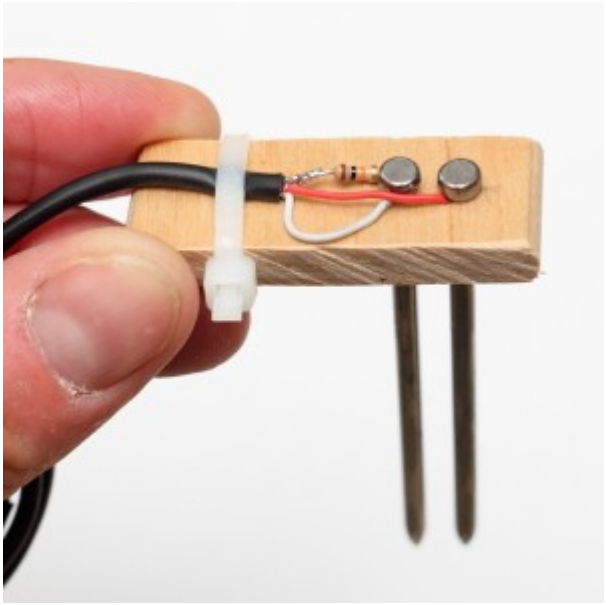
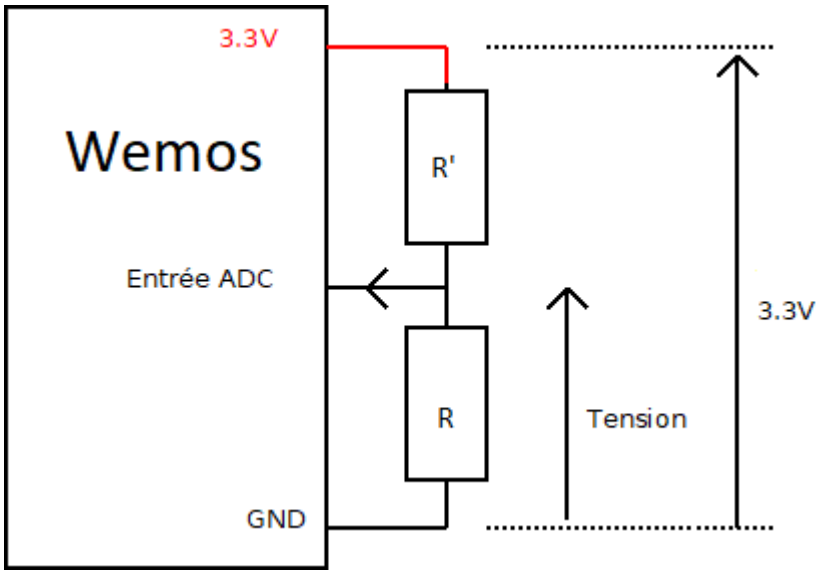


Figure 2: Capteur d'humidité de sol

Alimenté par la carte Lilygo Wemos XI, il nous permet de générer une valeur de tension entre 0 et 3,3V sur une entrée analogique, et ainsi simuler un capteur.



1.2 Carte Lilygo Wemos XI

Carte électronique développée par l'entreprise chinoise Lilygo (http://www.lilygo.cn/).
Elle utilise un SoC compatible avec l'ATMega328P qui équipe les Arduino Uno.
Ses avantages : taille réduite (2,5 x 5 cm) et très faible coût (1€50)

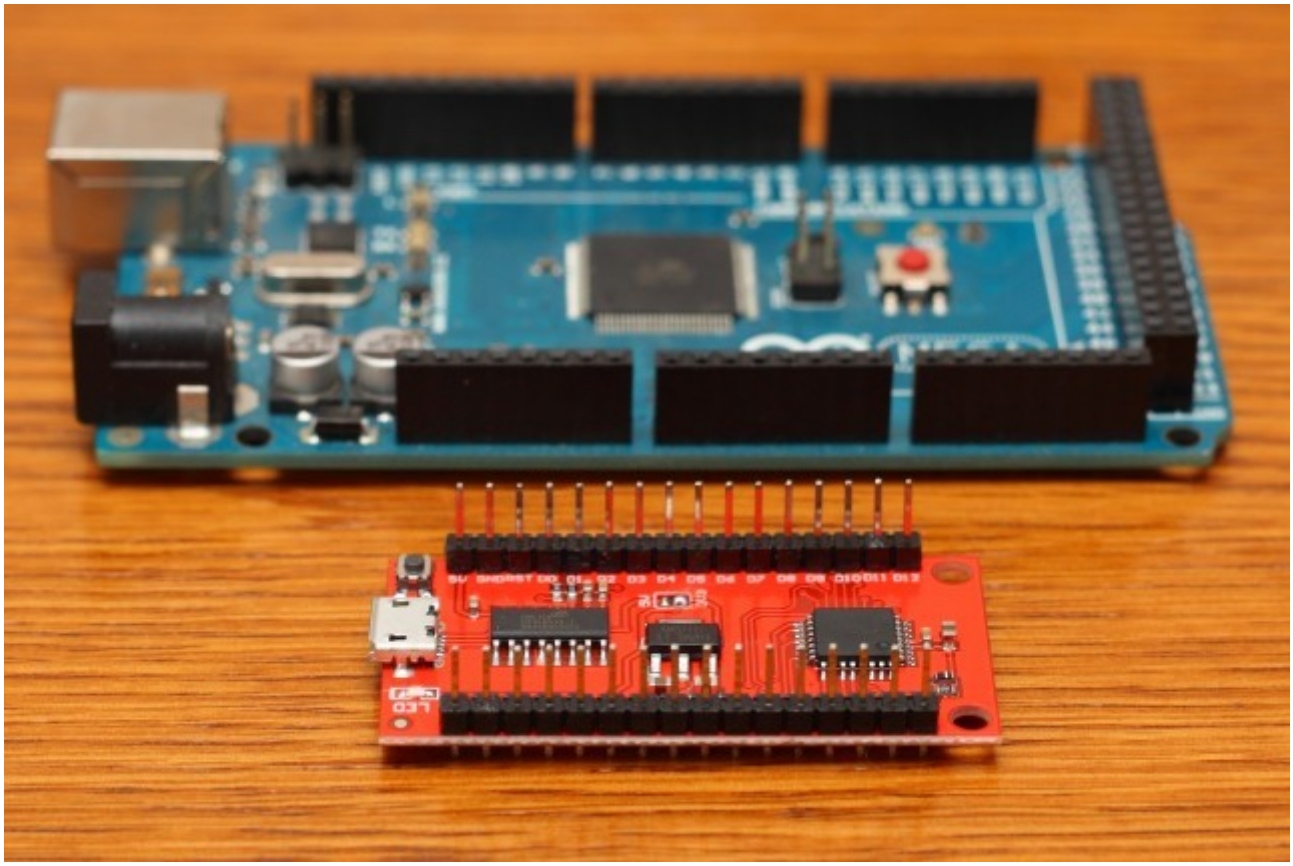
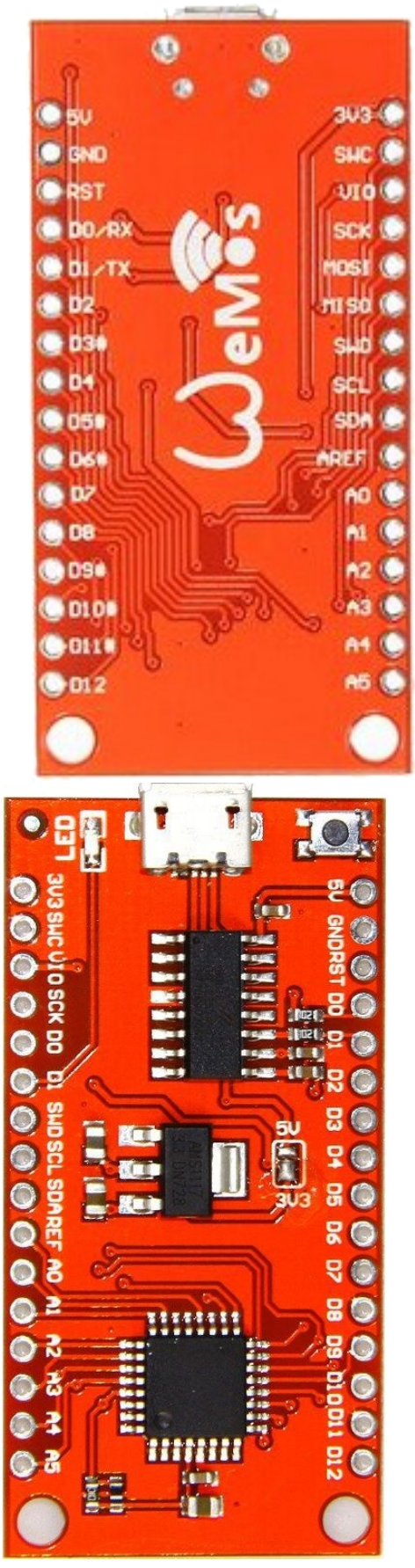


Figure 3: Lilygo Wemos XI (en rouge) et Arduino Mega 2560 (en bleu)

Caractéristiques de la carte Lilygo Wemos XI :

• MCU	• LGT8F328P
• FLASH	• 32Kbytes
• SRAM	• 2Kbytes
• E2PROM	• 0K/1K/2K/4K/8K(FLASH Share)
• PWM	• 8
• Frequency	• 16MHz（Maximum 32MHz）
• ADC	• 6 passageway12 position
• DAC	• 1passageway8 position
• UART	• 1
• SPI	• YES
• TWI（I2C）	• YES
• GUID（All serial numbers）	• YES
• Internal benchmark	• 1.024V/2.048V/4.096V ±0.5%
• System logic level	• Factory 3V3 (switch from pad to 5V)

Vue recto/verso de la Wemos XI:



Broche **Fonction**

A0 à A5	Entrées analogiques (également utilisable en entrées/sorties numériques)
D3# D5# D6# D9# D10# D11#	Sorties analogiques (également utilisable en entrées/sorties numériques)
D0 à D12	Entrées/sorties numériques
Rx et Tx	Liaison série RS232
SCL et SDA	Bus I2C
SCK MOSI MISO	Bus SPI

1.3 Driver USB

En cas de besoin, le driver USB permettant de communiquer avec la carte Wemos peut être téléchargé sur le github officiel de Lilygo : https://github.com/LilyGO
Lien driver : https://github.com/LilyGO/TTGO-XI-8F328P-U-driver/blob/master/usbbridgesetup_ca.zip

1.4 IDE Aduino

La carte Wemos se programme en utilisant l'IDE Arduino, téléchargeable sur le site officiel : https://www.arduino.cc/

1.5 Librairie Lilygo Wemos XI

Par défaut, l'IDE Arduino ne reconnaît pas la carte Wemos XI. Il est donc nécessaire d'ajouter une librairie.
Lien librairie : https://github.com/LilyGO/Arduino_XI/archive/master.zip

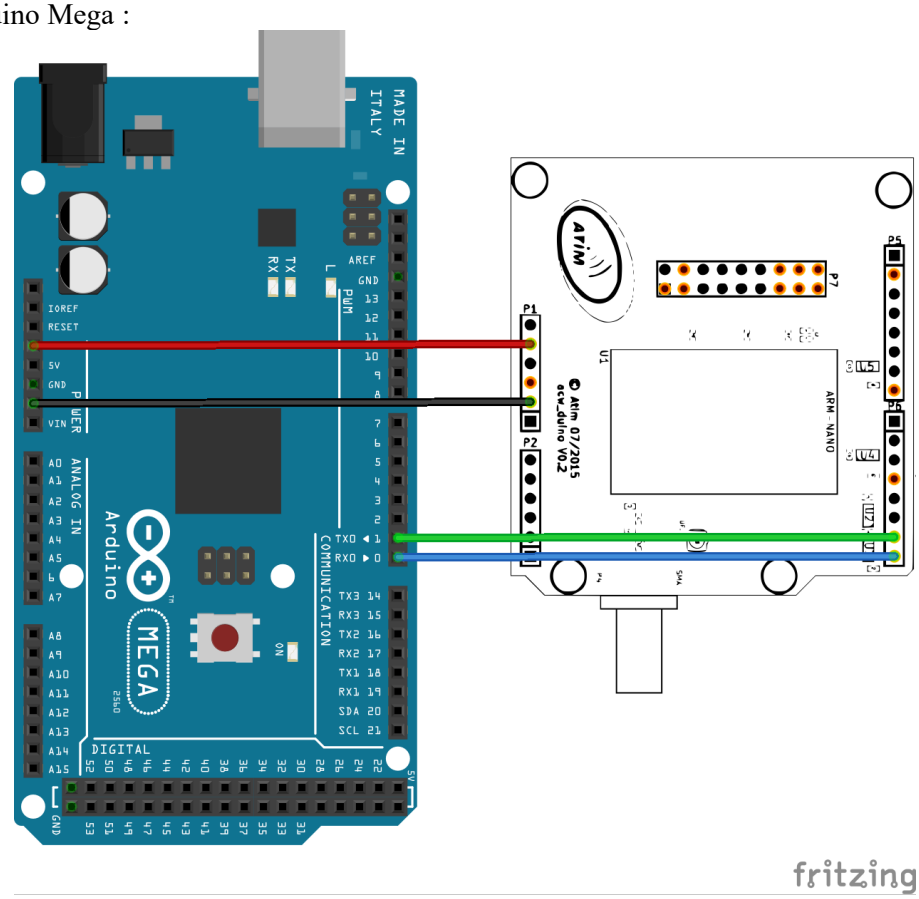
1.6 Shield LoRaWAN ATIM ACW DUINO (ou ACW PI)

Le shield ATIM peut être monté sur une Arduino et permet alors de communiquer en LoRaWAN.
Disponible par exemple chez https://yadom.fr/ pour environ 60€ (shield + antenne)

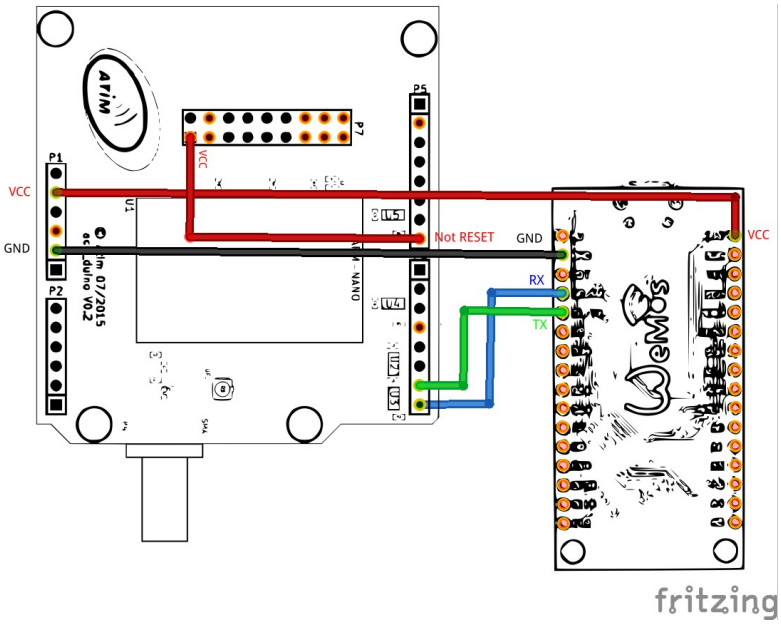
Binôme	Shield utilisé
Binôme 1	SNIR DUINO 01
Binôme 2	SNIR DUINO 02
Binôme 3	SNIR DUINO 03
Binôme 4	SNIR DUINO 04
Binôme 5	SNIR DUINO 05
Binôme 6	SNIR DUINO 06
Binôme 7	SNIR PI 01
Binôme 8	SNIR PI 02
Binôme 9	SNIR PI 03
Binôme 10	SNIR PI 04
Binôme 11	SNIR PI 05
Binôme 12	SNIR PI 06

La communication entre l'Arduino et le shield se fait via la liaison série RS232 (broches Rx et Tx)
NB : La liaison USB de l'Arduino et les broches Rx Tx utilisent le même UART
Lors du chargement du programme depuis le PC vers l'Arduino (ou la Wemos) il donc est nécessaire de débrancher le fil bleu (Rx côté Arduino)

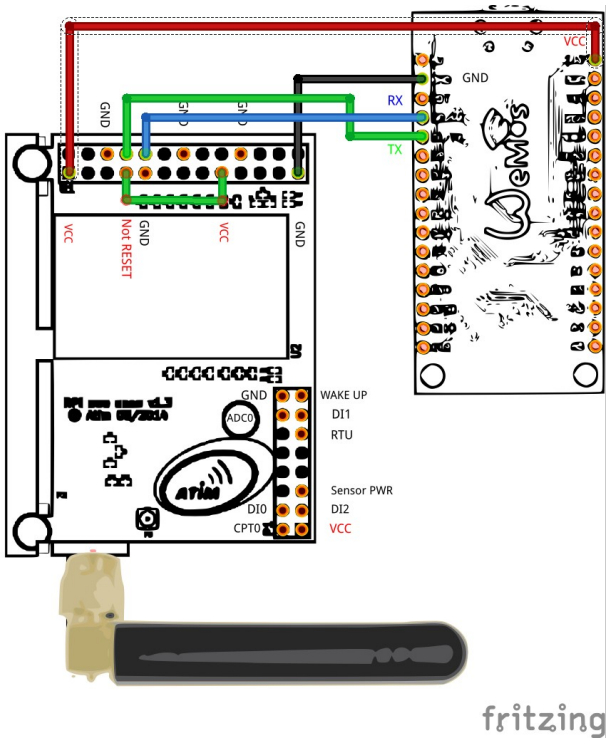
Avec une Arduino Mega :



Avec notre carte Lilygo Wemos XI :

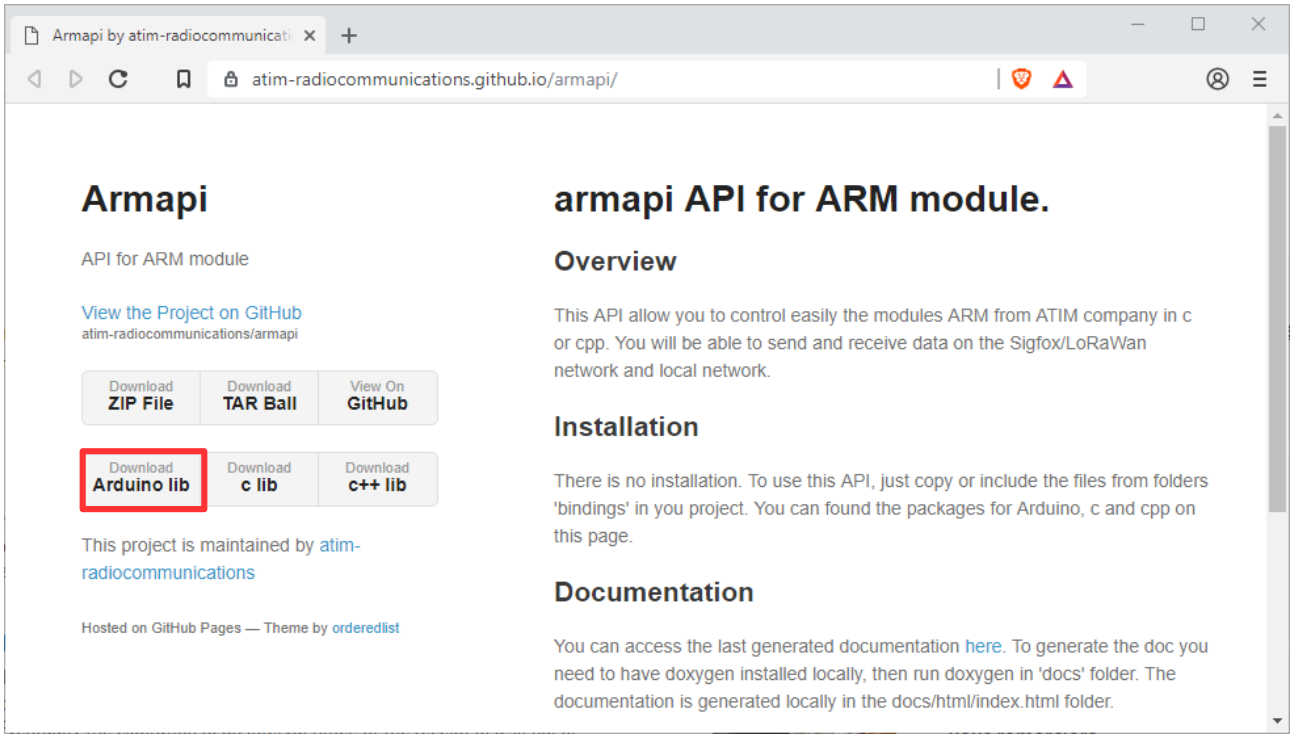


Carte Lilygo Wemos XI et le shield ACW-RPI :



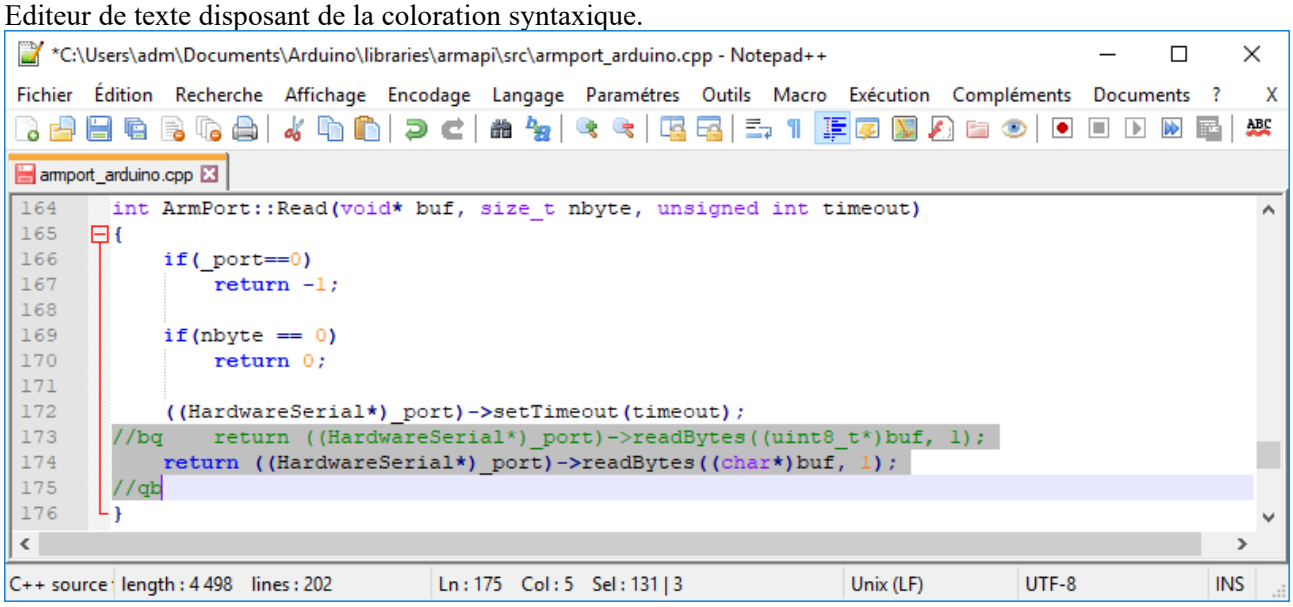
1.7 Librairie ATIM

ATIM fournit un fichier de librairie arduino_armapi_v1.0.4.zip
Celui-ci est nécessaire pour utiliser le shield ATIM depuis un sketch Arduino

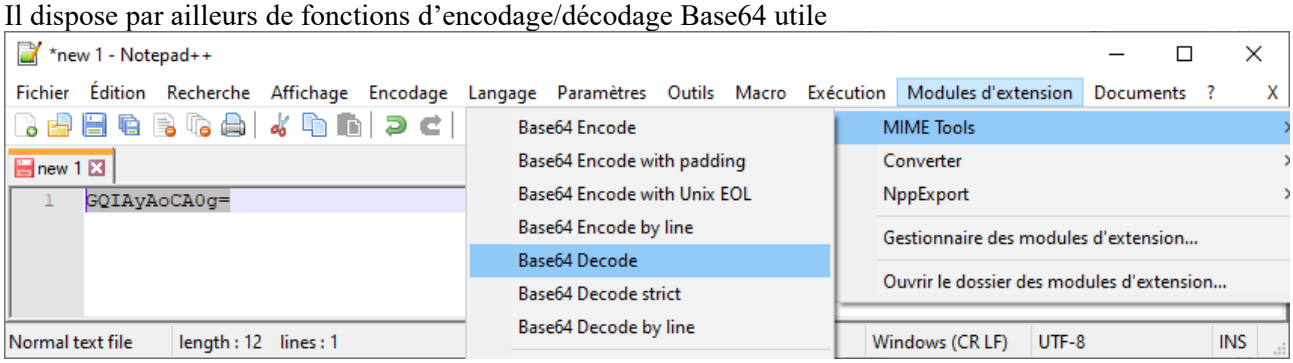


1.8 notepad++

Editeur de texte disposant de la coloration syntaxique.



Il dispose par ailleurs de fonctions d'encodage/décodage Base64 utile



1.9 Infrastructure LoRaWAN

Pour ce TP, l'infrastructure est composée de 2 gateways LoRaWAN et d'un network serveur ChirpStack installé sur un PC.
Les Gateways sont connectées au NetworkServer via un réseau Ethernet privé.

Figure 4: Gateway LoRaWAN Kerlink iFemtoCell

Figure 5: Gateway LoRaWAN DIY IMST + Rapsberry

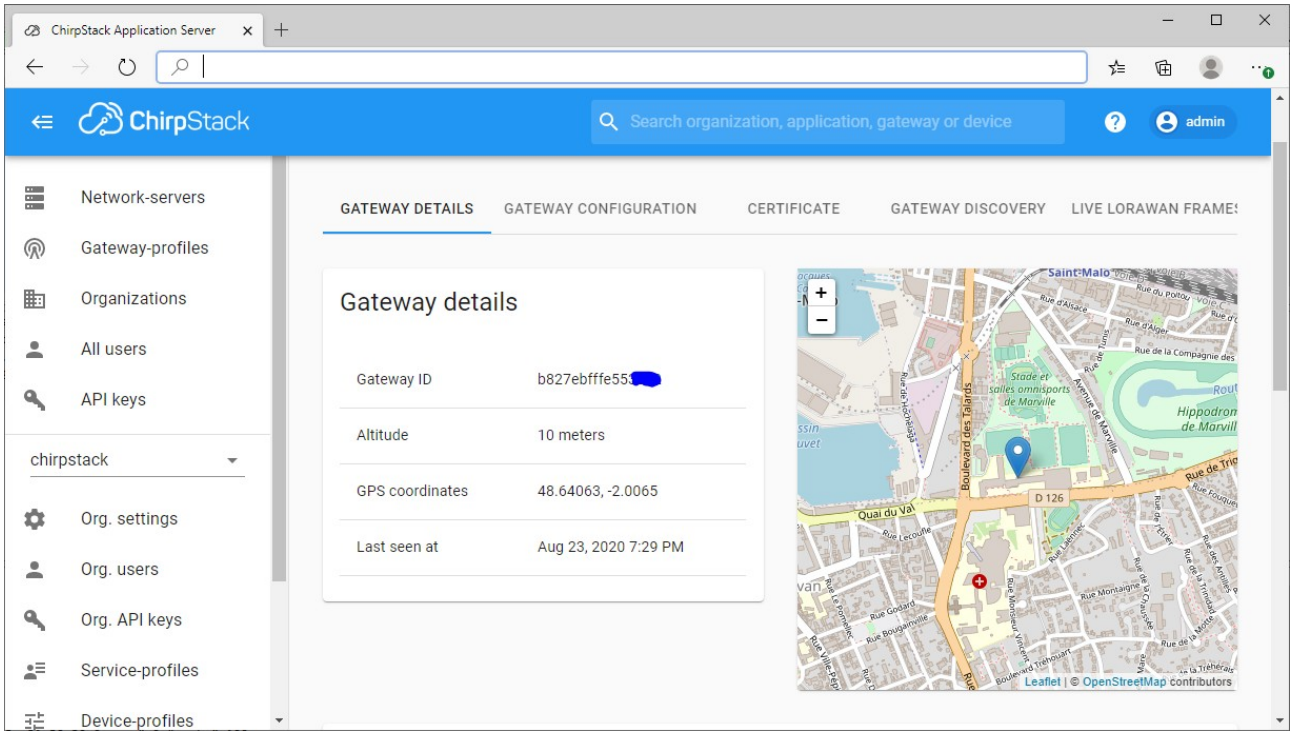


Figure 6: Interface web ChirpStack

Figure 7: Routeur (faisant également point d'accès WiFi)

1.10 WAMP server

WAMPserver est une distribution pour Windows contenant :

- Le serveur Web Apache
- Les serveurs de données (SGBD) MySQL et MariaDB
- L'interpréteur PHP
- Un outil graphique simplifiant l'administration de Base de Données : PhpMyAdmin

WAMP server 64 bits est téléchargeable ici : https://sourceforge.net/projects/wampserver/

NB : l'installation de Microsoft vcredist_x64 (2012 VC11) est requise avant d'installer WAMP server 64 bits

Figure 8: WAMP = Apache + MySQL et MariaDB + PHP

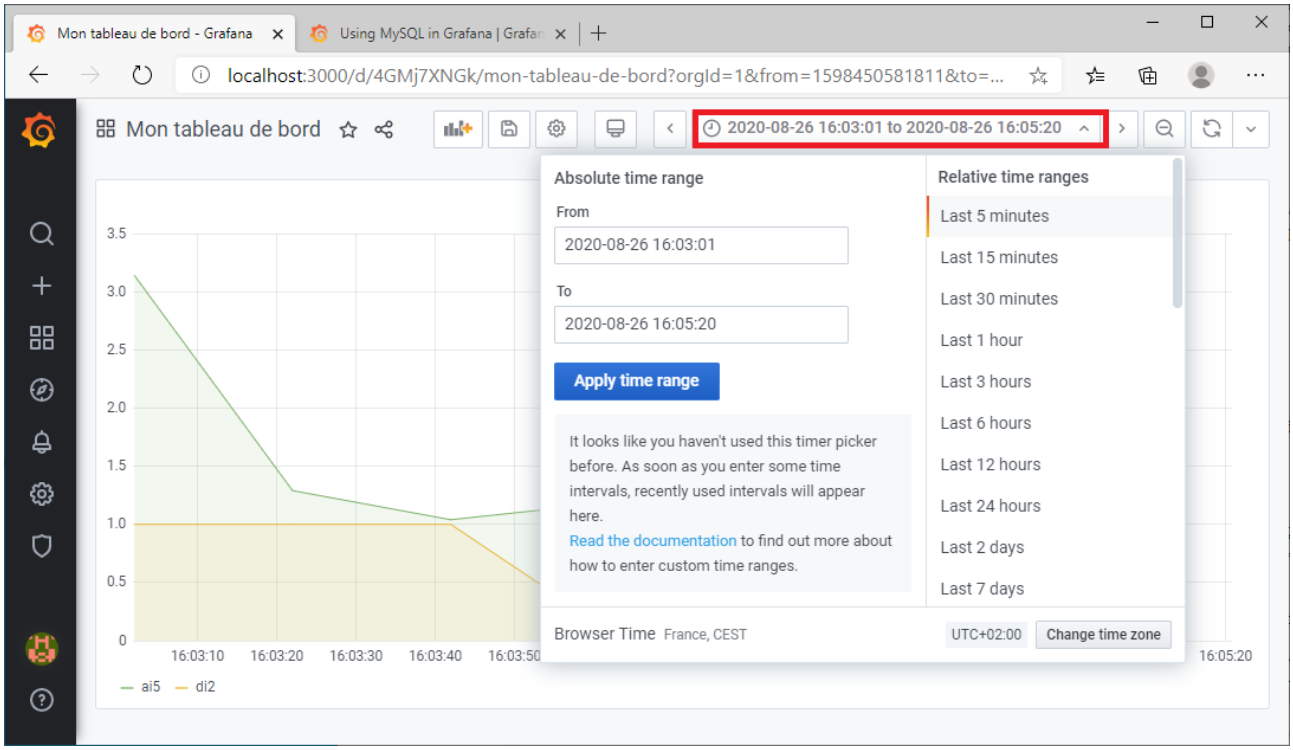
datation	di2	ai5
2020-08-26 16:03:02	1	3.140
2020-08-26 16:03:22	1	1.290
2020-08-26 16:03:42	1	1.040
2020-08-26 16:04:04	0	1.200
2020-08-26 16:04:22	0	1.230

Figure 9: Stockage des données dans la table "mesures" de la BdD

Nous utiliserons WAMP pour développer la partie Serveur Applicatif en langage PHP.
Notre script PHP va recevoir les données envoyées par le Network Server. Il va également les stocker dans une Base de Données (BdD) MariaDB

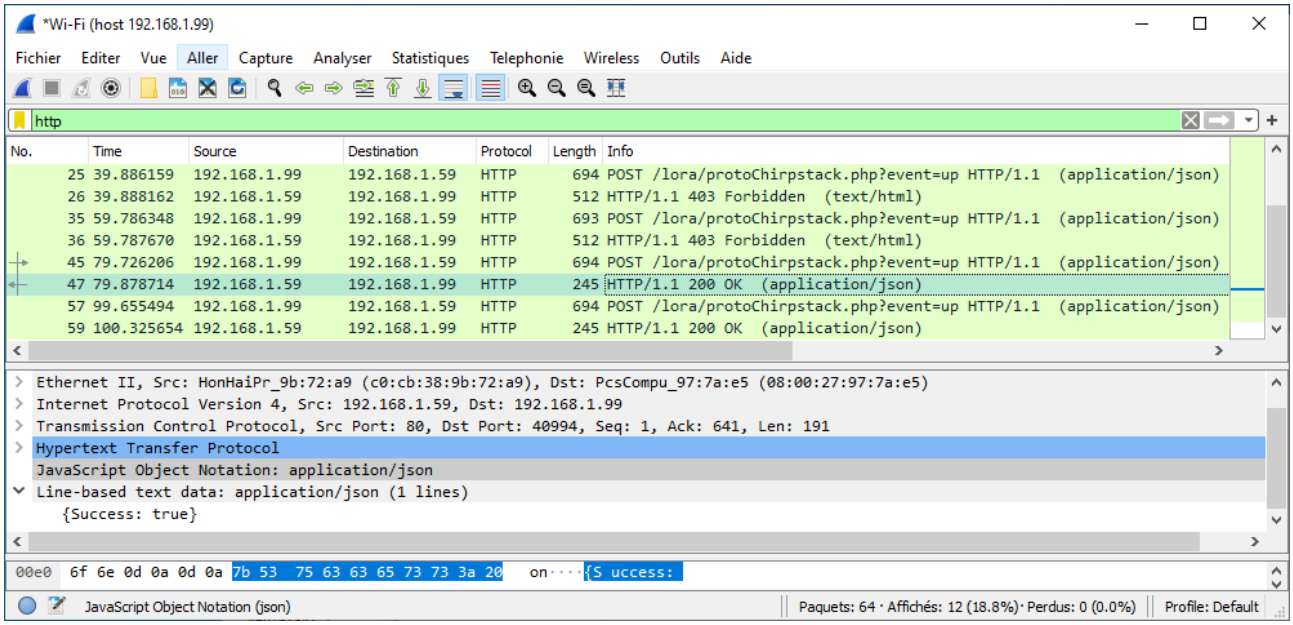
1.11 Grafana

Pour permettre aux utilisateurs de visualiser les mesures stockées dans la BdD, nous allons mettre en place un serveur Web. Nous utiliserons pour cela l'outil libre Grafana.
Grafana est téléchargeable ici : https://grafana.com/grafana/download



1.12 Wireshark

Wireshark permet d'analyser les échanges réseau et détecter les pannes de communication.



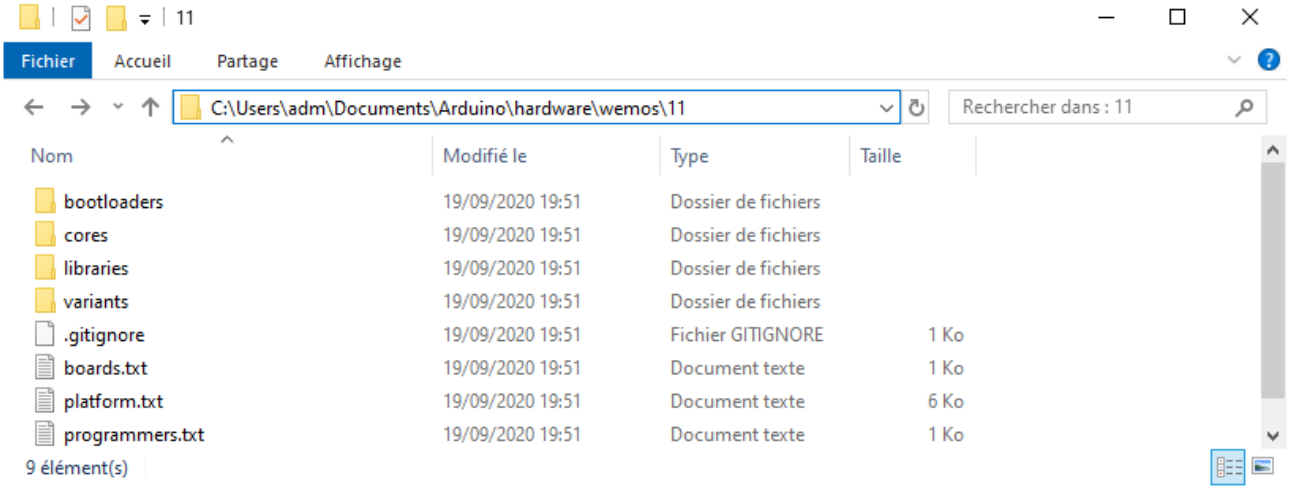
2 Mise en œuvre du device (carte Wemos XI + shield ATIM LoRaWAN)

2.1 Connexion de la carte Wemos XI au PC

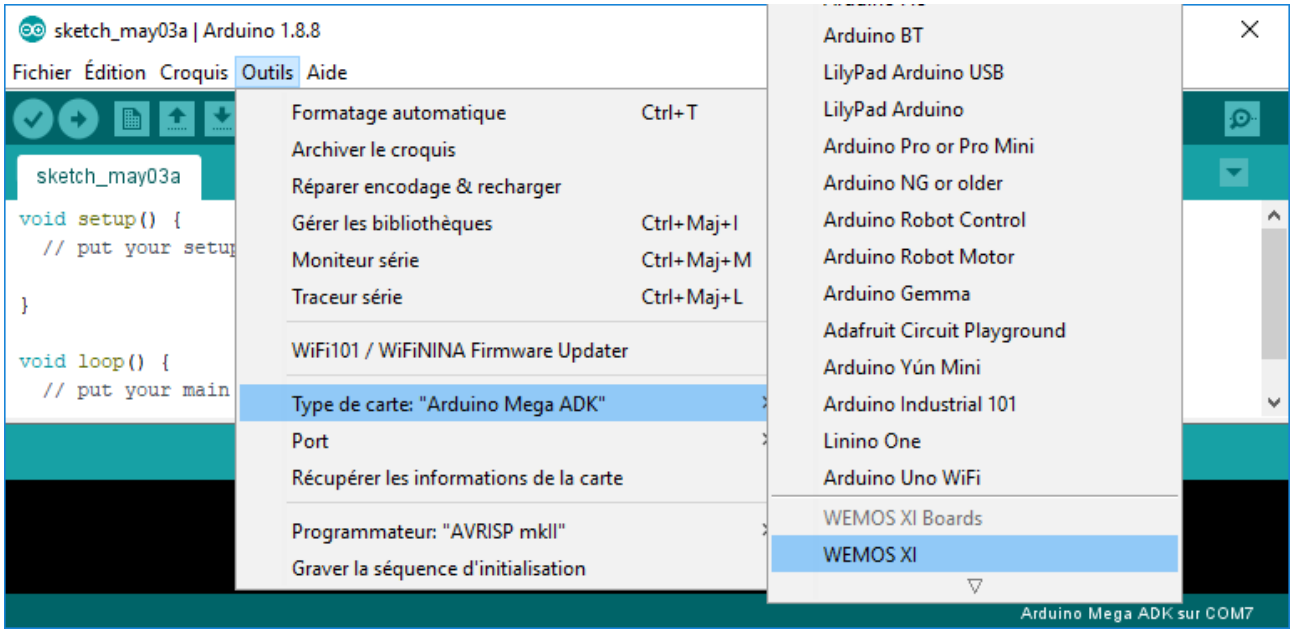
Sur Windows 10, le driver USB pour Wemos XI est a priori automatiquement installé
Sur Windows 7, installer le driver contenu dans le fichier usbbridgesetup_ca.zip
Brancher la carte Wemos au PC avec le câble micro USB
NB : c'est une carte à base de processeur ATmega328P comme l'Arduino UNO
Un port COM a été automatiquement créé. Il est associé à la carte Wemos XI
Au besoin, dans gestionnaire de périphériques, faire clic droit puis "mettre à jour le pilote"

2.2 Configuration de l'IDE Arduino pour la carte Wemos XI

Installer l'IDE Arduino, s'il n'est pas déjà installé.
Dans le dossier **Mes Documents**, il y a un dossier **Arduino** qui contient déjà un dossier **libraries**
Dans ce dossier **Arduino**, créer une arborescence **hardware\wemos** pour y décompresser le fichier Arduino_XI-2019-04-24.zip (librairie de la carte Wemos).
Renommer le dossier Arduino_XI-marser en 11.
IMPORTANT : L'arborescence obtenue dans "Mes Documents" doit être EXACTEMENT **hardware\wemos\11** comme dans l'illustration ci dessous :



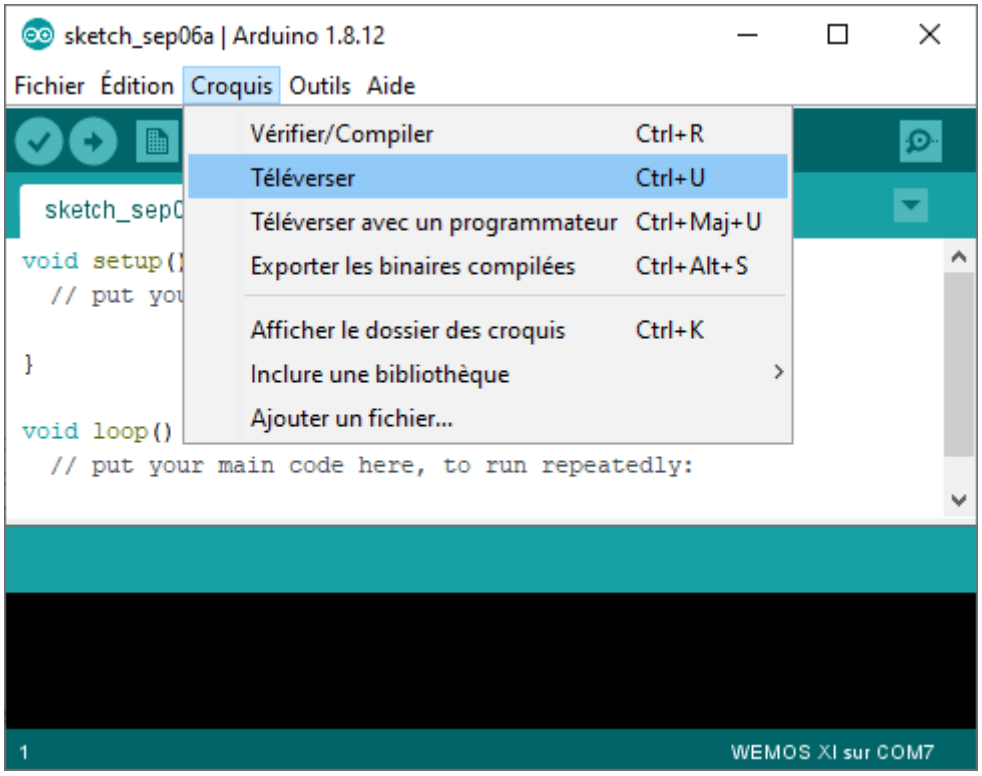
Après avoir redémarré l'IDE Arduino, il est maintenant possible de choisir le type de carte "WEMOS XI"



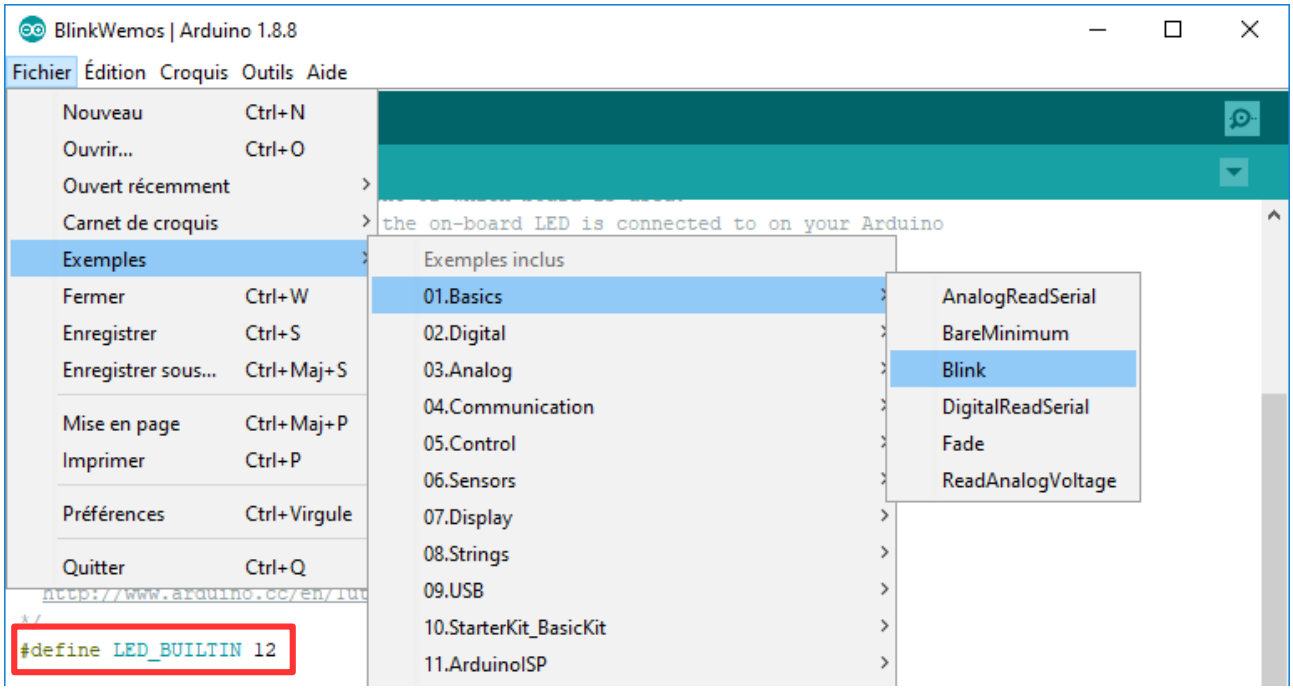
Choisir également le bon Port COM qui est associé à la carte Wemos (menu Outils / Port)

Vérifier qu'un sketch vide compile avec ce type de carte, et qu'il peut être téléversé vers la carte.
IMPORTANT : Avant de téléverser, **débrancher temporairement le fil bleu** (Rx sur la carte Wemos).
En effet, l'UART de la carte Wemos est à la fois utilisé pour l'USB et pour la communication via les broches Tx et Tx.

Vérifier qu'il n'y a pas d'erreur lors du téléversement du programme vers la carte.

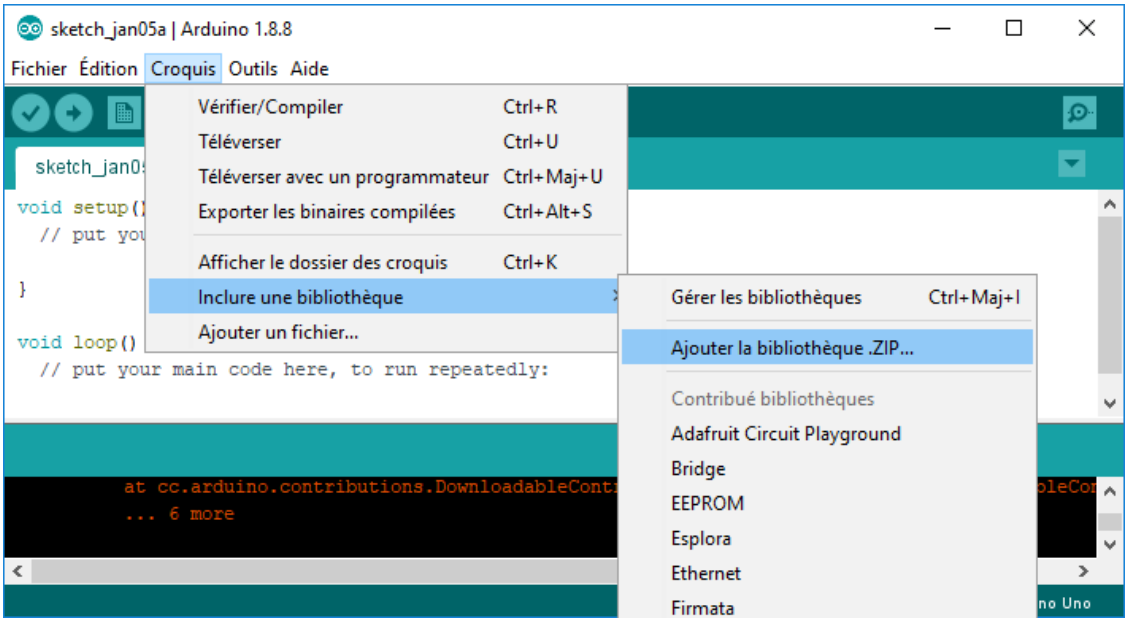


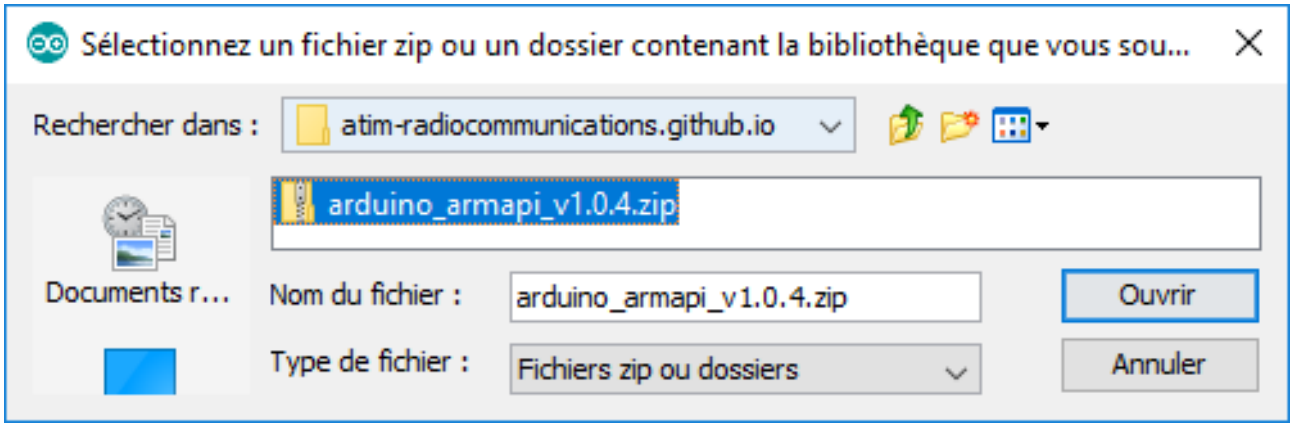
Il serait également possible de vérifier que le programme exemple "Blink" fonctionne également sur la carte Wemos XI (la Led de la carte doit clignoter)
NB : Avec cet exemple, penser à modifier le n° de la sortie qui est reliée à la LED (D12 sur Wemos XI)



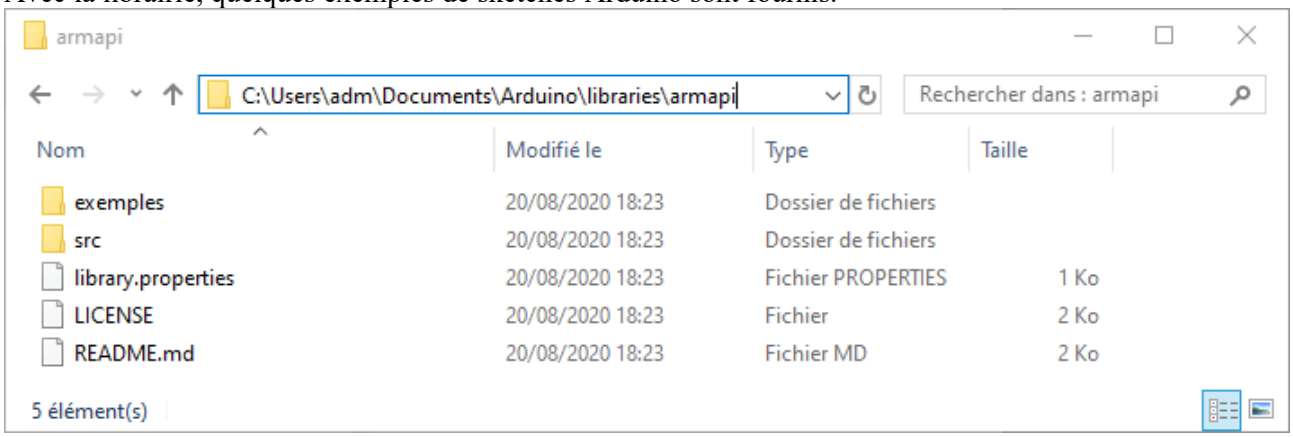
2.3 Configuration de l'IDE Arduino pour utiliser le shield LoRaWAN ATIM

ATIM fournit une bibliothèque qui permet d'utiliser le shield depuis un sketch Arduino.
Dans l'IDE Arduino, inclure la bibliothèque fournie par ATIM arduino_armapi_v1.0.4.zip



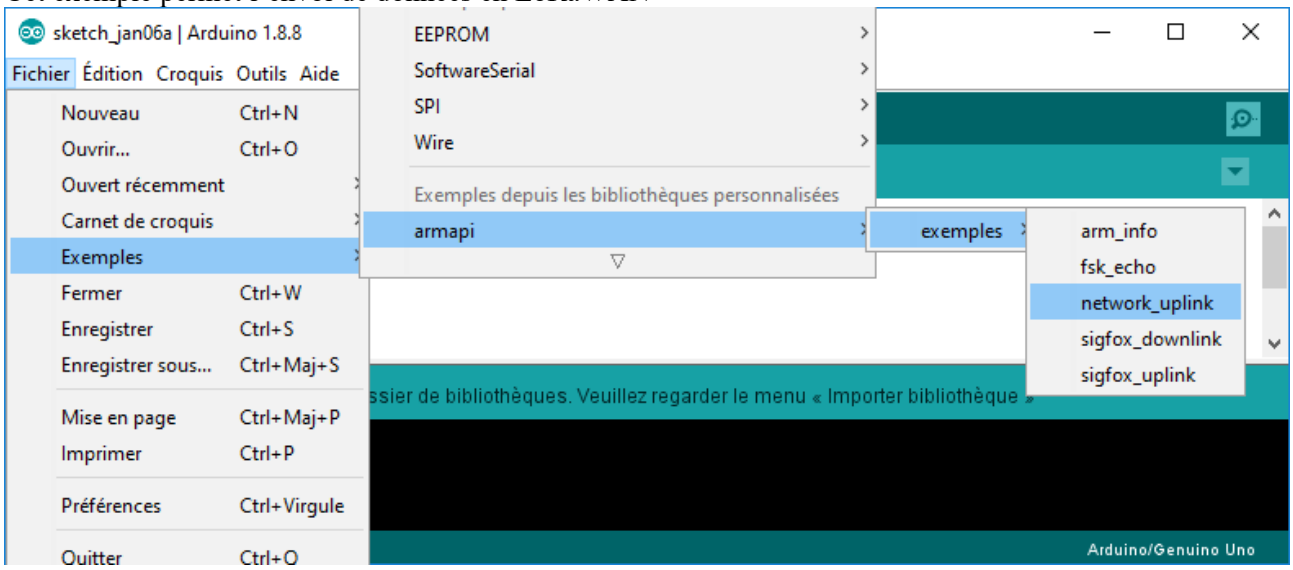


NB : l'installation de la librairie crée un dossier "armapi" dans le dossier des librairies de l'IDE Ardiuino.
Avec la librairie, quelques exemples de sketches Arduino sont fournis.

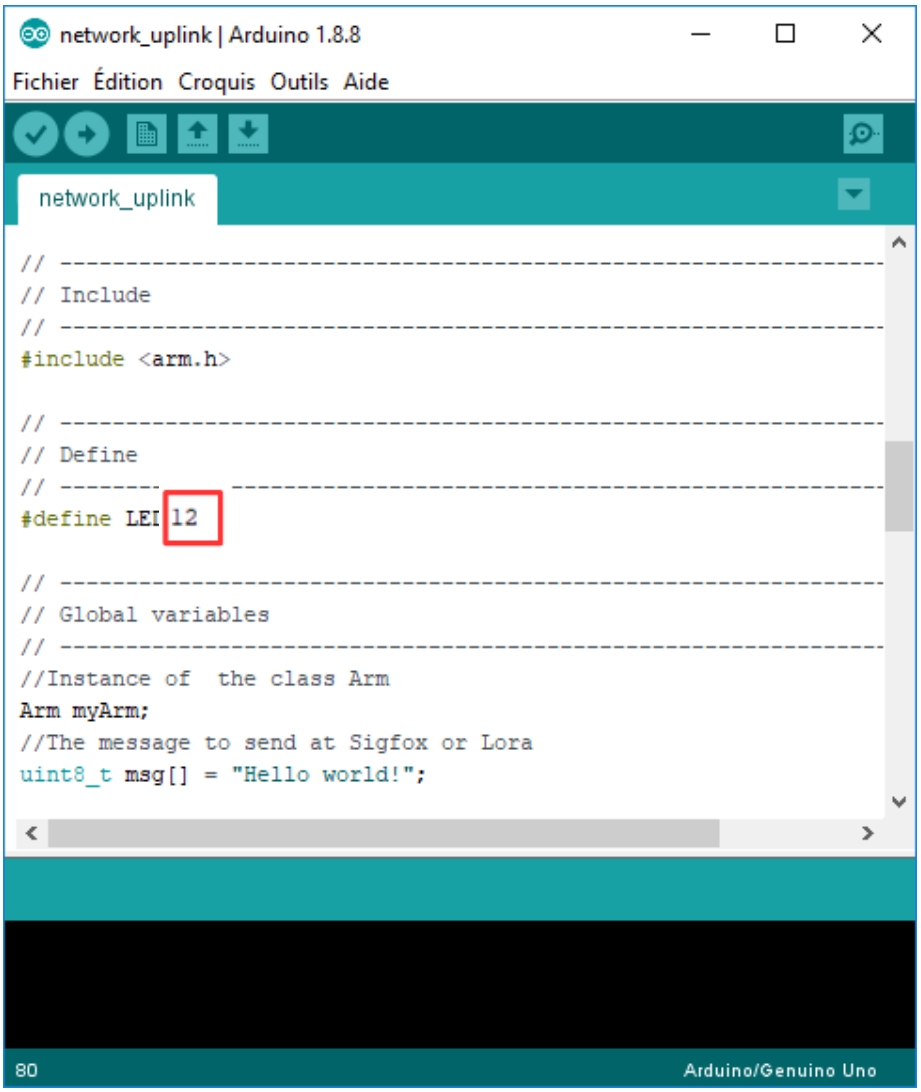


2.4 Programme exemple ATIM

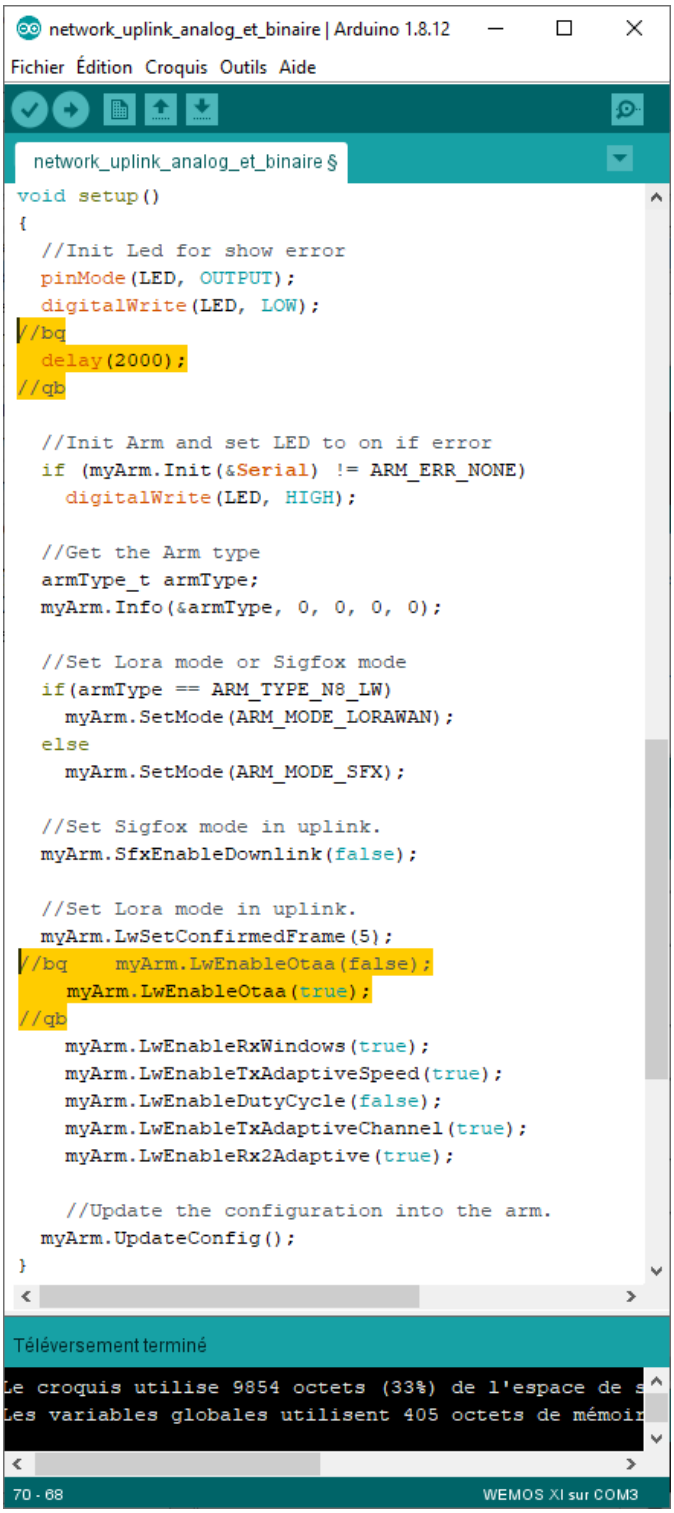
Dans l'IDE Arduino, ouvrir l'exemple "network_uplink" fourni.
Cet exemple permet l'envoi de données en LoRaWAN



Modifier le n° de LED (12 sur la carte Wemos au lieu de 13 habituellement)
NB : Cette modification n'est pas nécessaire sur une Arduino UNO
Le sketch utilise



Dans la fonction setup() insérer un temps de pause pour laisser le temps au shield de démarrer avant de lui envoyer une commande
Modifier également une ligne pour activer l'OTAA (cf figure ci-dessous)
Rappel : l'activation OTAA est plus sûre que l'activation ABP car les clés de chiffrement et de signature cryptographique sont régénérées à chaque session.

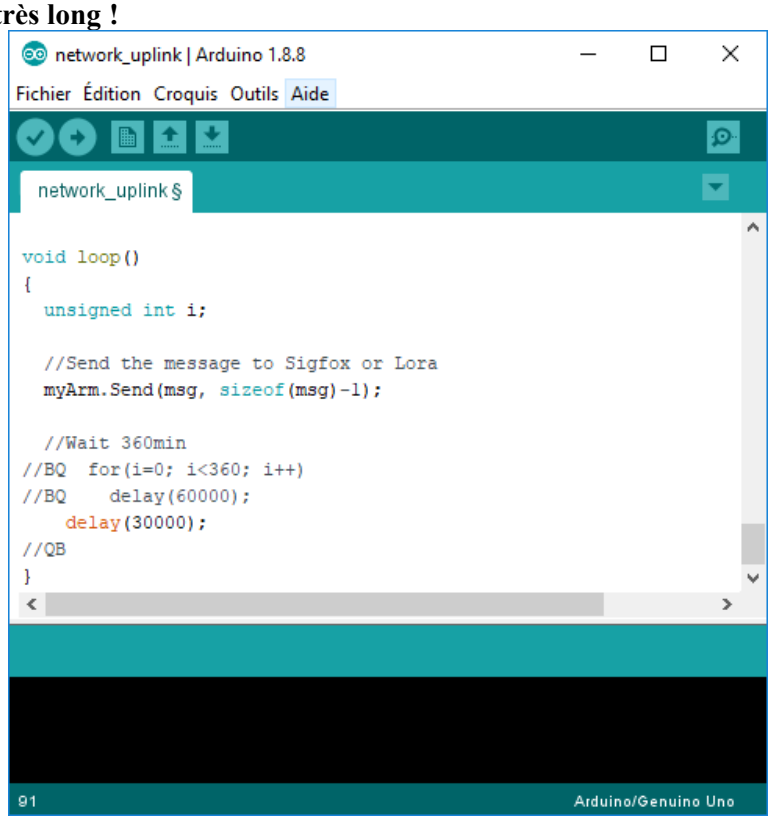


Dans la fonction loop() il y a une boucle for() qui appelle 360 fois la fonction delay(60000)
Cela a pour effet de faire une pause de 360 x 60 s = 360 minutes

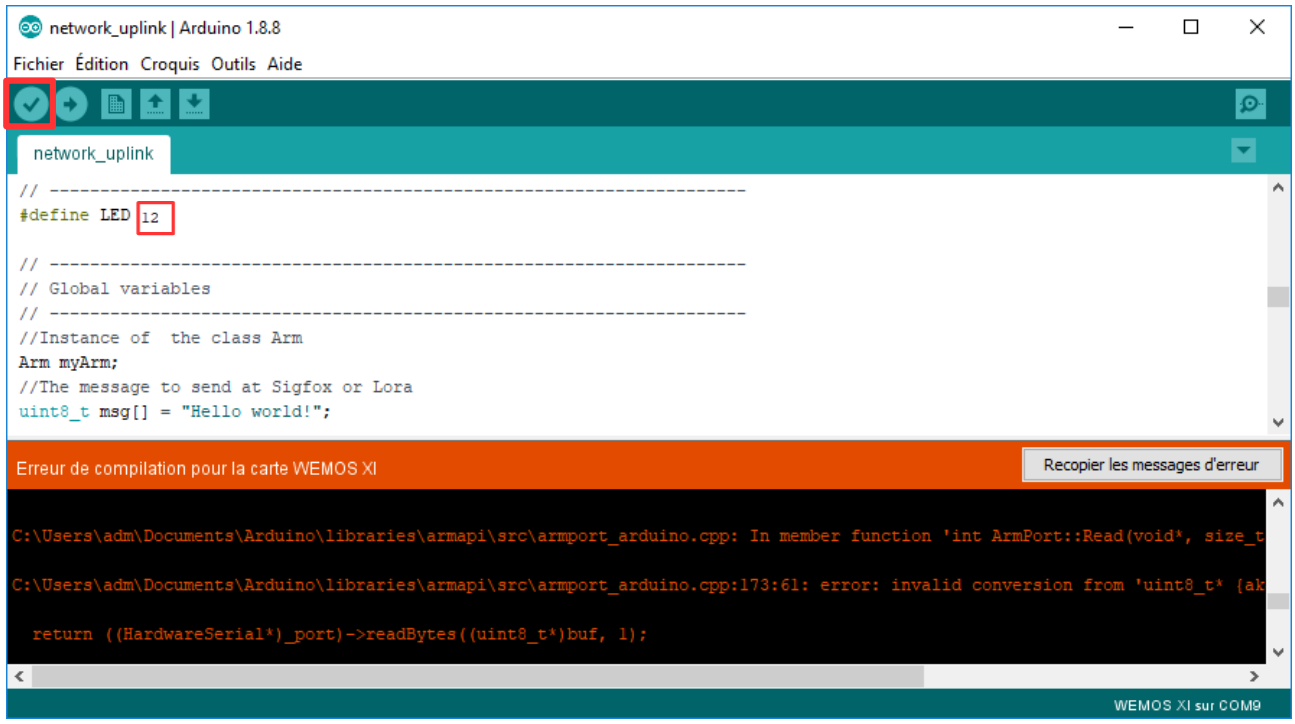
Modifier cette partie de programme pour limiter la pause à 30s soit 30 000 ms (cf figure page suivante).
NB : cette valeur réduite est intéressante lors des premières mises au point.

Une fois que le projet est au point il est nécessaire d'avoir une temps plus long à la fois pour réduire la consommation du device, mais également pour éviter de saturer la bande de fréquences.

NB : pour ce TP, ne pas oublier de désactiver la ligne for(…) en ajoutant // en début de ligne sinon, le délai d'attente sera très long !

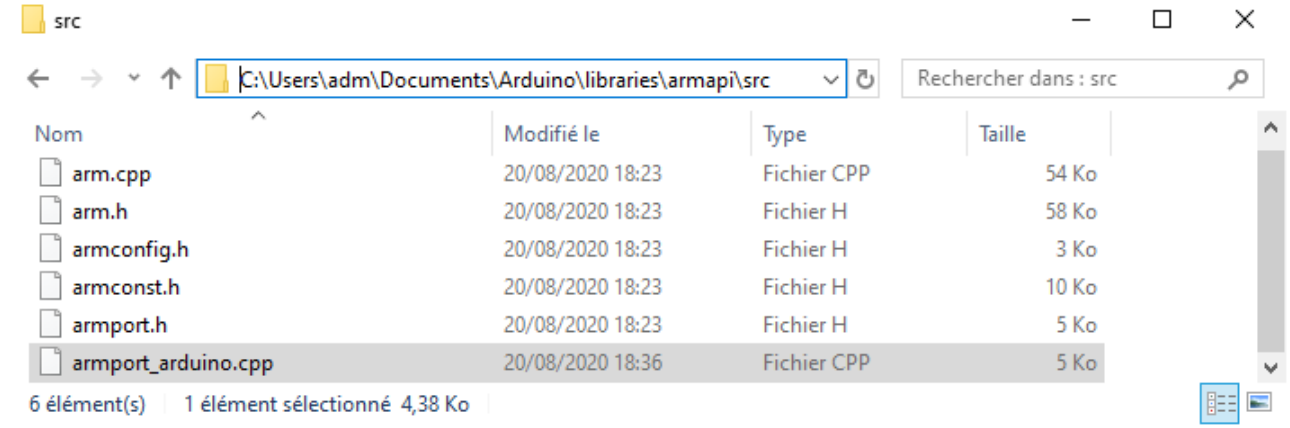


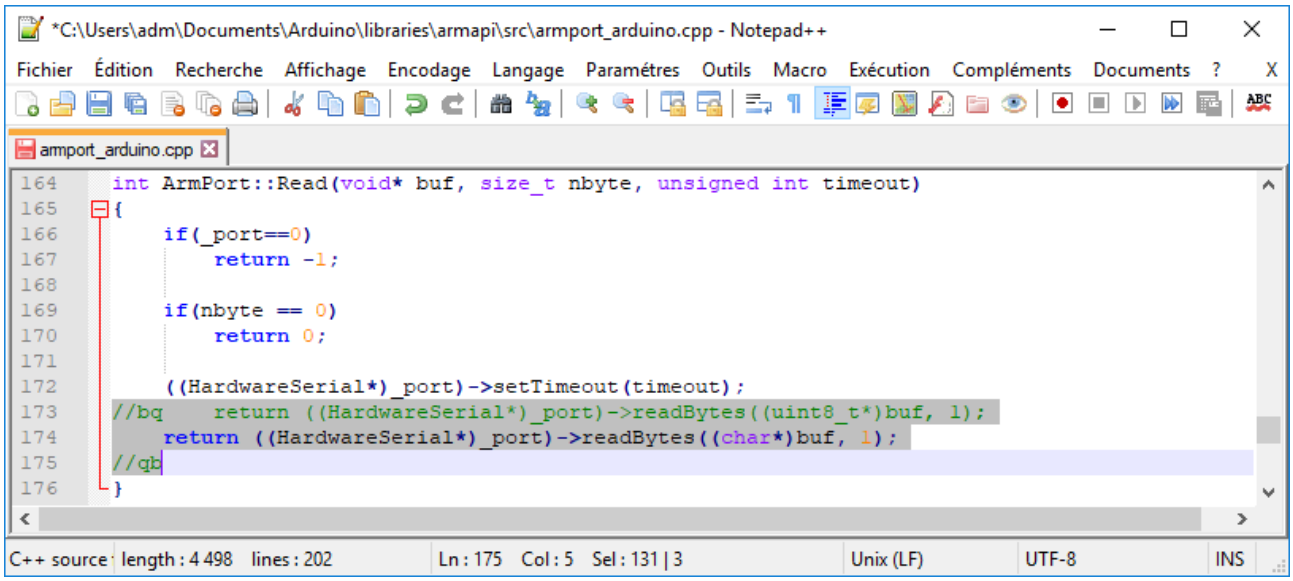
Compiler le sketch. Une erreur de compilation apparaît dans la librairie armapi.



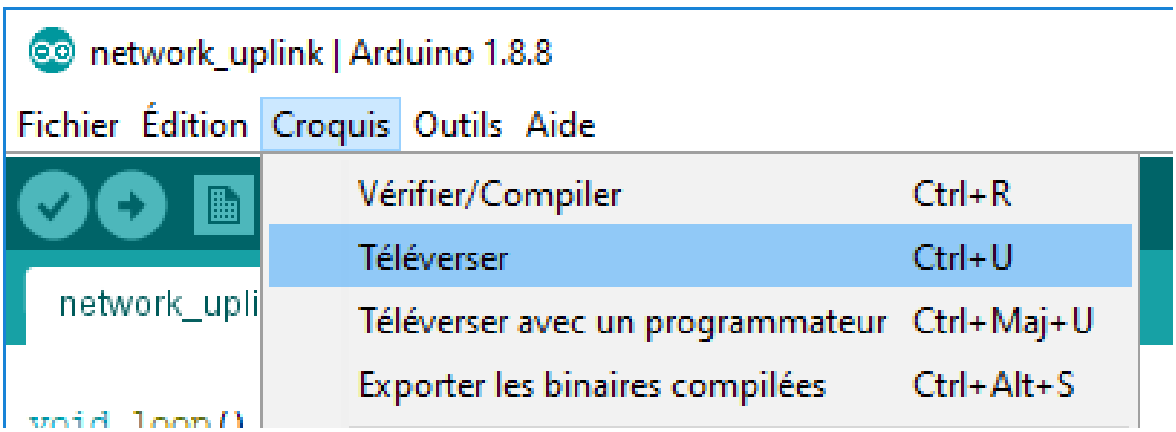
Il s'agit d'une erreur de type lors d'un appel à la méthode readBytes(), qu'il est possible de corriger (cf page suivante)

A l'aide d'un éditeur (comme par exemple notepad++), corriger l'erreur ligne 173 dans le fichier armport_arduino.cpp (cf figure page suivante)
Il s'agit en fait de remplacer (uint8_t*) par (char*) pour respecter le type de pointeur attendu par la méthode readBytes()



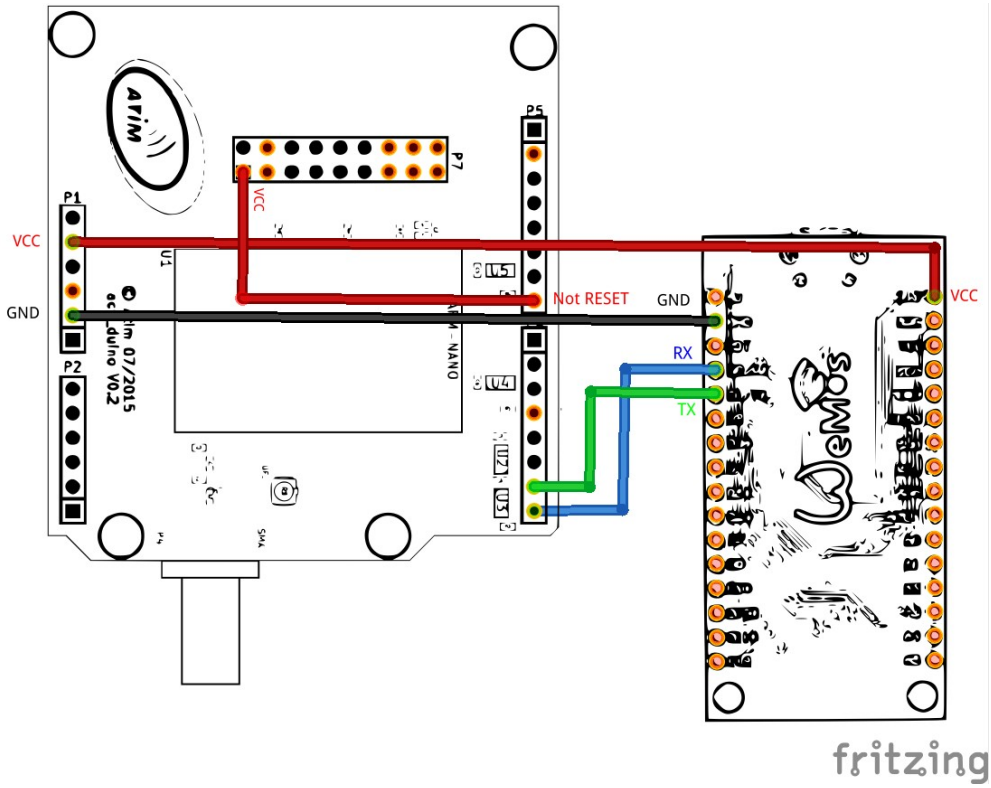


Compiler de nouveau le sketch. Le sketch se compile désormais sans erreur.
Téléverser alors le sketch vers la carte Wemos (ne pas oublier de débrancher le fil bleu Rx le temps du téléchargement)



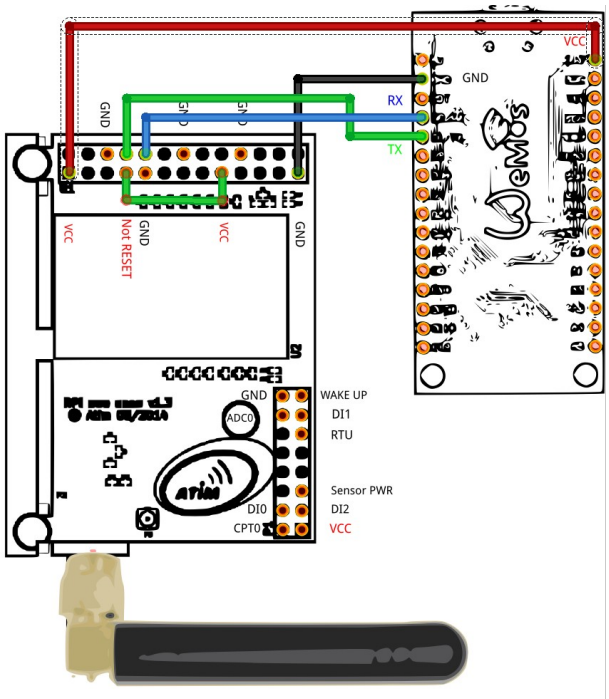
2.5 Câblage

Vérifier que la carte Wemos est bien reliée au shield ATIM comme sur le schéma ci-dessous.
Note importante : sur la carte Wemos (tout comme sur une Arduino d'ailleurs), l'UART est connecté à la fois à l'USB (port COM virtuel) ainsi qu'aux broches Rx et Tx.
Pour téléverser le sketch via l'USB, il est donc nécessaire de débrancher le fil bleu (Tx côté ATIM)
Pour tester le sketch, rebrancher ce fil bleu pour permettre la communication entre la carte Wemos et le device ATIM. Puis débrancher et rebrancher également l'USB pour redémarrer.
Cette manipulation assez contraignante doit être faîte à chaque fois que l'on souhaite modifier le sketches



Note : avec une carte Arduino (Uno, Mega,...) lorsque le sketch est au point, le shield peut se monter directement sur l'Arduino, sans fil volant.

Montage avec le shield ACW-RPI :

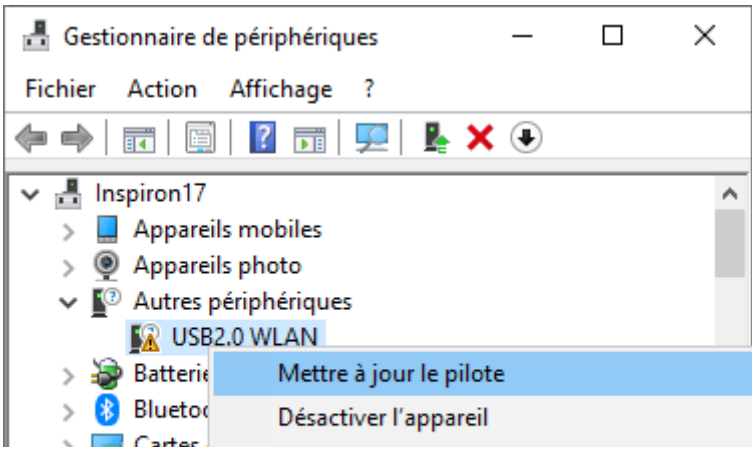


3 Network server ChirpStack

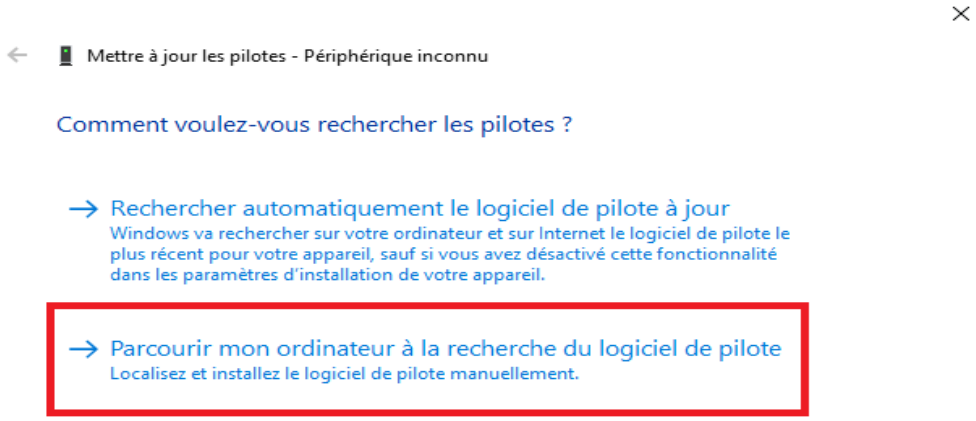
Les gateways LoRaWAN et le network serveur ChirpStack sont installées sur un réseau privé 192.168.1.0/24 , auquel il est possible d'accéder en WiFi.

3.1 Clé WiFi SMC

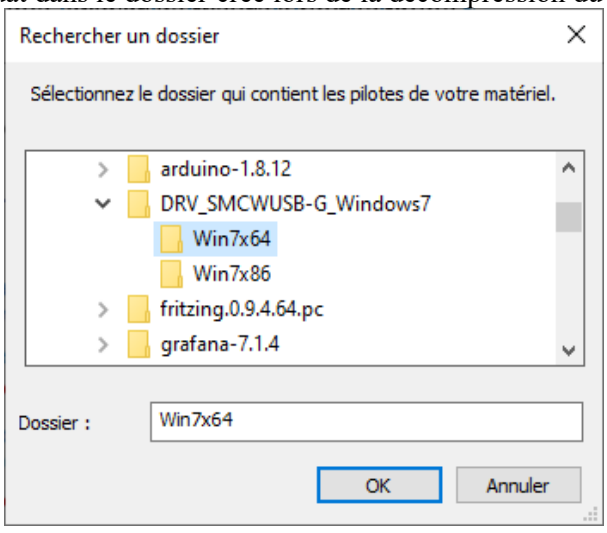
Si votre ordinateur dispose du WiFi, passer au chapitre suivant.
Si votre ordinateur n'est pas équipé de WiFi, il est possible d'utiliser une clé WiFi
Ex : clé SMS Ez Connect G

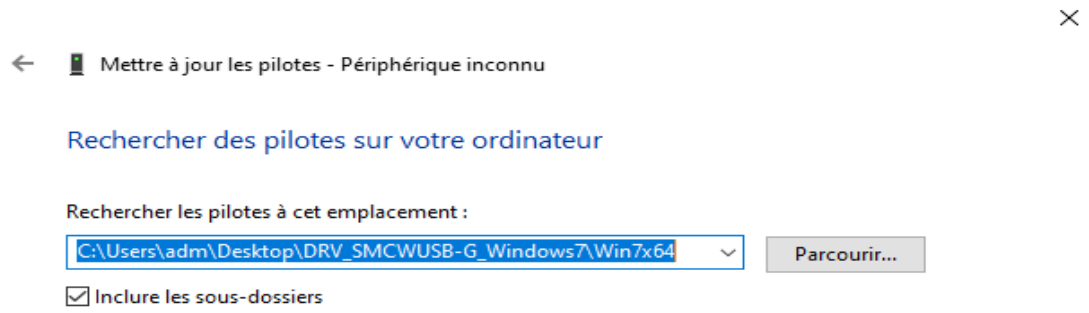


Pour cette clé, il est nécessaire d'installer le driver. Pour cela, décompresser le fichier DRV_SMCWUSB-G_Windows7.zip
Dans le gestionnaire de périphérique, choisir de mettre à jour le pilote.

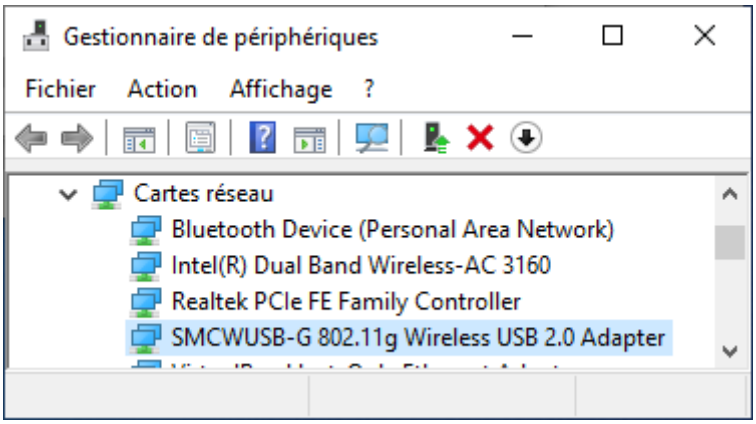


Choisir le sous dossier adéquat dans le dossier créé lors de la décompression du zip.





Une fois le driver installé, il est possible de se connecter en WiFi

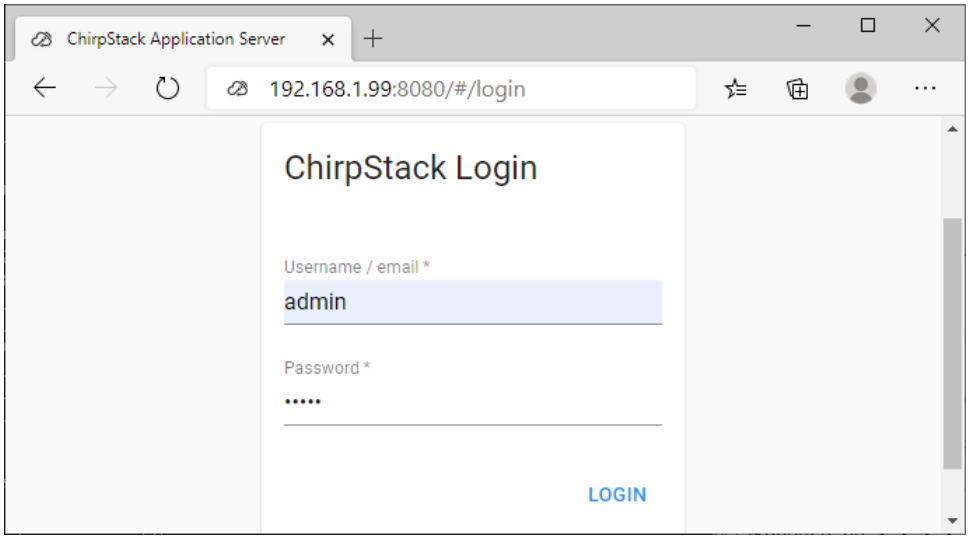


3.2 Connexion au réseau

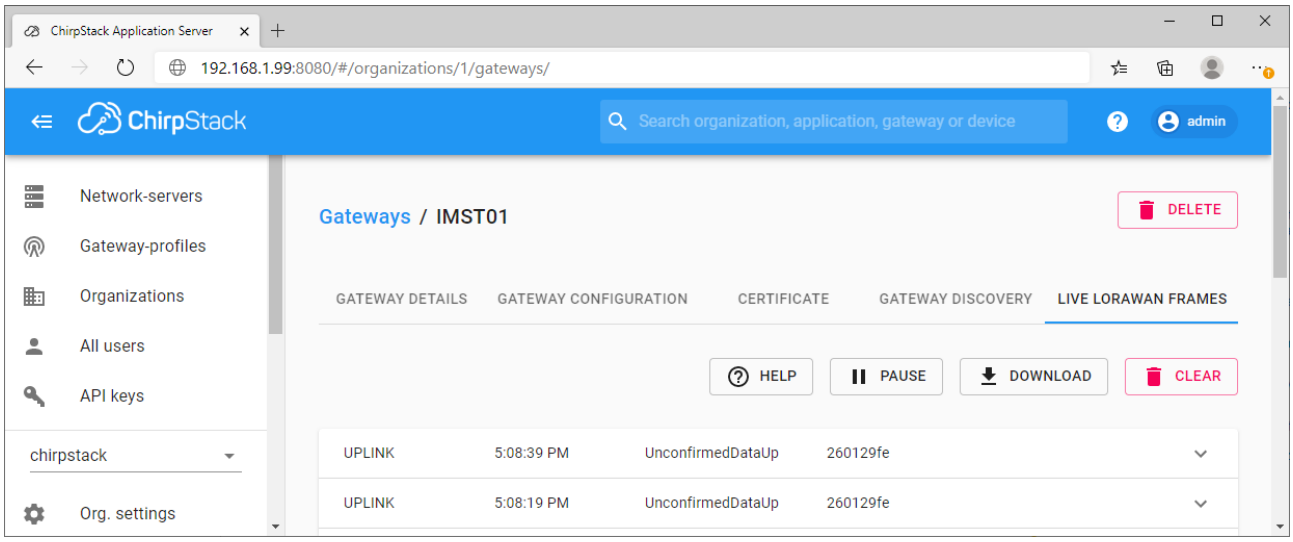
Point d'accès WiFi : Sweex
Mot de passe : (demander au formateur)

3.3 Accès au Network Server Chirpstack

Une fois connecté au réseau, accéder au network server ChirpStack
username : admin
password : admin

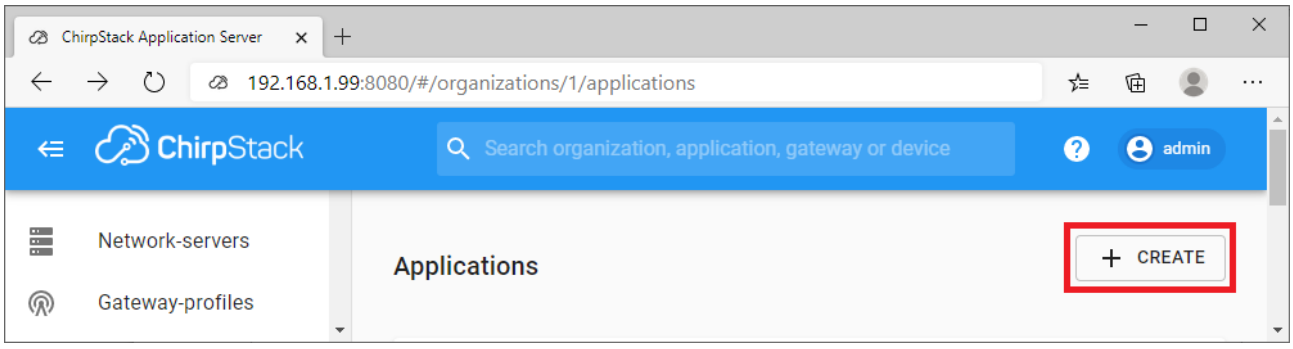


Le menu "Gateways" permet de voir l'activité des différentes gateways.

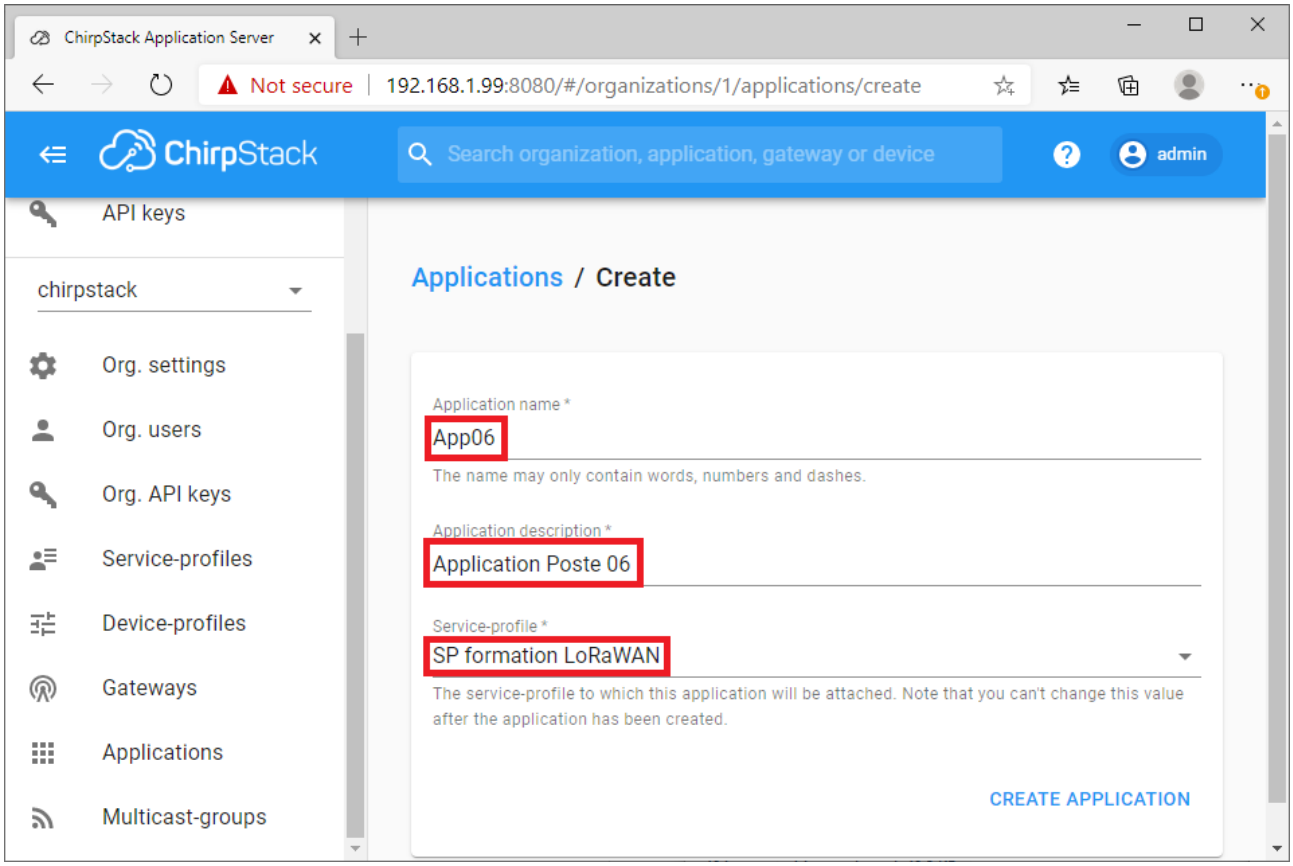


3.4 Application

Une application permet de regrouper l'ensemble des devices d'un projet.
Sur un même serveur ChirpsStack, on peut créer plusieurs applications (une par projet)
Nous allons créer une application par binôme



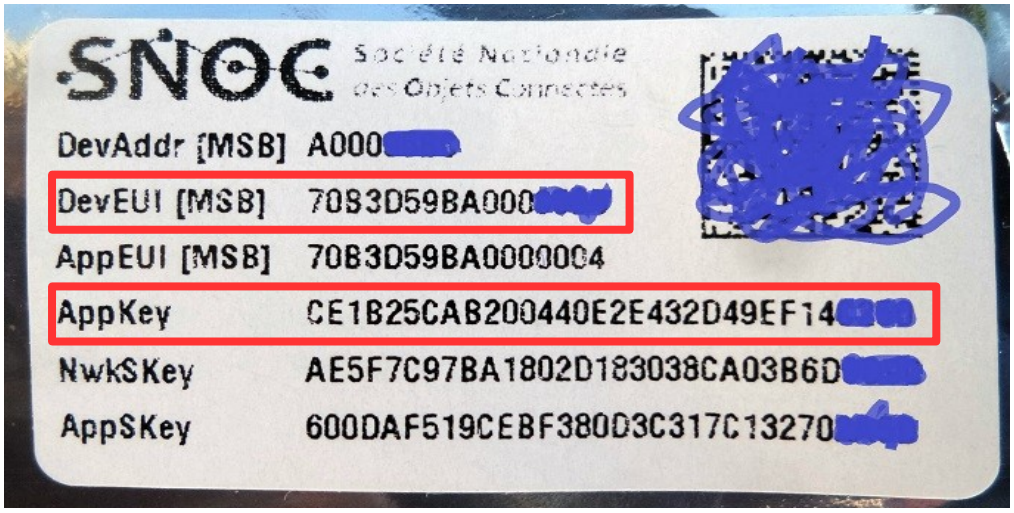
Exemple ci dessous, avec le poste 06.
NB : sélectionner le Service profile qui a été préalablement créé.



C'est également dans le menu Application que se règle l'intégration avec le serveur applicatif, ce que nous ferons ultérieurement.

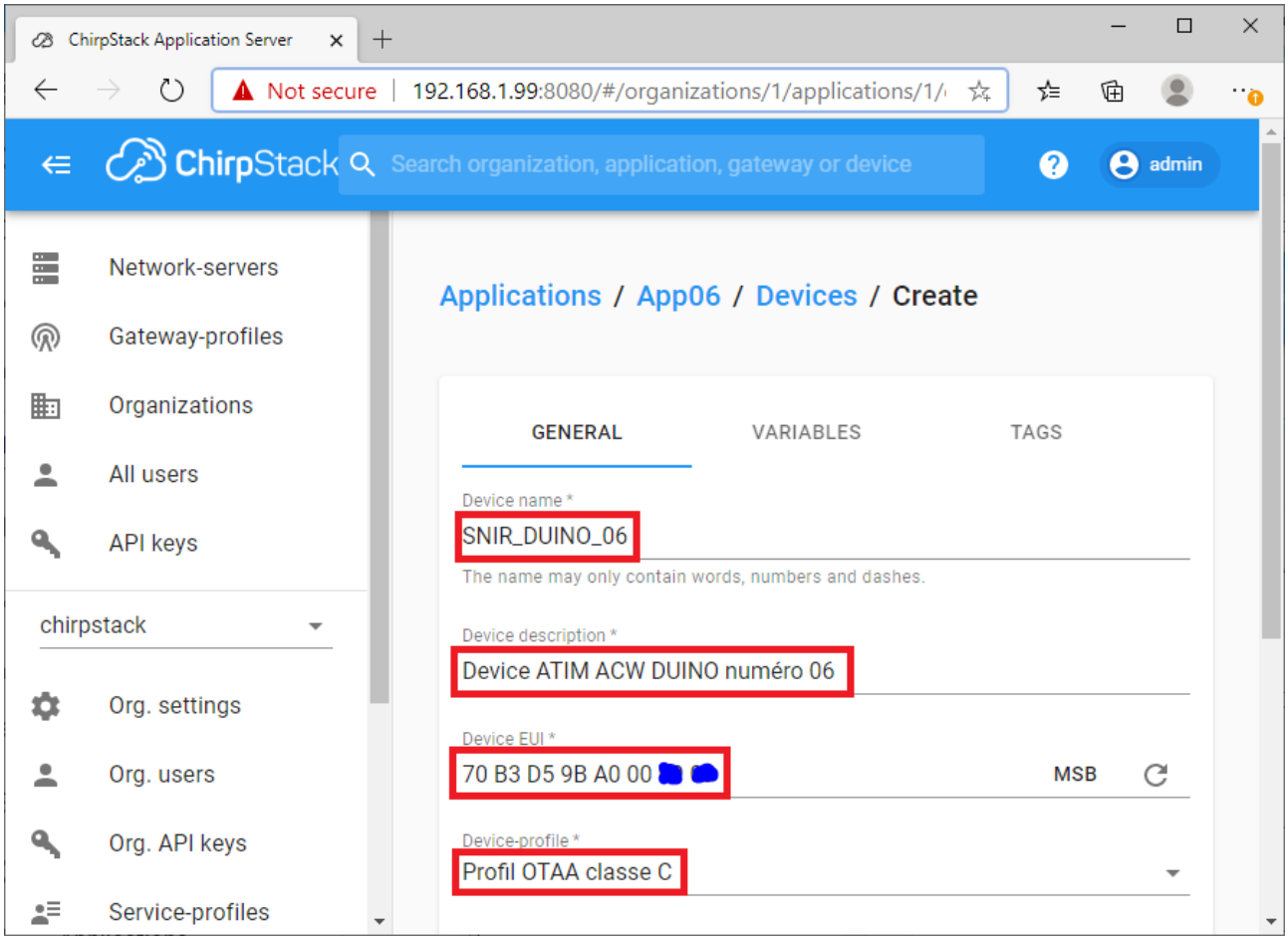
3.5 Enregistrement du device sur le Network Server

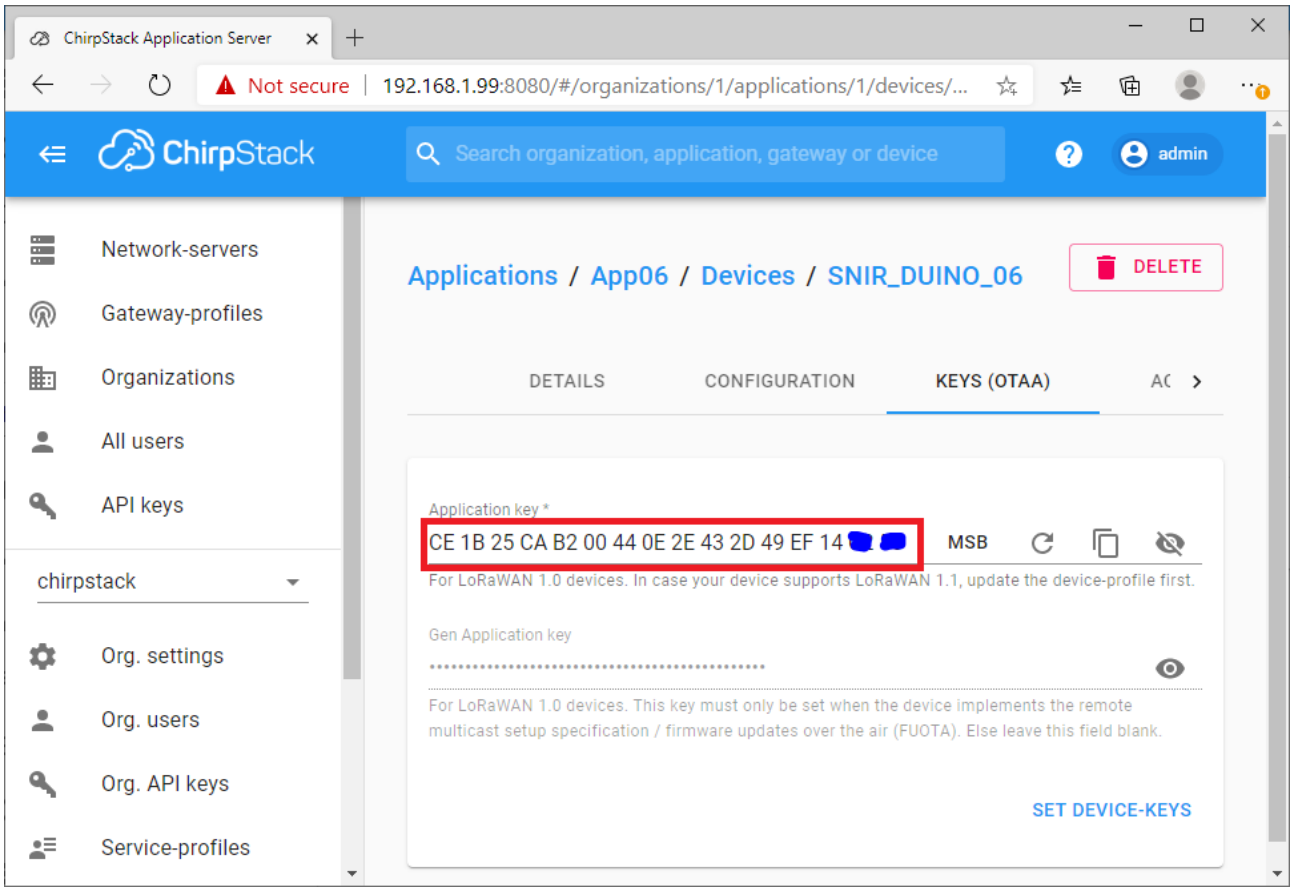
Avec le device, sont nécessairement fournis par le distributeur, des identifiants et clés d'enregistrement.
En OTAA (Over The Air Activation), seuls DevEUI, App EUI et AppKey sont requis.
Les autres identifiants sont utilisés uniquement en ABP (Activation By Personnalisation)
NB : le network server ChirpStack ne requiert pas la saisie de l'AppEUI, contrairement à la plupart des network servers (TTN par exemple)

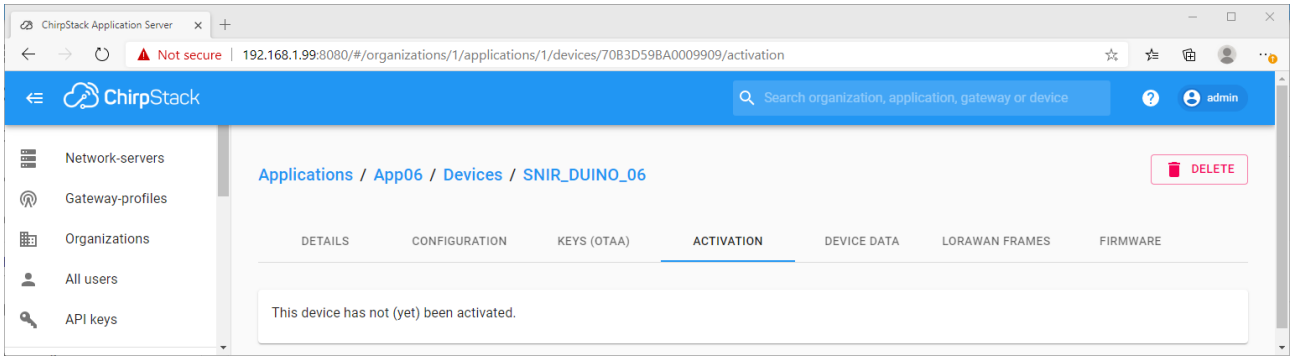


Relever DevEUI et AppKey de votre device.

Dans votre application, enregistrez votre device. Exemple avec le poste 06 :



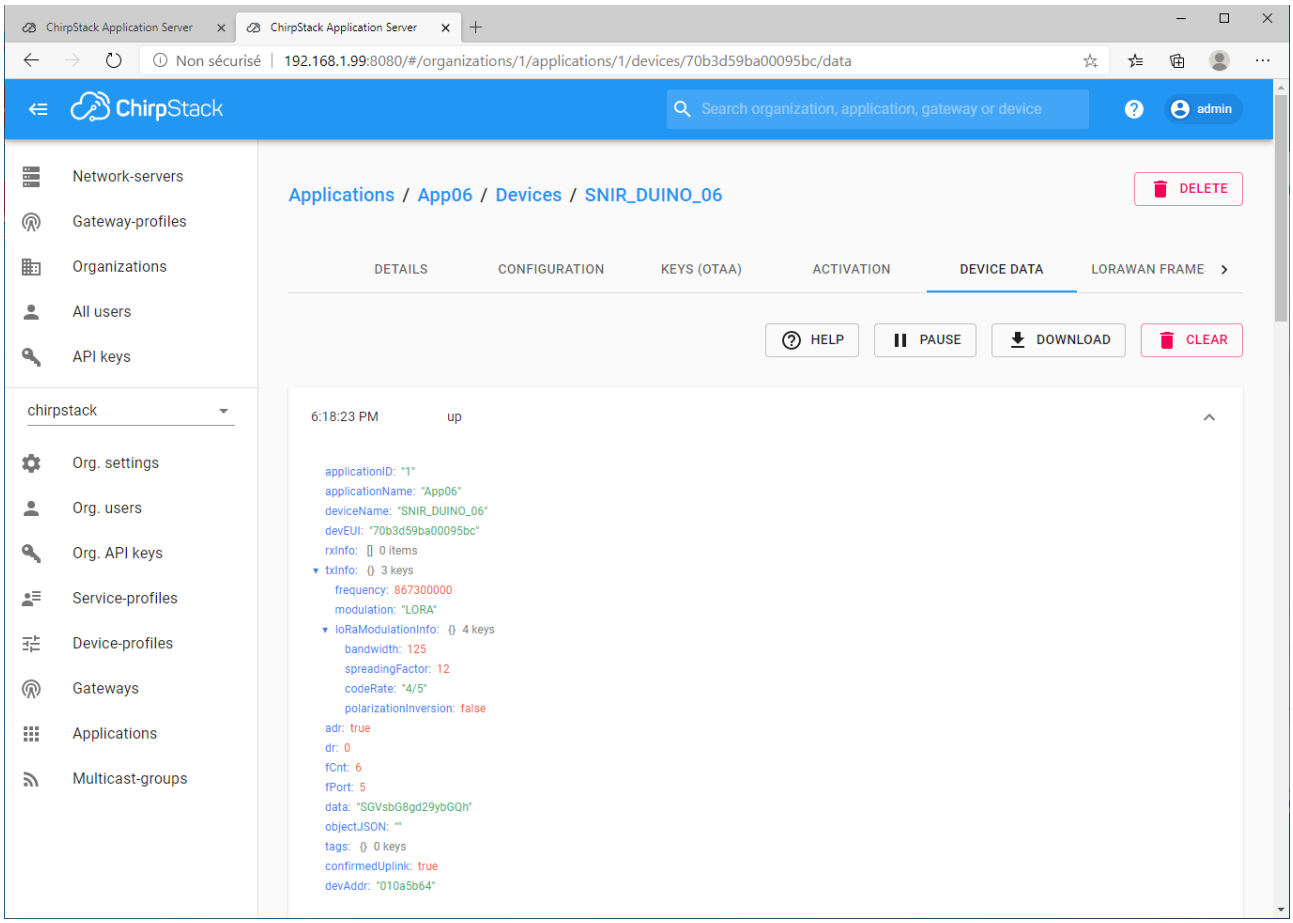




Si on fait un reset de la carte Wemos (débrancher / rebrancher l'USB), les premiers messages apparaissent au bout de quelques secondes. Cliquer sur "Device data" pour les observer.
Un message de type Uplink est normalement visible **toutes les 30 secondes**. Il est possible de cliquer sur un message pour en observer les informations.

La LED verte sur le shield doit être éteinte, mais clignoter de temps en temps (toutes les 30")
Si la led reste fixe (erreur de reset sur les shields pour Pi) débrancher puis rebrancher un des fils d'alimentation entre la carte Wemos et le shield. Ou ajouter un temps de pause en début de setup()

NB : Sur le network server, les messages sont susceptible de s'afficher avec la mention "error" car le payload "Hello World" ne respecte pas le format Cayenne qui a été paramétré.



3.6 Analyse d'un message

Chaque message contient différentes informations comme les paramètres radio LoRa qui ont été utilisées.
Certaines informations sont encodées en base64. Exemple : DevEUI, data, devAddr
L'encodage base64 n'est en aucun cas un chiffrement (cryptage). Il s'agit simplement d'un codage permettant de transmettre sous forme de caractères, des payloads contenant potentiellement des octets de valeur quelconque.

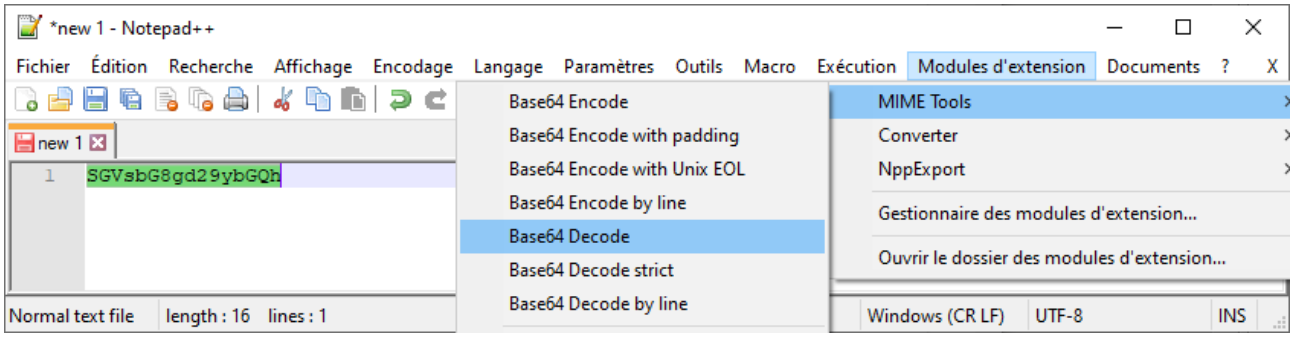
Dans le sketch exemple fourni par ATIM, le payload est constitué d'une chaîne de caractères.
Voir début du sketch juste avant setup().

```
//The message to send at Sigfox or Lora
uint8_t msg[] = "Hello world!";
```

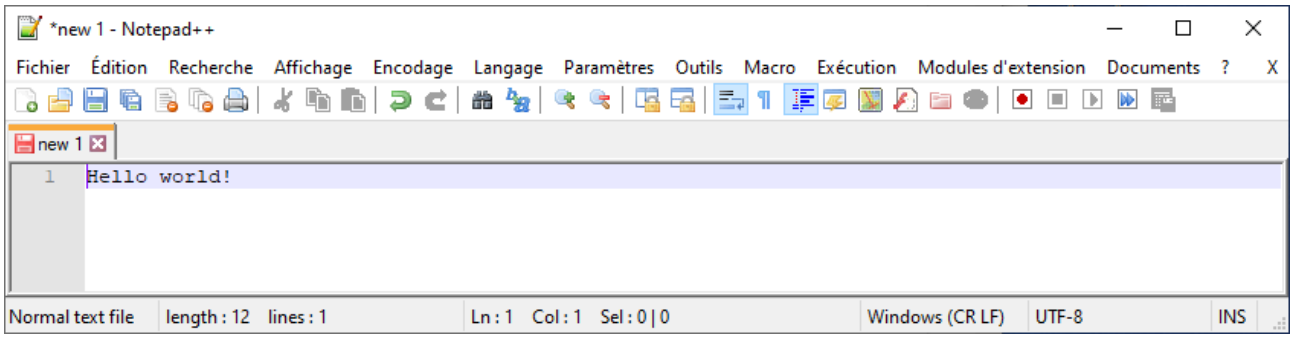
Exemple de message reçu :


```
"applicationID": "1",
"applicationName": "App06",
"deviceName": "SNIR_DUINO_06",
"devEUI": "cLPVm6AAlbw=",
"rxInfo": [],
"txInfo": {
    "frequency": 867300000,
    "modulation": "LORA",
    "loRaModulationInfo": {
        "bandwidth": 125,
        "spreadingFactor": 12,
        "codeRate": "4/5",
        "polarizationInversion": false
    }
},
"adr": true,
"dr": 0,
"fCnt": 6,
"fPort": 5,
"data": "SGVsbG8gd29ybGQh",
"objectJSON": "",
"tags": {},
"confirmedUplink": true,
"devAddr": "AQpbZA=="
```


Utilisation de la fonction de décodage "base64" incluse dans notepad++



On retrouve bien le payload transmis en LoRaWAN par le sketch



4 Modification du device

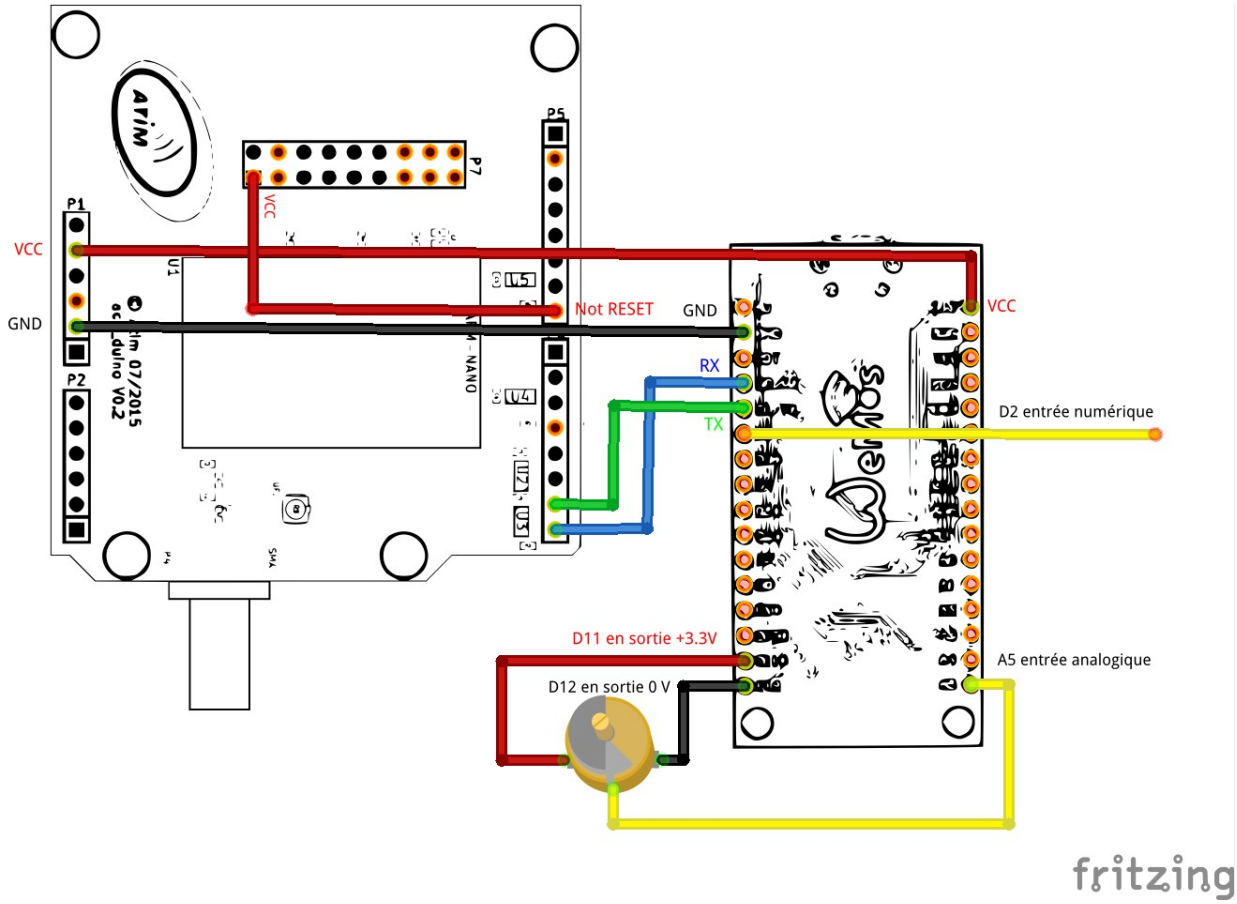
4.1 Câblage

Pour simuler le capteur d'humidité, on utilise un potentiomètre.
Le potentiomètre permet de générer une tension comprise entre 0 et 3,3V

Le nombre de broches de masse et de VCC étant limitées sur la carte Wemos, on utilisera D12 en sortie 0V et D11 en sortie 3,3V pour alimenter le potentiomètre.

Par la même occasion, on pourra tester la mise en œuvre d'entrée numérique TOR (Tout Ou Rien). On utilisera pour cela l'entrée D2.

Vérifier que le câblage correspond au schéma ci-dessous.



4.2 Sauvegarde du sketch

Sauvegardez votre sketch dans un dossier de travail pour le pas risquer d'altérer le sketch original.

4.3 Modification du sketch

Modifier la déclaration du tableau msg, qui est utilisé pour stocker les payloads
On utilise un codage LPP.
Sur les 7 octets du tableau :

- les 4 premiers octets seront utilisés pour transmettre la valeur de l'entrée analogique A5
- les 3 derniers seront utilisés pour transmettre la valeur de l'entrée numérique DI2

```
uint8_t msg[7] = {5,2,0,0,2,0,0}; // ch5 AI 00 00 ch2 DI 00
```

Dans la fonction setup, ajouter 3 lignes pour déclarer D11 et D12 en sorties, et A5 en entrée (lignes en gras)

```
void setup()
{
      //Init Led for show error
      pinMode(LED, OUTPUT);
      digitalWrite(LED, LOW);

  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(A5, INPUT);
  delay(2000);
```

Modifier également le contenu de la fonction loop() pour permettre :

1. la lecture des entrées
2. la préparation du payload au format LPP
3. l'envoi des 7 octets de payload en LoRaWAN

NB : le tableau msg[ ] permet de préparer le payload au format LPP Cayenne.
Ce payload est envolé par la méthode Send(…) de l'objet myArm

msg[ ]

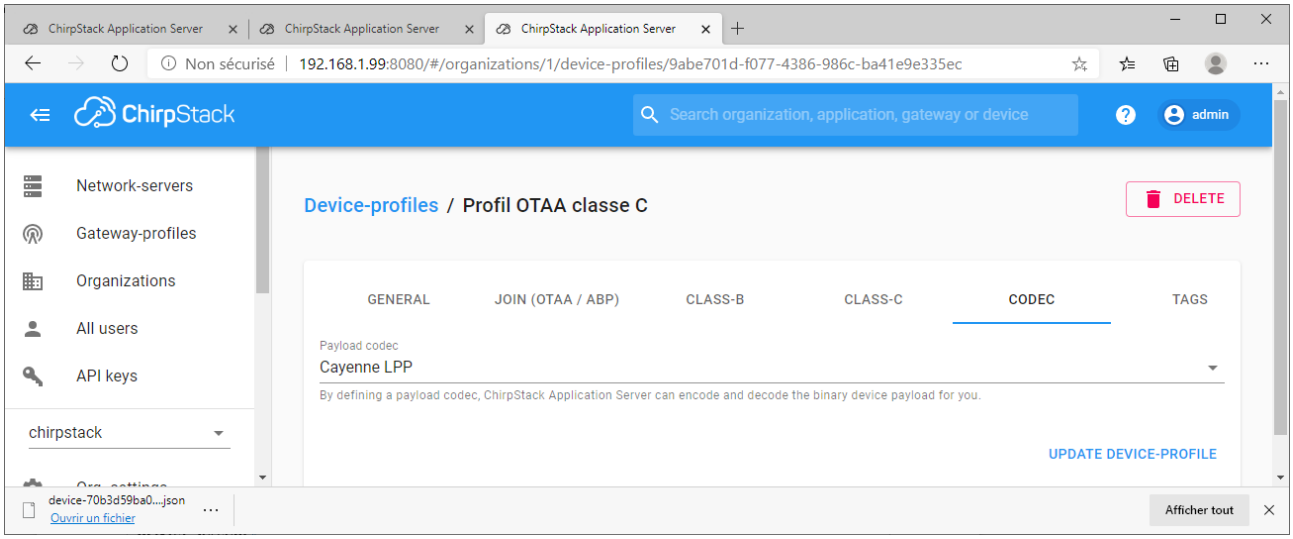
0	05	Canal n°5
1	02	Type Analogique
2	00 à FF	Tension/100 (poids FORT)
3	00 à FF	Tension/100 (poids faible)
4	02	Canal n°2
5	00	Type numérique
6	00 à 01	Niveau sur entrée 2

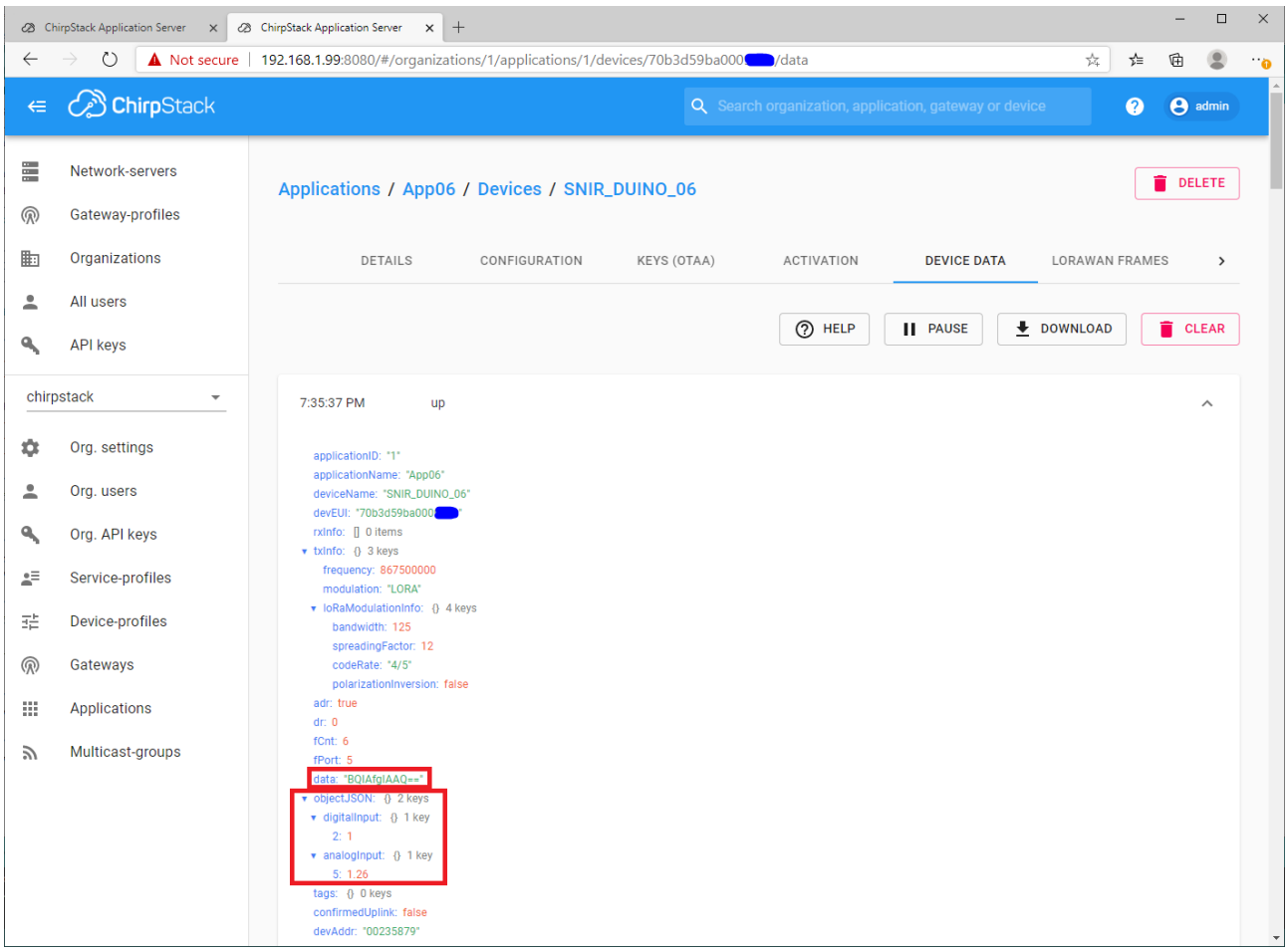
```
void loop()
{
  digitalWrite(11,HIGH); // 3.3
  digitalWrite(12,LOW); // 0
  float f5=analogRead(A5)*3.3/1024.0;
  int i5=f5*100.0+0.5;
  msg[2]=i5/256;
  msg[3]=i5%256;
  msg[6]=digitalRead(2);

  myArm.Send(msg,sizeof(msg)); // on enlève le -1
  delay(30000);
}
```

Téléverser le sketch modifié vers la carte Wemos, et tester le bon fonctionnement.

4.4 Visualisation sur ChirpStack

Le codec Cayenne LPP a été choisi dans le device profile. NB : ne pas modifier ce réglage.
Ainsi, ChirpsStack décode automatiquement les trames au format Cayenne LPP.



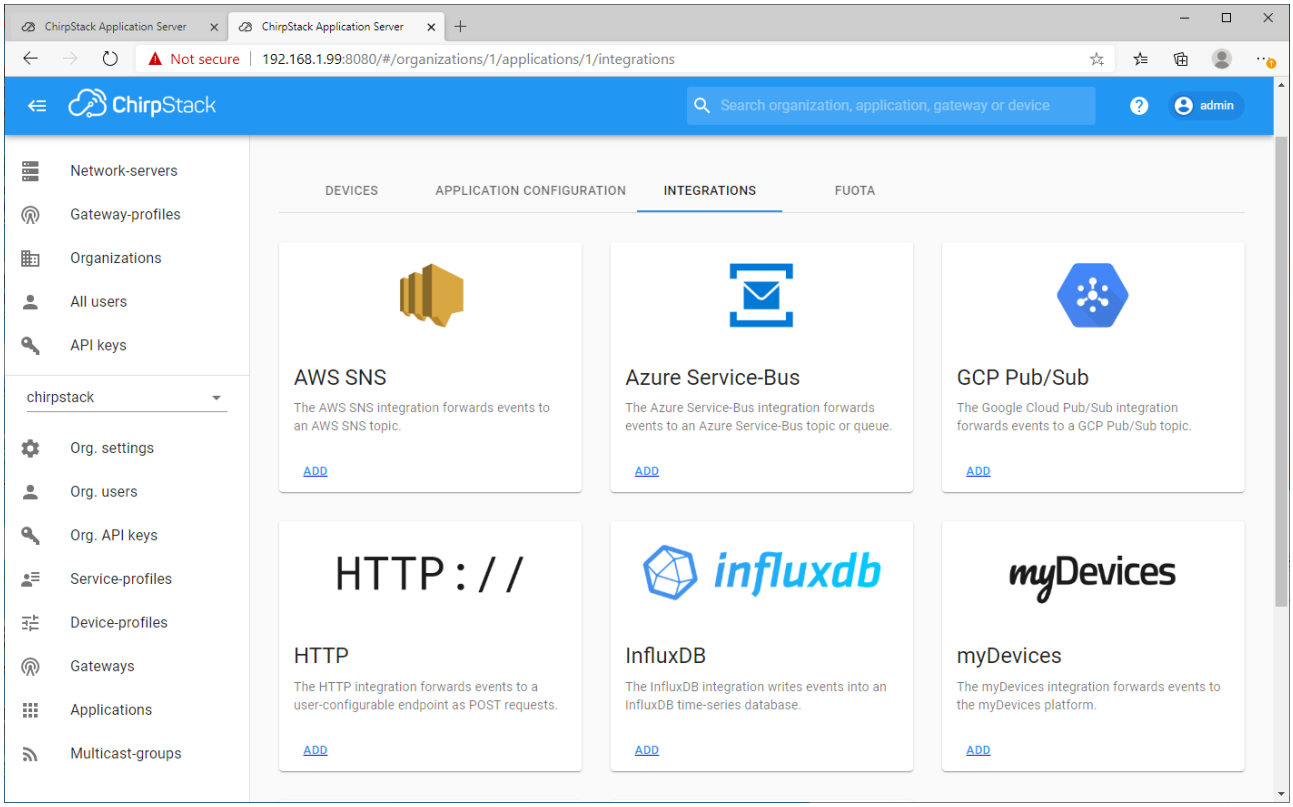


Vérifier que les valeurs uploadées sont bien décodées.
Agir sur le potentiomètre pour vérifier que la tension peut être modifiée entre 0 (roue côté noir) et 3V (roue côté rouge)
Pour tester l'entrée numérique D2, il est possible de mettre en contact le fil jaune relié à D2, tantôt à 3,3V (ex : broche du potentiomètre reliée au fil rouge), tantôt à 0V (ex : broche du potentiomètre reliée au fil noir).

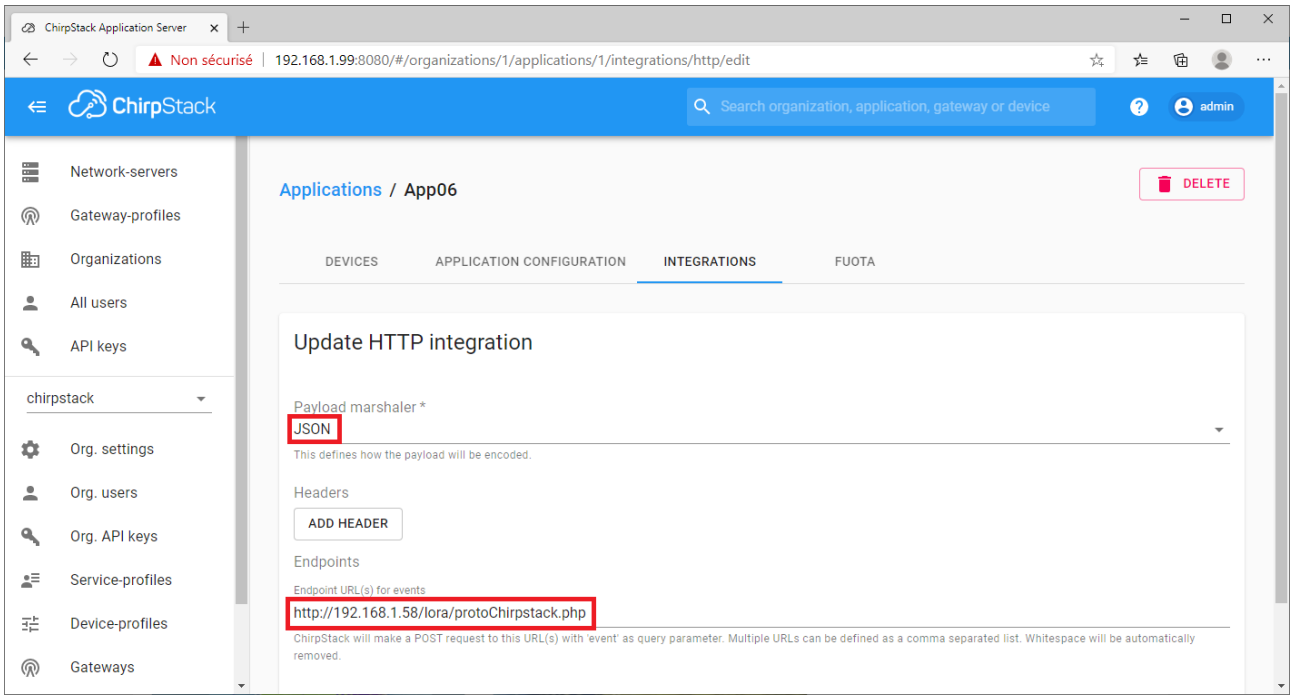
5 Serveur applicatif

5.1 Ajout intégration dans ChirpStack

Vous allez installer un serveur applicatif sur votre machine.
Pour que votre serveur applicatif reçoive les données, il faut configurer ChirpStack.
Dans votre application ChirpStack, choisir intégrations puis HTTP

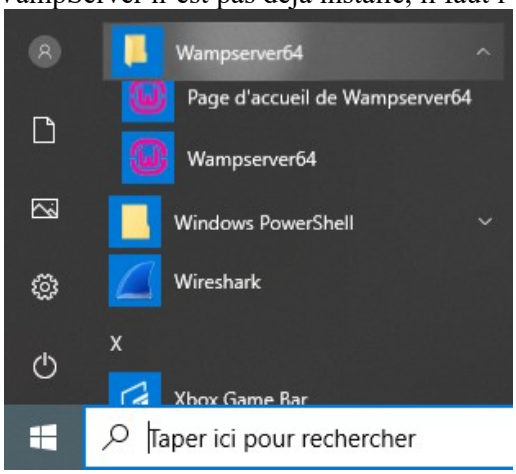


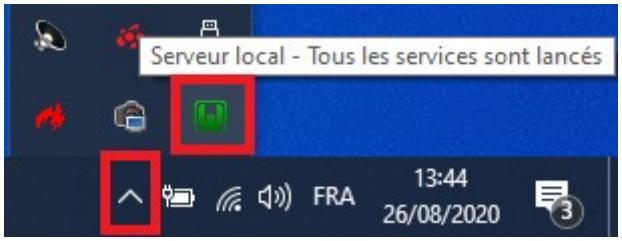
Choisir de transmettre les données en JSON. Relevez l'adresse IP de votre PC.
Saisir l'URL : `http://(votre IP)/lora/protoChirpStack.php`
C'est à cette URL que ChirpStack transmettra les données de votre device.
ChirpStack agira en client HTTP vis à vis de votre serveur applicatif.



5.2 Installation de WAMPserver

Si WampServer n'est pas déjà installé, il faut l'installer



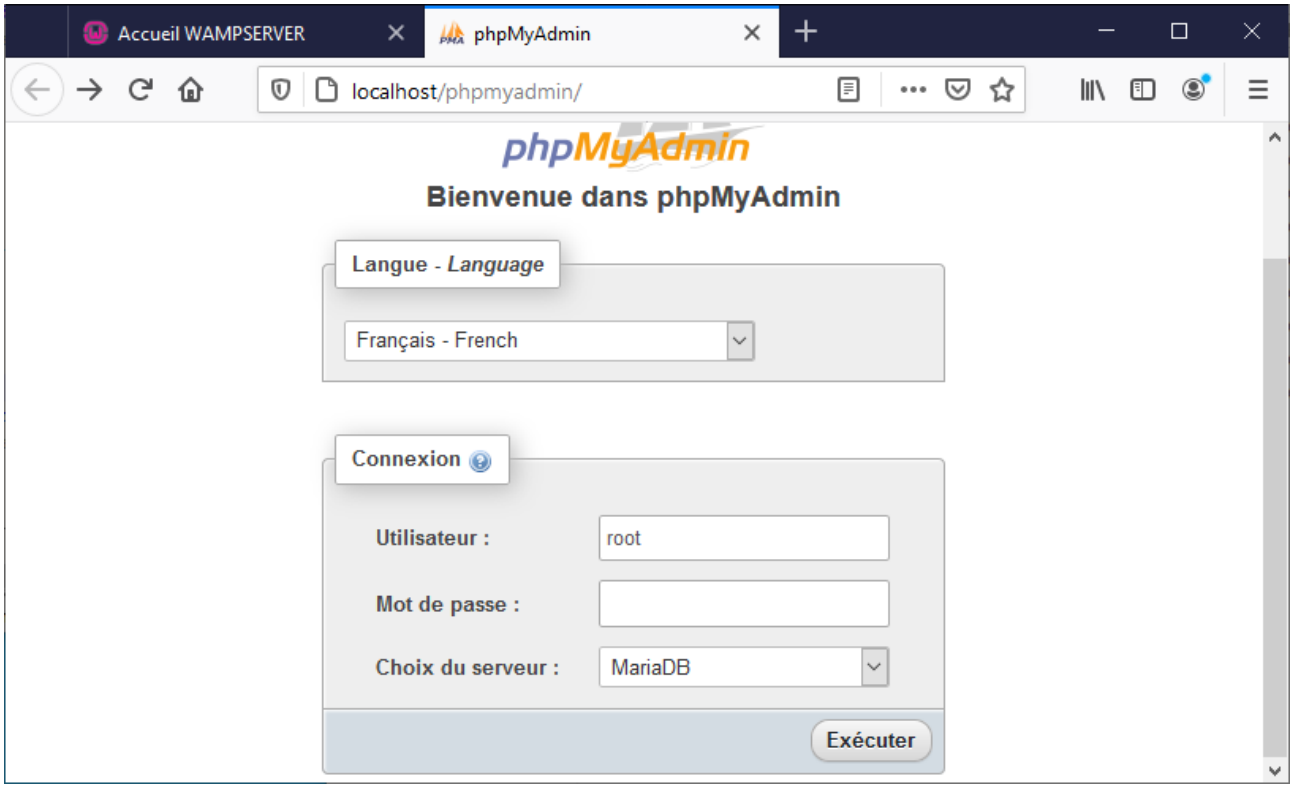


NB : Installer d'abord vcredist qui est requis par WAMPserver
Installer ensuite WAMPserver. Un redémarrage de la machine est possiblement requis.
Lorsque WAMPserver est démarré, une icône apparaît en bas à droite, et une page d'accueil est disponible

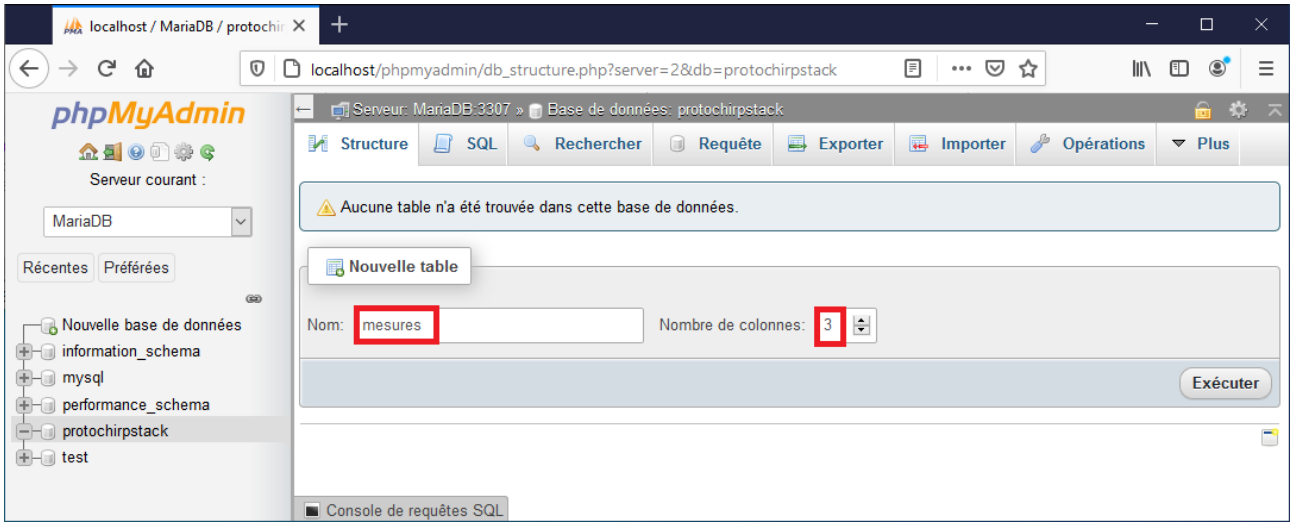
5.3 Création d'une Base de Données

Nous allons mettre en place une BdD pour stocker les mesures envoyées par le device LoRaWAN

Se connecter au serveur MariaDB (ou MySQL)
Par défaut : compte root sans mot de passe



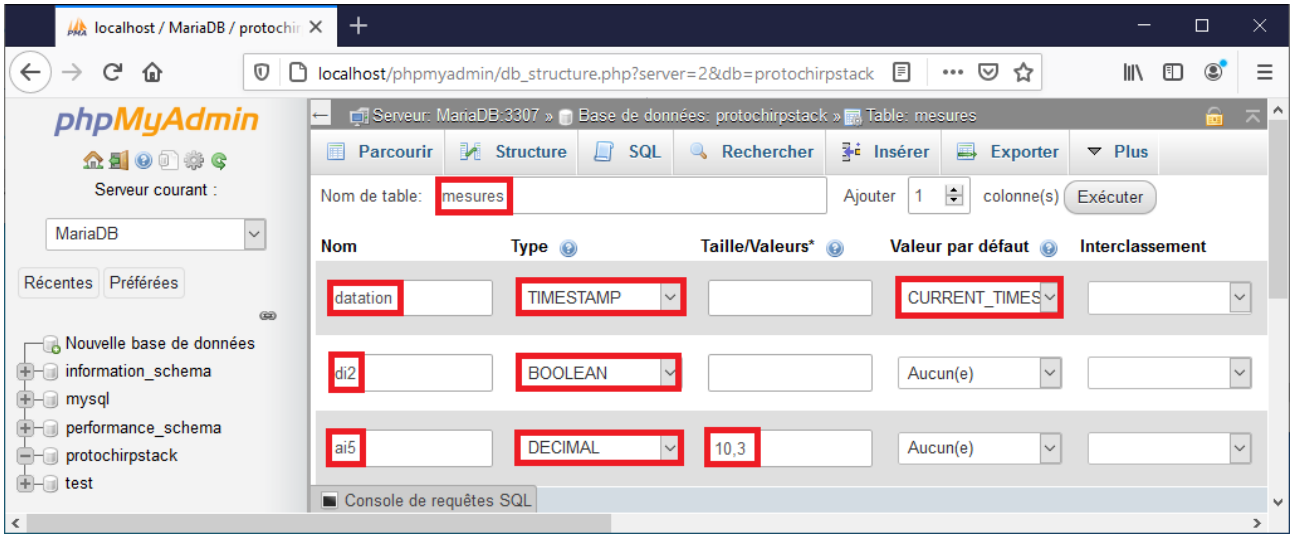
Créer une Base de Données "protochirpstack".
Dans cette BdD, créer une table "mesures".



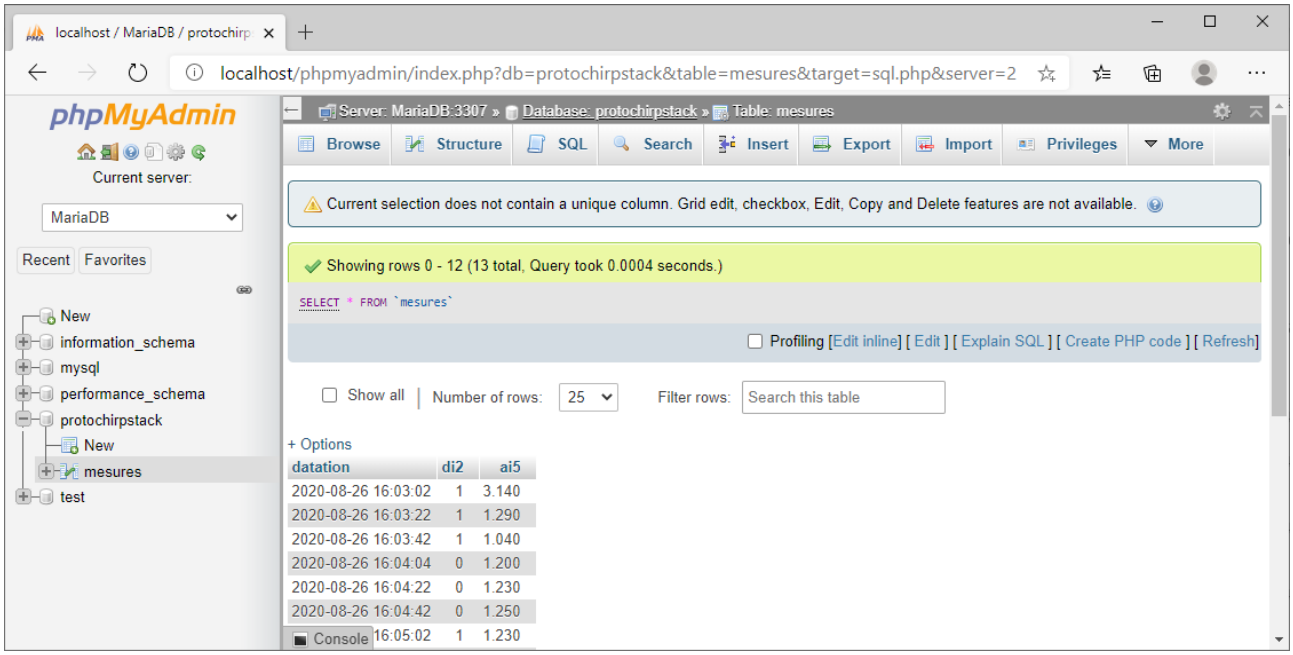
Créer 3 champs dans cette table :

- datation : timestamp
- di2 : booléen
- ai5 : décimal

Cf copie d'écran ci après.

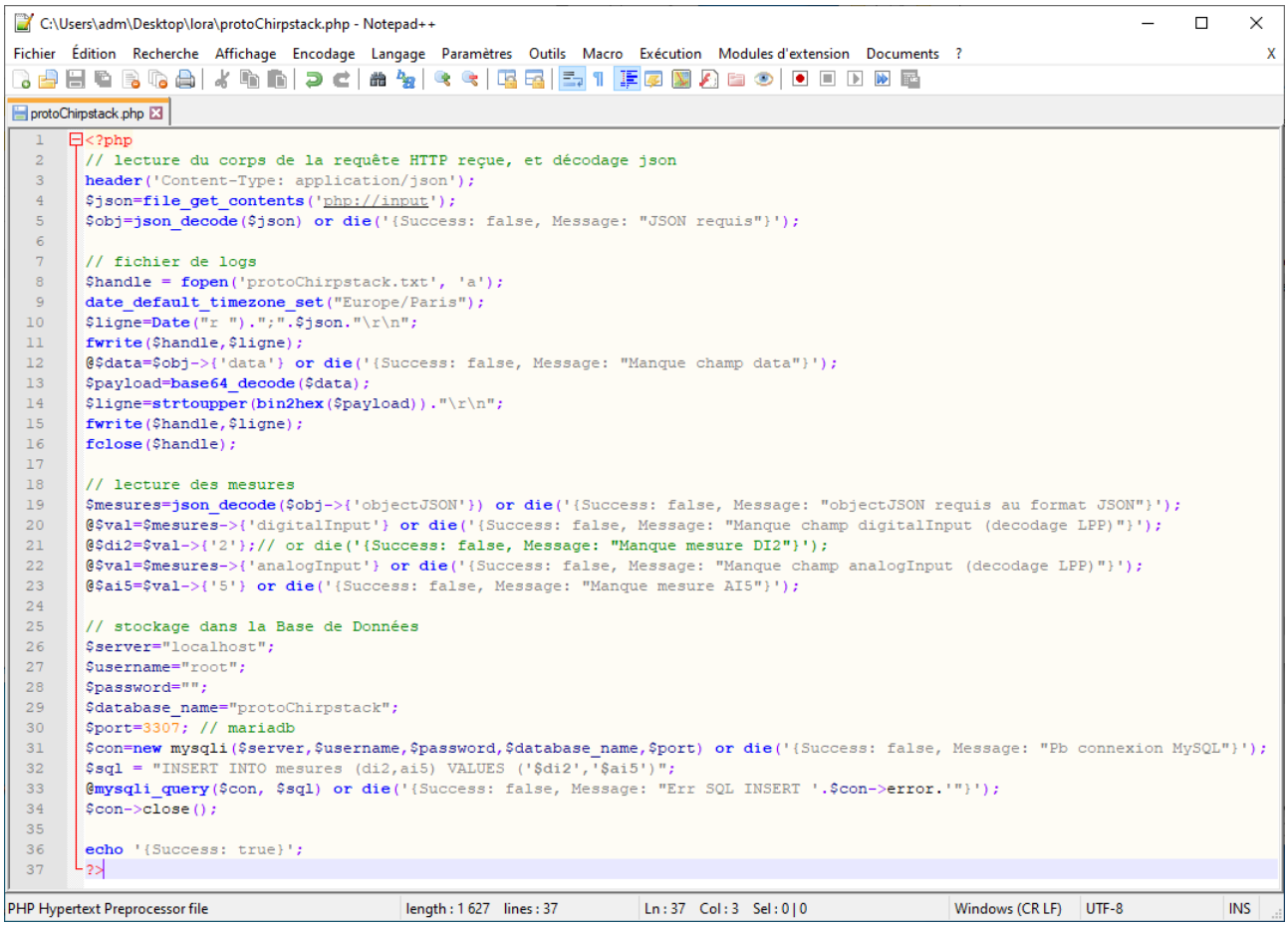


Lorsque le script PHP sera mis en place, la table sera remplie au fur et à mesure des messages reçus :



5.4 Script PHP

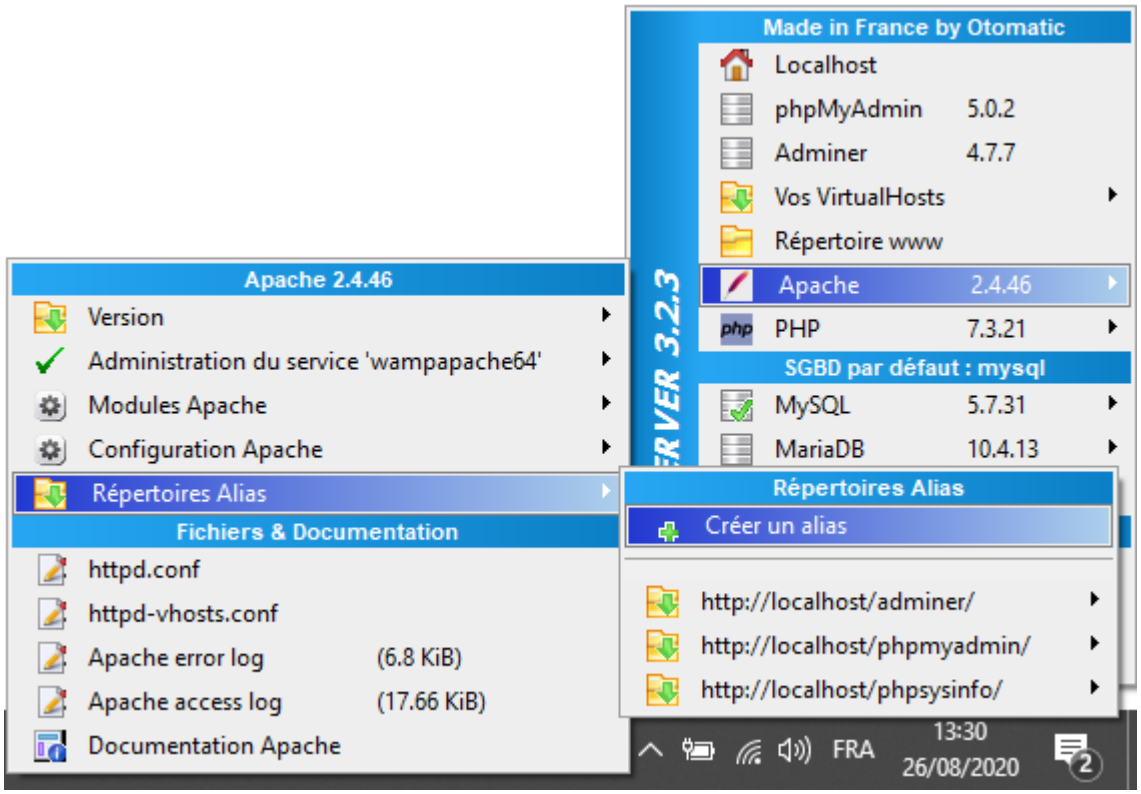
Placer le script suivant dans un dossier. Exemple : dossier "lora" sur le bureau
NB : le nom du script (`protoChirpStack.php`) doit être le même que dans l'URL configurée sur ChirpStack : `http://(votre IP)/lora/protoChirpStack.php`



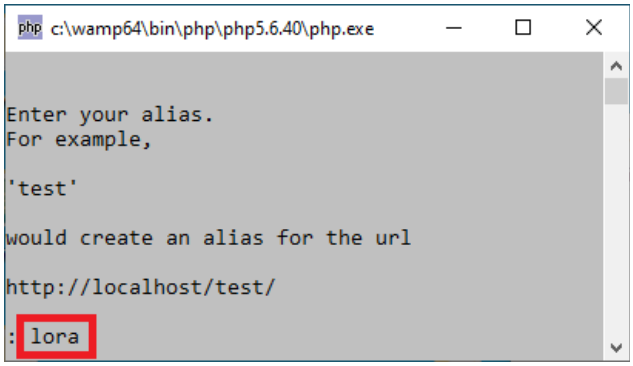

```php
<?php
// lecture du corps de la requête HTTP reçue, et décodage json
header('Content-Type: application/json');
$json=file_get_contents('php://input');
$obj=json_decode($json) or die('{Success: false, Message: "JSON requis"}');

// fichier de logs
$handle = fopen('protoChirpstack.txt', 'a');
date_default_timezone_set("Europe/Paris");
$ligne=Date("r ").";".$json."\r\n";
fwrite($handle,$ligne);
@$data=$obj->{'data'} or die('{Success: false, Message: "Manque champ data"}');
$payload=base64_decode($data);
$ligne=strtoupper(bin2hex($payload))."\r\n";
fwrite($handle,$ligne);
fclose($handle);

// lecture des mesures
$mesures=json_decode($obj->{'objectJSON'}) or die('{Success: false, Message: "objectJSON requis au format JSON"}');
@$val=$mesures->{'digitalInput'} or die('{Success: false, Message: "Manque champ digitalInput (decodage LPP)"}');
@$di2=$val->{'2'};// or die('{Success: false, Message: "Manque mesure DI2"}');
@$val=$mesures->{'analogInput'} or die('{Success: false, Message: "Manque champ analogInput (decodage LPP)"}');
@$ai5=$val->{'5'} or die('{Success: false, Message: "Manque mesure AI5"}');

// stockage dans la Base de Données
$server="localhost";
$username="root";
$password="";
$database_name="protoChirpstack";
$port=3307; // mariadb
$con=new mysqli($server,$username,$password,$database_name,$port) or die('{Success: false, Message: "Pb connexion MySQL"}');
$sql = "INSERT INTO mesures (di2,ai5) VALUES ('$di2','$ai5')";
@mysqli_query($con, $sql) or die('{Success: false, Message: "Err SQL INSERT '.$con->error.'"}');
$con->close();

echo '{Success: true}';
?>
```

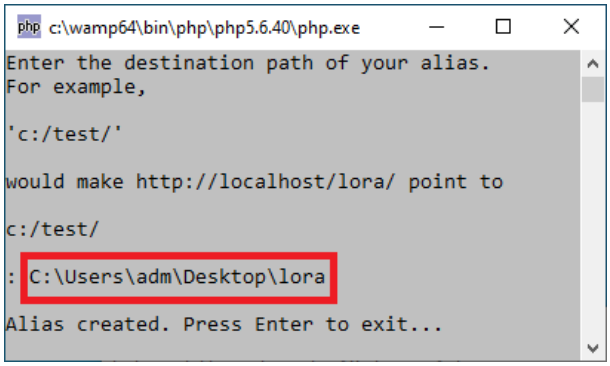
5.5 Création d'un alias pour le script PHP

Via l'icône de WAMP, choisir "créer un alias"

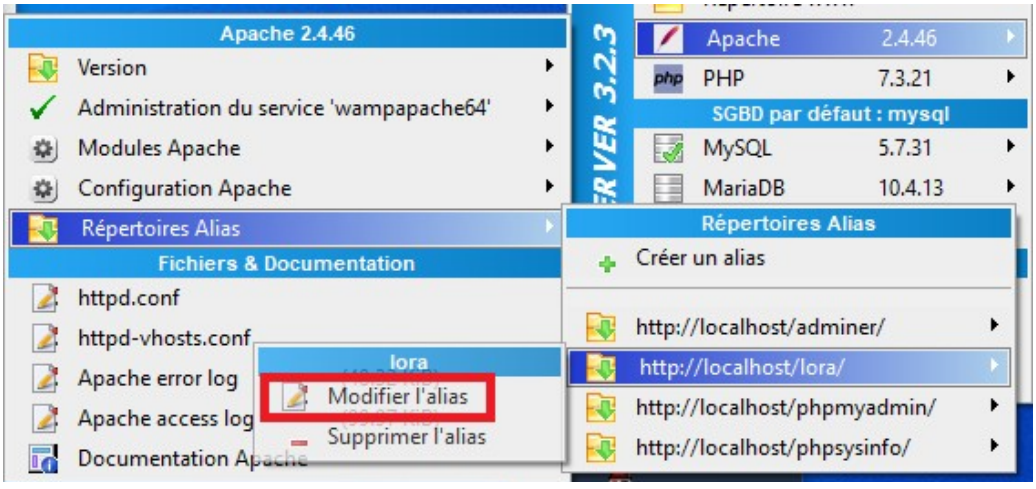


Choisir l'alias "lora". Celui-ci doit être le même que dans l'URL configurée sur ChirpStack :
`http://(votre IP)/lora/protoChirpStack.php`
A la deuxième question, copier/coller le nom du dossier où le script PHP est stocké.

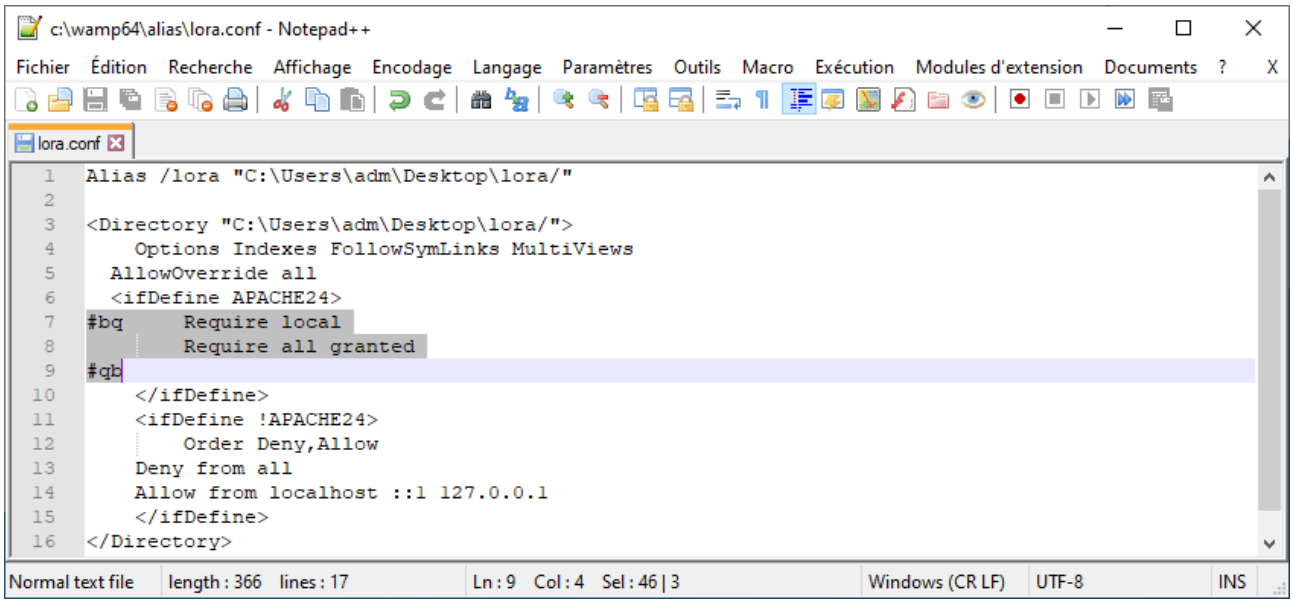




Choisir de modifier le fichier d'alias nouvellement créé

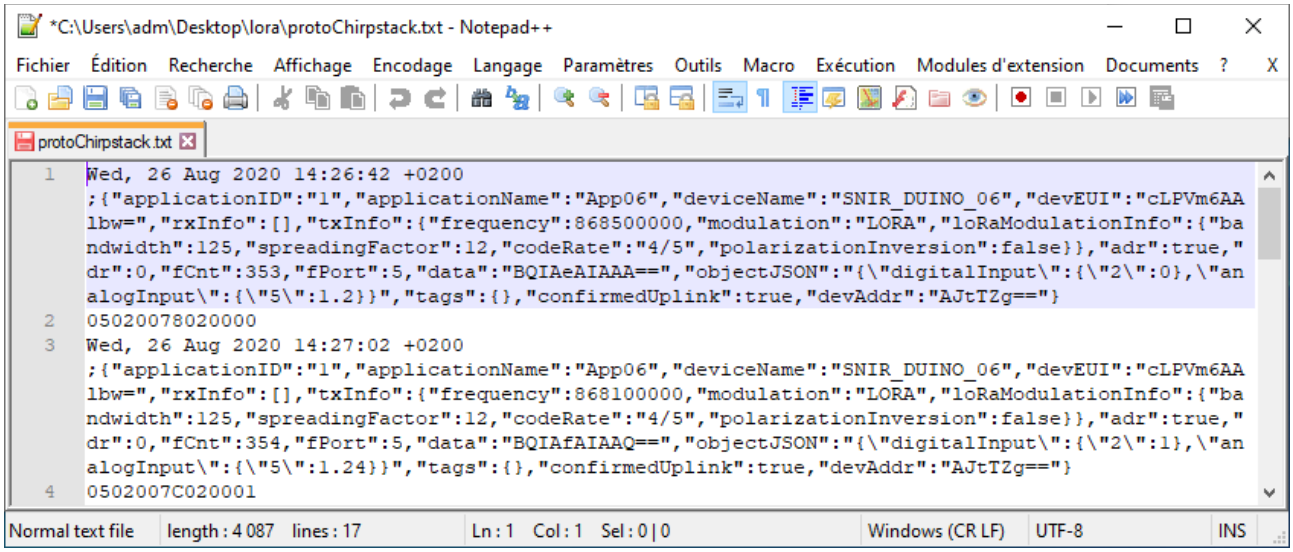


Dans le fichier lora.con qui vient d'être créé, remplacer "Require local" par "Require all granted"
Cela permet à votre serveur Web d'accepter des requêtes venant de toute autre machine.
Cela autorise le network server ChirpStack à envoyer des requêtes HTTP vers votre serveur Web.



Redémarrer Apache. Vérifier que la Base de Données, ainsi que le fichier de logs se remplissent au fur et à mesure des messages montant

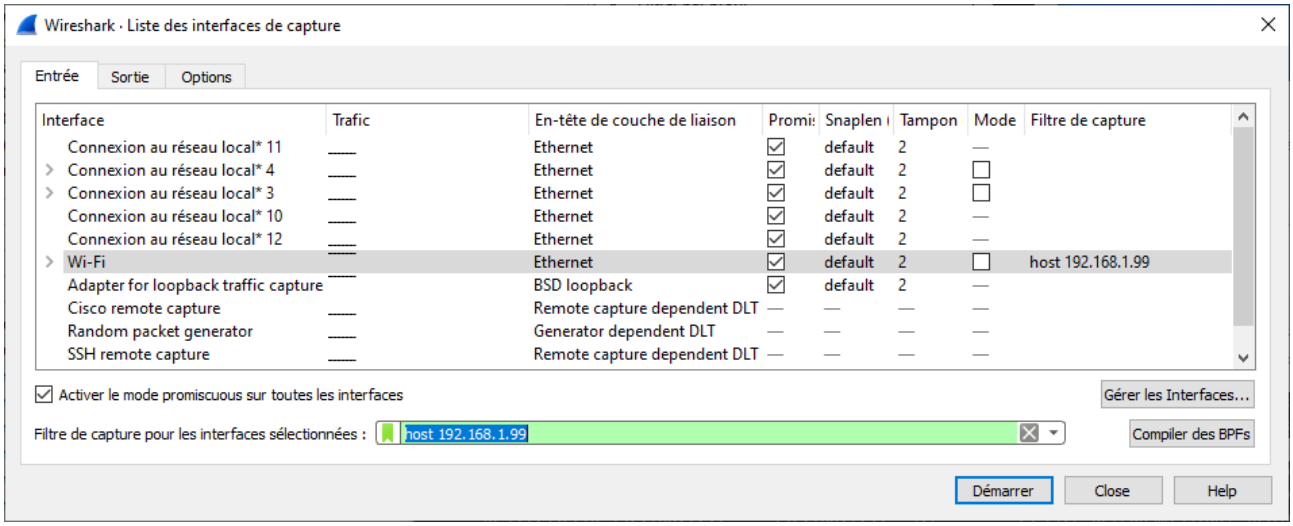
En cas de dysfonctionnement, jeter un œil au chapitre suivant



6 En cas de dysfonctionnement du script PHP

6.1 Wireshark

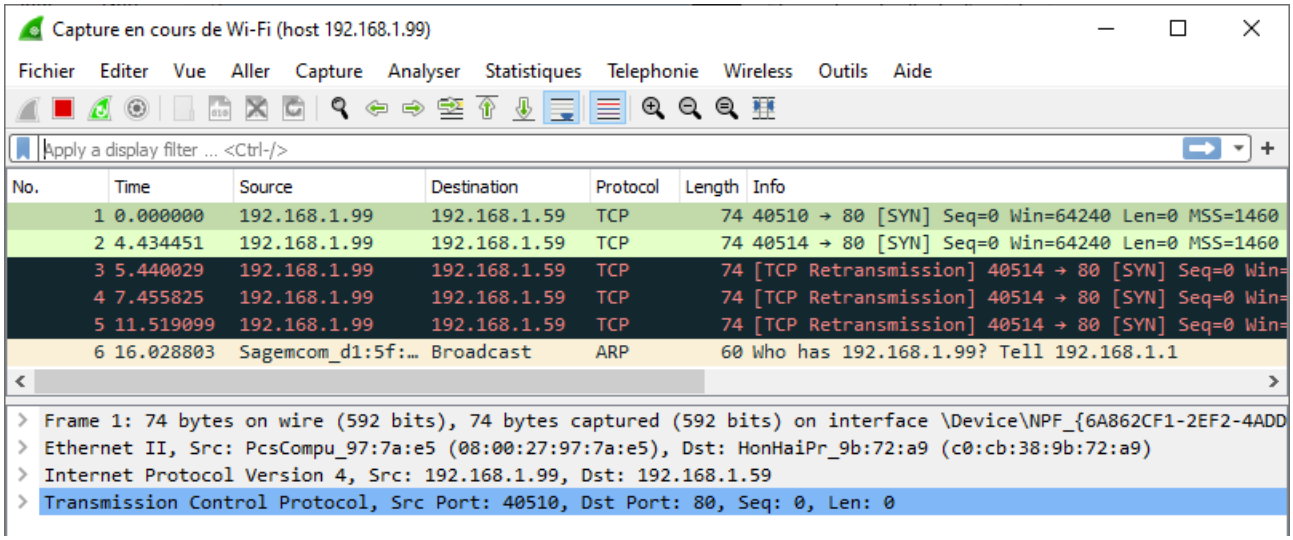
S'il n'est pas déjà installé, installer Wireshark.
En démarrant la capture, configurer un filtre de manière à limiter la capture aux échanges entre le network server ChirpStack et votre serveur applicatif. Utiliser pour cela la directive "host"



6.2 Problème de pare-feu

Dans la capture suivante, le network server ChirpStack essaye d'établir une connexion TCP sur le port 80 pour se connecter à votre serveur web Apache. Il essaye à plusieurs reprises sans succès.
Causes possibles :

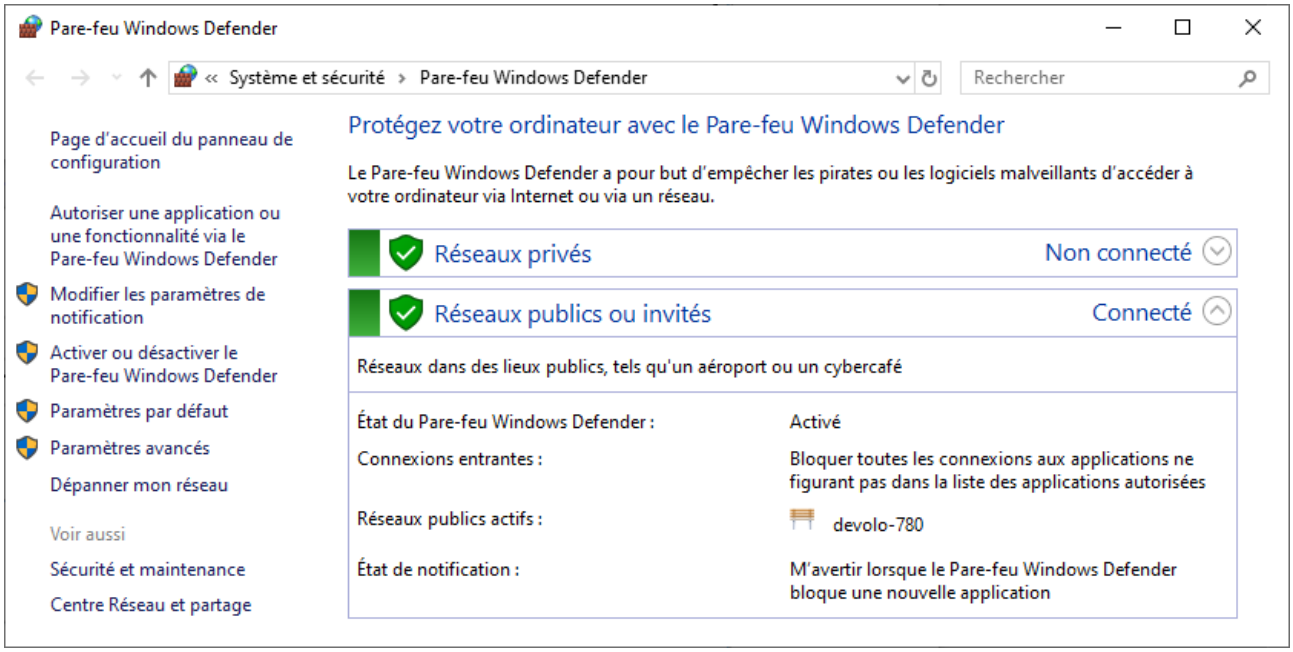
- Votre serveur web Apache est arrêté
- Votre pare-feu empêche les connexions entrantes sur votre PC



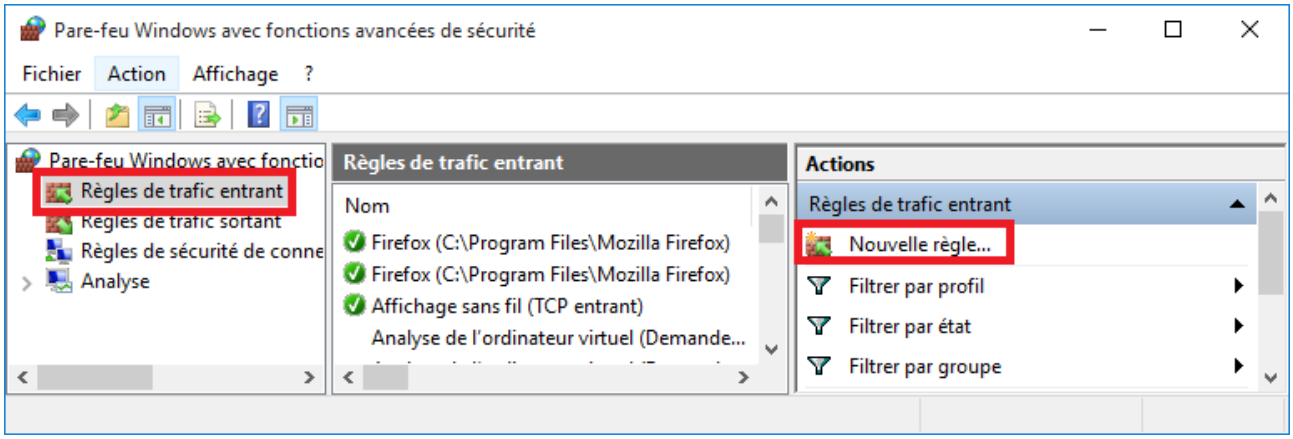
Pour résoudre le problème :

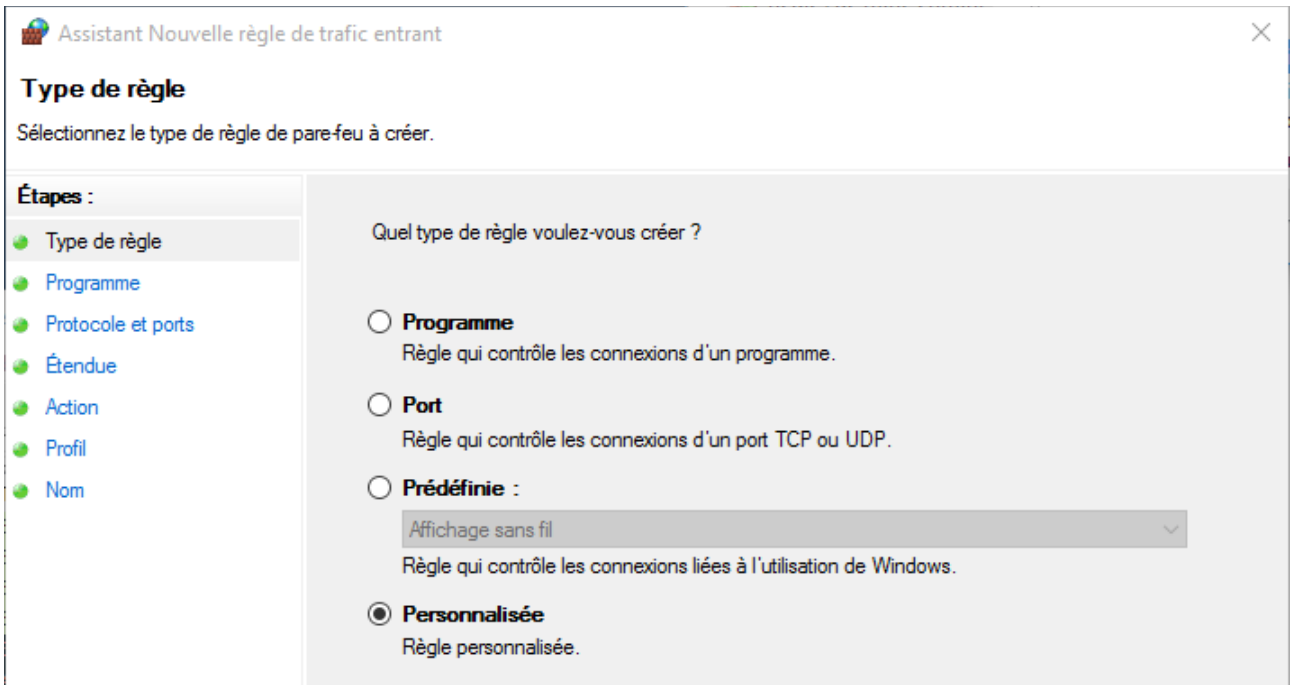
1. désactiver temporairement le pare-feu
2. vérifier que la connexion se fait bien sur le port 80
3. réactiver le pare-feu
4. créer une règle spécifique sur le pare-feu

Pour créer une règle, ouvrir les paramètres du pare-feu, et aller dans "paramètres avancés"

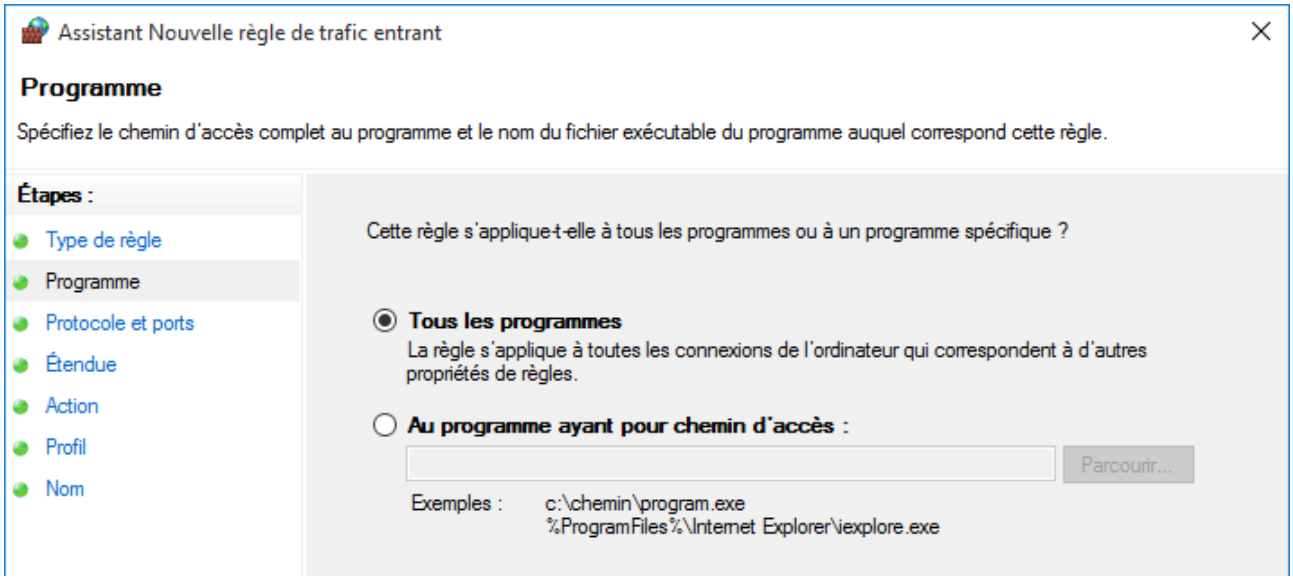


Créer une règle pour le trafic entrant. Choisir une règle de type "personnalisée"

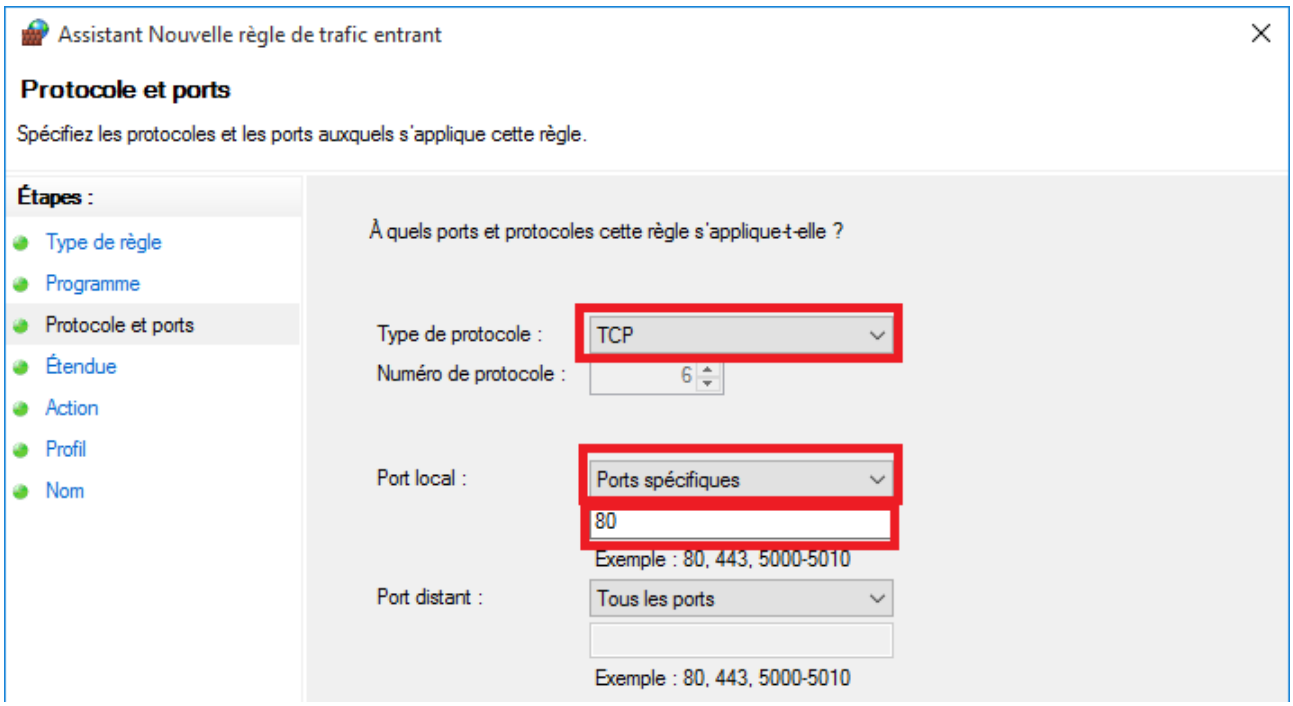




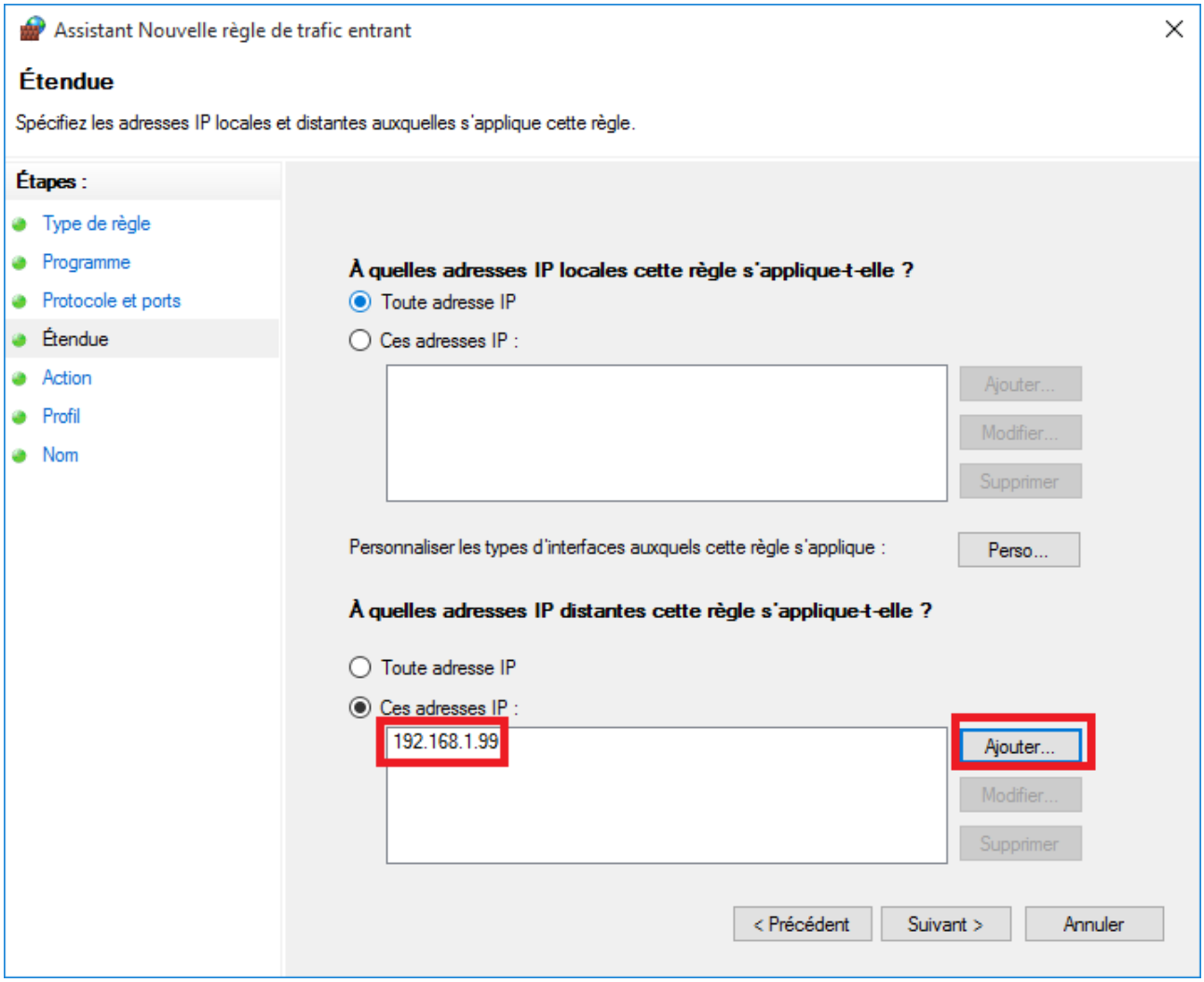
Choisir "tous les programmes"

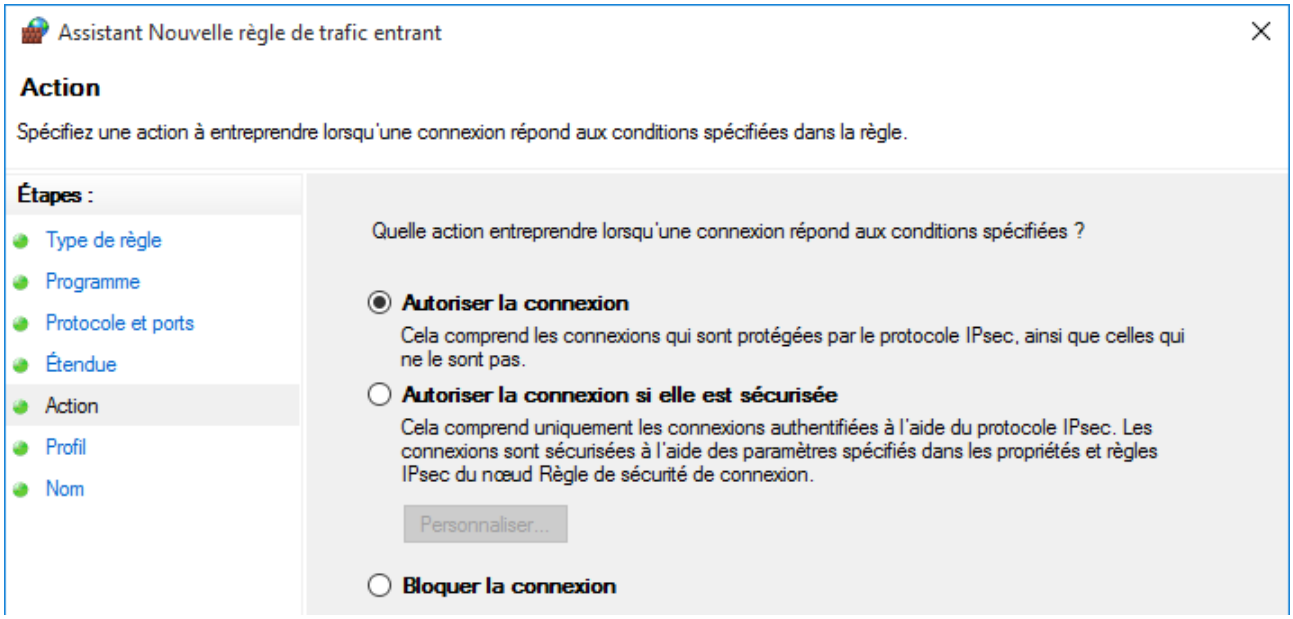


Port TCP local 80

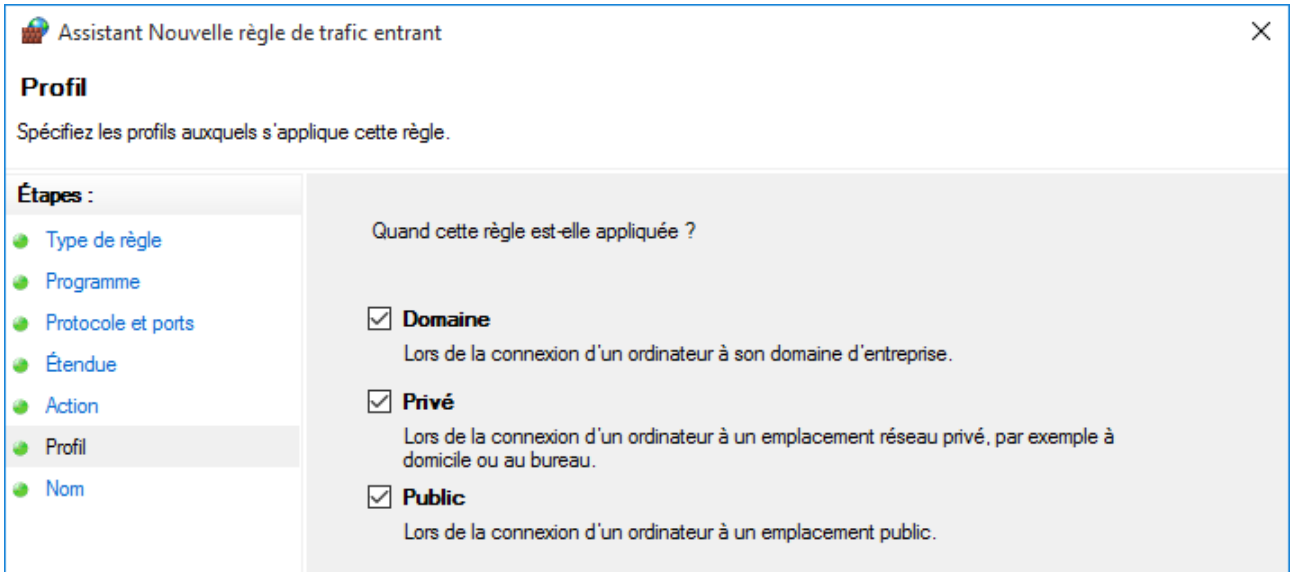


Autoriser l'adresse IP du network server

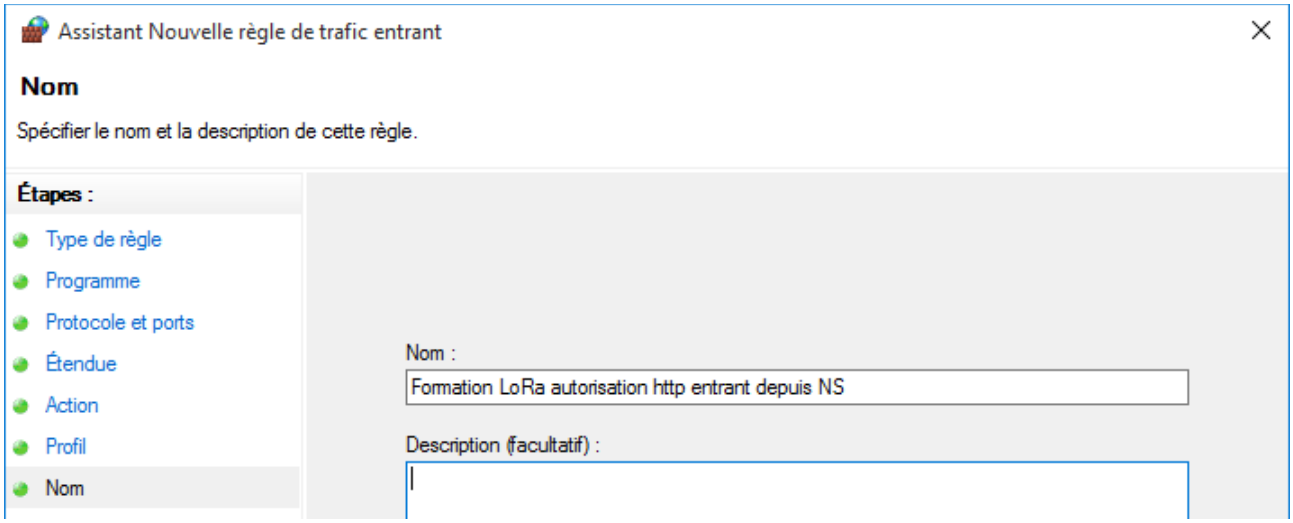




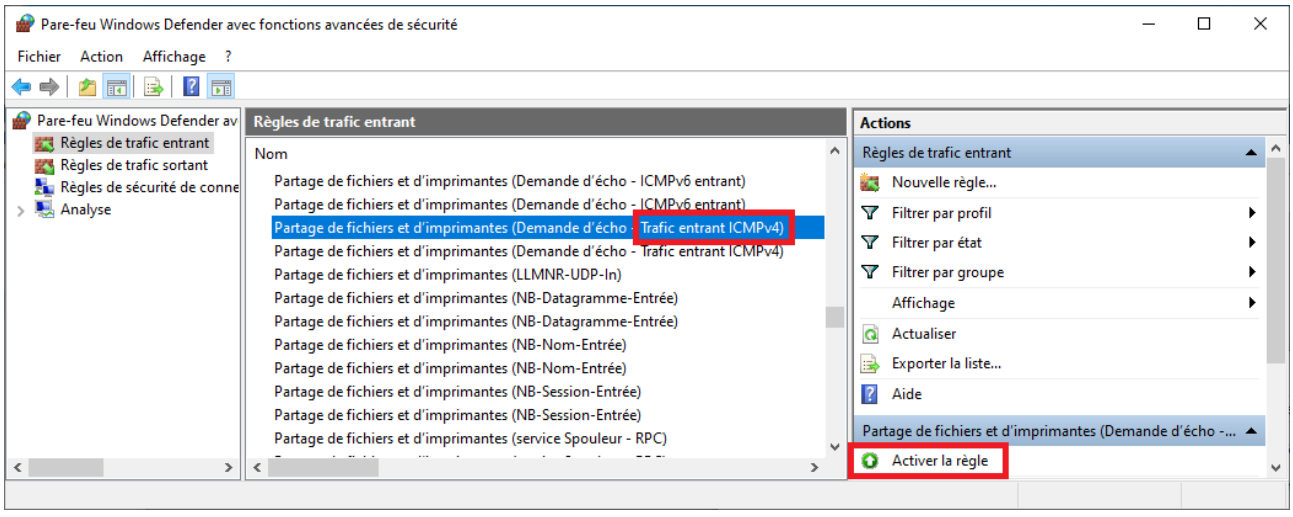
Appliquer la règle pour tous les profils réseaux



Saisir un nom pour la règle

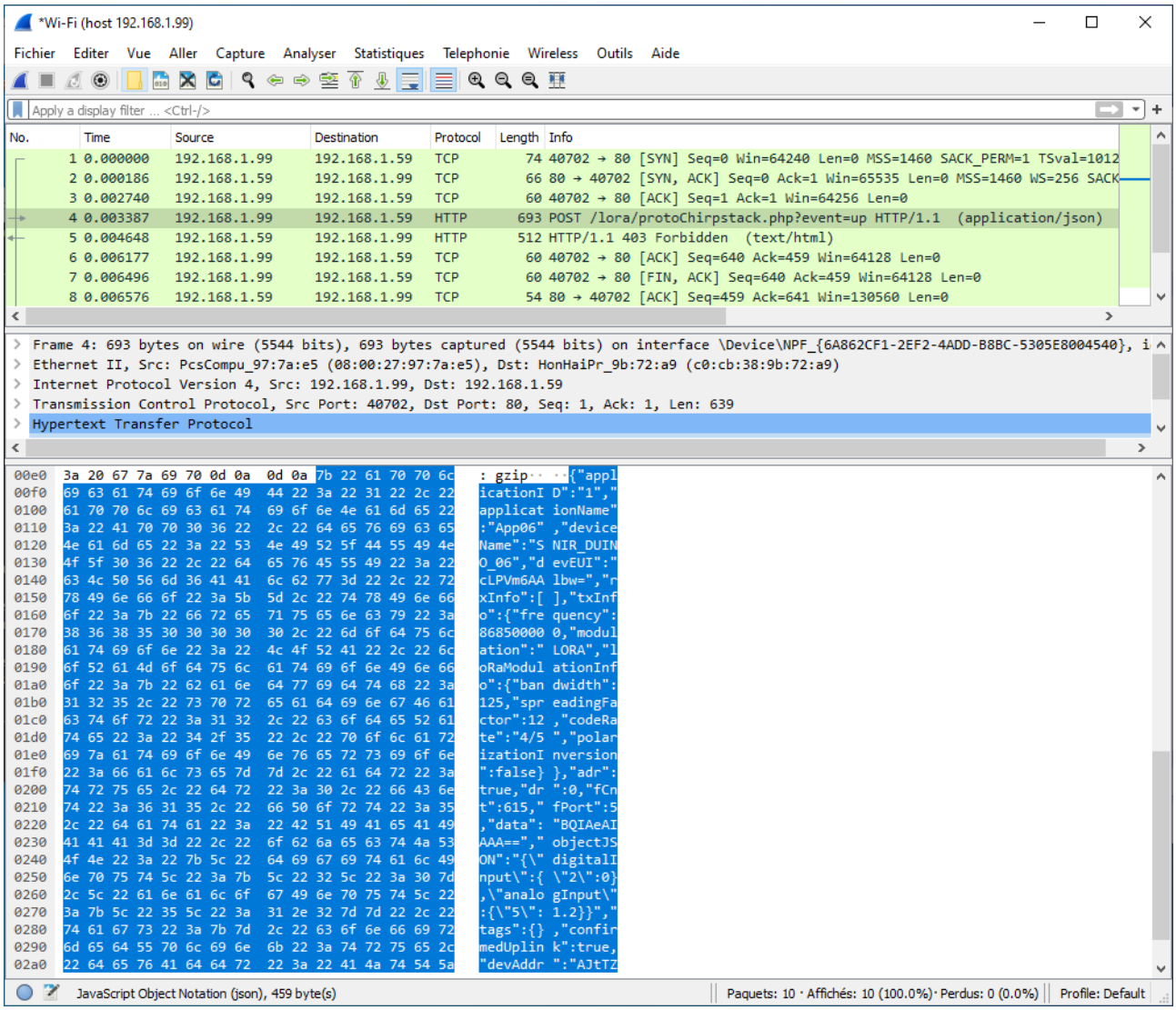


Vérifier que l'application de la règle résout le problème de connexion du network server à votre serveur applicatif
A propos de pare-feu, il est parfois intéressant d'activer la règle aui autorise les pings entrants :



6.3 Problème de réglage Apache

Dans la capture suivante, on voit que la connexion TCP se passe bien (trames 1 à 3) et que le network server "pousse" bien les données vers le serveur applicatif (trame 4)
Cependant, on voit une erreur "403 forbidden" en trame 5. Il s'agit d'un code d'erreur renvoyé par le serveur web Apache.
Il s'agit vraisemblablement ici d'une erreur de configuration de l'Alias (cf p 38 chapitre 5.5)



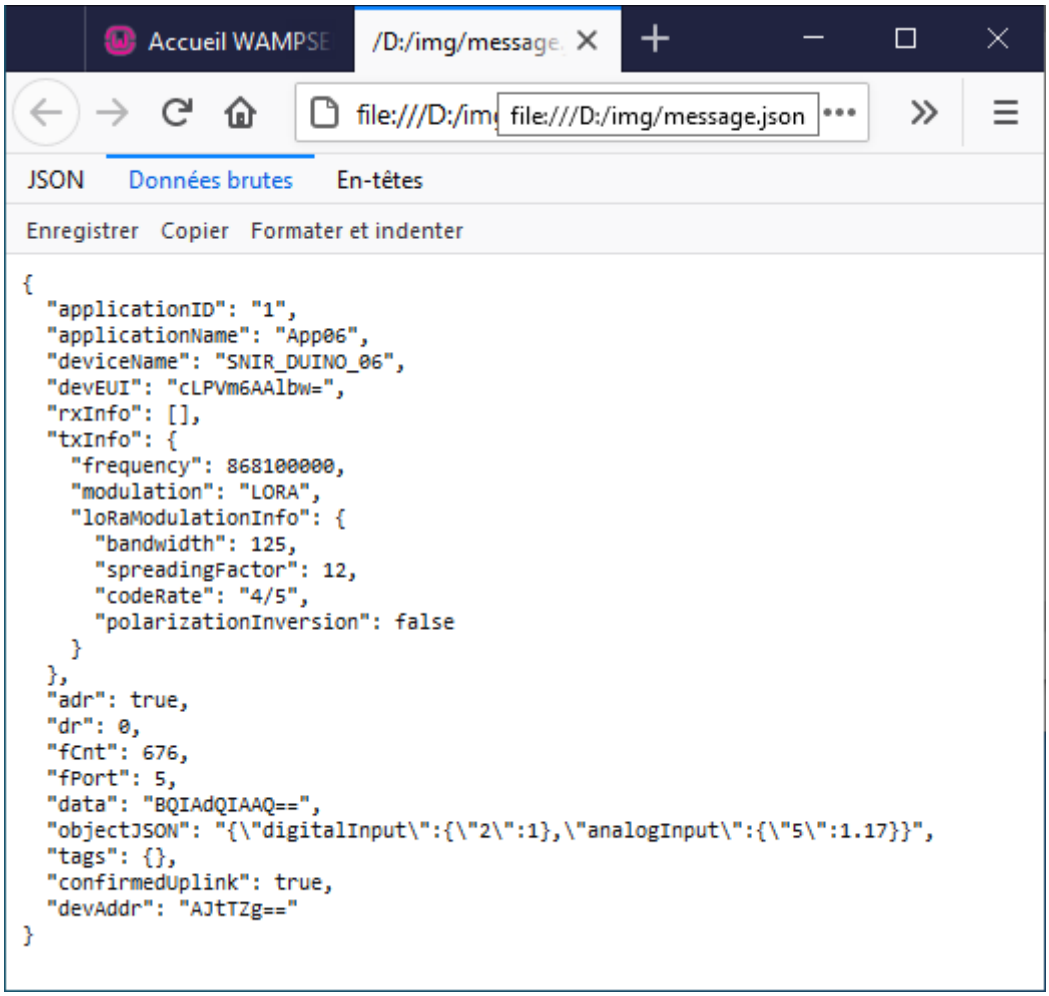
6.4 Si rien ne fonctionne

Il est possible de tester le script PHP en simulant avec la commande curl, le push HTTP du network server ChirpStack vers votre serveur applicatif.

```
curl -d @message.json -H "Content-Type: application/json" http://192.168.1.115/lora/protoChirpstack.php
```

En cas d'erreur du script php, les messages d'erreur affichés sont alors ici visibles.

fichier message.json contenant un objectJSON (message décodé LPP par ChirpStack)

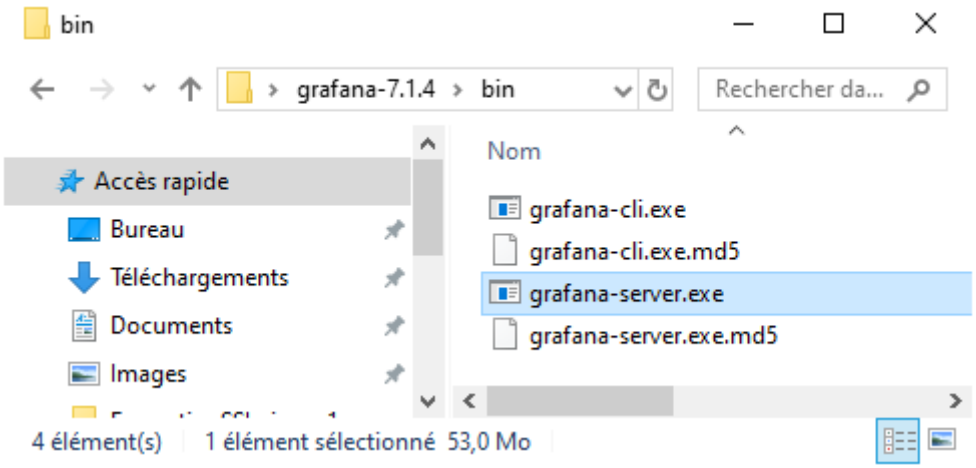


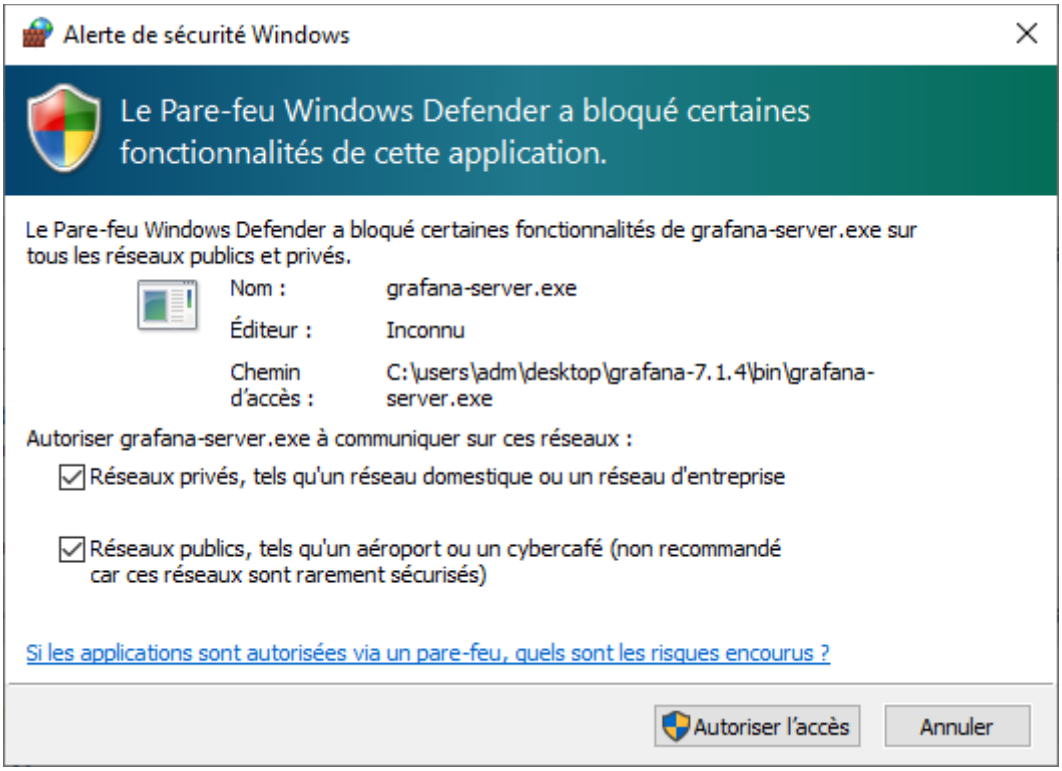
7 Pour aller plus loin : exploitation de la BdD

Il est tout à fait possible de développer une application PHP pour visualiser le contenu de la BdD.
Il est également possible d'utiliser des solution open source existantes, ce que nous allons faire avec Grafana.

7.1 Installation de Grafana

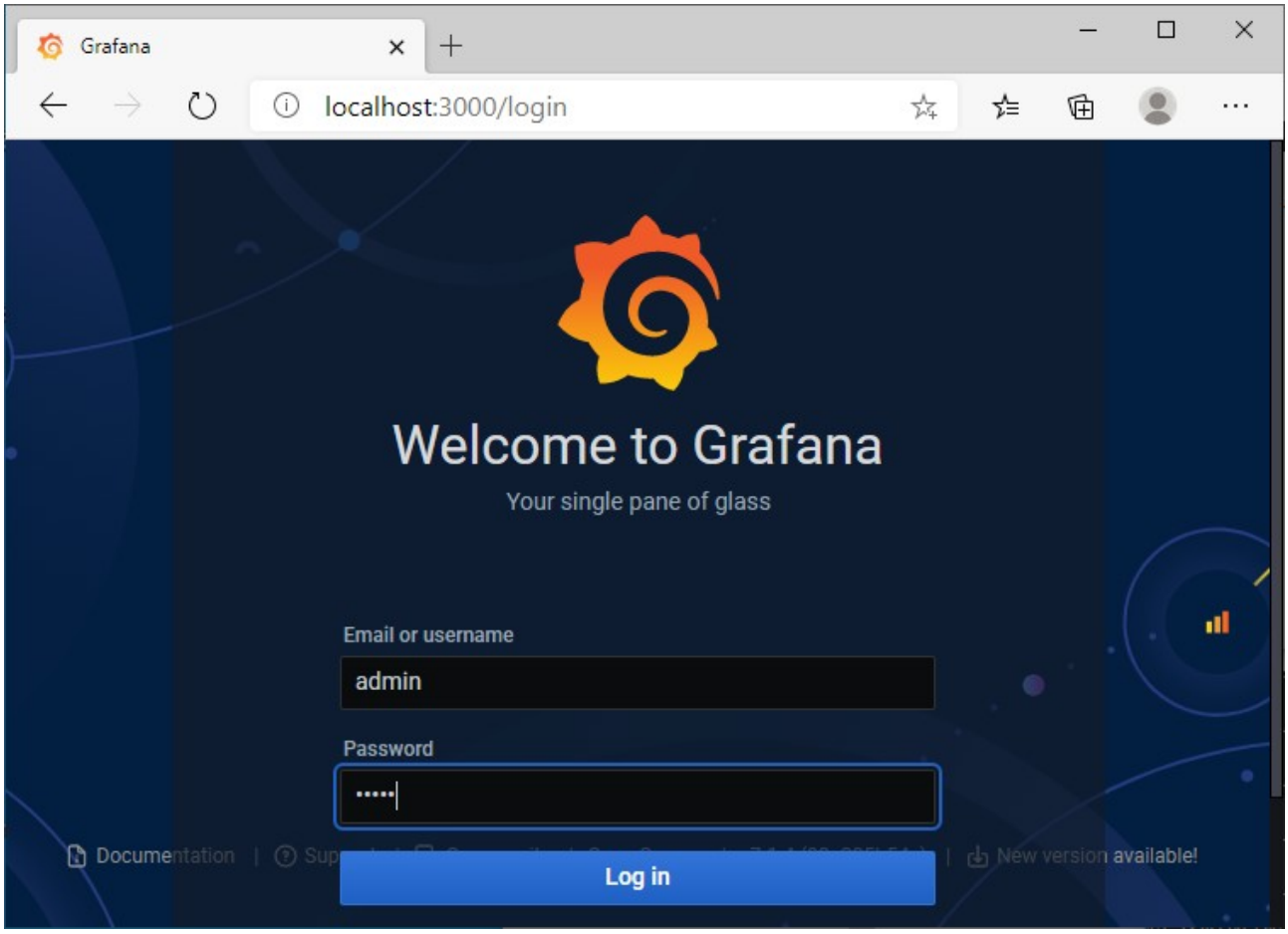
Décompresser le fichier Zip et démarrer Grafana-server.exe. Autoriser l'application à accéder au réseaux





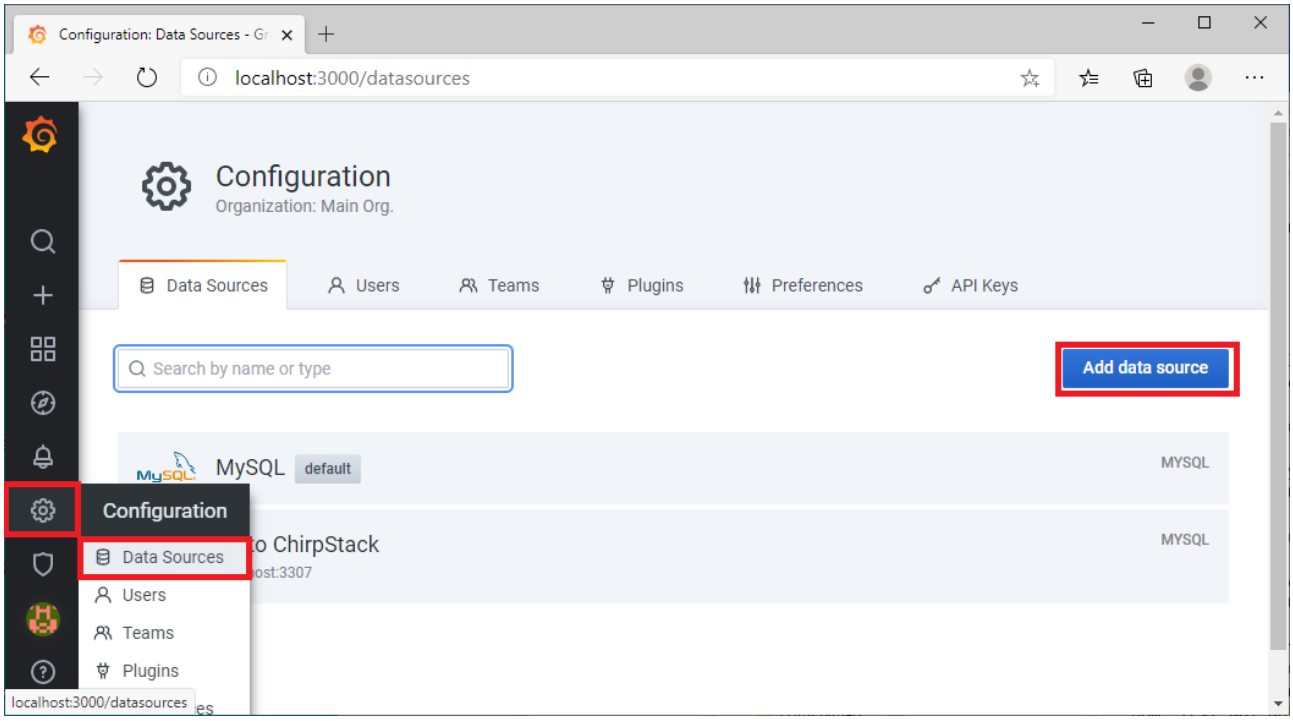
Pour se connecter à Grafana :
user : admin
password : admin

Une fois connecté, Grafana propose de changer le mot de passe, mais il est possible de cliquer sur "skip" pour ignorer

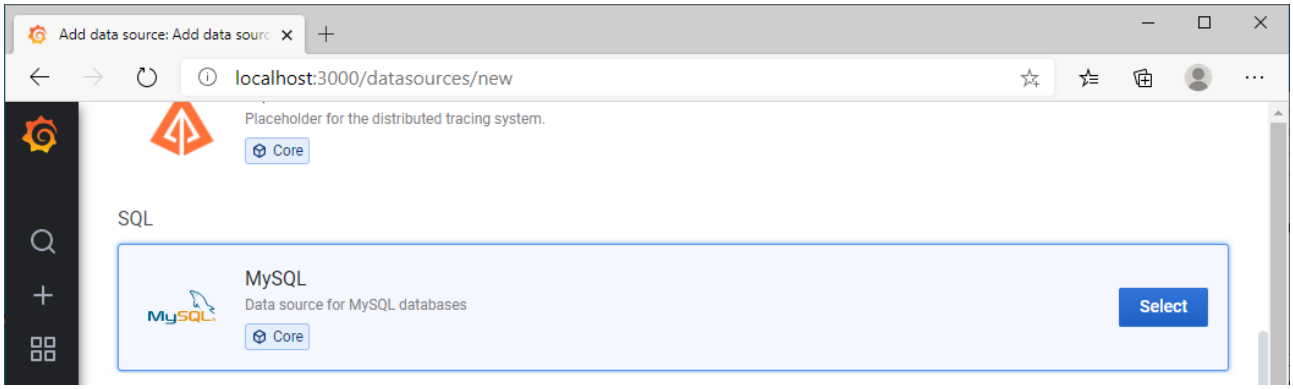


7.2 Création d'une source de données

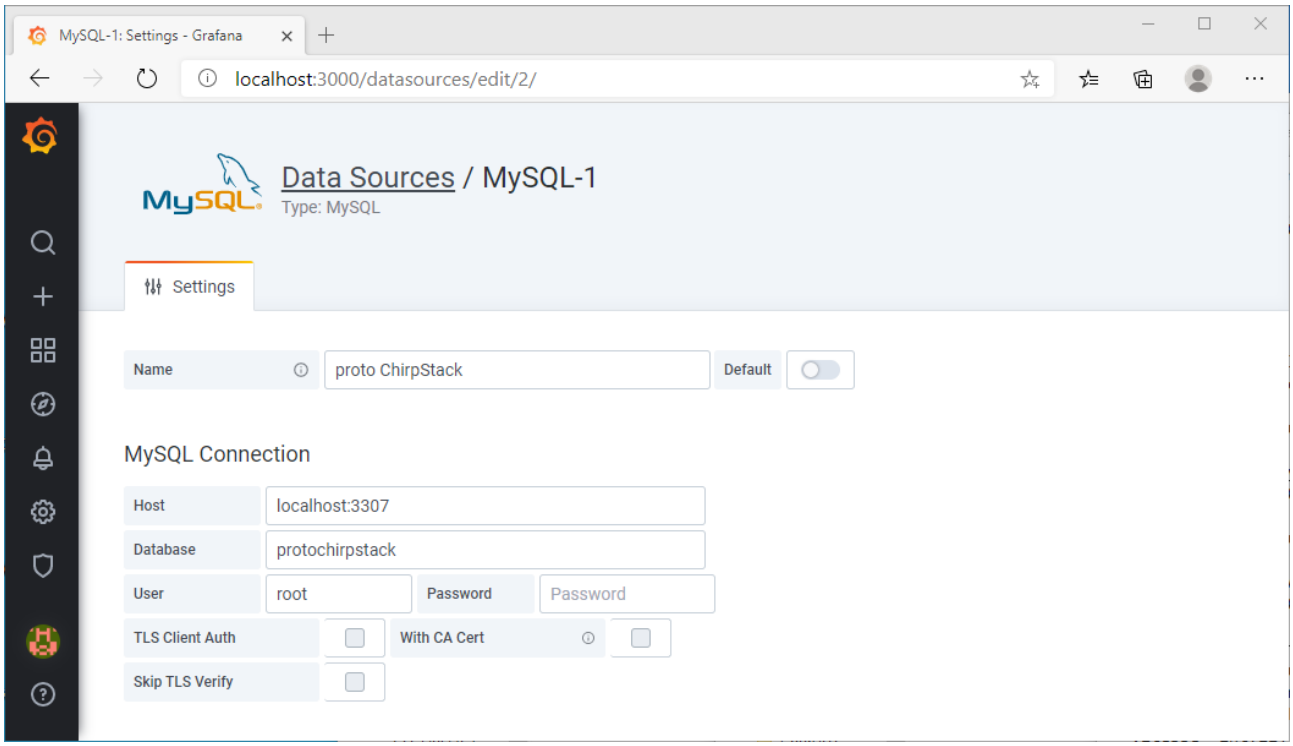
Choisir le menu "data sources"

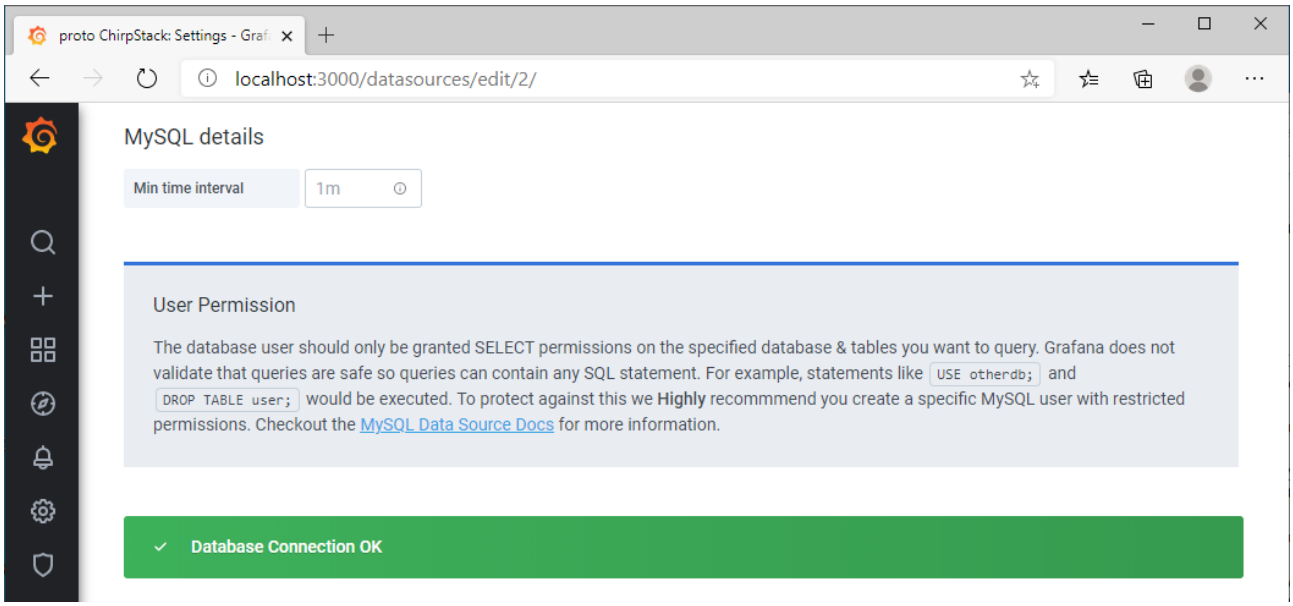


Choisir "mysql" pour un SGBD MySQL ou MariaDB

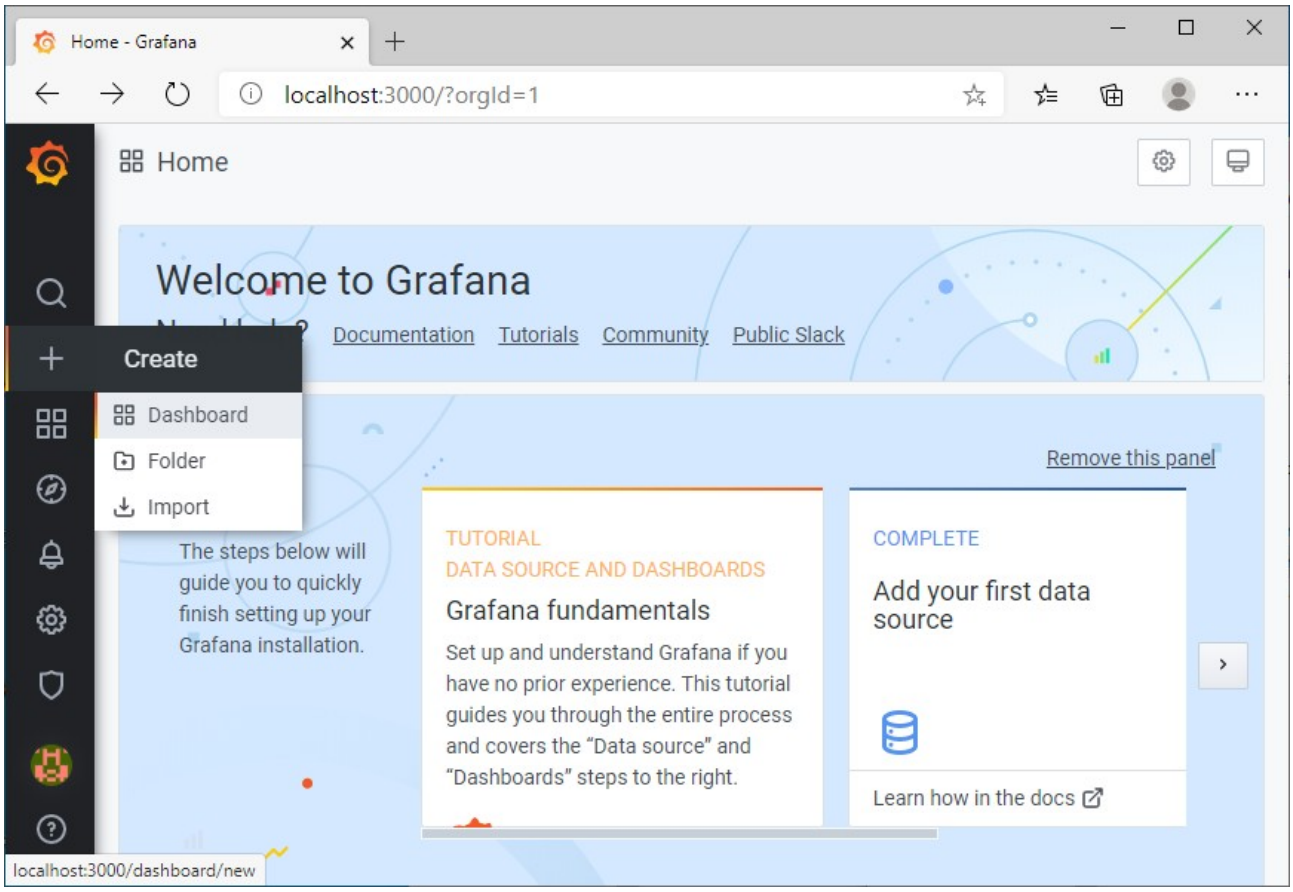


Régler les paramètres d'accès à la BdD

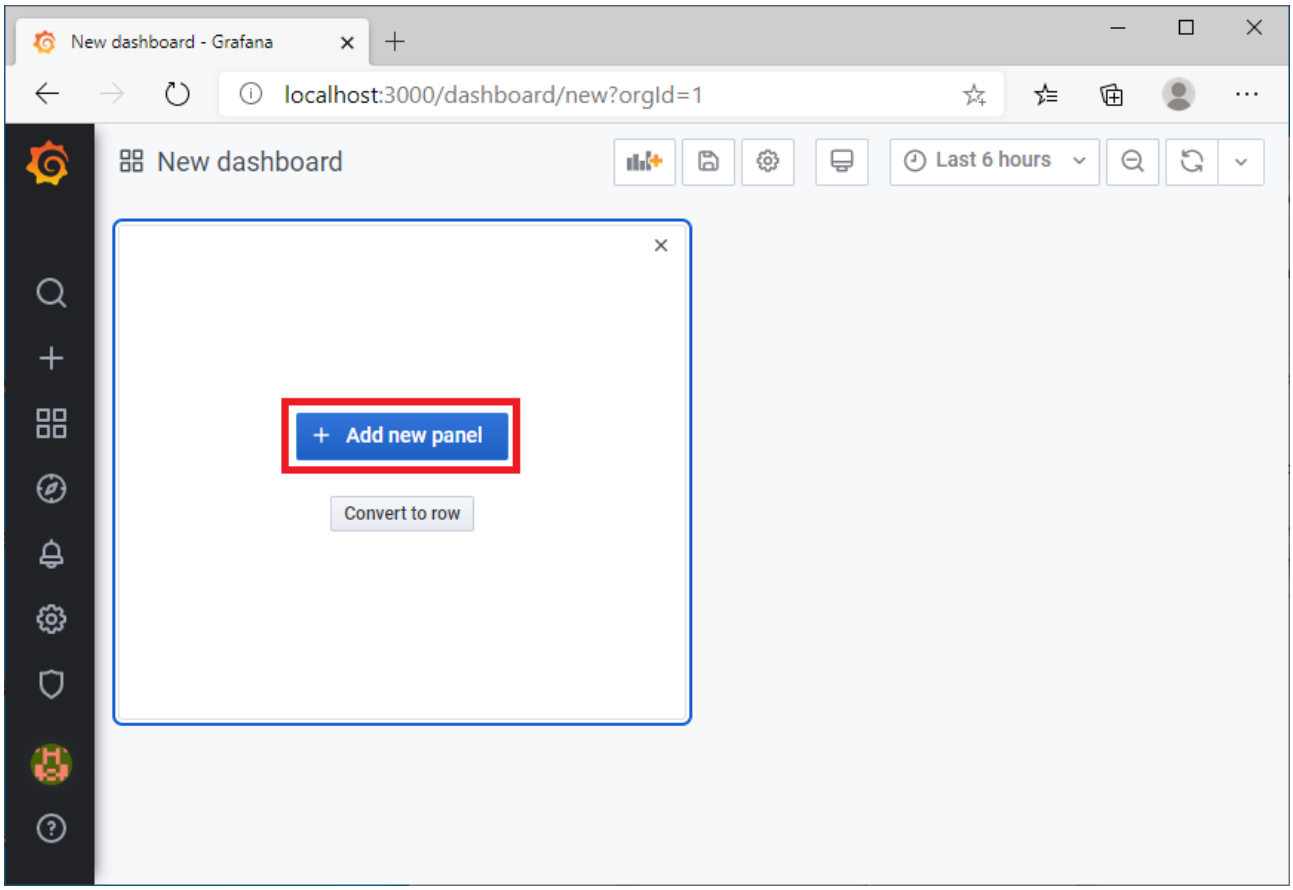




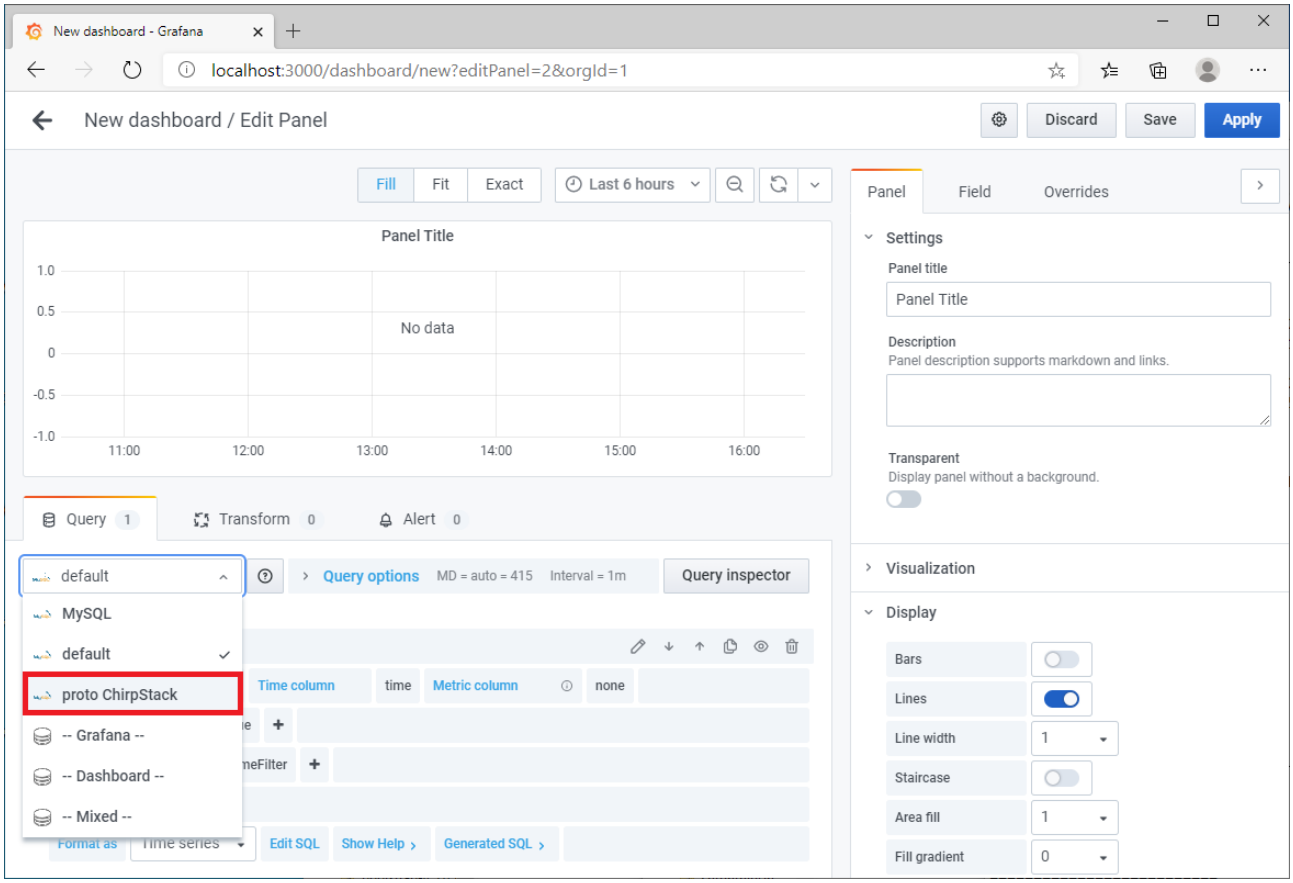
7.3 Création d'un tableau de bord



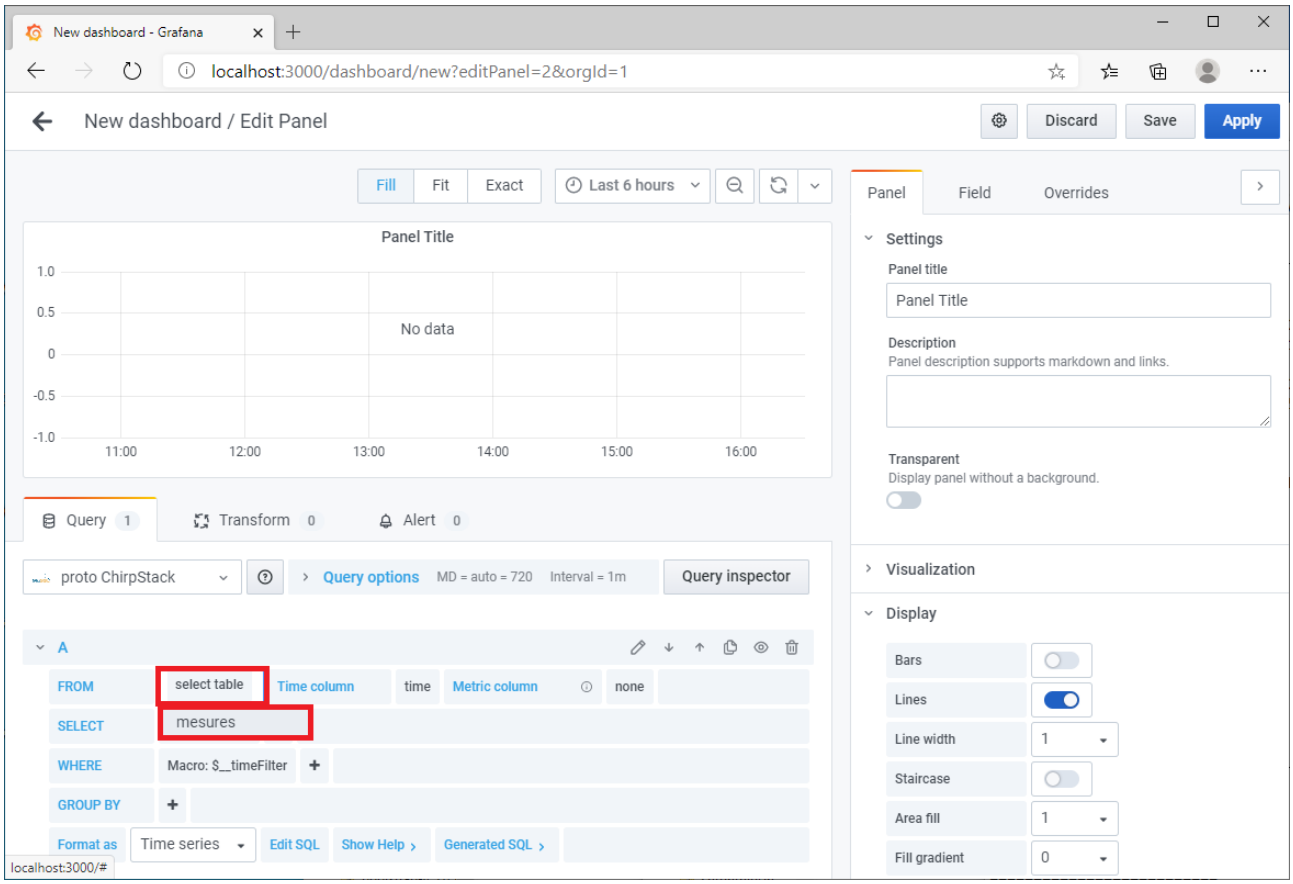
Créer un panneau



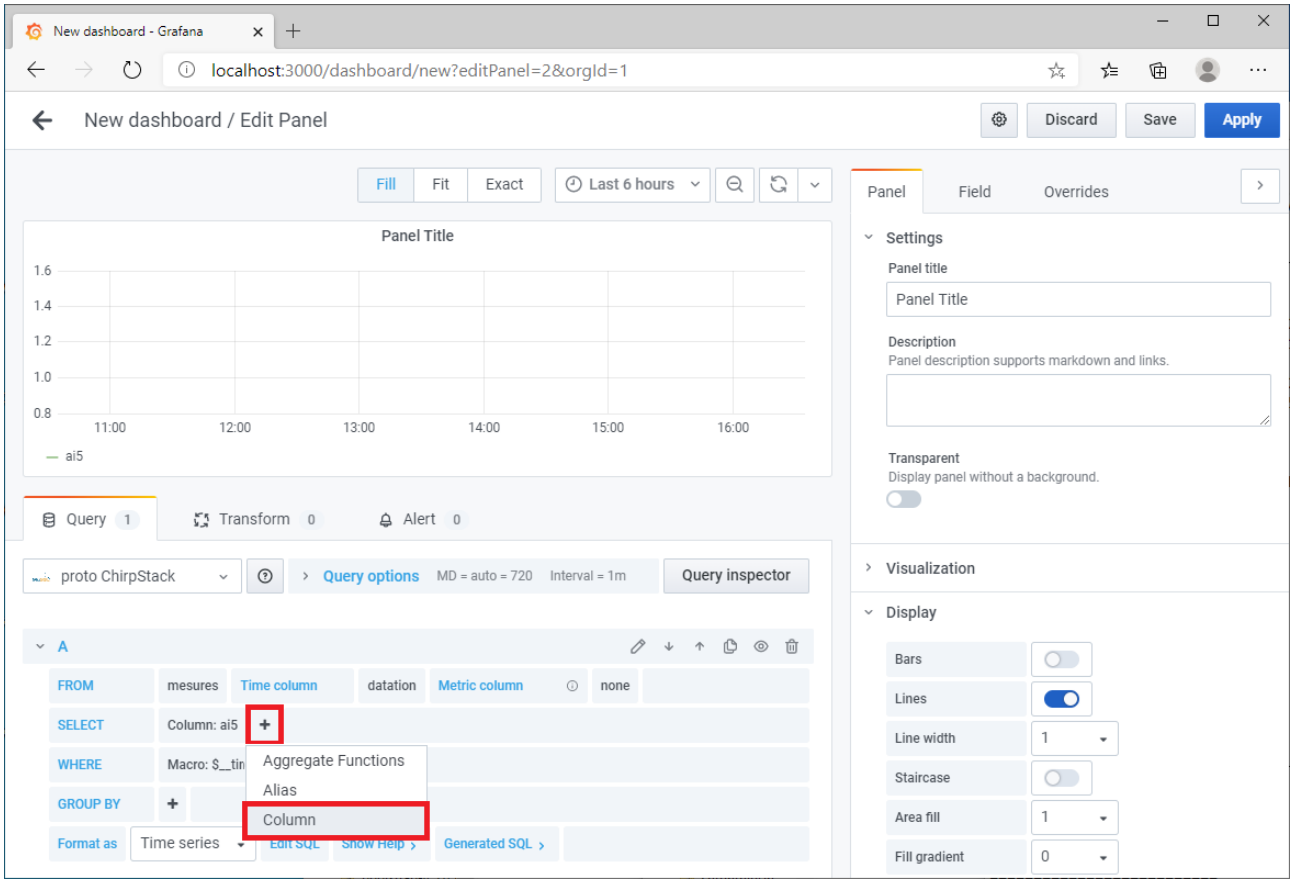
Choisir la source de données préalablement créée

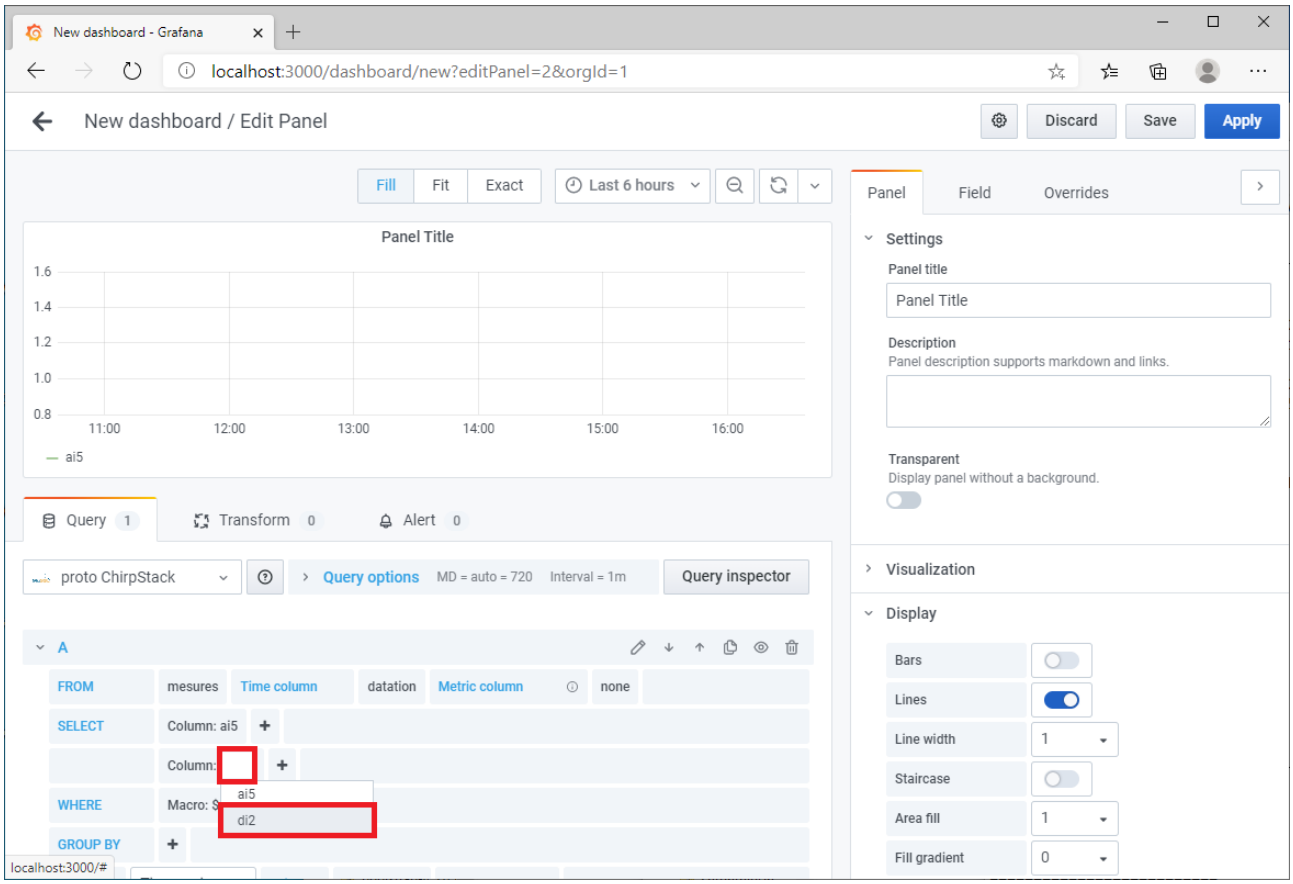


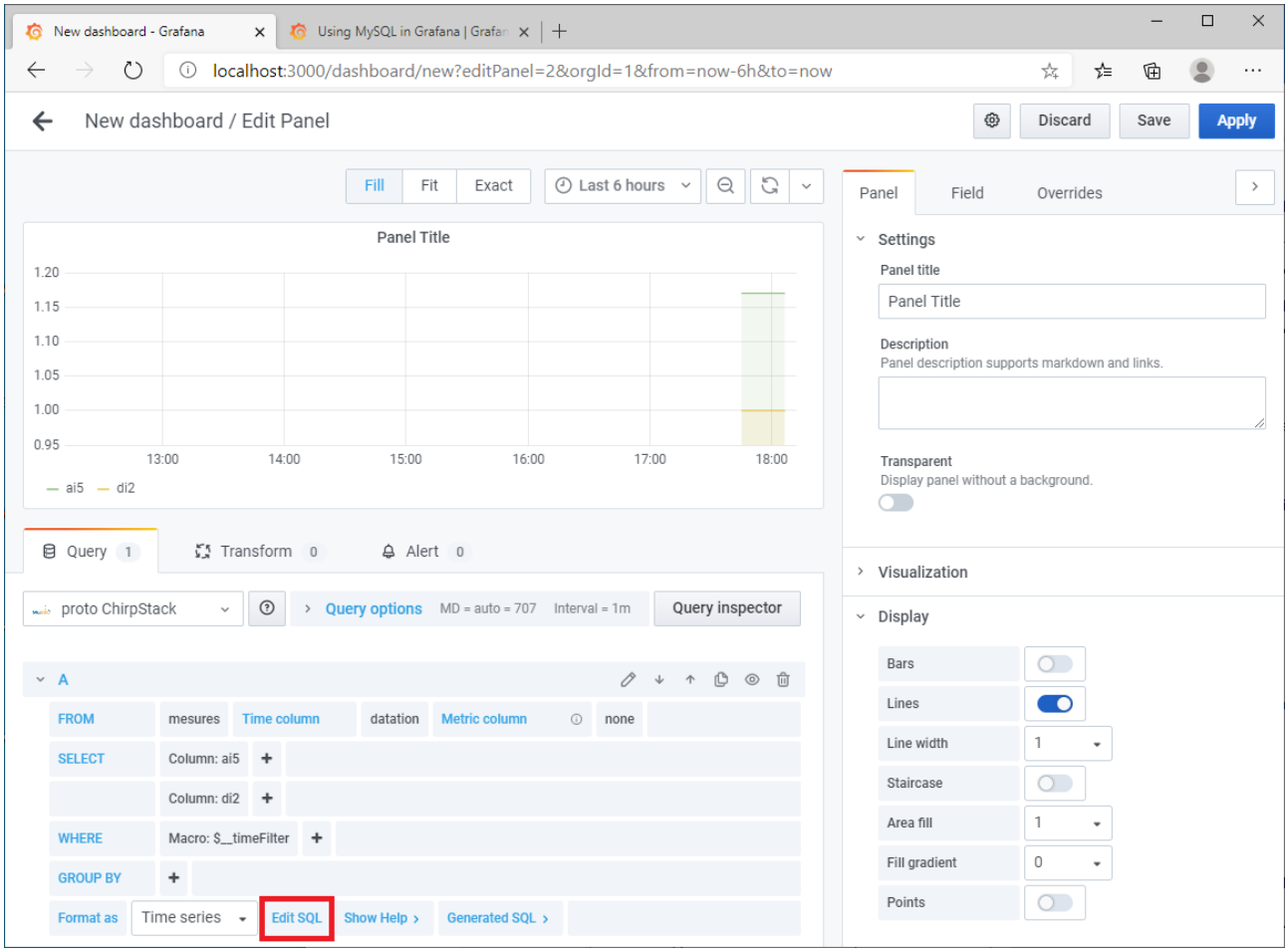
Choisir la table "mesures"

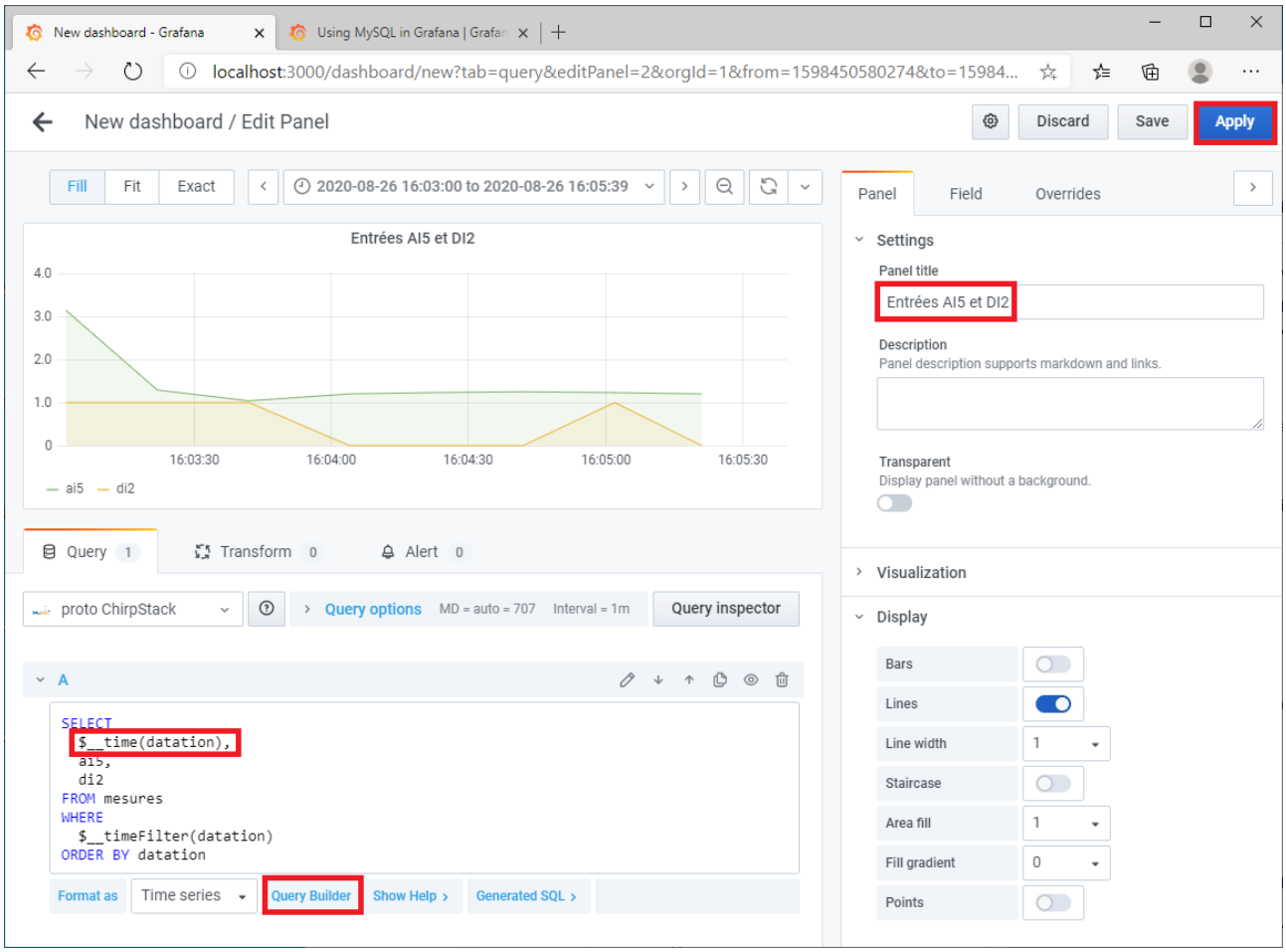


Ajouter les champs correspondants aux entrées AI5 et D2

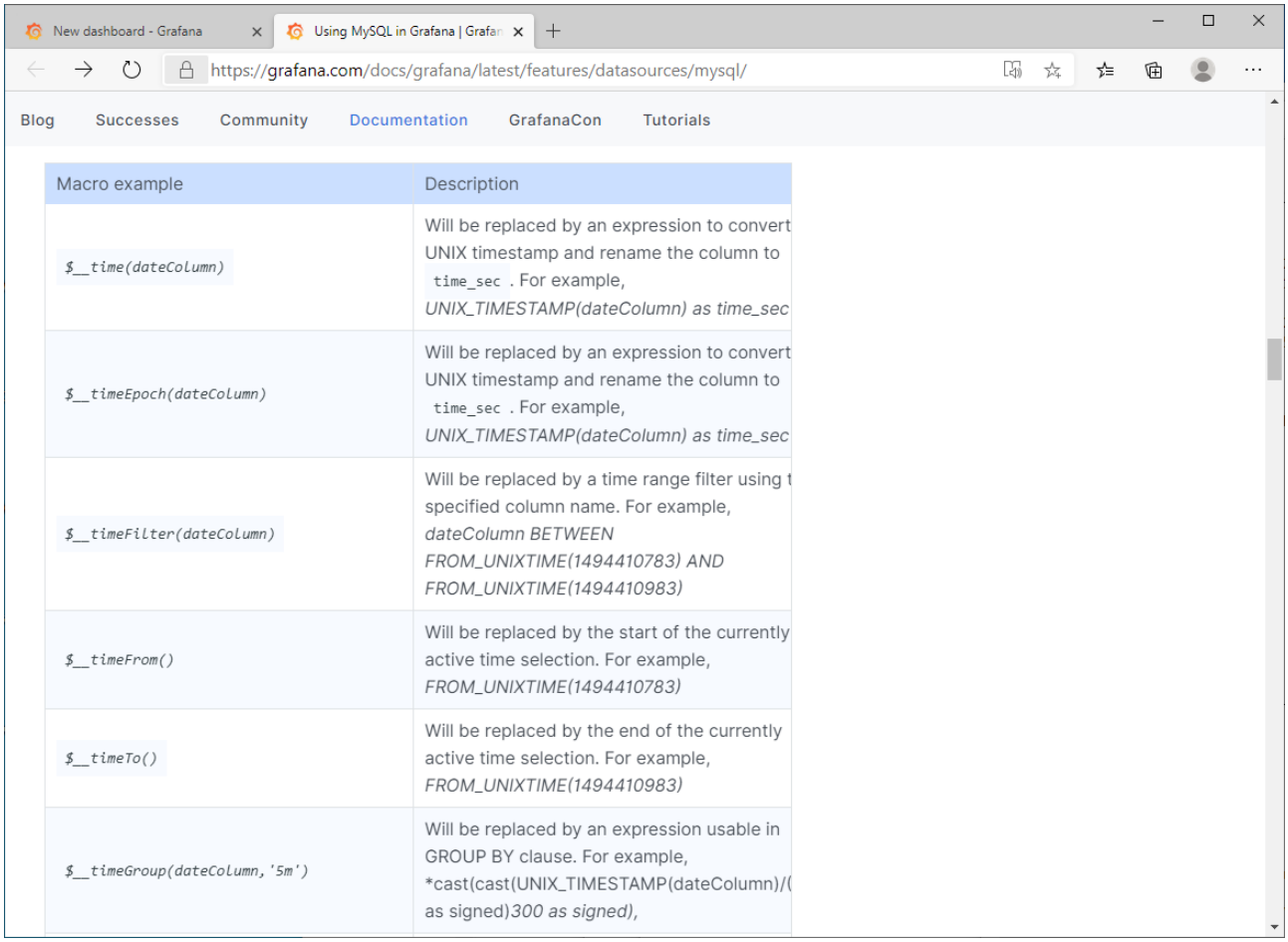




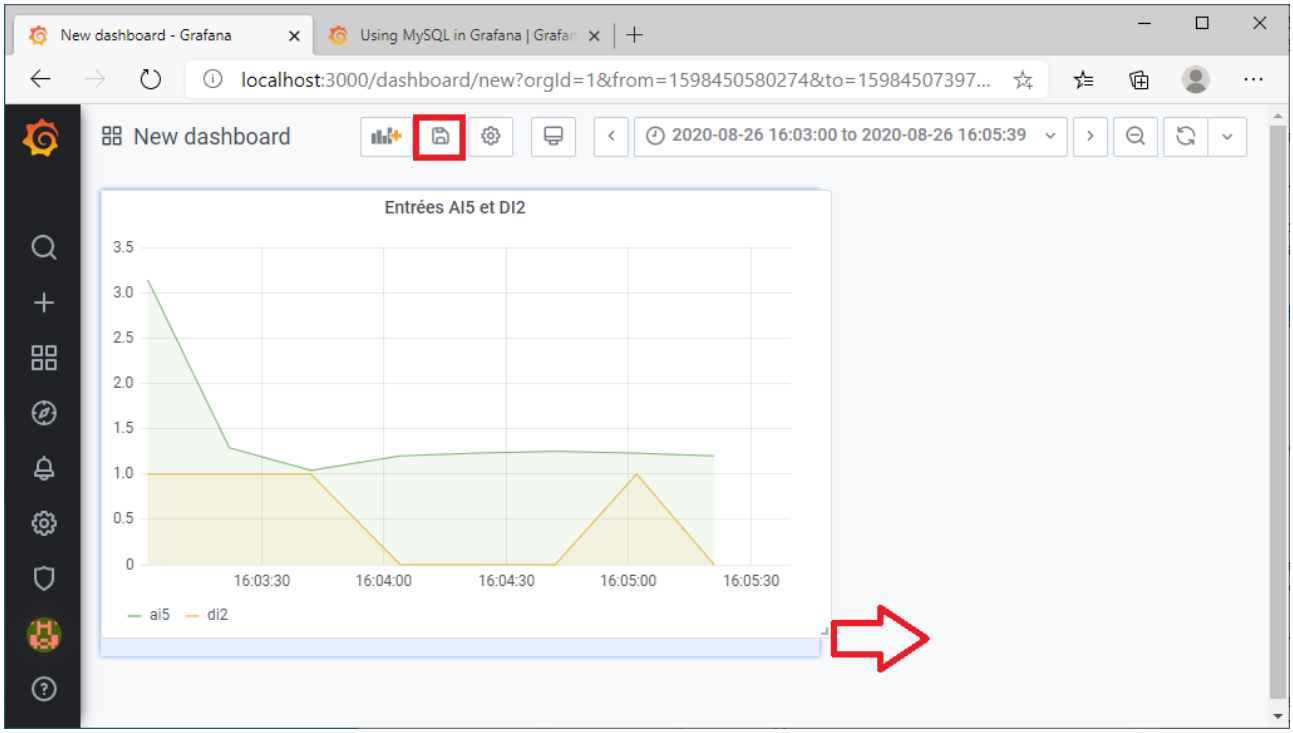

New dashboard - Grafana
Using MySQL in Grafana | Grafana
localhost:3000/dashboard/new?editPanel=2&orgId=1&from=now-6h&to=now
New dashboard / Edit Panel
Discard
Save
Apply
Fill
Fit
Exact
Last 6 hours
Panel
Field
Overrides
Panel Title
1.20
1.15
1.10
1.05
1.00
0.95
13:00
14:00
15:00
16:00
17:00
18:00
ai5
di2
Query 1
Transform 0
Alert 0
proto ChirpStack
Query options
MD = auto = 707
Interval = 1m
Query inspector
A
FROM
mesures
Time column
datation
Metric column
none
SELECT
Column: ai5
Column: di2
WHERE
Macro: $__timeFilter
GROUP BY
Format as
Time series
Edit SQL
Show Help
Generated SQL
Settings
Panel title
Panel Title
Description
Panel description supports markdown and links.
Transparent
Display panel without a background.
Visualization
Display
Bars
Lines
Line width
1
Staircase
Area fill
1
Fill gradient
0
Points

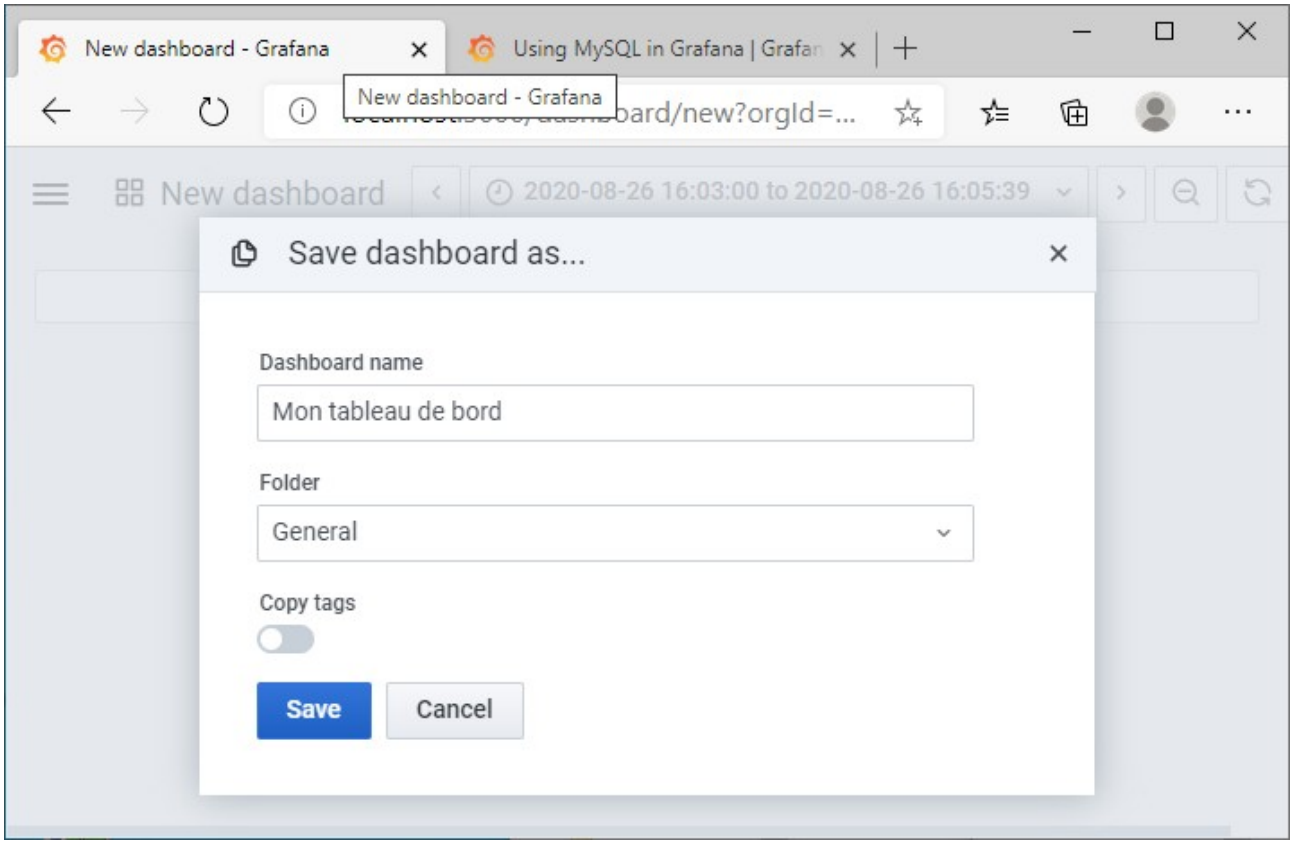

New dashboard - Grafana
Using MySQL in Grafana | Grafana
localhost:3000/dashboard/new?tab=query&editPanel=2&orgId=1&from=1598450580274&to=15984...
New dashboard / Edit Panel
Discard
Save
Apply
Fill
Fit
Exact
2020-08-26 16:03:00 to 2020-08-26 16:05:39
Panel
Field
Overrides
Entrées AI5 et DI2
4.0
3.0
2.0
1.0
0
16:03:30
16:04:00
16:04:30
16:05:00
16:05:30
ai5
di2
Query 1
Transform 0
Alert 0
proto ChirpStack
Query options
MD = auto = 707
Interval = 1m
Query inspector
A
SELECT
$__time(datation),
ai5,
di2
FROM mesures
WHERE
$__timeFilter(datation)
ORDER BY datation
Format as
Time series
Query Builder
Show Help
Generated SQL
Settings
Panel title
Entrées AI5 et DI2
Description
Panel description supports markdown and links.
Transparent
Display panel without a background.
Visualization
Display
Bars
Lines
Line width
1
Staircase
Area fill
1
Fill gradient
0
Points

L'utilisation de la macro $_ _time() permet d'exploiter le timestamp de la BdD
NB : il y a 2 fois le caractère souligné (tiret du 8)



Il est possible d'agrandir le panneau dans le tableau de bord..
Sauvegarder le tableau de bord.





La visualisation des données fonctionne. Elle peut être paramétrée