

LoRa et LoRaWAN pour l'IOT

ESE 3A

2023/2024

TP développé par Steve NKOLO et Michael BOCQUET

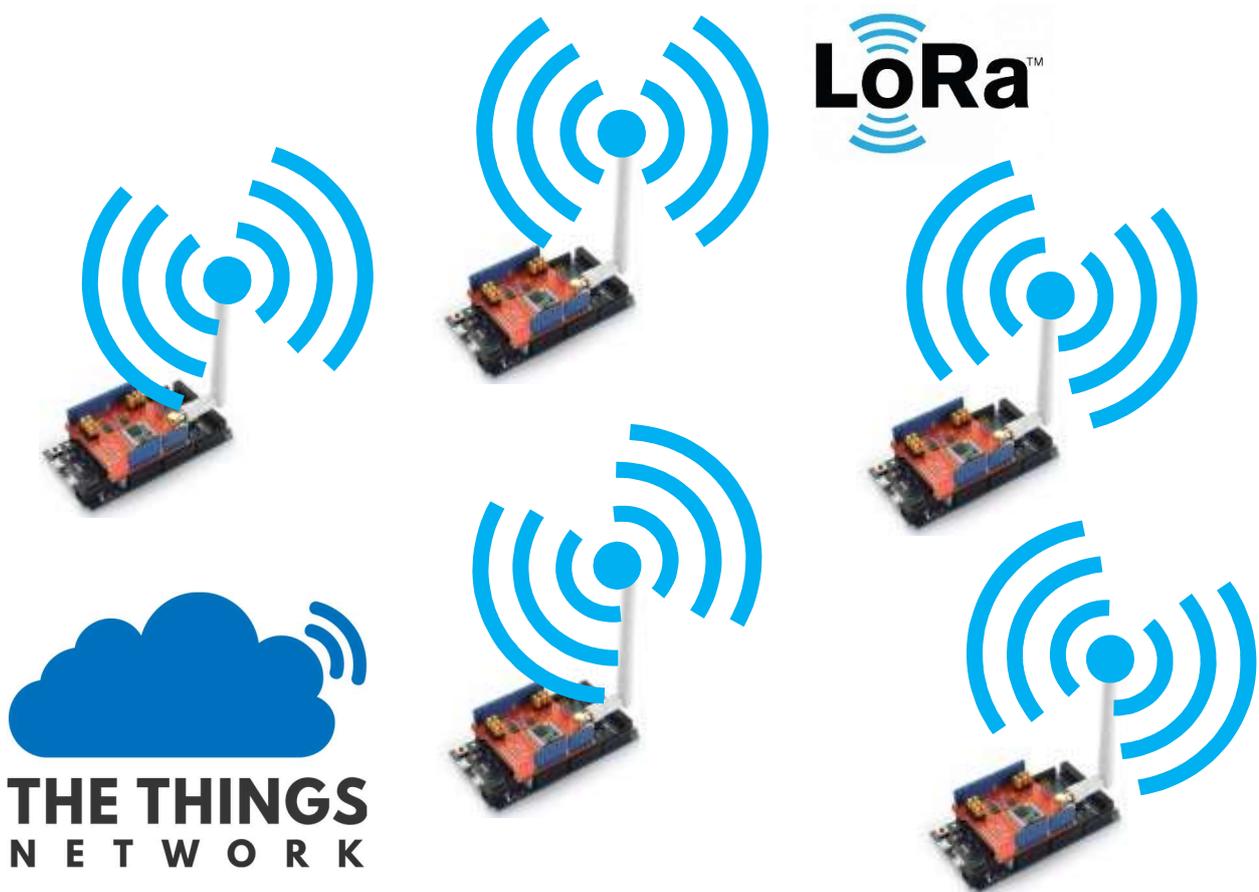


Table des matières

LoRa et LoRaWAN pour l'IOT	1
1 Objectif du TP	2
2 Matériels Utilisés.....	3
3 Logiciels/applications et sites web utilisés:	3
4) Bibliographie	4
5) Communication LoRa point à point	4
5_1) Matériels utilisés.....	4
5.2 Prérequis	4
5.3 Paramètres.....	6
5.4 Questions	7
6) Communication en Broadcast.....	7
7) planification Radio.	8

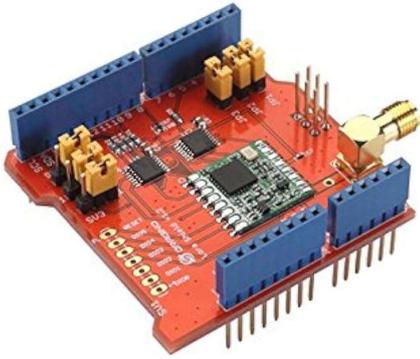
1 Objectif du TP

L'objectif de ce TP est d'étudier et de comprendre les principes de base de LoRa et LoRaWAN.

Vous apprendrez :

- a) A établir une communication entre deux nœuds LoRa
- b) A configurer la communication inter-nœuds
- c) De quelle manière remonter une information issue d'un ou de plusieurs capteurs
- d) Comment configurer une solution de broadcast
- e) A établir une planification radio.

2 Matériels Utilisés



Shield LoRa



Carte Arduino Uno



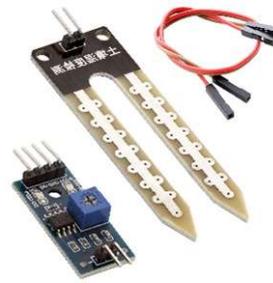
Antenne LoRa



LED



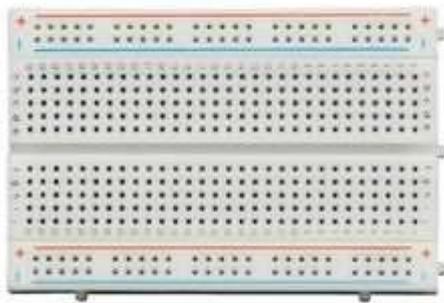
Capteur de emperature



Capteur h'Humidité



Captuer de lumiere



Breadboard



Cables

3 Logiciel/application utilisés:



Arduino

Note : Avant de mettre le shield LoRa (GPS) sous tension, assurez-vous que les antennes sont correctement connectées !

Note2 : Tous les matériels doivent être replacés correctement dans les boîtes à la fin du TP.

Les étudiants qui ne respectent pas ces règles seront pénalisés.

4) Bibliographie

Afin de comprendre le fonctionnement du LoRa, vous devez proposer une synthèse des techniques de ce type de communication. Vous explicitez le principe, les variantes de la communication de son déploiement, les intérêts et les contraintes de cette technologie.

Cette synthèse n'excédera pas 3 pages.

5) Communication LoRa point à point

5_1) Matériels utilisés



1 Arduino + LoRa (GPS) shield utilisé en mode émetteur

1 Arduino + LoRa (GPS) shield utilisé en mode récepteur

5.2 Prérequis

- Dernière version d'Arduino
- Installation des bibliothèques suivantes :

Arduino LMIC Master

DHTlib

L'environnement Arduino est simple à utiliser. Nous allons télécharger un programme simple (scénario) pour montrer comment il fonctionne.

La carte est connectée à l'ordinateur en utilisant un port USB qui émule une liaison série RS232.

Sélectionnez d'abord le bon port : Arduino IDE > Tools > Port.

Sélectionnez ensuite la carte Arduino utilisée (Tools>Card Type> Arduino Uno).

Le scénario est composé de deux parties différentes :

- Une fonction setup() qui ne sera exécutée qu'une seule fois au démarrage.
- Une fonction loop() qui sera exécutée en boucle.

Afin de vérifier si la carte fonctionne, ouvrez l'exemple blink et déversez-le dans la carte. (Fichier>Exemple>01.Basics>Blink).

Dans la suite, nous utiliserons le code suivant pour l'émetteur :

```
#include <SPI.h>
#include <RH_RF95.h>

// Singleton instance of the radio driver
RH_RF95 rf95;
float frequency = XXX.X;

void setup()
{
  Serial.begin(9600);
  //while (!Serial) ; // Wait for serial port to be available
  Serial.println("Start LoRa Client");
  if (!rf95.init())          7
  Serial.println("init failed");
  // Setup ISM frequency
  rf95.setFrequency(frequency);
  // Setup Power, dBm
  rf95.setTxPower(XX);

  // Setup Spreading Factor
  rf95.setSpreadingFactor(X);

  // Setup BandWidth, option
  rf95.setSignalBandwidth(XXXXXXX);

  // Setup Coding Rate: 5 (4/5), 6 (4/6), 7 (4/7), 8 (4/8)
  rf95.setCodingRate4(5);
}

void loop()
{
  Serial.println("Sending to LoRa Server");
  // Send a message to LoRa Server
  uint8_t data[] = "XXXXXXXXXXXXXXXXXXXXXXXXX";
  rf95.send(data, sizeof(data));

  rf95.waitPacketSent();
  // Now wait for a reply
  uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
  uint8_t len = sizeof(buf);

  if (rf95.waitAvailableTimeout(4000)) //(rf95.waitAvailableTimeout(3000))
  {
    // Should be a reply message for us now
    if (rf95.recv(buf, &len))
    {
      Serial.print("got reply: ");
      Serial.println((char*)buf);
    }
    else
    {
      Serial.println("recv failed");
    }
  }
  else
  {
    Serial.println("No reply, is LoRa server running?");
  }
  delay(3000);
}
```

Code pour le nœud émetteur LoRa (à compléter)

5.3 Paramètres

Pour cette partie, vous utiliserez les paramètres suivants :

- SF = 7
- BW = 125KHz
- CR = 5 (par défaut ne changera pas au cours de ce travail)
- F = 868.1 MHz

```
#include <SPI.h>
#include <RH_RF95.h>
// Singleton instance of the radio driver
RH_RF95 rf95;
float frequency = XXX.X;
void setup()
{
    Serial.begin(9600);
    while (!Serial) ; // Wait for serial port to be available
    Serial.println("Start Sketch");
    if (!rf95.init())
        Serial.println("init failed");
    // Setup ISM frequency
    rf95.setFrequency(frequency);
    // Setup Power, dBm
    rf95.setTxPower(XX);

    // Setup Spreading Factor
    rf95.setSpreadingFactor(XX);

    // Setup BandWidth, option
    rf95.setSignalBandwidth(XXXXXXX);

    // Setup Coding Rate: 5 (4/5), 6 (4/6), 7 (4/7), 8 (4/8)
    rf95.setCodingRate4(5);

    Serial.print("Listening on frequency: ");
    Serial.println(frequency);
}

void loop()
{
    if (rf95.available())
    {
        // Should be a message for us now
        uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
        uint8_t len = sizeof(buf);
        if (rf95.recv(buf, &len))
        {
            RH_RF95::printBuffer("request: ", buf, len);
            Serial.print("got request: ");
            Serial.println((char*)buf);

            // Send a reply
```

```
uint8_t data[] = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX";
rf95.send(data, sizeof(data));
rf95.waitPacketSent();
Serial.println("Sent a reply");

}
else
{
    Serial.println("recv failed");          9
}
}
}
```

Code pour le nœud récepteur LoRa (à compléter)

5.4 Questions

- 1) Réglez la puissance conformément aux réglementations de l'UE (argumentez votre choix) et complétez le code en conséquence.
- 2) Choisissez une petite phrase secrète entre les groupes (un pour l'émetteur, un pour le récepteur). Ouvrez le moniteur série des deux cartes et copiez les messages de sortie des deux cartes en indiquant clairement qui est l'émetteur et qui est le récepteur. Que se passe-t-il ?
- 3) Eprouver cette communication par l'envoi de différents caractères, etc...
- 4) Pour différentes topologies, il est nécessaire de connaître la force du signal reçu par le récepteur. Développer une solution étayée par des arguments pour obtenir cette donnée.
- 5) En faire de même avec le SNR. Quel intérêt celui-ci a sur la communication ?
- 6) Votre solution fonctionnant, nous voulons maintenant commander une Led sur le récepteur, développer le code nécessaire à cette action (la Led doit clignoter à la réception du message).

Attention, la Led ne supporte qu'un courant limité !!!

- 7) Nous voulons également récupérer l'information d'un capteur de température type DHT. Réaliser cette fonction.

PS : Merci de détailler chaque partie de code, et chaque solution par des captures d'écran, figures, graphique,... qui colorerons votre compte rendu de TP.

6) Communication en Broadcast.

Pour cette partie, merci de travailler avec un autre groupe que vous stipulerez sur votre compte rendu.

- a) Etablir la communication de type broadcast.
- b) Que constatez-vous au niveau des récepteurs ?
- c) Que constatez-vous au niveau de l'émetteur ? est-ce normal
- d) Proposer une solution pour résoudre ce problème.
- e) Votre réseau fonctionnant maintenant correctement, développer une topologie capable de remonter vers la tête de réseau les informations issues d'au moins 3 capteurs communicants différents.

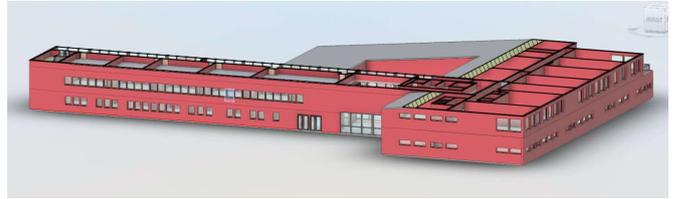
7) Planification Radio.

Dans cette partie, vous allez proposer et expérimenter une solution afin de relever le niveau de réception ainsi que la qualité du signal dans le bâtiment où se déroule le TP.

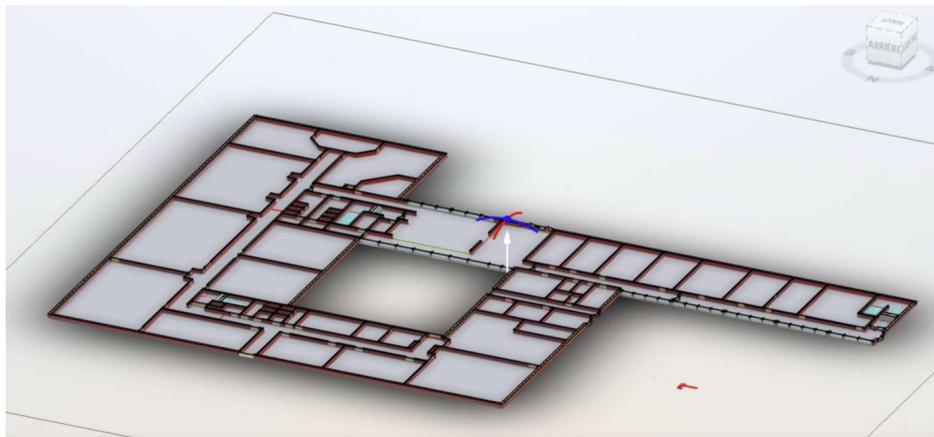
Pour vous aider, voici les plans du bâtiment Abel de Pujol 3



AB3 vu du parking



AB3 vu du tramway sans le 2^{ème} étage



AB3 vu du tramway sans le 2^{ème} étage

- Proposer une méthodologie de mesure.
- Effectuer les relevés dans les zones accessibles.
- Proposer une cartographie du niveau de réception (type fingerprinting) et en qualité de réception.
- Faire un bilan.