

# Table des matières

- Introduction
- Première partie:
  - a. SpreadingFactor
  - b. Bandwidth
  - c. Payload Size
- Deuxième partie
- Conclusion

### • INTRODUCTION:

LoRaWAN est un protocole de télécommunication permettant la communication à bas débit, par radio, d'objets à faible consommation électrique communiquant selon la technologie LoRa et connectés à l'Internet via des passerelles(cf wikipedia)

LoRa wan utilise une modulation a étalement de spectre(868MHz) qui offre de bonne performance et à une portée importante

Le but de ce TP est d'observer une modulation LoRa sous la forme d'un spectrogramme.

En modifiant les paramètres tels que le SpreadingFactor, la largeur de la bande (Bandwidth) et la taille de la payload (Payload Size) nous observerons leurs influences.

Ensuite nous verrons comment recevoir un packet LoRa sur notre carte.

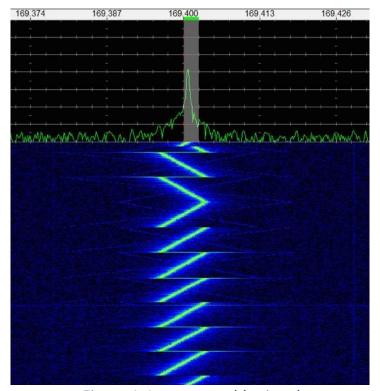


Figure 1: Lora pre-amble signal

# • PREMIÈRE PARTIE :

Grâce au programme Arduino donne sur le tp nous pouvons modifier les parametres afin d'observer sur le spectre les modifications, nous utilision le programme « LoRaSender » qui nous permet d'observer sur le spectre le LoRa pre-amble

```
© Little Edition County Children Lills

| County Children Children County Children Children
```

Figure 2: Programme Arduino

Dans un premier temps nous avons choisis une fréquence afin de pouvoir l'observer sur le spectrogramme. La fréquence choisit est 866.0Hz.



Figure 3: LoRa pre-amble (fréquence 866Hz)

#### 1) SPREADINGFACTOR:

Le SpreadingFactor représente la longueur du code envoyée. Nous avons d'abord commencé par modifier le SpreadingFactor. Les valeurs vont de 7 à 12.

Sur le spectre ci-dessus on peut lire que plus le SpreadingFactor est fort plus l'inclinaison du chirp est faible, en théorie plus le SpreadingFactor est élevé, plus le débit de données plus faible, la sensibilité est élevée et plus la portée est longue.

Tandis qu'un un spreading Factor plus bas signifie un débit de données plus élevé, une sensibilité plus faible et une portée plus courte.

Figure 4: Programme Arduino (Spreading factor=7)

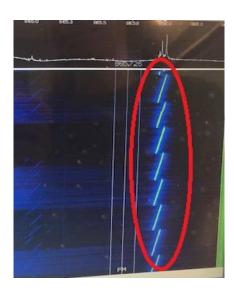


Figure 5: LoRA pre-amble (SpreadingFactor=7)

#### 2) BANDWIDTH:

Le bandwidth est tout simplement la bande passante et la portée d'une communication et le débit d'une voie de transmission.

Nous avons modifié le paramètre Bandwidth, sur ce TP on a pu la modifier de 7,8E3 à 500KHz, on constate que plus on augmente le Bandwidth plus la fréquence sera grande et donc la largeur du spectre sera grande.

Théoriquement une grande bande passante aura un débit très fort (beaucoup plus de signal) le problème est que plus la fréquence est grande plus la portée de la communication sera faible.

```
int counter - v;

// Parameters you can play with :

int txPower = 14; // from 0 to 20, default is 14

int spreadingPactor = 12; // from 7 to 12, default is 12

long signalBandwidth = 500E3; // 7.8E3, 10.4E3, 15.6E3, 20.8E3, 31.25E3,41.7E3,62.5E3,125i

int codingRateDenominator=5; // Numerator is 4, and denominator from 5 to 8, default is 5

int preambleLength=8; // from 2 to 20, default is 8

String payload = "Lion A1 "; // you can change the payload

$define RST 8

$define RST 8

$define BAND 866E6 // Here you define the frequency carrier

void setup() {

Serial.begin(115200);

while (!Serial);
```

Figure 6: Programme Arduino (Bandwidth :500E3 Hz)

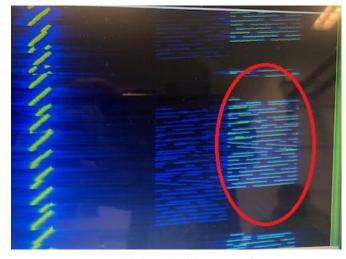


Figure 7 : LoRA pre-amble (Bandwidth :500E3 Hz)

### 3) PAYLOAD SIZE:

Le Payload Size est la charge utile, il permet d'établir la taille du message.

La taille du message maximum est de 255 bits.

Donc on a modifié le Payload, sur le spectre plus le message sera long (en respectant la taille maximal) plus la couleur du pre-amble sera clair (voir photo ci-dessous)

Plus le message sera long plus la transmission sera longue.

Figure 9: LoRA pre-amble (Payload Size)

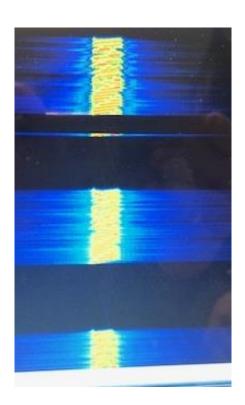


Figure 8: Programme Arduino( Grand Payload Size)

## • 2ÈME PARTIE DU TP

Dans la seconde partie du TP, nous sommes rentrés en communication avec des élèves dans une salle différente à l'aide des programmes Arduino donné par le professeur et au LoRa modulation. La classe a était diviser en deux (groupe A et B), tous les groupes se sont connectés sur la même fréquence afin de trouver un binôme complémentaire, les groupes A devait communiquer avec le groupe B1. Nous étions le groupe B1 donc nous avons communiquer avec le groupe A1

Ensuite, nous avons communiqué une fréquence sur laquelle l'autre binôme devait se calibrer pour pouvoir leur poser la question "Si votre binôme était un animal de compagnie ?".

Arduino Figure 10 : Programme Arduino (Sender)

Après leur avoir posé la question, le groupe A1 nous a répondu sur la même fréquence "Ecureuil". On peut voir la réponse sur le Moniteur Série grâce au programme Arduino (LoRaReceiver\_Phy).

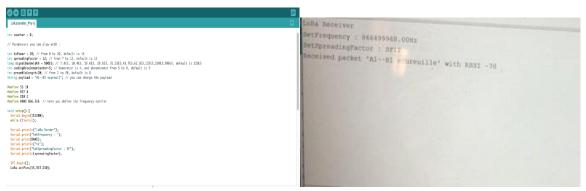


Figure 12 : Programme Arduino du groupe A1

Figure 11 : Moniteur Série

Le groupe A1 nous ont aussi envoyé une question pour entrer en contact avec nous, il est donc possible de communiquer grâce au LoRa.

## • CONCLUSION :

En modifiant plusieurs paramètres le LoRa modulation change comme par exemple le SpreadingFactor, l'inclinaison change en fonction de la donnée (de 7 à 12).

On a aussi pu modifier le Bandwidth et le Payload Size sur le LoRa modulation.

En changeant les paramètres il y a donc un affichage différent.

On a ensuite pu rentrer en communication avec un groupe à distance en se positionnant sur la même fréquence.

On a pu envoyer et recevoir des messages.

Il est donc possible de communiquer à distance grâce au LoRa modulation.