



## Présentation de la bibliothèque LoRa

Installation :

- Aller dans Croquis → Inclure une bibliothèque → Gérer les bibliothèques...
- Dans la barre de recherche, taper "LoRa"
- Trouver "LoRa by Sandeep Mistry" (c'est la bibliothèque standard)
- Cliquer sur Installer

Dans le sketch utilisant LoRa, il faut inclure <SPI.h> et <LoRa.h>

La communication LoRa est gérée à travers les méthodes de l'objet global LoRa

Initialisation et Démarrage

```
bool begin(long frequency);
```

```
// Exemple :
if (!LoRa.begin(868100000)) {
  Serial.println("Échec de l'initialisation LoRa!");
}
```

Configuration des Broches

```
// Configurer les broches personnalisées
void setPins(int ss = 10, int reset = 9, int dio0 = 2);
```

```
// Exemple :
LoRa.setPins(10, 9, 2); // SS, RST, DI00
```

Paramètres de Transmission

```
// Spreading Factor (SF7 à SF12)
void setSpreadingFactor(int sf);
// Exemple : LoRa.setSpreadingFactor(7);

// Bandwidth (7.8E3, 10.4E3, 15.6E3, 20.8E3, 31.25E3, 41.7E3, 62.5E3, 125E3,
250E3, 500E3)
void setSignalBandwidth(long sbw);
// Exemple : LoRa.setSignalBandwidth(125000);

// Coding Rate (5=4/5, 6=4/6, 7=4/7, 8=4/8)
void setCodingRate4(int denominator);
// Exemple : LoRa.setCodingRate4(5);

// Puissance d'émission (2 à 20 dBm)
void setTxPower(int level, int outputPin = PA_OUTPUT_PA_BOOST_PIN);
// Exemple : LoRa.setTxPower(14);
```

## TP 1

Développer des objets connectés	Version	1.0
Chiheb Ameur ABID		

Implémenter une communication bidirectionnelle sans fil entre deux cartes Arduino équipées de modules LoRa, en utilisant le protocole LoRa dans la bande de fréquences 868 MHz.

La connexion entre le shield LoRa et la carte Arduino Uno est comme suit :

LoRa Shield → Arduino Uno

VCC	→ 3.3V
GND	→ GND
MISO	→ D12
MOSI	→ D11
SCK	→ D13
NSS	→ D10
RST	→ D9
DI00	→ D

L'objectif est de faire communiquer deux cartes Arduino via LoRa en implémentant un protocole simple de demande-réponse.

Consignes :

1. La station A envoie un message "hello" toutes les 10 secondes
2. La station B répond automatiquement par "bonjour" lorsqu'elle reçoit "hello"
3. Chaque message doit contenir l'ID de l'émetteur
4. Utiliser la fréquence 868.1 MHz avec les paramètres LoRa suivants :
  - Spreading Factor: 7
  - Bandwidth: 125 kHz
  - Coding Rate: 4/5



#### Paramètres de Réception

```
// Mot de synchronisation (0x12 à 0xFF)
void setSyncWord(int sw);
// Exemple : LoRa.setSyncWord(0x34);
```

#### Émission de Données

```
// Commencer un nouveau paquet
int beginPacket();

// Ajouter des données au paquet
size_t print(const String &s);
size_t print(const char* text);
size_t print(uint8_t data);

// Terminer et envoyer le paquet
int endPacket();

// Exemple complet :
LoRa.beginPacket();
LoRa.print("Hello LoRa!");
LoRa.endPacket();
```

#### Réception de Données

```
// Vérifier si un paquet est disponible
int parsePacket();

// Lire les données reçues
int available();
int read();
int peek();

// Exemple complet :
if (LoRa.parsePacket()) {
  while (LoRa.available()) {
    char c = LoRa.read();
    Serial.print(c);
  }
}
```

#### Squelette du sketch du maître

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



```
#include <LoRa.h>

// Paramètres LoRa
#define LOCAL_ID "1234" // Identifiant unique
#define SF 7 // Spreading Factor
#define BW 125E3 // Bandwidth (125 kHz)
#define CR 5 // Coding Rate (4/5)

// Pins pour Dragino LoRa Shield
#define LORA_SS 10
#define LORA_RST 9
#define LORA_DI00 2

void setup() {
  Serial.begin(115200);
  while (!Serial); // Attendre que le port série soit prêt

  // Initialiser LoRa avec les pins du Dragino Shield
  ...

  // Initialiser le module LoRa à 868.1 MHz
  ...

  // Configurer les paramètres LoRa
  ...
  LoRa.setSyncWord(0x34); // Sync Word pour communication privée
}

void loop() {
  // Construire le Payload à envoyer
  ...
  // Envoyer le message "Hello"
  ...
  // Attendre 10 secondes avant le prochain envoi
  delay(10000);
}
```

#### Squelette du sketch de récepteur

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

// Paramètres LoRa
```



```
#define LOCAL_ID "5678" // Identifiant unique pour le récepteur
#define SF 7 // Spreading Factor
#define BW 125E3 // Bandwidth (125 kHz)
#define CR 5 // Coding Rate (4/5)

void setup() {
  Serial.begin(115200);
  while (!Serial); // Attendre que le port série soit prêt

  // Initialiser le module LoRa à 868.1 MHz
  ...

  // Configurer les paramètres LoRa
  ...
}

void loop() {
  // Vérifier si un paquet est reçu
  ...
  Si un paquet est reçu {
    // Lire le message
    ...
    // Afficher le message reçu
    ...

    // Vérifier si le message contient "Hello"
    if (received.indexOf("Hello") != -1) {
      // Répondre avec "<5678> Bonjour"
      ...
    }
  }
}
```

Modifier le code du Maître pour qu'il affiche la réponse après avoir envoyé son message.