

**UE INF-IoT**  
**« Internet of Things »**

**Sujet de TP : Transmission de données sans fil en modulation OOK**

L'objectif de cette séance de travaux pratiques est de mettre en œuvre des transmissions sans fil entre capteurs et afficheurs, ces transmissions reposant sur de la modulation OOK (*On-Off Keying*).

La modulation OOK est une forme simplifiée de modulation d'amplitude. En modulation d'amplitude, un module émetteur génère une porteuse sur une fréquence donnée, et l'entrée *data* de cet émetteur permet d'agir sur l'amplitude du signal émis. En modulation OOK, l'entrée *data* agit en tout ou rien : la porteuse est émise (à pleine puissance), ou n'est pas émise. Au niveau du récepteur, la porteuse est donc reçue ou pas, et si elle l'est c'est avec un certain niveau d'amplitude. Il convient donc d'examiner le niveau de réception pour interpréter le signal reçu et le distinguer du bruit ambiant (toujours présent).

Dans le cas présent, vous allez utiliser un MCU (le modèle exact vous sera indiqué en cours de séance), et allez y connecter un module émetteur et un module récepteur OOK. Vous allez ensuite développer de petits programmes permettant d'exploiter ces modules pour faire de la transmission de données.

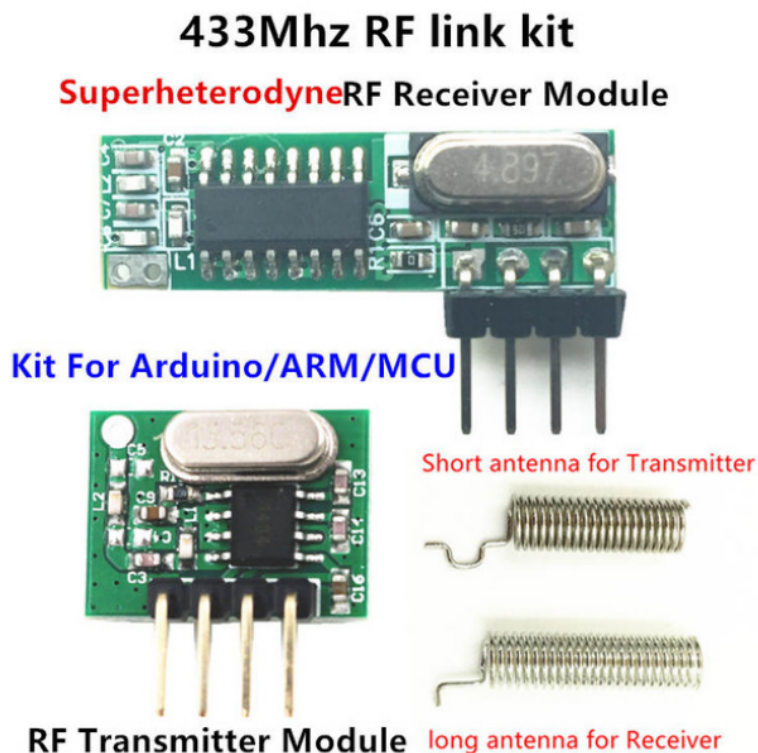


FIGURE 1 – Modules récepteur (en haut) et émetteur (en bas) permettant la modulation OOK dans la bande ISM 433 MHz

## 1 Aperçu des modules émetteurs et récepteurs OOK

Il existe sur le marché un grand nombre de modules peu coûteux permettant d'effectuer de l'émission OOK, de la réception OOK, ou les deux à la fois<sup>1</sup>. La plupart du temps ces modules sont en fait des modules ASK/OOK : ils sont donc capables de fonctionner indifféremment en modulation ASK (i.e., modulation

1. Un module capable de fonctionner à la fois en tant qu'émetteur et de récepteur est appelé un *transceiver* (contraction de *transmitter-receiver*).

d'amplitude) ou en modulation OOK (i.e., en tout ou rien). Par ailleurs chaque module est conçu pour fonctionner spécifiquement dans une bande de fréquences donnée, comme par exemple l'une des bandes de fréquences ISM 433 MHz, 868 MHz, 915 MHz<sup>2</sup>, 2.4 GHz, etc. Les transmissions ne sont en général possibles qu'à faible portée (quelques dizaines de mètres en terrain dégagé) et faible débit (au plus quelques dizaines de kbit/s).

Dans le cadre de cette séance de TP vous allez utiliser des modules émetteurs et récepteurs distincts : le module récepteur sera un module RX470B (ou équivalent), et le module émetteur sera un module WL102-341 (ou équivalent). Ces deux types de modules sont représentés dans la figure 1. Ils fonctionnent dans la bande 433 MHz. Plus précisément, le module émetteur permet de produire une porteuse modulée autour de la fréquence centrale de 433.92 MHz, avec une bande passante d'environ 150 kHz. Dans ces conditions le débit de transmission maximal est de l'ordre de 20 kbit/s. Le module récepteur est bien sûr conçu pour recevoir ce type de modulation.

On peut noter que dans notre vie quotidienne, bon nombre de systèmes de transmission sans fil reposent sur de la modulation OOK dans la bande ISM 433 MHz. C'est par exemple le cas de beaucoup de stations météo domestiques, de télécommandes pour portes de garage, portails de jardin, volets roulants, et même parfois portes de voitures. Le module récepteur que vous allez utiliser ici permettrait de recevoir ces signaux, et le module émetteur de les reproduire.

Les modules émetteur et récepteur offrent chacun quatre broches (plus une cinquième pour l'antenne). Deux broches sont dédiées à l'alimentation électrique ( $V_{CC}$  et GND), une broche est dédiée à la transmission de données, et la quatrième broche peut être laissée libre.

Attention : le module émetteur s'alimente dans une plage de tensions de 2.0 V à 3.6 V. Vous veillerez donc à l'alimenter en 3.3 V, et **surtout pas en 5 V** ! Cela pourrait lui être fatal. Le module récepteur s'alimente quant à lui dans une plage de tensions de 2.2 V à 5 V. Vous pourrez donc l'alimenter en 3.3 V ou 5 V, mais ses performances seront sans doute meilleures avec 5 V.

## 2 Aperçu de la librairie RadioHead

Il est tout à fait possible de piloter un module émetteur en agissant directement sur la broche *data* de ce module, et d'exploiter de même un module récepteur en consultant périodiquement l'état de sa broche *data*. Le codage de signaux, la maîtrise d'une certaine rapidité de transmission, etc. doivent alors être gérés explicitement dans le code source de l'application.

La librairie *RadioHead*<sup>3</sup> est une librairie C++ permet de simplifier grandement le développement d'applications exploitant des modules émetteurs ou récepteurs. Elle fournit d'une part des pilotes (*drivers*) adaptés à plusieurs types de module radio, et d'autre part des gestionnaires (*managers*) offrant des services optionnels de plus haut niveau et pouvant de ce fait répondre aux besoins propres de chaque application. Ces services de haut niveau couvrent notamment la transmission en mode datagramme (avec adressage, en mode fiable ou non fiable), ainsi que le routage entre nœuds (manuel ou avec découverte automatique des routes).

Attention : certains de ces services de haut niveau nécessitent que les nœuds du réseau soient chacun dotés d'un ensemble émetteur + récepteur (ou d'un *transceiver*). C'est par exemple le cas du service en mode datagramme fiable, puisqu'il repose sur des accusés de réception pour assurer la retransmission éventuelle des messages. Les services de routage nécessitent de même que les transmissions entre nœuds soient bi-directionnelles. Comme nos deux montages « capteur » et « afficheur » comporteront, l'un uniquement un module émetteur, l'autre uniquement un module récepteur, nous devons nous passer des services de haut niveau dans le cadre de ce TP.

Pour piloter les modules radio utilisés dans cette séance de TP, vous utiliserez le pilote RH\_ASK de la librairie *RadioHead*. Commencez donc par lire très attentivement la documentation de ce pilote.

Vous noterez que le pilote RH\_ASK ne permet de transmettre que des paquets de taille réduite (la taille maximale admissible est indiquée dans la documentation du pilote), et qu'il n'est pas prévu d'embarquer d'adresses en en-tête de paquet. Pour ce TP vous ferez donc en sorte d'embarquer dans la charge utile de chaque paquet

---

2. Attention : la bande ISM 915 MHz n'est utilisable qu'en Amérique du Nord ! Dans le reste du monde, et en Europe notamment, cette bande chevauche les bandes de fréquences réservées pour la téléphonie cellulaire (GSM).

3. <http://www.airspayce.com/mikem/arduino/RadioHead>

l'identité du montage émetteur et celle du montage récepteur. Ceci permettra au montage « afficheur » de présenter à l'écran l'identité de la source des données qu'il reçoit. Le montage « capteur » pourra quant à lui adresser ses données, soit à un afficheur donné, soit à l'ensemble des afficheurs situés aux alentours.

### 3 Montage

Deux types de montage seront réalisés pour cette séance de TP, sur des platines d'assemblage distinctes : un montage « capteur », et un montage « afficheur ».

Le montage « capteur » sera constitué d'un micro-contrôleur auquel seront connectés une sonde d'hygrométrie (e.g., DHT11<sup>4</sup> ou DHT22<sup>5</sup>), une sonde de luminosité (alias photo-résistance<sup>6</sup>), et un émetteur radio. Le micro-contrôleur de ce montage aura pour rôle de consulter périodiquement l'état de chacune des deux sondes, et d'envoyer les résultats par voie radio vers un montage « afficheur ».

Le montage « afficheur » sera constitué d'un micro-contrôleur connecté à un récepteur radio et à un écran OLED. Son rôle sera de recevoir les données provenant d'un ou plusieurs montages « capteurs », et de les afficher à l'écran.

Dans le montage « capteur », la broche de transmission de données (*data*) du module émetteur sera connectée à une broche d'entrée-sortie *numérique* du micro-contrôleur. Vous veillerez à alimenter ce module émetteur en 3.3 V (et **surtout pas en 5 V!**). Dans ce montage vous intégrerez également les sondes d'hygrométrie et de luminosité. La sonde d'hygrométrie (DHT11 ou DHT22) utilise un protocole propriétaire, assez semblable au protocole *1-Wire* mais non compatible avec ce dernier. Il existe des bibliothèques toutes faites pour lire ce type de sonde, comme par exemple la bibliothèque *DHTstable* développée par Rob Tillaart<sup>7</sup>. Notez que pour être précise une mesure d'hygrométrie doit être « compensée en température ». Les sondes DHT11 et DHT22 mesurent donc à la fois la température et l'humidité ambiantes.

La sonde de luminosité (basée sur une photo-résistance) doit quant à elle être intégrée dans un « pont diviseur » : une des pattes de la sonde est connectée à  $V_{CC}$ , l'autre à une entrée analogique du micro-contrôleur. Une résistance de valeur fixe (typiquement, 10 k $\Omega$ ) est insérée entre la même entrée analogique et GND. En consultant l'entrée analogique, on obtient une valeur qui dépend de la luminosité perçue par la sonde.

Dans le montage « afficheur », la broche *data* du module récepteur sera connectée à une broche d'E/S *analogique* du micro-contrôleur. Vous pourrez alimenter le module récepteur en 5 V. Le montage « afficheur » sera également doté d'une carte d'affichage de référence GM009606. Cette carte supporte un petit écran OLED de définition 128x64, contrôlé par un circuit SSD1306<sup>8</sup>. L'interaction entre micro-contrôleur et écran s'effectue via un bus I<sup>2</sup>C. Les bibliothèques *u8g2*<sup>9</sup> et *u8x8* (incluse dans la précédente) permettent de piloter aisément la plupart des écrans monochrome, dont ceux bâtis sur le circuit SSD1306.

En effectuant ces montages, veillez à toujours respecter les polarités et niveaux de tension électrique ! Et comme toujours faites contrôler chaque montage par un(e) autre étudiant(e), puis par un enseignant, **avant** de le mettre sous tension.

---

4. <https://akizukidenshi.com/download/ds/aosong/DHT11.pdf>

5. <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>

6. *LDR : Light-Dependant Resistor*. Composant électronique dont la résistivité varie en fonction de la lumière incidente.

7. <https://github.com/RobTillaart/Arduino/tree/master/libraries/DHTstable>

8. <http://www.upgradeindustries.com/media/ebay/Photos/OLED/datasheet.pdf>

9. <https://github.com/olikraus/u8g2>

## 4 Développement de programmes pour les montages « capteur » et « afficheur »

En utilisant l'éditeur Visual Studio Code avec l'extension PlatformIO, ainsi que le *framework* Arduino, vous allez développer un programme pour chaque type de montage.

### 4.1 Programme pour le montage « capteur »

La fonction de ce programme est simplement de lire à intervalle régulier (par exemple toutes les 15 secondes) les deux sondes du montage « capteur », et d'embarquer les valeurs lues dans un paquet qui est alors émis sur le canal radio. Ce paquet a pour charge utile une structure de données de la forme suivante :

```
struct Payload_t {  
    uint8_t sdr;  
    uint8_t rcvr;  
    float temperature;  
    float humidity;  
    uint16_t light;  
}
```

Dans cette structure de données, les champs `sdr` et `rcvr` servent à indiquer l'identité de l'émetteur et du destinataire du paquet (sous forme d'un entier non signé, exprimé sur 8 bits). Vous veillerez à donner une identité propre à chaque montage, en vous assurant qu'aucun autre montage dans la salle n'utilise la même identité. L'identité 255 (ou 0xFF) sera réservée pour désigner « tous les montages afficheurs ». On pourra ainsi choisir d'adresser les données, soit à un afficheur spécifique, soit à tous les afficheurs.

Les champs `temperature`, `humidity`, et `light`, ont bien sûr vocation à porter les valeurs issues de la lecture des sondes.

### 4.2 Programme pour le montage « afficheur »

Ce programme se contente de réceptionner les paquets qui lui parviennent depuis le canal radio, et d'en afficher le contenu sur l'écran OLED. Vous ferez en sorte que l'identifiant fixé en paramètre du programme lui permette de filtrer les paquets qui ne lui sont pas explicitement destinés (sauf bien sûr s'il s'agit de paquets diffusés à l'intention de tous les afficheurs).