

XBee

Table des matières

1. Présentation du XBee.....	2
1.1. Applications.....	2
1.2. Réseaux Zigbee.....	3
2. Commandes de configuration.....	3
2.1. Commandes AT.....	3
2.2. Principales commandes AT.....	4
3. Association en réseau.....	4
3.1. Réseaux sans fils.....	4
3.2. Communication avec l'ordinateur.....	5
3.3. Adressage.....	6
4. Mise en place d'une communication.....	6

"XBee" est une famille de composants sans-fil prêts à l'emploi développés qui implémentent différents protocoles, dont 802.15.4 et sa version de plus haut-niveau ZigBee, ainsi que des protocoles de réseau ad-hoc ("mesh") spécifique.



1. Présentation du XBee

Les produits XBee sont des modules de communication sans fil certifiés par la communauté industrielle ZigBee Alliance. La certification Zigbee se base sur le standard IEEE 802.15.4 qui définit les fonctionnalités et spécifications des réseaux sans fil à dimension personnelle (Wireless Personal Area Networks : WPANs).

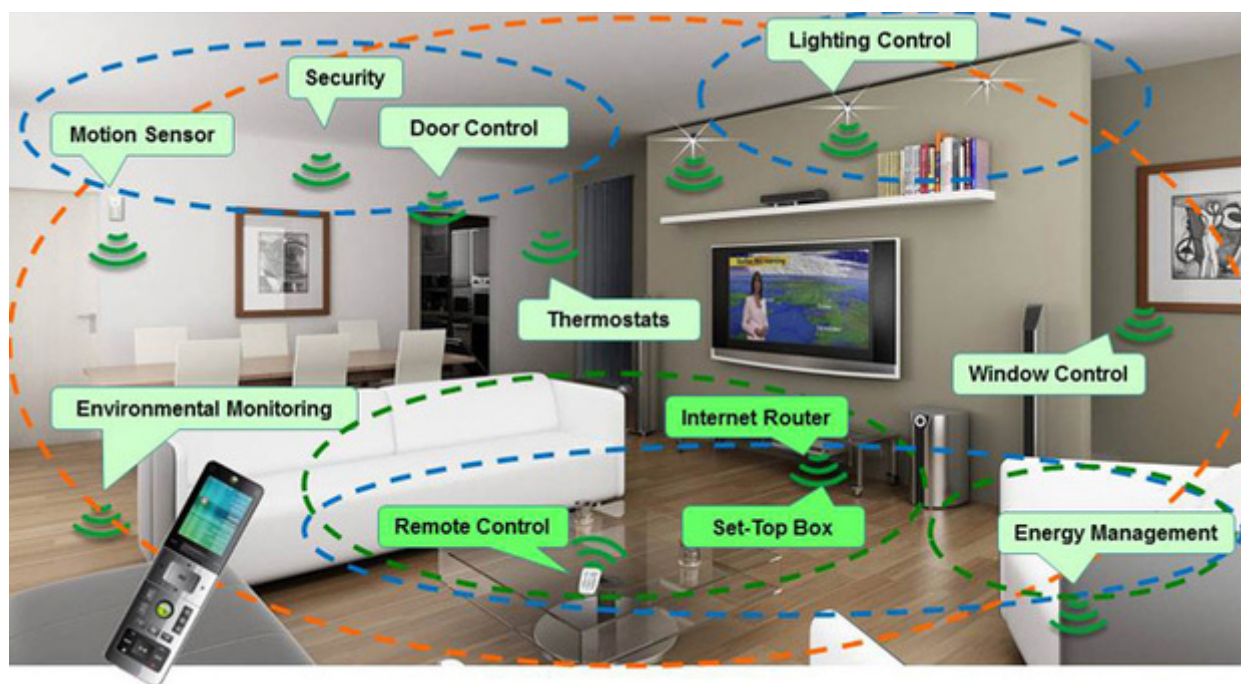


Les principales caractéristiques du XBee :

- fréquence porteuse : 2.4Ghz
- portées variées : assez faible pour les XBee 1 et 2 (10 - 100m), grande pour le XBee Pro (1000m)
- faible débit : 250kbps
- faible consommation : 3.3V @ 50mA (inférieure à 10 μ A en mode "sleep").
- faible coût : ~ 25€
- simplicité d'utilisation : communication via le port série
- ensemble de commandes AT et API
- topologies de réseaux variées : maillé, point à point, point à multipoint

1.1. Applications

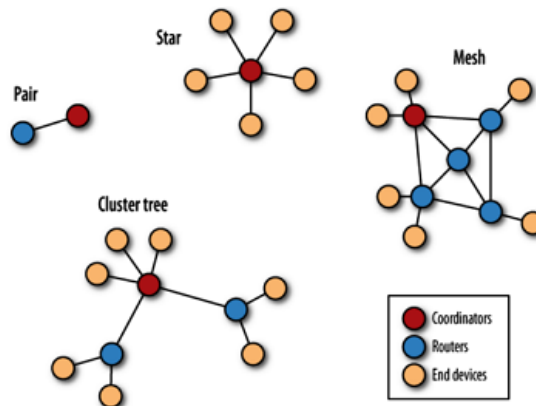
Le ZigBee a été conçu pour réaliser l'Internet des objets, un ensemble d'objets communicants voir "autonomes", une extension d'Internet aux objets physiques. La domotique est l'exemple le plus parlant.



1.2. Réseaux Zigbee

Les réseaux Zigbee peuvent présenter plusieurs types de topologie :

- Point à point (Pair).
- Etoile (Star).
- Arbre (Clustertree).
- Maille (Mesh).



2. Commandes de configuration

Pour modifier ou lire les paramètres du module, on va dialoguer par des commandes "AT".

Il faut tout d'abord passer dans le mode "commande" en envoyant 3 fois le caractère "+" en moins de 1 seconde. On doit respecter un temps de garde (de 1 seconde) avant et après l'envoi de ces 3 caractères. Le module répond par "OK" + "CR" (Carriage Return).



2.1. Commandes AT

Dans les télécommunications, l'ensemble de commandes Hayes est un langage de commandes spécifiques développé pour le modem Hayes en 1981. Les commandes sont une série de mots courts qui permettent de contrôler le modem avec un langage simple : composer un numéro de téléphone, connaître l'état de la ligne, régler le volume sonore, etc. Ce [jeu de commandes](#) s'est ensuite retrouvé dans tous les modems produits.

Elle est constituée des 2 caractères ASCII : "A" et "T" suivis de 2 caractères spécifiques à la commande, puis suit ou pas le caractère "Espace" et enfin suit un paramètre optionnel. On termine la commande par un "CR". Le module répond par "OK" suivi d'un "CR". Pour lire un paramètre, il suffira de laisser le champ paramètre en blanc. C'est le module qui renvoie alors la valeur de son paramètre.

"AT" + "ASCII commande" + "Espace" (option) + Paramètre (option) + "CR"

Si aucune commande AT n'est parvenue au module après son passage en mode commande pendant un temps de TIME OUT de 10 secondes, le module retourne en mode IDLE.

Pour quitter le mode commande avant les 10 secondes du Time OUT, il faut envoyer la commande AT suivante : ATCN et le module répond alors par "OK"

Exemple :

ATBD 3 Cette commande fixe la vitesse de transmission à 9600 bauds. Le module répond par "OK".

On peut envoyer plusieurs commandes à la suite.

Exemple :

ATDB 3,WR,CN Cette commande fixe la vitesse de transmission à 9600 bauds, puis sauve les paramètres dans la mémoire EE PROM et fait sortir le module du mode AT.
Le module répond par "OK","OK","OK".

2.2. Principales commandes AT

ATCN : Pour quitter le mode commande.

ATWR : Sauve les paramètres dans la mémoire non volatile. Il faut impérativement attendre la réponse "OK" du module avant de lui envoyer une nouvelle commande.

ATCH + paramètre (0x0C à 0x17) : Modifie ou lit le canal utilisé dans la bande 2,4 GHz. Par défaut il y a 0x0C.

ATID + paramètre (0xFFFF) : Modifie ou lit l'adresse du Pan ID. Il faut que cette valeur soit la même pour que les modules puissent communiquer entre eux.

ATDL + paramètre (0xFFFFFFFF) : Modifie ou lit les 32 bits LSB de l'adressage destinataire. Par défaut il y a 0x00000000.

ATMY + paramètre (0xFFFF) : Modifie ou lit les 16 bits de l'adressage source. Par défaut il y a 0x0000.

ATBD + paramètre (0 à 7) : Modifie ou lit la vitesse en Baud de la liaison RS232. Par défaut on a 3 soit 9600 bauds.

0	1200 Bauds
1	2400 Bauds
2	4800 Bauds
3	9600 Bauds
4	19200 Bauds
5	38400 Bauds
6	57600 Bauds
7	115200 Bauds

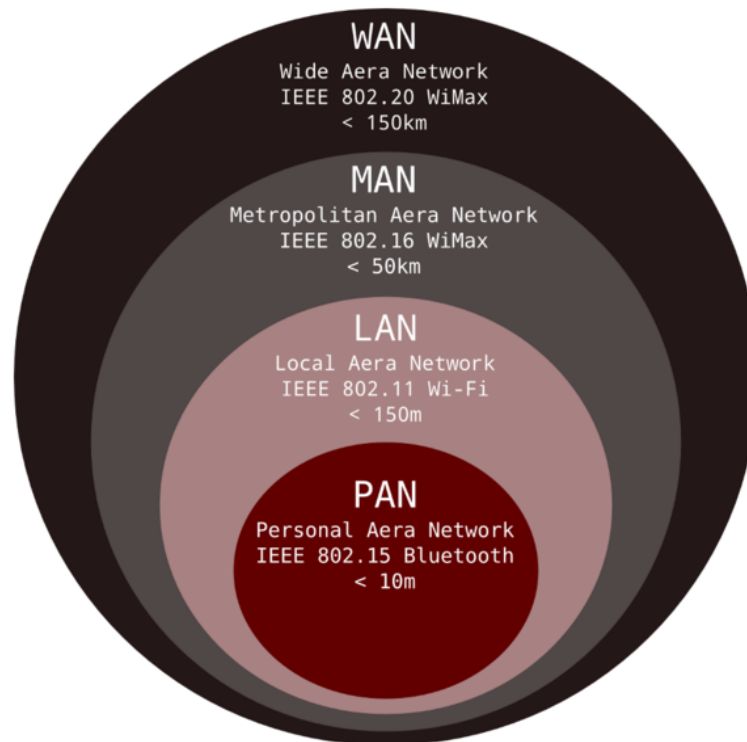
ATRE Restaure les paramètres par défaut du module. Cette commande ne réinitialise pas le champ ID

3. Association en réseau

3.1. Réseaux sans fils

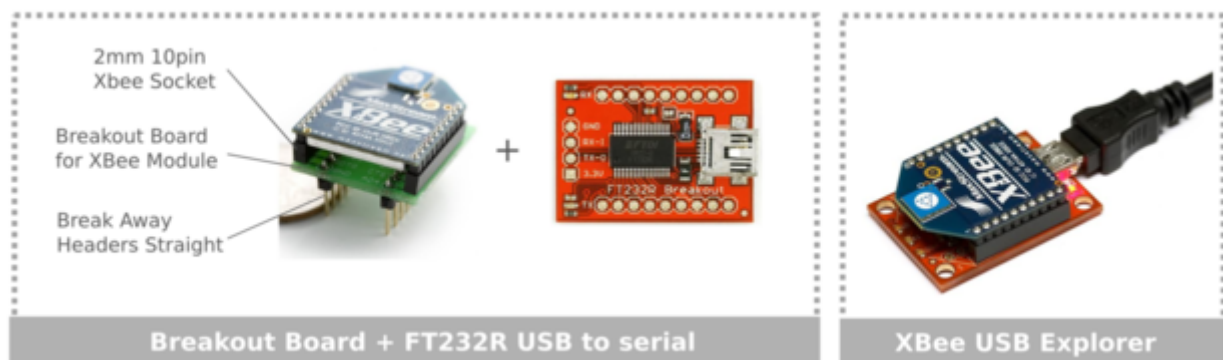
Quand on aborde les réseaux sans fil, on est confronté au paramètre de portée, c'est-à-dire jusqu'à

quelle distance l'information peut-elle être transportée en bonne état. Des catégories de réseau ont ainsi été créées pour différencier les zones géographiques : PAN, LAN, MAN, WAN. Cela ne concerne pas uniquement les réseaux sans fils puisqu'un réseau Ethernet, très commun dans les entreprises ou les associations, peut être un PAN ou un LAN.



3.2. Communication avec l'ordinateur

Pour établir une communication avec l'ordinateur, il y a deux options : l'assemblage de différents éléments ou le XBee USB Explorer.



La communication en direct sans passer par une Arduino permet de configurer rapidement le XBee.

Les paramètres importants sont :

- **ID (Identification Network)** : Identifiant du réseau personnel. Cet identifiant doit être le même pour les modules XBee qui doivent appartenir au même réseau.
- **CH** pour donner le n° du canal de communication. Cet n° doit être le même pour les modules XBee qui doivent communiquer sur la même fréquence.

- MY pour donner l'adresse source sur 16 bits.
- DH qui donne l'adresse du destinataire.
- BD (Baud Rate) : Vitesse de transmission en bit/s.

3.3. Adressage

Pour tout XBee, il faut impérativement définir l'adresse du réseau ATID, son adresse personnelle ATMY et si besoin, l'adresse de destination des paquets ATDL.

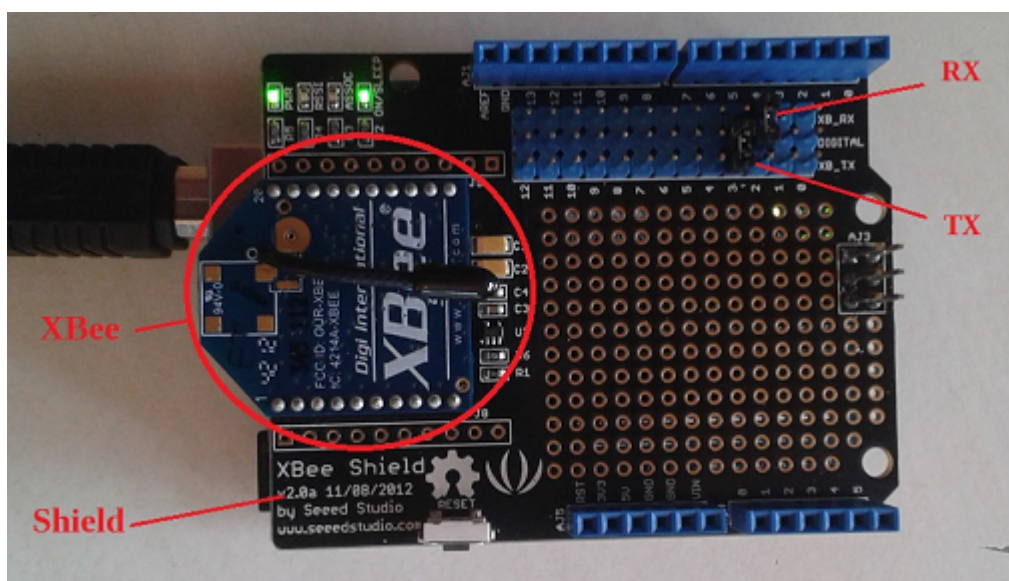


4. Mise en place d'une communication

Ouvrir le dossier « test_communication ».

Utiliser les programmes situés dans les répertoires « xbee_rcv » pour la partie réception et « xbee_snd » pour la partie émission.

1. Définir le Xbee émetteur et le Xbee récepteur.
2. Positionner les cavaliers du shield pour la réception et l'émission des données.



3. Modifier les programmes Arduino pour configurer le n° des broches conformément à celles du shield.

```
const int RX = ...;    // pin Shield Arduino pour reception
const int TX = ...;    // pin Shield Arduino pour transmission
```

4. Configurer les adresses PANID, MYID et DLID afin d'établir un canal de communication entre deux Xbee.

```
const char *PANID = "...";    // adresse reseau
const char *MYID = "...";     // adresse du XBee
const char *DLID = "...";     // adresse de destination
const char *CHID = "...";     // n° de canal
```

5. Configurer le terminal Arduino avec la vitesse utilisée par les programmes et sur "retour chariot".

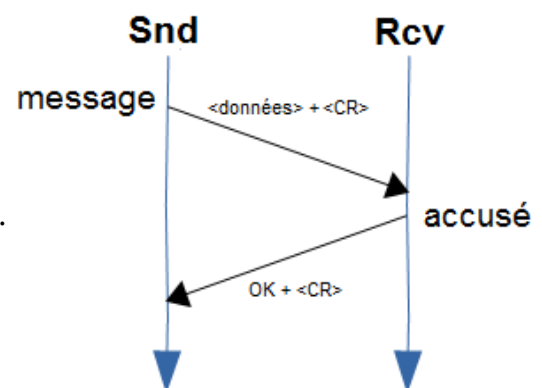


Un Xbee émetteur envoie des caractères reçus au travers du terminal Arduino côté PC vers un Xbee récepteur. Les caractères reçus par le récepteur sont affichés sur un deuxième terminal Arduino.

Un accusé de réception « OK » + <CR> est envoyé si la transmission s'est effectuée sans erreur (cf schéma ci-contre).

En informatique, le retour chariot (en anglais carriage return, CR) est l'un des caractères de contrôle des normes de codage de caractère ASCII qui indique une fin de ligne.

Le code ASCII du retour chariot est 13 (0x0D). Ce caractère non affichable est représenté par le symbole « \r » en langage C.



5. Commande à distance

Ouvrir le dossier « commande_distante ».

Utiliser les programmes situés dans les répertoires « xbee_rcv » pour la partie réception et « xbee_snd » pour la partie émission.

On désire communiquer avec différents équipements :

- deux LED (blanche et rouge) : allumer et éteindre
- un servomoteur : déplacer de 0° à 180°
- un thermomètre : lecture
- un luxmètre : lecture

La syntaxe pour piloter les LEDs blanche et rouge du récepteur est la suivante :

LED [R|B]=[ON|OFF]

- R : Rouge
- B : Blanche
- ON : allumer la LED
- OFF : éteindre la LED

exemple : LED B=ON, allume la LED blanche.

La syntaxe pour piloter le servomoteur est la suivante :

SERV=[0..180] : positionne le servomoteur

SERV : retourne la position du servomoteur

La syntaxe pour connaître la température est la suivante :

TEMP

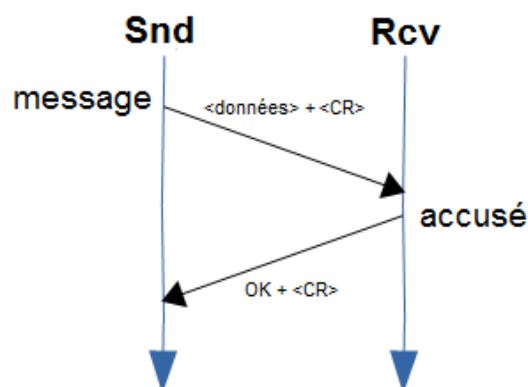
La syntaxe pour connaître la luminosité est la suivante :

LUX

Un Xbee émetteur envoie la commande saisie à travers le terminal Arduino. La commande reçue par le récepteur est traitée si sa syntaxe est correcte.

Un accusé de réception « OK » + <CR> est envoyé si la commande est correcte (cf schéma ci-contre).

En cas d'erreur, le récepteur envoie le message « ERROR » + <CR>.



1. Modifier le programme récepteur pour affecter les broches des différents équipements.
2. Faire un test de communication.
3. Tester les différents équipements.
4. Modifier le programme pour utiliser un télémètre.

La syntaxe à utiliser pour recevoir la distance en cm est : TELE.