

# Note d'application

---

## La mise en veille des modules Xbee Série 2

**Auteur: LAKBAIDI Said**

**Tuteur technique : LAFFONT Jacques**

**Projet : P12AB01**

**Janvier 2012**

## **Introduction :**

Dans cette note d'application nous allons expliquer comment mettre en basse consommation un module de communication ZigBee de type Xbee série 2. Suivant deux méthodes testées, et avant tout on mettra en évidence tous les registres et les paramètres nécessaires pour établir une communication ZigBee.

## I. Introduction aux modules Xbee Serie2

Les modules XBee serie 2 se présentent sous la forme de circuits tels que sur l'image à côté. Le circuit est équipé de sa propre antenne planaire ou filaire ou dispose d'une sortie antenne sur connecteur pour les applications nécessitant le déport de l'antenne externe.



Les modules Xbee fabriqués par Maxstream (Digi) sont conçus pour opérer avec le protocole Zigbee, une communication radio 2.4Ghz.

Ces modules utilisent des versions de firmwares qui peuvent être mis à jour par téléchargement manuel depuis le site de Digi ou à travers le logiciel X-CTU.

Les modules XBee Serie 2, en version normale ont une portée 30 à 100m en terrain découvert. La version PRO à d'une puissance supérieure est destiné à couvrir des distances de 100 à 1600m.

Les deux versions sont totalement compatibles. Et peuvent communiquer entre elles mais le plus judicieux serai d'utiliser la même version et pour une consommation minime il est conseillé d'utiliser les Xbee S2 version normale.



Portée intérieur : jusqu'à 30 m

Portée extérieur champs libre: jusqu'à 100m

Puissance d'émission RF: 1 mW

Courant de repos: < 10  $\mu$ A

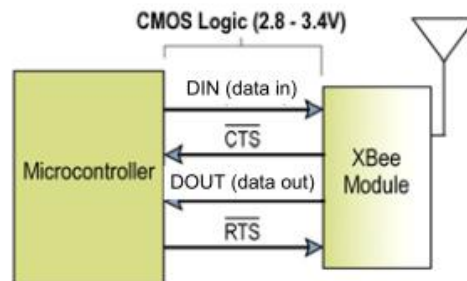
Fréquence de fonctionnement: 2.4 GHz

Vitesse de transmission RF: 250Kbps

Tension d'alimentation entre 2,8 et 3,4V

Il existe deux modes de fonctionnements du modules Xbee d'abord, le mode AT qui utilise les commandes AT pour configurer le module et l'envoi ainsi que la réception se font de façon simple. Le deuxième mode se nomme mode API, nécessite de concevoir les trames à envoyer par l'utilisateur même. Et pour des raisons de simplicité (temps de configuration et taille du programme) nous avons adopté le mode AT pour la suite de cette note d'application.

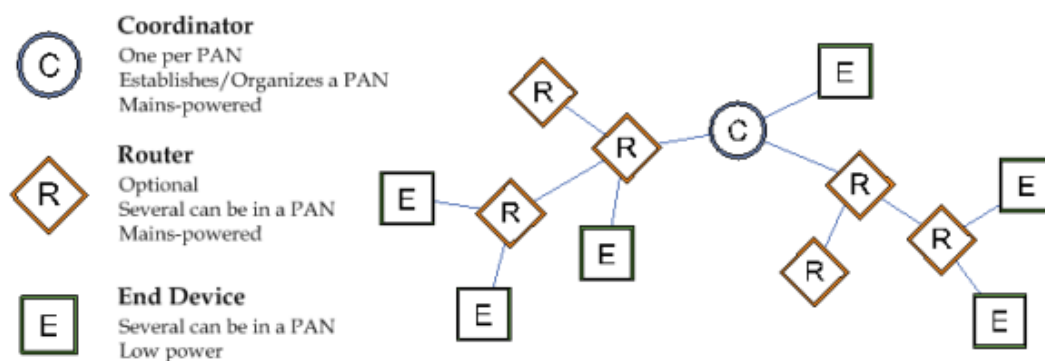
## 1. Configuration de liaison série :



Le module XBee S2 utilise une liaison série asynchrone pour communiquer avec le microcontrôleur. Une liaison qu'on configure (Vitesse, parité, start et stop, contrôle de flux) à l'aide du logiciel X-CTU ou grâce aux commandes AT à travers le terminal de communication.

## 2. Les réseaux ZigBee :

Pour la suite on va adapter nos modules à fin de construire un réseau ZigBee maillé.



Pour cela, nous avons choisi deux modules à qui on assigne respectivement les fonctions suivantes :

**Coordinateur :**

Ce module assure les fonctions tel que l'authentification, l'initiation de la communication, la sécurité et l'ajout des nœuds au réseau ... sur le même réseau on peut avoir qu'un coordinateur. Ce dernier doit être actif en permanence pour répondre à tout moment aux requêtes des autres éléments du réseau. Il est donc alimenté à plein temps.

**Les routeurs :**

Ces modules sont indispensables pour étendre le réseau par acheminements des trames d'un module à un autre. Ils permettent aussi aux autres modules de s'enregistrer sur le même réseau, et non exclusivement chez le coordinateur, le nombre d'éléments du réseau peu atteindre les 65536 modules.

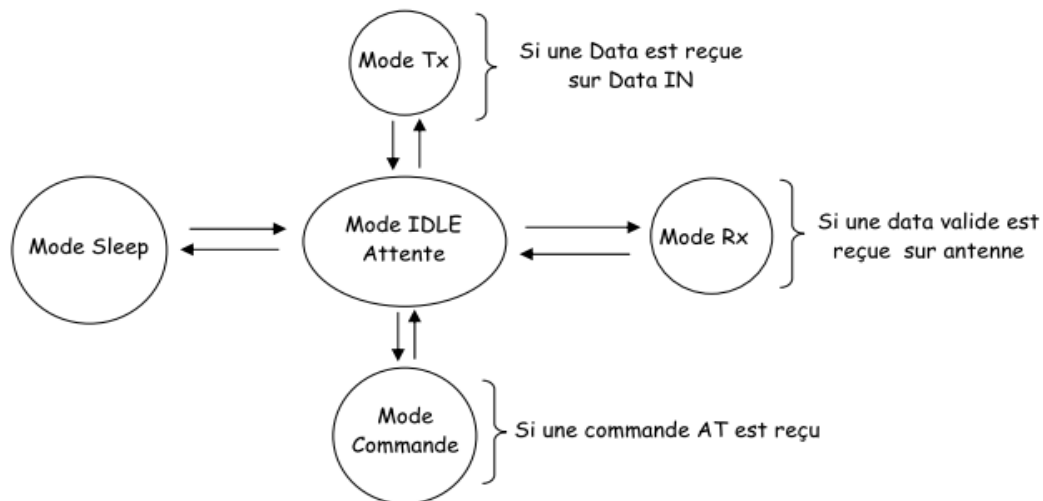
**Les End-Devices :**

C'est des modules qui ont presque les mêmes caractéristiques que les routeurs, sauf la fonction d'acheminement de données, ils offrent la possibilité de mise en veille à fin de gagner en consommation. Cette fonction est surtout utilisée en industrie pour des capteurs, qui livrent des informations par moment.

Une solution logicielle offerte par des Firmwares spéciaux permet de rassembler les fonctionnalités des routeurs et des end-devices en une seule configuration de module Xbee S2 afin de pouvoir gagner d'avantage en consommation sur les routeurs en permettant de les mettre en veille occasionnellement. Mais cela exige une maîtrise totale des temps de veille sur l'ensemble du réseau maillé.

### 3. Les modes de fonctionnements des modules Xbee S2

Tout module xbee préconfiguré fonctionne selon le schéma ci-dessous :



Pour la suite nous allons préciser la signification des paramètres essentiels à configurer afin d'assurer une communication ZigBee basse consommation.

#### 4. Les paramètres à configurer:

Pour faire communiquer deux modules Xbee S2, il est nécessaire de configurer au moins les paramètres suivants:

- **PAN ID** : Représente un identifiant qui va permettre au module de communiquer seulement avec les modules qui se trouvent sur le même canal et qui portent le même PAN ID, ainsi on empêche les interférences avec d'autres modules non concernés par la communication.
- **CH** : Contient le numéro du canal utilisé par les modules pour communiquer, c'est le coordinateur qui fait un balayage des canaux de communication jusqu'à trouver son correspondant (routeur/end-device).
- **SH et SL** : Contiennent les numéros de série usine des modules. Et qui représente aussi leurs propres adresses.
- **DH et DL** : Contiennent les parties hautes et basses des adresses des correspondants. (pour une communication en broadcast il faut mettre DH=0x0000 et DL=0xFFFF)
- **PL** : Power Level, désigne la puissance du signal d'envoi, (0= -7dBm, 1= -3dBm, 2= -1dBm, 3= +1dBm, 4= +3dBm.)
- **PM** : Power Mode, S'il est activé, le mode boost améliore la sensibilité de 1 dB et puissance de sortie augmente de 2 dB.
- **BD** : Spécifie le « baud rate » choisi pour la liaison série.
  - 0 : 1200 bps.
  - 1 : 2400 bps
  - 2 : 4800 bps
  - 3 : 9600 bps

- 4 : 19200 bps
  - 5 : 38400 bps
  - 6 : 57600 bps
  - 7 : 115200 bps
- **D7** : à 1 ça permet l'activation du contrôle de flux sur CTS.

## II. La mode veille:

Pour augmenter la durée de vie des batteries, tout en conservant un réseau ZigBee fonctionnel et réactif, on configure les paramètres suivants :

**SM** : un paramètre qui désigne le mode veille adopté

- **SM = 0 : Désactive le mode veille.**
- **SM 1 : Pin Hebernate**, lorsqu'une tension de 3,3v est fournie à cette la broche 9, le module fini de recevoir ou de transmettre les données encours puis se met en veille et se réveille lorsqu'on applique une tension 0v à cette broche 9. Le module consomme environ 10µA en veille et le réveil prend 13 ms.
- **SM = 2 : Pin Doze**, mise une veille du module de la même façon que le pin hebernate, cette méthode garde le module partiellement éveillé ce qui explique sa consommation légèrement supérieure à 40µA mais avec un temps de réveil plus court (2 ms).
- **SM = 4 : Cyclic sleep**, le module se met en veille régulièrement durant un temps dit SP (Sleep period) et se réveille au minimum pendant une durée ST (Time before sleep), durant ce mode de veille le module consomme environ 50µs et le réveil prend 2 ms. Ce mode est plutôt adapté aux capteurs.

**Les paramètres qui suivent sont surtout dédié au cyclic sleep** (sauf pour le SP sur les modules parents)

- **SP : Sleep cyclic period** pour les routers/end devices, représente durée de veille pour le cyclic sleep, qui prend une valeur en hexadécimal comprise entre 0x20 et 0xAF0 qu'on multiplie par 10 ms pour donner la durée de veille en millisecondes, donc cette durée au final sera comprise entre 320 ms et 28 secondes.

Ce paramètre pour les coordinateurs et les routeurs parents, représente la durée pendant laquelle un parent doit garder le message en mémoire en attendant qu'il soit réclamé par le module en veille. Pour éviter les pertes de données, SP (du parent) ≥ SP du module en veille régulier.

- **ST : Time before sleep**, représente le temps minimal avant de se remettre en veille en absence d'activité (envoi ou réception de données), on l'appelle aussi le temps de silence, sa valeur est comprise entre 0x01 (1 ms) et 0xFFFE (65s)

Ce tableau résume les valeurs des paramètres principaux :

AT	Description	Paramètre	Default
SM	Sleep mode	0, 1, 2 4,	0
SP	Sleep Period	0x20 - 0xAF0	0x20
ST	Time bf Sleep	1 - 0xFFFE	0x1388

## 1. Exemples:

Dans cette partie on va faire la mise en veille d'un module routeur Xbee S2 pendant 99% du temps avec deux méthodes testées. Les modules communiquent en unicast avec un coordinateur qui lui restera toujours alimenté (fonctionnement en mode normal).

On précisera les valeurs hexadécimales des paramètres essentiels qu'on configure à l'aide de X-CTU (dans l'onglet Modem Configuration) ou avec les commandes AT à travers un terminal.

### a) Méthode 1 : Cyclic sleeping :

Configuration des modules :

Paramètre	Coordinateur (valeurs en hexadécimal)	Routeur (valeurs en hexadécimal)
PAN ID	99	99
SH	13A200	13A200
SL	404A6F2B	40341287
DH	13A200	13A200
DL	40341287	404A6F2B
PL	4	0
PM	1	0



<b>BD</b>	<b>2</b>	<b>2</b>
<b>D7</b>	<b>1</b>	<b>1</b>
<b>SM</b>	<b>NA</b>	<b>4</b>
<b>ST</b>	<b>NA</b>	<b>32 (50 ms)</b>
<b>SP</b>	<b>3E8 (10 s)</b>	<b>1FE (4950 ms)</b>
<b>SN</b>	<b>NA</b>	<b>1</b>

Avec cette configuration on met on veille le routeur pendant 4,95 secondes et on le réveille pendant 50 millisecondes, si aucune activité n'a était détectée sur le canal de communication, le module se remet en veille. Si le coordinateur avait un message destiné au routeur, il le garde en mémoire pendant 10 secondes maximum en attendant que le routeur le réclame en fin de sa période de veille. On a activé le contrôle de flux CTS sur les deux modules de façon logicielle, pour éviter l'écrasement de données dans les buffers d'envoi.

Une fois le module est configuré pour se mettre en veille régulièrement, il devient difficile de communiquer à travers une liaison série pour le reconfigurer par exemple, pour remédier à cela il est conseillé d'utiliser le logiciel X-CTU et passer par l'onglet Modem Configuration, qui va nous demander d'appuyer sur le bouton Reset manuellement avant d'exécuter n'importe quelle commande, un moyen pour forcer le modem à se réveiller pour qu'on puisse opérer normalement.

## Méthode 2 : PIN Hebernaté :

### Configuration des modules :

<b>Paramètre</b>	<b>Coordinateur</b> (valeurs en hexadécimal)	<b>Routeur</b> (valeurs en hexadécimal)
<b>PAN ID</b>	<b>99</b>	<b>99</b>
<b>SH</b>	<b>13A200</b>	<b>13A200</b>
<b>SL</b>	<b>404A6F2B</b>	<b>40341287</b>

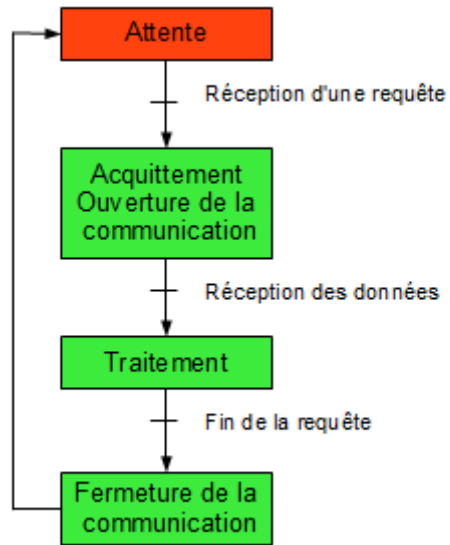
<b>DH</b>	<b>13A200</b>	<b>13A200</b>
<b>DL</b>	<b>40341287</b>	<b>404A6F2B</b>
<b>PL</b>	<b>4</b>	<b>0</b>
<b>PM</b>	<b>1</b>	<b>0</b>
<b>BD</b>	<b>2</b>	<b>2</b>
<b>D7</b>	<b>1</b>	<b>1</b>
<b>SM</b>	<b>NA</b>	<b>1</b>
<b>ST</b>	<b>NA</b>	<b>Paramètres non pris en compte qu'en cyclic sleep</b>
<b>SP</b>	<b>3E8 (10 s)</b>	
<b>SN</b>	<b>NA</b>	<b>1</b>

Avec cette configuration on autorise la mise en veille du module routeur par la mise de la broche 9 à 3,3 Volts. Et le réveil se fait en absence de cette tension sur la broche 9.

Donc la gestion des temps de veille et réveil se fait dans les programmes associés aux systèmes qui sollicitent ces modules (Micro contrôleurs...). Pour avoir un fonctionnement à 99% du temps en veille on prend les mêmes valeurs de veille que le mode précédant, qui sont de 4950 millisecondes en veille et 50 millisecondes de réveil.

On évite de mettre un temps de silence (ST time before sleep) inférieur à 50 millisecondes pour laisser le temps au module routeur de se mettre en route et demander au module parent s'il a des données en mémoire qui lui sont destinées.

Pour une sûreté de communication, on évite l'envoi des données critiques directement sur les buffers d'envoi du modem parent (coordinateur dans ce cas), donc on envoie une requête d'ouverture de port de communication de façon répétitive en attendant le réveil du routeur. Le schéma suivant explique notre façon de faire.



*Protocole de communication implémenté*