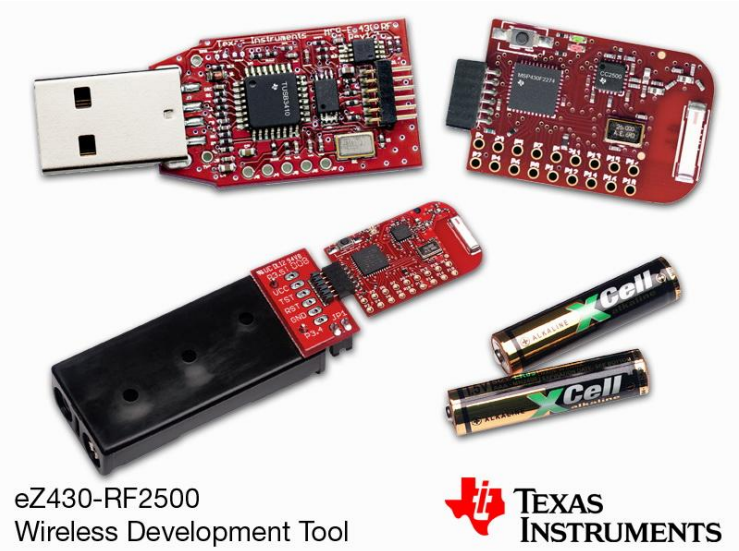
Projet Spécialité



Sensor network

Élèves : HUANG yongkan & KANJ mohamad

Prof : Frank Rousseau & Olivier Alphand

Date : 10-06-2011

Sommaire

Figure

1 Objectif

Le but de développer un protocole multi-sauts économe en énergie très simplifié et le but final c'est de former un réseau interopérable avec tous les capteurs de tous les groupes.

2 Plate-forme

2.1 Matériel

Le projet se déroule sur les kits de développement Texas Instruments eZ430-rf2500. Ces kits contiennent :

- 2 capteurs équipés d'un micro contrôleur basse consommation TI MSP430 et d'une radio CC2500 opérant dans la bande des 2.4Ghz
- Une interface de développement USB
- Un conteneur de piles pour alimenter un capteur de manière autonome

2.2 Logiciels

- Linux (ou n'importe quel UNIX like) .
- Compilateur MSPGCC .
- Utilitaire MSPDEBUG.
- Bibliothèques MRFI (Minimal RF Interface) et BSP (Board Support Package) de TI.
- Gestionnaire de versions GIT.

3 Synchronisation

3.1 Timer pour synchronisation

pour la synchronisation , nous avons utilisé deux timers TimerA et TimerB

TimerA:

- 1, surveille le réseau,si le créateur du reseau tombe en panne donc après un certaine temps , le capteur va tiré une valeur aléatoire puis il va créer son propre réseau .

- 2, vérifier si il a toujours reçu des beacons des voisins, si dans 3 secondes il n'a pas reçu le beacon d'un voisin , il va le supprimer avec les destination qui sont associé à ce voisin comme next_hop de sa table de routage.
- 3, chaque 3s , nous mettons un variable SIP_Prepared = 1, dans l'etat de wait_message , si le SIP_Prepared == 1,une packet de rip va être envoyé et apres on met SIP_Prepared = 0.

TimerB:

- Synchronisation et tous les changements des états
- un automate pour le duty cycle

3.2 Format des paquets

Pour les formats des paquets , nous avons défini 3 type de trame qui sont identifié par le flag.

3.2.1 Paquet de beacon

Frame de Beacon

1B	4B	4B	1B	1B	1B	4B
Len	Src	Dst	Flag	ID_ Reseau	ID_ slot	Voisin

Figure 1- frame de beacon

3.2.2 Paquet de RIP

Frame de RIP

1B	4B	4B	1B	1B	1B	1B	
Len	Src	Dst	Flag	Dst	Next_ hop	Metric	……

Figure 2- frame de RIP

3.2.3 Paquet de data

Frame de Data

Figure 3- frame de data

3.3 Automate de synchronisation

Au début , tout les capteurs commencent à l'état de WAIT_SCAN, si le capteur détecte un réseau , il va le rejoindre , si non il va créer son propre réseau avec son MAC comme le ID_Network.

3.3.1 Pour le créateur

Pour le créateur du reseau,il a 4 état :WAIT_SCAN, WAIT_SYNCHRONE , WAIT_MESSAGE, WAIT_SLEEP.

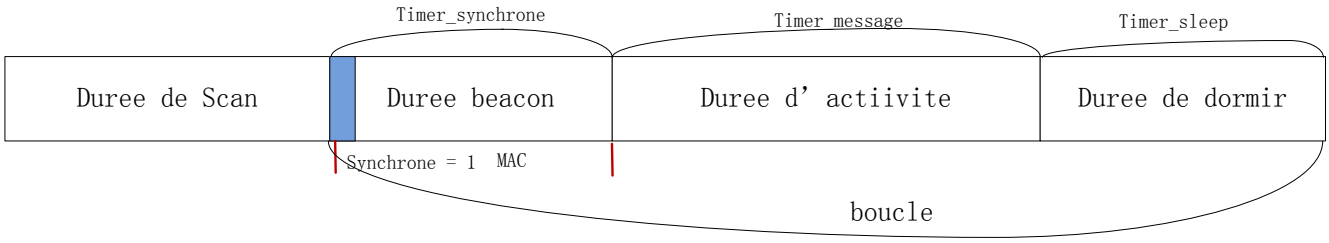


Figure 4- scenario pour le cas 1

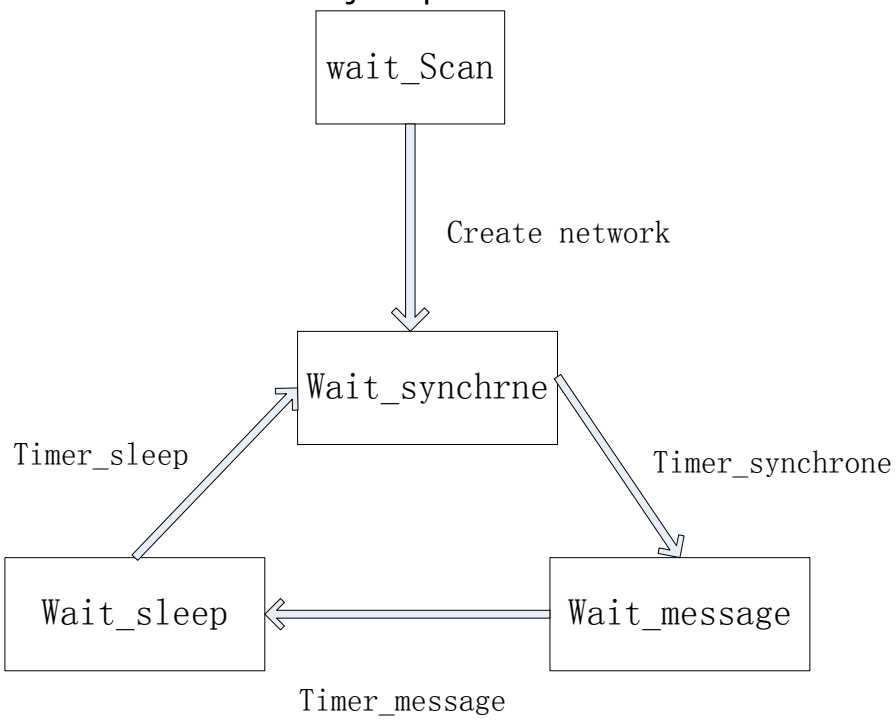


Figure 5- automate pour créateur

3.3.2 Pour le capteur qui a MAC > ID_slot recu

Pour le capteur normal(qui n'est pas le créateur du reseau), il a 5 état:WAIT_SCAN, WAIT_BEACON, WAIT_SYNCHRONE , WAIT_MESSAGE, WAIT_SLEEP.

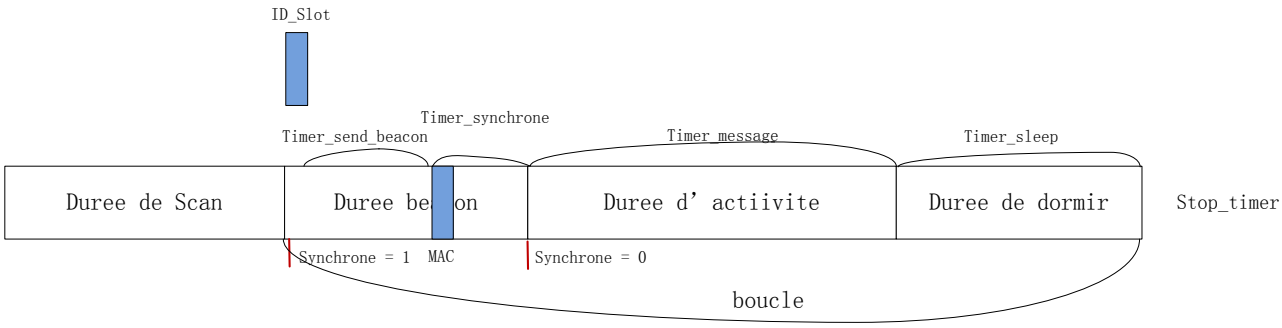


Figure 6- scenario pour le cas 2

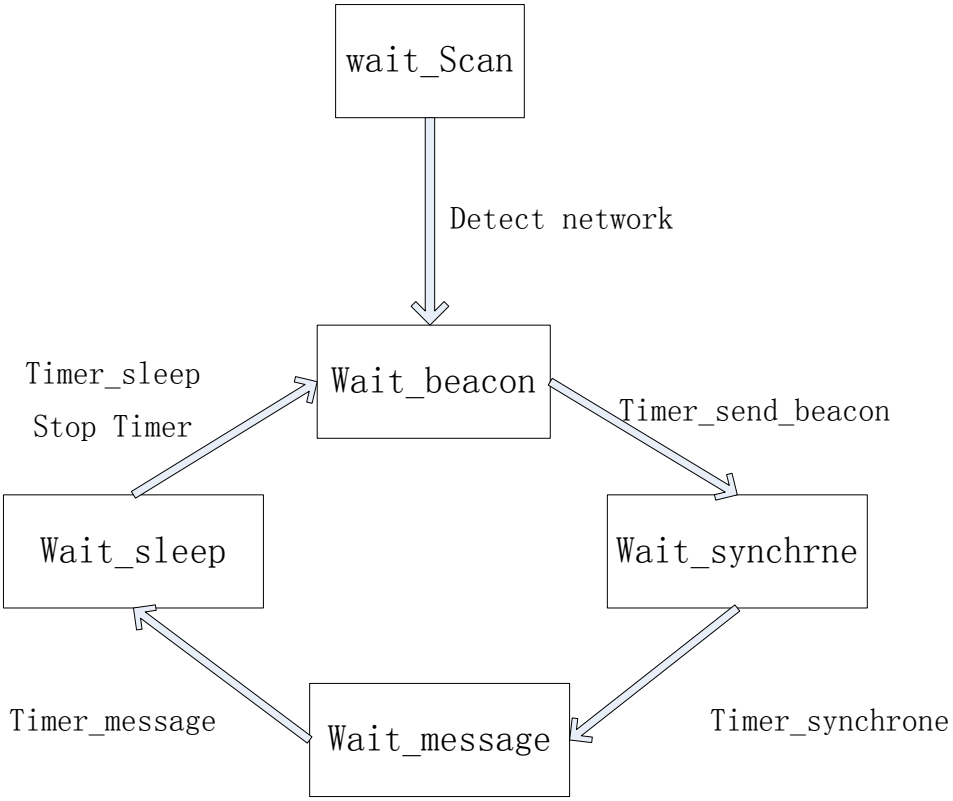


Figure 7 -automate pour le cas 2

3.3.3 Pour le capteur qui a MAC < ID_slot recu

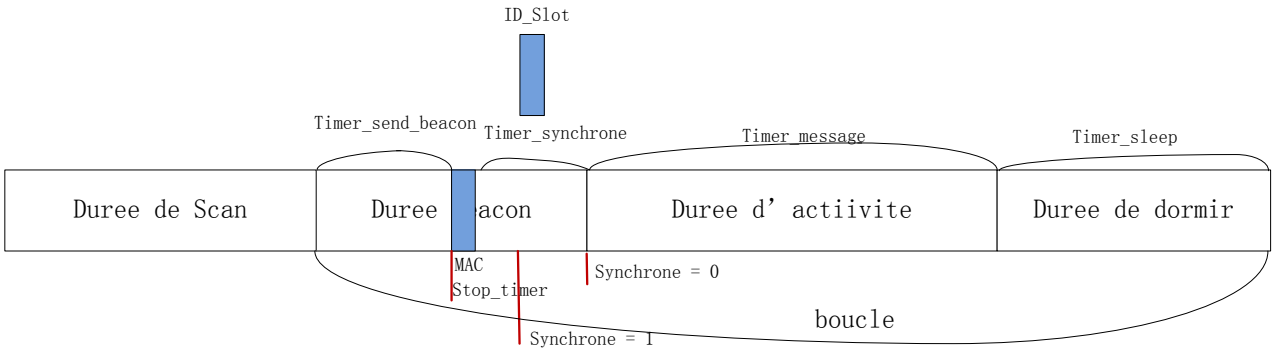


Figure 8- scenario pour le cas 3

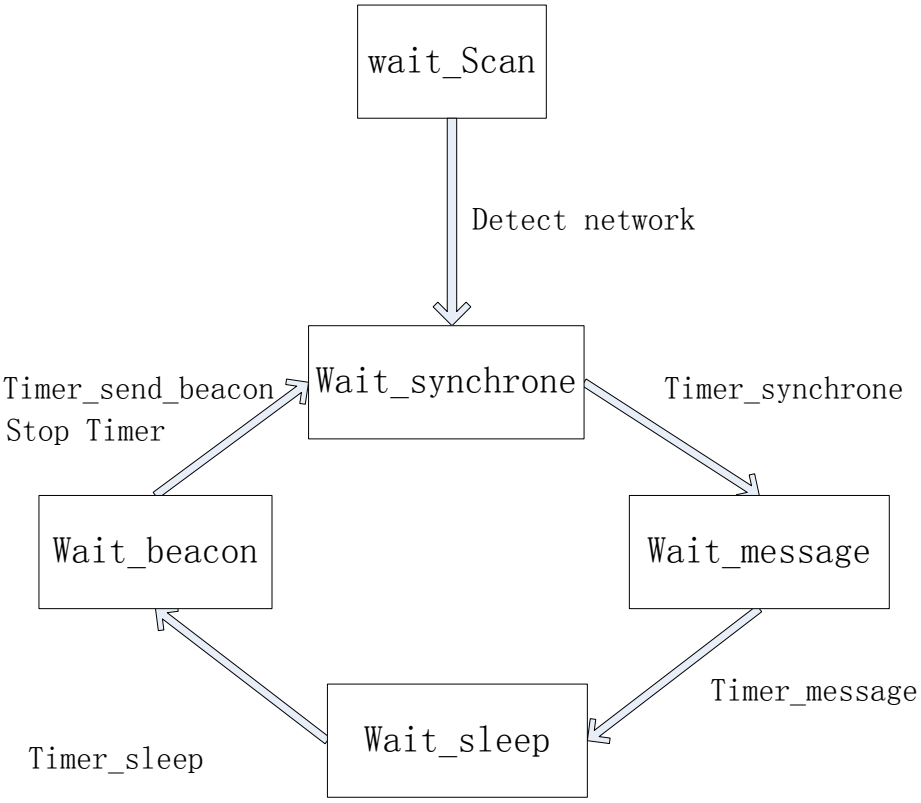


Figure 9- automate pour le cas 3

4 Routage

4.1 Voisinage

on a utilisé 32 bits pour indiquer le voisinage, on met le bit Neme bit a 1 selon le Mac du voisin, cette manière on peut échanger les 32bit voisin et établir le table de routage au départ pour un réseau simple.

pour décider si un capteur est voisin ou pas , on a pris la métrique dans le paquet et lire le RSSI , on prend un seuil pour le voisin .

4.2 Protocol RIP

Pour le réseau plus compliqué , on a utilisé le Protocol RIP pour envoyer le table de routage chaque 3s pour que tous ses voisins puissent partager leur table de routage.

DST	Next_hop	Metric
07	07	0
12	12	1
...		

Figure 10- format de RIP

4.3 Mise à jour le table de routage

- Si un capteur est perdu(tombe en panne ou déplacé a une autre region du reseau) , son voisin va détecté qu'il est perdu grâce au Timer A, donc il va supprimer se voisin et les destination qui sont associé à ce voisin comme next_hop de sa table de routage.
- Quand on a reçu un paquet de RIP , on va mettre à jour le table de routage. si le capteur reçoit un paquet de RIP de son voisin ,si une destination existe chez son voisin mais n'existe chez lui , donc il va vérifier si le next_hop dans le paquet de RIP est lui même , si oui, il ne va pas ajouter , si non il va l'ajouter .en revanche ,si une destination existe chez lui mais n'existe pas chez son voisin , il va vérifier le next_hop aussi.

5 Communication

5.1 FIFO

pour le communication

- On a un FIFO pour envoyer et recoivre les message.
- Dans l'etat de wait message , on va appeler la fonction Send_message et Recieve_message.

Send_message:

écrire le message dans FIFO_Send avec la destination, si il a détecté le taille de FIFO est supérieur que la mrif_max_frame il va couper et envoyer le message avec longueur mrif_max_frame , si il a détecté '\r' , c'est-a-dire il y a déjà une phrase complet , il va envoyer le phrase. si non ,il va attendre jusqu'à des condition.

Recieve_message:

lire le message dans le FIFO_Recieve et vérifier s'il y a un '\r' dans le FIFO , si oui ,il va imprimer le message , si non il va attendre jusqu'à le '\r'.

2, lorsqu'on tape un caractère , il va être stocké dans le FIFO_Send , quand on fini une phrase on type '\r' (enter) donc il va être envoyé.

5.2 UART

Pour la communication , on a utilisé le UART comme une interface d'interaction avec le capteur.

le menu de manipulation

command:

o: who is on line

v: voisin

r: router table

i: sysinfo

ESC: help

pour chaque commande on a traité le cas , si on tape 'o' il va nous affiché les capteurs qu'on puisse communiquer avec eux,puis on choisit avec qui on veut communiquer , pour quitter la conversation on tape ESC,si on tape 'v' il va nous affiché et la liste des voisins,si on tape 'r' il va nous affiché la table de routage et si on tape 'i' il va affiché les information du reseaux et du capteur.

5.3 Multi-saut

Pour communiquer avec un capteur à qui on ne peut pas envoyer des message directement , donc il faut cherche la route dans le table de routage.

on a pu faire fonctionner le multi-saut.

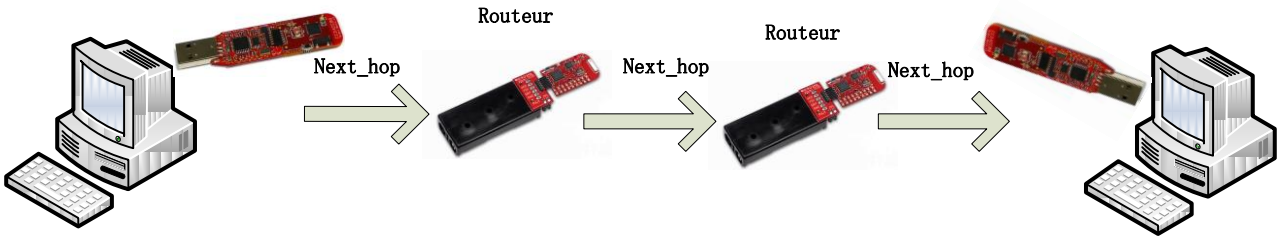


Figure 11- multi-saut

6 Etape realisée

- Synchronisation.
- Routage des paquets.
- Partage de la table de routage.
- Communication et envoi des messages entre les capteurs.
- Changement des positions des capteurs .
- Mis a jour de la table de routage dans le cas du deplacement d'un capteur,ou si un capteur tombe en panne.
- Maintenance du reseau meme si le createur du reseau tombe en panne, si le créateur du réseau est perdu , tous les capteur vont attendre un certaine temps aléatoire et créer son propre réseau , le capteur le plus vite devient le créateur du nouveau réseau.
- Collisionnement de deux réseaux , le capteur qui a le MAC le plus petits va etre le createur du reseau.

7 Problème

7.1 Choix de solution

Pour les choix des solution ,à cause de mauvais solution , on a perdu beaucoup de temps.

- Solution de synchronisation , au début , on a choisi une solution de synchroniser une seul fois , mais pas resynchroniser chaque cycle , mais ça marche pas bien.
- Solution de fifo , au début on a utilisé le FIFO de chaine, mais on a trouvé ça marche pas bien sur MSP430 , donc on a changé à utilisé le tableau comme le FIFO .

7.2 Programmation

- CCA et Timer A, quand on transmet les message avec csma/ca, donc le timerA va être éteint , donc on est obligé de remettre le timer A chaque fois.
- LPM3 , on a pas pu de entrer dans le mode LPM3 (low puissance mode 3) , il nous a fait beaucoup de problèmes. Si on entre dans LPM3, a cause de la décalage du temps , si il est dans le sleep mais il reçoit un paquet , dans ce cas la il va rater le beacon.

7.3 A réaliser

Pour économiser l'énergie, donc on a proposé une solution pour cela , après la durée de message , on va entrer dans le mode LPM3 , après certaine de temps ; on va réveille le timer B , de cette manière on va pas rater le beacon ,mais ça va augmenter la complexité du programme.

8 Remerciment

Ce sujet nous a enrichis beaucoup, malgre qu'on a eu beaucoup des problèmes ,mais on a réussi de les résoudre , on est content de voir la communication multi-saut entre les capteurs , et on a appris beaucoup de choses dans ce projet. Merci à vous , Monsieur Frank Rousseau et Monsieur Olivier Alphand !

Bonne vacance !