REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur
Et de la Recherche Scientifique

Université Dr. YAHIA FARES de *Médéa*

Faculté des Sciences et de la Technologie
Département du Génie électrique et de l'Informatique

Mémoire de Projet de Fin d'Etude pour
L'Obtention du Diplôme de Master (LMD)
En Génie Electrique
Option: Systèmes de Télécommunications

Thème :

Développement d'une plate forme de réseau de capteur sans fil (WSN) pour applications sportives

Proposé et dirigé par :

Mr : CHADLI Samir .

Réalisé par :

Mr : OULD BABA ALi Essam
Mr : BAKIR TErki Othmane

Promos : 2012 - 2013

ملخص

هذا العمل يتمثل في تجسيد منصة لشبكة الاستشعارات اللاسلكية القائمة على أداة EZ430RF2500 من تكساس إنسترومنتس

هذا التطبيق يتكون من ثلاث نهايات عقد استشعار، اثنان منهما موصولين مع أجهزة قياس التسارع التناظرية، والعقدة الثالثة موصولة ببوصلة الكترونية متصلة ب I2C

المحطة الأساسية متصلة بجهاز كمبيوتر لاسترداد بيانات القياس من العقد الاستشعار الثلاث

ويهدف هذا التطبيق لمجال الرياضة من أجل الدراسة، والتحليل الكمي للحركة الفنية للرياضي.

الكلمات المفتاحية:

لشبكة الاستشعارات اللاسلكية، الدارة المجملة المدمجة ، أداة، أدوات الاستشعار.

Résumé :

Ce travail présente la mise au point d'une plateforme d'un réseau de capteur sans fil à base d'un Kit WSN l'EZ430-RF2500 de Texas Instruments. L'application réalisée est constituée de 3 nœuds de capteurs dont deux d'entre eux sont interfacés à des accéléromètres analogiques et le troisième nœud est relié à un compas électronique via une liaison I2C. Une station de base reliée à un PC récupère les données de mesures des 3 nœuds de capteurs.

Cette application est destinée pour le domaine du sport pour l'étude, l'analyse et la quantification du geste technique du sportif.

Mot clés :

Réseau de capteur sans fil, circuit inter-intégré, Kit, Capteurs,

Abstract:

This work presents the development of a platform for a wireless sensor network based on a WSN Kit the EZ430-RF2500 Texas Instruments. The application is made up of 3 sensor nodes two of them are interfaced to analog accelerometers and the third node is connected to an electronic compass via I2C connection. A base station connected to a PC recovers the measurement data from the sensor nodes 3.

This application is intended for the field of sport for the study, analysis and quantification of the technical gesture sports.

Key word :

Wireless sensor network, Inter - Integrated Circuit, Kit, sensors.

Remerciements

Louange à notre Seigneur « ALLAH » qui nous a dotés de la merveilleuse faculté de raisonnement. Louange à notre créateur qui nous a incités à acquérir le savoir.

Au terme de ce modeste travail, nous voudrions adresser nos vifs remerciements à nos professeurs de l'université de Dr.yahia Fares de Médéa qui ont contribué à notre formation.

Nous exprimons notre profonde gratitude à notre encadreur M. CHADLI Samir, nous le remercions pour son aide et ces précieux conseils.

Nos remerciements s'adressent également aux membres des jurys pour l'honneur qu'ils nous font d'assister à notre soutenance.

Pour finir nous remercions tous ceux qui ont contribué de prés ou de loin à l'aboutissement de ce travail.

Othmane ET Essam

DEDICACES

A mes très chers parents pour leur soutien,

A mes grands parents

A mes frère « Mohamed, Adbdelhak et Walid », et tous mes cousins, et toute ma famille
Et tous les Professeurs, Docteurs et Encadreurs de l'université YAHIA FARES

A mes chers amis « Mohamed, Hamza, Nassiba et Hadjer » et tous mes amis de promo de cette année
A mon collaborateur, ami et frère « Othmane »

Essam

A tous ceux qui nous aiment, nous ont encouragée et ont cru en nous !!

DEDICACES

A mes parents que dieu les garde

A ma chère femme et mon cher fils Mohamed Cherif

A mes sœurs et frères et leurs enfants

Aux membres de ma promotion

A tous ceux qui m'ont soutenue de près ou de loin

Othmane

Sommaire

Sommaire

Liste des figures

Liste des tableaux

Introduction Générale ……. 01

Chapitre I : Généralités

I.1. Introduction...…….…… 03

I.2. Les capteurs ……... 03

I.2.1. Définition ……... 03

I.2.2. Les familles de capteurs ……... 04

I.2.2.1. Le capteur Tout Ou Rien (TOR) …… 04

I.2.2.2. Le capteur analogique …… 04

I.2.2.3. Le capteur numérique …… 05

I.2.3. Type des capteurs …….. 06

I.2.4 Les caractéristiques des capteurs ……. 06

I.2.5. Les capteurs intelligents …… 07

I.2.6. Les micro capteurs ……... 07

I.3. Les réseaux …… 08

I.4. Les réseaux de capteurs sans fil ……... 09

I.5. Architecture des réseaux de capteurs sans fil …… 10

I.6. Classification des RCSF …….. 11

I.7. Applications des RCSF (WSN) …… 12

I.7.1. Applications militaires …….. 12

I.7.2. Applications dans le domaine de sécurité ……. 13

I.7.3.Applications environnementales …… 13

I.7.4. Applications médicales …….. 14

I.7.5. Applications commerciales ……. 15

I.8. Contraintes …… 15
I.9. Les différentes problématiques dans les réseaux de capteurs ……... 17
I.10. Les Standards dans les WSNs …… 18

Chapitre II : Conception du système de mesure

II.1. Introduction…….……..... 19
II.2. Cahier de charge……..……... 19
II.3. La solution adoptée…….……. 20
II.4. Choix du kit WSN……...……. 21
II.5. Présentation du kit eZ430 RF2500……..……... 22
II.5.1. Contenu de kit eZ430-RF2500……...……... 23
II.5.2. Caractéristiques des composants de bases…… 25
II.6. Le protocole SimpliciTI…….. 30
II.7. L'outil de développement utilisé……... 31
II.8. Schéma synoptique du réseau de capteurs élaboré……... 31
II.8.1. Le module ADC10……. 32
II.8.1.1. Les entrées analogiques……. 33
II.8.1.2. Le convertisseur A/N……. 34
II.8.2- Le module USCI_B0…….. 34
II.8.2.1. Présentation……... 34
II.8.2.2. Caractéristiques électrique ……... 39
II.8.2.3. Les bases du protocole …… 39
II.8.2.4. L'adressage ……... 40
II.8.2.5. Trames de Communications ……... 41
II.8.2.6. La communication par octet …… 42
II.8.2.7. Les registres de contrôle de la communication I2C……. 42
II.8.3- Le programme du nœud station de base (point d'Accès)…… 44
II.9. Conclusion…… 45

Chapitre III : Mise en marche et Tests

III.1- Introduction…….. 46

III.2- La simulation…… 46

III.3- Les tests pratiques …….. 50

III.3.1- Les accéléromètres utilisés …… 50

III.3.1.1 Fonctionnement de l'accéléromètre…… 50

III.3.2. Le compas électronique …… 53

III.4- L'interface de visualisation sur PC …… 53

III.5- Conclusion …… 56

Conclusion Générale …….. 57

Annexe

Bibliographie

Glossaire

Liste des figures

Figure I-1: Le principe d'un capteur…….. 5
Figure I-2: Fonctionnement d'un capteur TOR…….. 6
Figure I-3: Fonctionnement d'un capteur analogique…… 7
Figure I-4: Fonctionnement d'un capteur numérique……..... 7
Figure I-5: Architecture générale d'un capteur intelligent…….…… 9
Figure I-6: Classifications des Réseaux de communications…… 10
Figure I-7: Les échanges dans les modes Infrastructure et Ad-hoc ……. 11
Figure I-8: Exemple d'une architecture d'un réseau de capteurs sans fil.…… 12
Figure I-9: Schéma de communication dans un WSN…….. 13
Figure I-10: Exemple d'une Applications militaires.……..... 14
Figure I-11: Exemples d'Applications environnementales……... 16
Figure I-12: Exemple d'une Applications médicales…… 16
Figure II-1: Schéma de principe du système à concevoir ……. 18
Figure II-2: Photos de quelques kits pour réseaux de capteurs sans fil …… 19
Figure II-3. Le kit eZ430-RF2500 de Texas Instruments…….. 21
Figure II-4. Les entrées/sorties du support batterie et du nœud eZ430-RF2500……... 21
Figure II-5. La consommation du courant pour chaque mode…….. 23
Figure II-6. Brochage du MSP430x22x4……... 23
Figure II-7. Schéma fonctionnel de MSP430x22x4 …… 24
Figure II-8. Schémas électriques de l'eZ430-RF2500……... 25
Figure II-9. Les couches logiques du protocole SimpliciTI.……..... 26
Figure II-10. Shémas Synoptique du WSN de notre application…….. 27
Figure II-11. Schéma fonctionnel d'ADC10……..... 29
Figure II-12. Le convertisseur A/N……..... 30
Figure II-13. Les registres de contrôle de l'ADC10……..... 30
Figure II-14. Le registre ADC10CTL0……..... 30
Figure II-15. Le registre ADC10CTL1 …….. 31

Figure II-16. Algorithme de L'End Device 1 et 2……. 33
Figure II-17. La communication sur le bus I2C…… 34
Figure II-18. L'état de repos du bus I2C…… 35
Figure II-19 La condition de départ du bus I2C…… 35
Figure II-20 La condition d'arret du bus I2C……. 36
Figure II-21 L'acquitement sur le bus I2C……. 36
Figure II-22. L'adressage sur le bus I2C…… 36
Figure II-23. Structure d'une trame de communication……. 37
Figure II-24. Écriture de donnés dans une mémoire EEPROM I2C……. 38
Figure II-25. Les registres de controle de la communication I2C……. 38
Figure II-26. L'organigramme du programme du noeud 3…… 39
Figure II-27 Organigramme du programme de l'Access Point……. 40
Figure III-1. Programme de conversion d'un canal analogique avec MSP430F2272……... 43
Figure III-2. Modification du format de fichier de sortie IAR…….. 44
Figure III-3. Configuration des options de Débougage……. 44
Figure III-4. Numérisation d'un signal continue d'amplitude 0.8 V…….. 45
Figure III-5. Numérisation d'un signal continue d'amplitude 1.06 V…… 45
Figure III-6. L'accéléromètre MMA7361 monté sur un PCB & brochage du capteur……. 47
Figure III-7. Schéma fonctionnel simplifié d'accéléromètre……. 48
Figure III-8. les modes de selections……. 49
Figure III-9. L'interface de visualisation des courbes sous Matlab……... 49
Figure III-10. Lorganigramme de l'interface de visualisation ……..... 50
Figure III-11. Le code Matlab de l'interface de visualisation…… 51

Liste des tableaux

Tableau I-1. Les différents types de capteurs……... 8

Tableau II-1. Les entrées/sorties du support batterie…… 21

Tableau II-2. Les entrées/sorties du connecteur d'un Noeud eZ430-RF2500….. 22

Tableau III-1. Sélection de g……. 48

Introduction générale

Introduction Générale

Depuis quelques décennies, le besoin d'observer et de contrôler des phénomènes physiques tels que la température, la pression ou encore la luminosité est devenue une tache essentielle pour de nombreuses applications industrielles et scientifiques, et pourrait considérablement améliorer la qualité de notre vie.

Les progrès réalisés ces dernières années dans les domaines de la microélectronique, de la micromécanique, et des technologies de communication sans fil, ont permis de produire avec un coût raisonnable des composants de quelques millimètres cubes de volume. Ces derniers, appelés micro capteurs, intègrent : une unité de mesure chargée de capter des grandeurs physiques (chaleur, humidité, vibrations), une unité de traitement informatique et de stockage de données et un module de transmission sans fil dans le but de transformer ces grandeurs physiques en grandeurs numériques.

De ce fait, les micro capteurs associés a ces systèmes sont devenues de véritables systèmes embarqués de mesures. Le déploiement de plusieurs d'entre eux, en vue de collecter et transmettre des données environnementales vers un ou plusieurs points de collecte, d'une manière autonome, forme des réseaux de capteurs sans fil ou WSN : Wireless Sensor Networks .

Cette solution est très intéressante car il y a beaucoup d'applications où il est nécessaire de réaliser des mesures dans un endroit, où l'utilisation de câble est problématique. En effet, protéger les câbles en les plaçant dans des gaines ou en les enterrant représente une surcharge de travail et demande beaucoup de temps. C'est parfois difficilement réalisable. Certaines réglementations liées à la protection contre le feu peuvent écarter l'utilisation de certains types de câble à l'intérieur des bâtiments. Dans certaines applications, il est nécessaire d'effectuer des mesures sur de grande distance, ce qui diminue la qualité de la mesure en raison de la longueur des câbles et représente un surcoût non négligeable. Lorsque la centrale de mesure n'a pas assez de voies disponibles pour connecter de nouveaux capteurs, les capteurs sans fil sont une alternative afin d'augmenter le nombre de capteurs de mesure.

Dans ce travail nous présentons le développement d'une plate forme de réseau de capteur sans fil à base d'un Kit WSN de Texas Instrument. Cette plate forme est destinée pour réaliser des mesures dans le monde sportif, elle est de très faible dimensions, d'un poids très réduit, d'une portée de quelques dizaines de mètres fonctionnant sur des batteries.

Ce polycopie est organisé de la manière suivante:

- Dans le premier chapitre nous commençons par la présentation générale des concepts de capteur puis nous passons aux réseaux de capteurs sans fil (WSN).

- Dans le deuxième chapitre nous présentons la conception de la plate forme et les choix réalisés pour aboutir aux objectifs fixés.

- Le troisième chapitre sera consacré à la mise en marche du système et aux tests réalisés

- Enfin nous terminons par une conclusion.

Chapitre I : Généralités

I.1.Introduction [7] *:*

En électronique, dans les applications de mesures, de contrôle et de commande, les capteurs sont des éléments de bases permettant de quantifier des grandeurs physiques diverses afin de fournir une information sur la variation de ces derniers en fonction du temps ou bien pour assurer une fonction de gestion, d'alerte, de contrôle ou autre en fonction de l'évolution de ces grandeurs.

I.2.Les capteurs [6] *:*

I.2.1. Définition :

Un capteur est un dispositif qui transforme la variation d'une grandeur physique observée en une variation d'une grandeur électrique exploitable, (tension, courant, charge électrique, ….etc). Souvent on fait (à tort) confusion entre capteur et transducteur : le capteur est au minimum constitué d'un transducteur.

Le capteur se distingue de l'instrument de mesure par le fait qu'il ne s'agit que d'une simple interface entre un processus physique et une information manipulable. Par opposition, l'instrument de mesure est un appareil autonome. Il dispose donc d'un outil d'affichage ou d'un système de stockage de données. Ce qui n'est pas forcément le cas du capteur. Les capteurs sont les éléments de base des systèmes d'acquisition de données, leur mise en œuvre est du domaine de l'instrumentation.

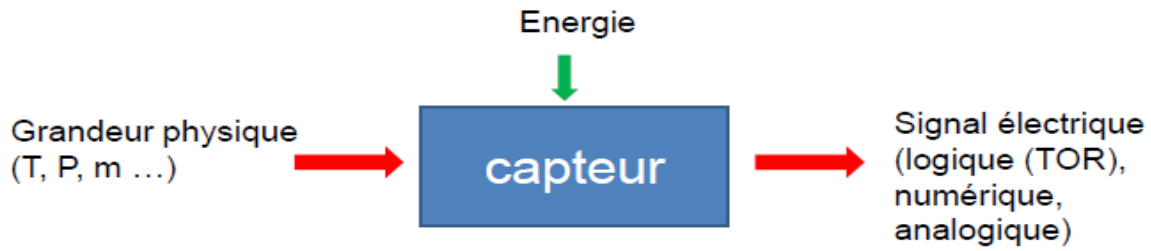


Figure I-1: Le principe d'un capteur

Généralement, on obtient une grandeur de sortie de type électrique, elle peut être soit :

- une charge électrique,
- une tension,
- un courant,
- une impédance (R, L, C); ....

I.2.2. Les familles de capteurs [8]:

Il existe 3 grandes familles de capteurs :

- Les capteurs Tout Ou Rien (TOR),
- Les capteurs analogiques,
- Les capteurs numériques.

I.2.2.1. Le capteur Tout Ou Rien (TOR) :

Ce type de capteur permet de détecter un événement ou un objet lié au fonctionnement d'un système donné. Le signal électrique en sortie de ce capteur est de type logique (signal acceptant deux niveaux : niveau logique 0 (NL0) ou niveau logique 1 (NL1)).

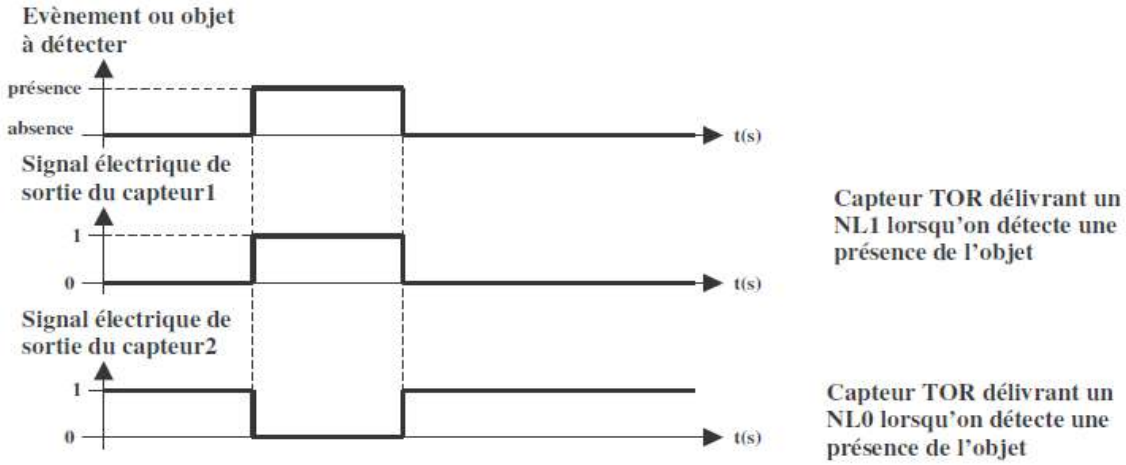


Figure I-2: Fonctionnement d'un capteur TOR

La contrainte de choix la plus importante de ce type de capteur est la vitesse d'apparition ou de disparition de l'objet à détecter.

I.2.2.2. Le capteur analogique :

Une variation de la grandeur physique d'entrée du capteur produit une variation de la caractéristique électrique du capteur (courant, tension, fréquence, valeur moyenne, rapport cyclique,…). La zone de variation définie l'étendue de mesure du capteur qui représente la plage ou le capteur fonctione et fournie une information sur la variation de la grandeur physique.

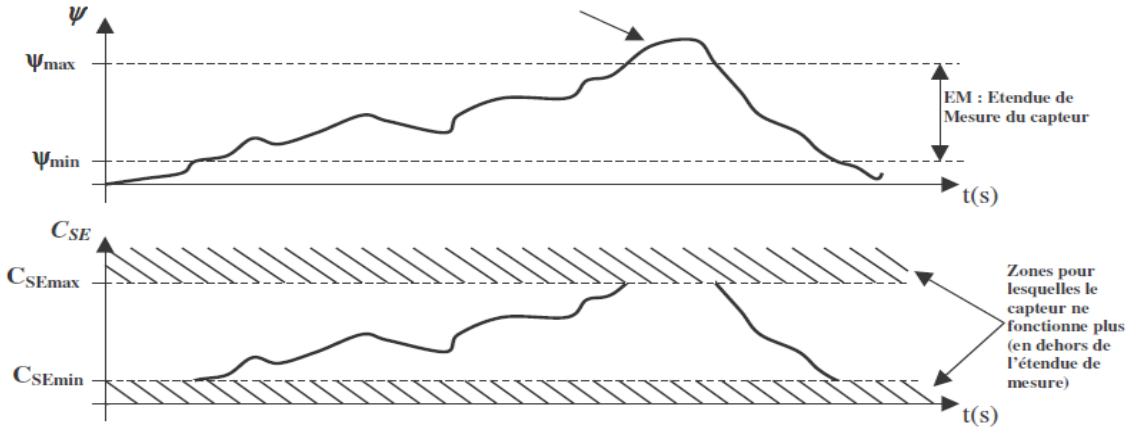
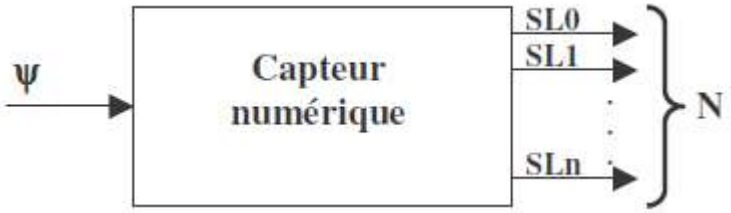


Figure I-3: Fonctionnement d'un capteur analogique

I.2.2.3. Le capteur numérique :

Ce type de capteur produit un nombre binaire N (suite de 0 et de 1) qui dépend directement de la grandeur physique à capter.



SL0 : signal logique 0 (poids faible);
SLn : signal logique n (poids fort);
N : nombre binaire (n signaux logiques).

Figure I-4: Fonctionnement d'un capteur numérique

Son avantage est que l'on peut directement le connecter à un microcontrôleur qui pourra exploiter le nombre N. Il n'y a pas de mise en forme analogique et/ou logicielle. Son inconvénient est qu'en sortie plusieurs fils sont disponibles (en fonction du nombre de bits du nombre binaire N), donc le nombre d'entrées du microcontrôleur devra être important. En général la documentation constructeur donne une relation liant N à ψ de la forme : **N = K ψ** ; K : constante dépendant de n (nombre de bits), de ψ_{max} et ψ_{min}. Cette constante est donnée par la documentation constructeur du capteur. Plus n (nombre de bits de N) est grand et plus la précision sera importante.

Les critères de choix les plus importants de ce type de capteur dépendent de l'exploitation du capteur dans le système de mesure selon :

- ***EM :*** l'Etendue de Mesure : qu'elle est la plage de mesures que l'on souhaite réaliser dans le système conçu ?
- ***Précision :*** de quelle précision a-t-on besoin dans le cadre du système technique ? Cette précision dépend de l'élaboration du cahier des charges..

I.2.3. Type des capteurs :

Il existe une grande variéte de capteurs en instrumentation, le tableau suivant présente certains types de capteurs utilisés en pratique :

Les capteurs de lumière	➢ La Photorésistance ➢ La Photodiode ➢ Le Phototransistor ➢ Le Capteur Photovoltaïque
Les capteurs de force	➢ les capteurs de mesures de flexions ➢ capteurs de mesures de tractions et de compression
Les capteurs de son	➢ Le microphone dynamique à bobine mobile ➢ Le microphone électrostatique à condensateur ➢ Le microphone électrostatique magnétique
Les capteurs de vitesse	➢ Tachymétrie ➢ Codeur incrémental
Les capteurs météorologiques	➢ Capteur de Température ➢ Capteur de pression ➢ Capteurs d'humidité
Les capteurs de position	➢ Capteur a effet hall ➢ Capteurs capacitifs ➢ Accéléromètres
Les capteurs de choc	➢ Accéléromètres ➢ Cellules de charge

Tableau I-1: Les différents types de capteurs

I.2.4. Les critères des capteurs[8] :

On caractérise un capteur selon plusieurs critères dont les plus courants sont :

- La grandeur physique observée (ou mesurée);
- Son étendue de mesure : plage de valeurs du mesurande (grandeur physique à mesurée) pour lesquelles le capteur répond aux spécifications du constructeur.
- Sa sensibilité (la plus faible grandeur détectable, elle correspond au quotient de la variation de la sortie électrique par la variation correspondante du mesurande (entrée);
- Sa précision : est le pourcentage d'erreur de son signal de sortie par rapport à la vraie valeur d'entrée (mesurande) ;
- sa linéarité;
- sa bande passante: la plage de fréquence de travail du capteur;
- sa plage de température de fonctionnement;
- sa dérive thermique;
- sa résolution, ...

I.2.5. Les capteurs intelligents[8] :

Les dernières années ont vu apparaître le concept de capteurs intelligents, en plus de leur faculté de mesurer une grandeur physique, il possèdent d'autres fonctionnalités telles que :

- fonctions configurables de traitement du signal (filtre, gains…);
- fonctions d'auto-test et d'auto-contrôle;
- étalonnage automatique;

sortie sur des bus de terrain.

Ainsi la fonction de capteur s'enrichit avec de nouvelles fonctions suplémentaires de conditionnement des signaux, d'autotest, de calcul et de traitement des données évoluées, mais aussi de communication bidirectionnelle avec l'extérieur. ces capteurs sont capables non seulement de détecter et de mesurer les phénomènes et paramètres physiques, chimiques ou biologiques mais encore de traduire leurs variations en informations partagées et de générer en fonction de celles-ci le déclenchement d'actions au sein de systèmes.

La figure suivante présente un exemple d'architecture de ce type de capteur :

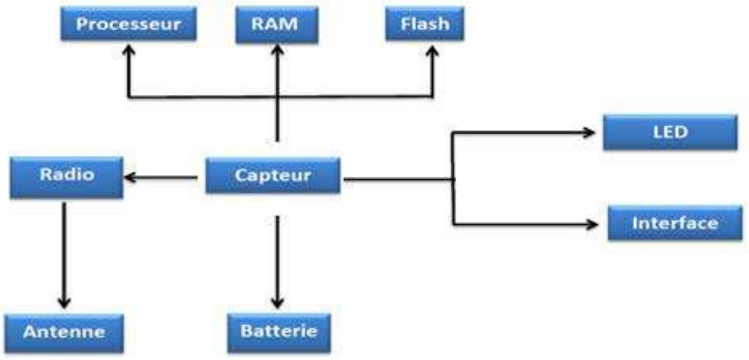


Figure I-5: Architecture générale d'un capteur intelligent

Le capteur constitue le noyau central auquel viennent se greffer différents composants, un processeur, une mémoire pour les données et une mémoire système (RAM et Flash), un système radio dotée d'une antenne, une batterie, quelque fois une diode électroluminescente (LED) et une interface de communication.

I.2.6. Les micro capteurs:

Le micro-capteur est un capteur intégré qui utilise la microélectronique. Ce capteur est constitué d'une plaque en silicium dans lequel on a fixé le capteur, le corps d'épreuve si besoin et d'autres composants électroniques qui peuvent servir à linéariser, amplifier, convertir le courant en tension, etc. Ce type de capteur est très utile vu qu'il fournit un signal linéaire avec une grande sensibilité, une miniaturisation et un coût faible.

Ces évolutions ont permis de constituer des ensembles (réseaux) de capteurs qui fonctionnent comme une seule entité afin d'améliorer la mesure et étendre sa zone d'action avec des couts accéptables.

I.3 Les réseaux [16]*:*

Aujourd'hui en instrumentation la notion du capteur classique isolé est dépasser, on parle de réseaux regroupant un ensemble (dizaines, centaines voir des miliers) de capteurs qui fonctionnent ensembles avec une fonction de communication à distance des mesures réalisées, la figure suivante résume les différents types de réseaux les plus connues en télecommunication et qui ont influencés le domaine des capteurs:

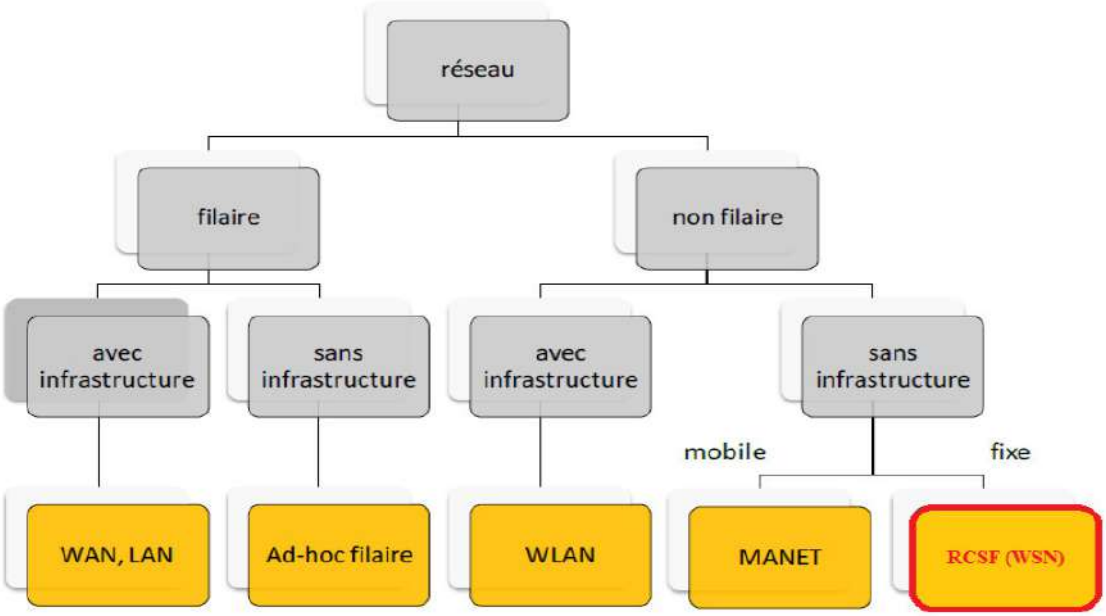


Figure I-6: Classifications des Réseaux de communications

Les réseaux Ad hoc:

Les réseaux Ad hoc sont des réseaux sans fil capables de s'organiser sans infrastructure définie préalablement. Ces réseaux, dans leur configuration mobile, sont connus sous le nom de MANet (pour Mobile Ad hoc NET Works) .

Chaque entité (**node:** noeud) communique directement avec sa voisine. Pour communiquer avec d'autres entités, il lui est nécessaire de faire passer ses données par d'autres qui se chargeront de les acheminer. Pour cela, il est d'abord primordial que les entités se situent les unes par rapport aux autres, et soient capables de construire des routes entre elles : c'est le rôle du protocole de routage. Ainsi, le fonctionnement d'un réseau ad hoc le différencie notablement d'un réseau comme le réseau GSM, les réseaux Wi-Fi avec des points d'accès : là où une ou plusieurs stations de base sont nécessaires à la plupart des communications entre les différents nœuds du réseau (**mode Infrastructure**), les réseaux ad hoc s'organisent d'eux-mêmes et chaque entité peut jouer différents rôles.

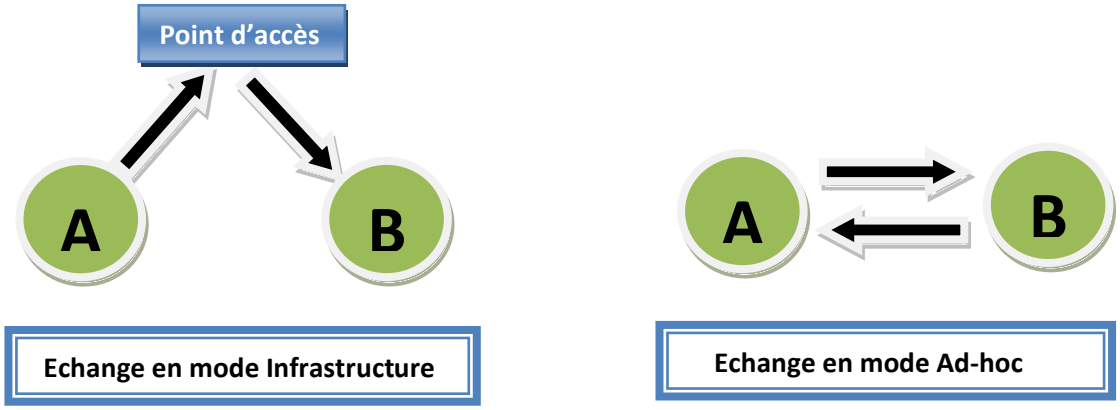


Figure I-7: Les échanges dans les modes Infrastructure et Ad-hoc

L'utilisation la plus simple et la plus courante des réseaux ad hoc est faite par les réseaux sans fil Wi-Fi en permettant une mise en place rapide d'une connexion réseau entre deux ordinateurs.

Les principaux problèmes des réseaux ad hoc **[1]** sont **:**

- Absence d'infrastructure;
- Bande passante limitée;
- Perte de données;
- Perte de routes;
- Contraintes de consommation d'énergie;
- Sécurité limitée;
- Erreur de transmission;
- Interférences;
- Nœuds cachés.

I.4.Les réseaux de capteurs sans fil [16]*:*

Grâce au progrès réalisés dans le domaine de la miniaturisation des capteurs et systèmes micro électromécanique (MEMS) et dans le marché des réseaux et des applications sans fil, s'est crée une nouvelle branche de réseaux mobiles : les réseaux de capteurs sans fil (Wireless Sensor networks) afin d'offrir des solutions économiquement intéressantes pour la surveillance à distance et le traitement des données dans des environnements complexes.

Ces nouvelles technologies permettent de réduire l'encombrement, le coût et la consommation d'énergie et d'augmenter la précision et les performances des capteurs, des processeurs et des circuits spécifiques liés aux capteurs. Un nombre très important de capteurs peut donc être envisagé, intégré et organisé en réseau.

Un réseau de capteurs sans fil (RCSF), Wireless Sensor Networks (WSN) est un ensemble de nœuds communicants composés chacun de quatre entités :

- Un module radio pour échanger des messages via le medium sans fil,
- Un ou plusieurs capteurs ayant une grande physique spécifique à mesurée comme la température, la pression, la vitesse, la position, ...etc,
- Un microcontrôleur qui se charge des traitements nécessaires,
- Une source d'énergie pour alimenter l'ensemble.

Les réseaux de capteurs sans fil sont de plus en plus d'éployés dans des domaines divers couvrant les applications militaires, la surveillance de sites naturels, les applications industrielles, systèmes de (nano) robots…etc. Cette technologie se doit de proposer des solutions autonomes, c'est à dire, capable d'économiser l'énergie et de s'auto configurer.

Les réseaux de capteurs sans fil sont considérés comme un type spécial de réseaux ad hoc. Les nœuds de ce type de réseaux sont reliés le plus souvent à un grand nombre de micro-capteurs capables de récolter et de transmettre des données environnementales d'une manière autonome.

I.5 Architecture des réseaux de capteurs sans fil [3]*:*

Les micro-capteurs , sont considérés comme la pierre angulaire permettant l'édification de réseaux dits de capteurs sans fil. Ils sont dispersés à travers une zone géographique, appelée champ de captage, qui définit le terrain d'intérêt pour le phénomène à étudié, outre les capteurs eux-mêmes, les RCSF possèdent le plus souvent des stations de contrôle appelées nœuds-puits ou stations de base.

Une station de base (passerelle) peut être un nœud fixe ou mobile capable de collecter et de stocker les informations issues des capteurs, et peut également relier le réseau de capteurs à l'Internet ou à une autre infrastructure de communications où un utilisateur peut avoir accès aux données collectées. Les stations de base possèdent beaucoup plus de capacités que les nœuds de capteurs tant au niveau de la mémoire que de la vitesse de traitement ou des réserves en énergie.

Le réseau intermédiaire qui sert à déporter le poste contrôle-commande du lieu de déploiement du réseau de capteurs peut très bien être le réseau GSM, un réseau sans fil comme le WiFi ou même un réseau filaire.

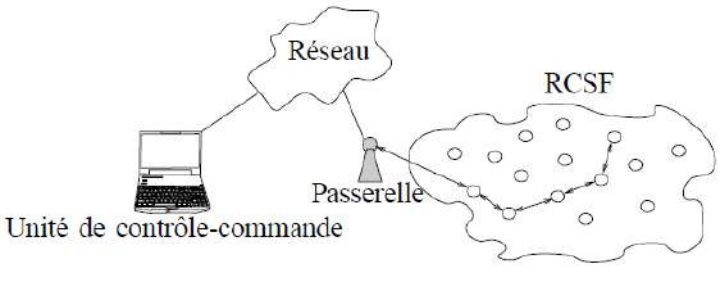


Figure I-8: Exemple d'une architecture d'un réseau de capteurs sans fil.

I.6 Classification des RCSF [3]*:*

Les réseaux de capteurs sans fil peuvent être classés suivant plusieurs critères :

- **Par rapport à la topologie du réseau :** la topologie détermine l'organisation logique des capteurs dans le réseau. Il existe deux principales topologies dans les réseaux RCSF:
  - Topologie plate;
  - Topologie hiérarchique.
- **Par rapport au schéma de communication:** dans un RCSF, la plus part des communications sont du type plusieurs-à-un, où un ensemble de nœuds de la zone d'événement diffuse les informations collectées vers une station de base pour les traiter.

En réalité, un WSN doit assurer trois schémas de communication

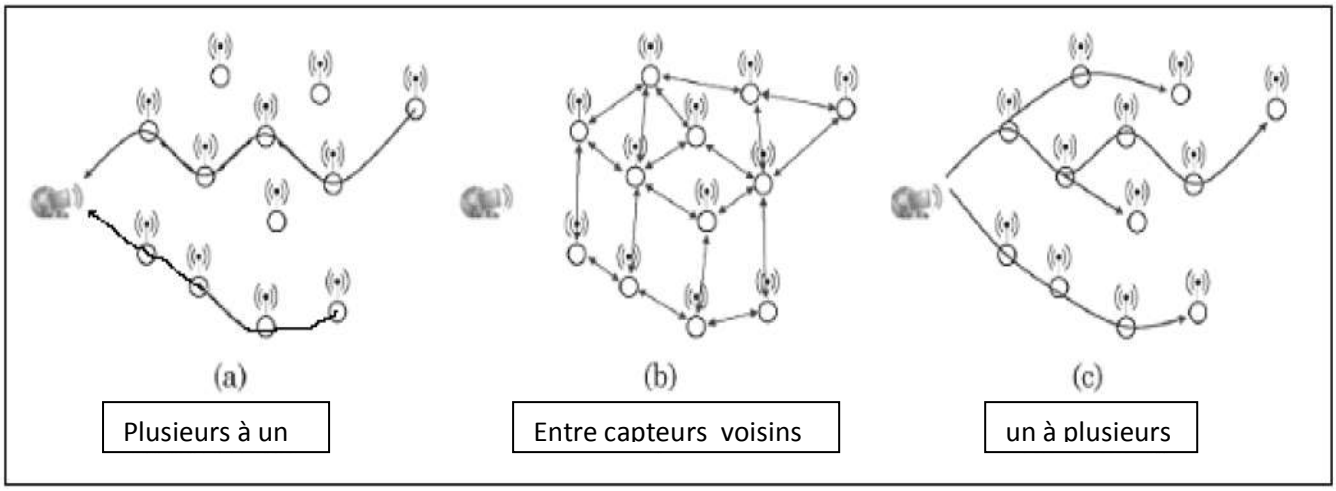


Figure I-9: *Schéma* ***de communication dans un RCSF***

- **Plusieurs-à-un:** il est généralement utilisé pour acheminer les rapports de données vers le puits. En raison des redondances, les lectures doivent être agrégées dans des zones particulières du réseau.
- **Communications locales:** ce sont les communications à un saut entre des capteurs voisins.
- **Un-à-plusieurs:** ce schéma est utilisé par le puits pour communiquer avec ses capteurs, afin de propager des requêtes ou contrôler le fonctionnement des nœuds.

Ces communications peuvent être générales ou ciblées: c-à-d destinées vers une certaine région du réseau.

I.7 Applications des RCSF [13]*:*

La taille de plus en plus réduite des micro-capteurs, le coût de plus en plus faible, la large gamme des types de capteurs disponibles (thermique, optique, vibrations,...) ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs d'envahir plusieurs domaines d'applications. Ils permettent aussi d'étendre les champs applications existantes et de faciliter la conception d'autres systèmes tels que le contrôle et l'automatisation des chaînes de montage.

Les réseaux de capteurs ont le potentiel de révolutionner la manière même de comprendre et de construire les systèmes physiques complexes. Les réseaux de capteurs peuvent se révéler très utiles dans de nombreuses applications lorsqu'il s'agit de collecter et de traiter des informations provenant de l'environnement. Parmi les domaines où ces réseaux peuvent offrir les meilleures contributions, nous citons les domaines : militaire, environnemental, domestique, santé, sécurité, infrastructures de génie civile, ...etc. Des exemples d'applications potentiels dans ces différents domaines sont exposés ci-dessous.

I.7.1. Applications militaires : Comme dans le cas de plusieurs technologies, le domaine militaire a été un moteur initial pour le développement des réseaux de capteurs. Le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui rendent ce type de réseaux un outil appréciable dans un tel domaine. Comme exemple d'application dans ce domaine, on peut penser à un réseau de capteurs déployé sur un endroit stratégique ou difficile d'accès, afin de surveiller toutes les activités des forces ennemies, ou d'analyser le terrain avant d'y envoyer des troupes (détection d'agents chimiques, biologiques ou de radiations). Des tests concluants ont déjà été réalisés dans ce domaine par l'armée américaine dans le désert de Californie.



Figure I-10: Exemple d'une Applications militaires.

I.7.2. Applications dans le domaine de sécurité : les altérations dans la structure d'un bâtiment, suite à un séisme ou au vieillissement, pourraient être détectées par des capteurs intégrés dans les murs ou dans le béton, sans alimentation électrique ou autres connexions filaires. Les capteurs doivent s'activer périodiquement et peuvent ainsi fonctionner durant des années, voire des décennies. Un réseau de capteurs de mouvements peut constituer un système d'alarme distribué qui servira à détecter les intrusions sur un large secteur. Déconnecter le système ne serait plus aussi simple, puisque il n'existe pas de point critique. La surveillance de voies ferrées pour prévenir des accidents avec des animaux et des êtres humains peut être une application intéressante des réseaux de capteurs.

La protection des barrages pourrait être accomplie en y introduisant des capteurs. La détection prompte de fuites d'eau permettrait d'éviter des dégâts. Les êtres humains sont conscients des risques et attaques qui les menacent. Du coup, ils mettent à disposition toutes les ressources humaines et financières nécessaires pour leur sécurité. Cependant, des failles sont toujours présentes dans les mécanismes de sécurisation appliqués aujourd'hui, sans oublier leur coût très élevé. L'application des réseaux de capteurs dans le domaine de la sécurité pourrait diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et à la protection des êtres humains tout en garantissant de meilleurs résultats.

I.7.3. Applications environnementales : des thermo-capteurs dispersés à partir d'un avion sur une forêt peuvent signaler un éventuel début d'incendie dans le champ de captage ; ce qui permettra une meilleure efficacité pour la lutte contre les feux de forêt. Dans les champs agricoles, les capteurs peuvent être semés avec les graines. Ainsi, les zones sèches seront facilement identifiées et l'irrigation sera donc plus efficace. Sur les sites industriels, les centrales nucléaires ou dans les pétroliers, des capteurs peuvent être déployés pour détecter des fuites de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, ...etc.) et alerter les utilisateurs dans un délai suffisamment court pour permettre une intervention efficace. Une grande quantité de capteurs peut être déployée en forêt ou dans un environnement de conservation de la faune afin de recueillir des informations diverses sur l'état du milieu naturel et sur les comportements de déplacement. Par exemple, l'université de Pise en Italie a réalisé des réseaux de capteurs pour le contrôle des parcs naturels (feux, animaux,..).

Il est ainsi possible "d'observer", sans déranger, des espèces animales difficiles à étudier dans leur environnement naturel et de proposer des solutions plus efficaces pour la conservation de la faune.

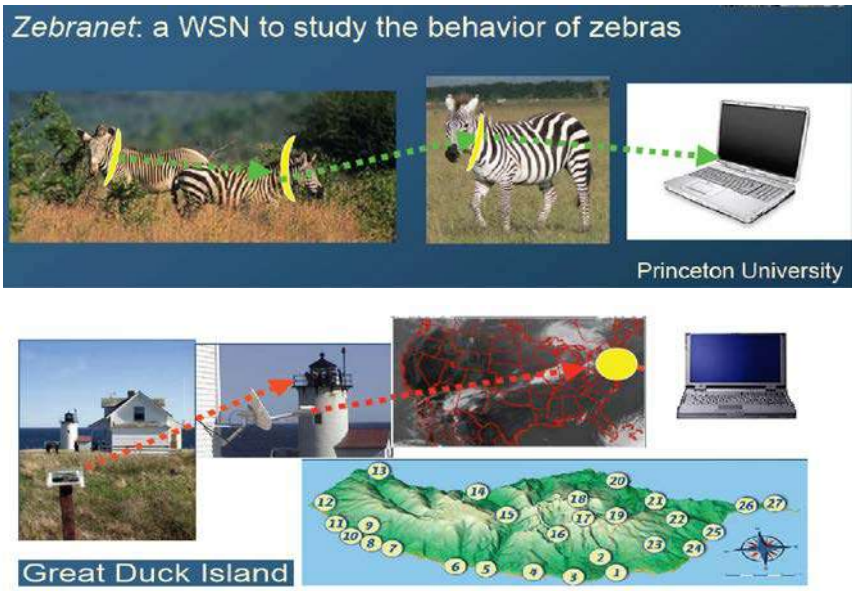


Figure I-11: Exemples d'Applications environnementales

I.7.4. Applications médicales : On pourrait imaginer que dans le futur, la surveillance des fonctions vitales de l'être humain serait possible grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau. Actuellement, des micro-caméras qui peuvent être avalées existent. Elles sont capables, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain avec une autonomie de 24 heures. Les auteurs d'une récente étude, présentent des capteurs qui fonctionnent à l'intérieur du corps humain pour traiter certains types de maladies. Leur projet actuel est de créer une rétine artificielle composée de 100 micro-capteurs pour corriger la vue. D'autres ambitieuses applications biomédicales sont aussi présentées, tel que : la surveillance du niveau de glucose, le monitoring des organes vitaux ou la détection de cancers. L'utilisation des réseaux de capteurs dans le domaine de la médecine pourrait apporter une surveillance permanente des patients et une possibilité de collecter des informations physiologiques de meilleure qualité, facilitant ainsi le diagnostic de quelques maladies.

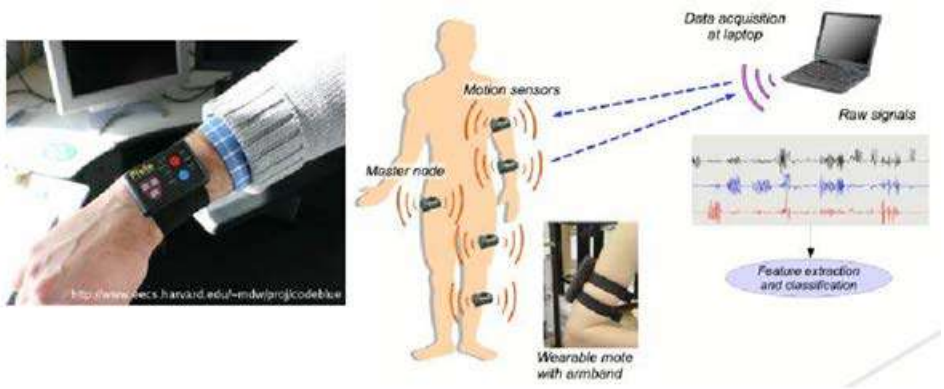


Figure I-12: Exemple d'une Applications médicales

I.7.5. Applications commerciales : Il est possible d'intégrer des nœuds de capteurs au processus de stockage et de livraison. Le réseau ainsi formé, pourra être utilisé pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison. Il devient alors possible pour un client qui attend la réception d'un paquet, d'avoir un avis de livraison en temps réel et de connaître la position actuelle du paquet. Pour les entreprises manufacturières, les réseaux de capteurs permettront de suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux réseaux de capteurs, les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts. Dans les immeubles, le système de climatisation peut être conçu en intégrant plusieurs micro-capteurs dans les tuiles du plancher et les meubles. Ainsi, la climatisation pourra être déclenchée seulement aux endroits où il y a des personnes présentes et seulement si c'est nécessaire. Le système distribué pourra aussi maintenir une température homogène dans les pièces.

Utilisée à grande échelle, une telle application permettrait de réduire la demande mondiale en énergie réduisant du même coup les gaz à effet de serre. Rien que pour les Etats-Unis, on estime cette économie à 55 milliards de dollars par an avec une diminution de 35 millions de tonnes des émissions de carbone dans l'air. Ainsi, dans un contexte mondial où le réchauffement de la planète devient une préoccupation grandissante, une telle conséquence environnementale serait un pas dans la bonne direction.

I.8 Contraintes [13]*:*

Dans le déploiement d'un réseau de capteur sans fil on doit prendre en considération les aspects suivants qui représentent des défis aux réseaux de capteurs:

- ***Lien radio :*** les capteurs possèdent le matériel nécessaire pour effectuer des communications par ondes radio. Toutefois, la diffusion de l'information via ces liens est peu aisée à cause de l'instabilité et du manque de fiabilité qu'ils présentent. En plus, l'utilisation d'un médium de communication partagé pour faire face aux interférences radio, réduit considérablement la capacité d'exploitation du canal.
- ***La dynamicité :*** la topologie des réseaux de capteurs sans fil mobiles est en constant évolution à cause de l'état d'activité des capteurs (extinction, mise en veille et actif) ainsi que leurs déplacements indépendamment les uns des autres. La conception d'un protocole d'auto-organisation qui s'adapte continuellement et rapidement aux changements, s'avère nécessaire pour assurer le bon fonctionnement du réseau.
- ***La limitation des ressources :*** les capteurs se caractérisent par une limitation de ressources en termes de calcul, de stockage et d'autonomie d'énergie.

La contrainte énergie dans ce type de réseaux étant très forte, puisque l'on considère généralement qu'il est difficile voire impossible de recharger les batteries des capteurs (capteurs déployés dans des zones hostiles et inaccessibles). L'ordonnancement d'activité et l'auto-organisation sont considérés comme des solutions de premier choix, dans le but d'éteindre l'équipement radio de certains capteurs et permettre seulement aux capteurs disposant de plus d'énergie de transmettre les données captées à la station de base.

- ***Localisation des décisions :*** un réseau de capteurs est potentiellement constitué d'un grand nombre de capteurs qui ont une portée de transmission très limitée. D'où, le processus d'acheminement d'une information de la source à la destination se fait généralement en multi-sauts par le biais de plusieurs capteurs. En outre, il est inconcevable de faire impliquer un grand nombre de capteurs pour une prise de décision d'un capteur. Ainsi, tout processus de prise de décision doit être localisé.
- ***Passage à l'échelle :*** la plupart des protocoles sont conçus pour des réseaux de capteurs d'une grande taille. Cependant, ces protocoles sont dits efficaces si les performances des réseaux ne doivent pas chuter d'une manière drastique quand le nombre de capteurs augmente dans le réseau.
- ***Auto-configuration :*** les réseaux de capteurs sont généralement déployés aléatoirement dans des zones d'intérêt hostiles (capteurs largués par un avion). Par conséquent, aucune intervention humaine ne peut être requise pour assurer leur organisation. L'auto configuration de ces réseaux s'avère nécessaire pour leur bon fonctionnement.
- ***Tolérance aux fautes :*** certains capteurs pourraient être défaillants, bloqués à cause de l'épuisement de leurs batteries, ou subissent des dommages physiques (écrasés par des animaux ou lorsqu'ils sont jetés par un avion). La défaillance de ces capteurs ne devrait pas avoir une influence sur le fonctionnement du réseau. La tolérance aux fautes est la capacité de soutenir les fonctionnalités d'un réseau de capteurs sans causer d'interruption lorsqu'un capteur cesse de fonctionner. Si les capteurs sont déployés dans un habitat, la tolérance aux fautes exigée peut être basse puisque ce type de réseaux n'est pas facilement endommagé, alors que si les capteurs sont déployés dans un champ de bataille pour la surveillance et la détection, la tolérance aux fautes devrait être élevée puisque les données surveillées sont critiques et les capteurs peuvent être détruits facilement par des actions hostiles.

Un réseau RCSF idéal doit passer a l'échelle, être robuste, économe en énergie, offrir des performances réseaux raisonnables en terme de délai et taux de perte, être fiable, pas chère et ne demande pas de maintenance à long terme. Un peu trop exigeant !! Trois limitations essentielles se posent : énergie, calcul et portée.

I.9 Les différentes problématiques dans les réseaux de capteurs [2]*:*

Les recherches dans le domaine des réseaux de capteurs ont révélé plusieurs problématiques, parmi lesquelles nous citons :

- ***Le Routage :*** concevoir un protocole de routage performant en termes de minimisation de la consommation de l'énergie, du choix des routes optimales pour l'acheminement de l'information d'un capteur à la station de base et vice versa, de réduction du délai de délivrance des paquets...etc. Ainsi le réseau doit passer à l'échelle sans que ses performances se dégradent.
- ***La couche MAC :*** la spécificité des réseaux de capteurs sans fil mobiles nécessite le développement de nouveaux protocoles MAC qui s'adaptent aux contraintes imposées par ces réseaux. Ceci dans le but d'améliorer le débit, minimiser la consommation d'énergie, optimiser le partage du médium ainsi que minimiser le délai de délivrance des paquets.
- ***La qualité de service :*** des protocoles au niveau de la couche MAC devraient être capables d'établir des priorités entre les flux, limiter les pertes de paquets vitaux pour la gestion du réseau, ou du moins en restreindre l'impact.
- ***La cross-layer :*** dans les modèles classiques, les concepteurs essaient d'optimiser les performances au niveau d'une couche indépendamment des autres couches. Par exemple, le routage et les fonctions de la couche MAC sont optimisés indépendamment les uns des autres. Cependant, une telle indépendance est communément considérée comme trop onéreuse pour les réseaux de capteurs. Par conséquent, une coopération permettant un compromis entre performances, dépendance et flexibilité doit être proposée pour optimiser les capacités du réseau en général.
- ***La diffusion de l'information :*** les protocoles de diffusion conçus pour les réseaux de capteurs doivent tenir compte de leurs spécificités ainsi que de leurs contraintes intrinsèques imposées. Ainsi, pour concevoir un protocole efficace, il faudrait assurer une couverture maximale des capteurs composant le réseau (taux d'accessibilité supérieur 90%), minimiser le nombre des réémetteurs et des réceptions redondantes ainsi que la consommation d'énergie.
- ***La sécurité :*** pour les applications qui exigent un niveau de sécurité assez élevé telles que les applications militaires, des mécanismes d'authentification, de confidentialité, et d'intégrité doivent être mis en place au sein de leur communauté. Les algorithmes de cryptographie conçus pour les réseaux de capteurs doivent tenir compte des ressources limitées que présentent ces réseaux.

I.10 Les Standards dans les RCSF [5]*:*

L'hétérogénéité dans les plates formes de réseaux de capteurs disponibles a comme conséquence des problèmes de compatibilité lors de la réalisation des applications. Par conséquent, la standardisation de certains aspects de communication est nécessaire. Pour cette finalité, l'organisme de normalisation IEEE à éditer une norme: 802.15.4. Cette norme est un protocole de communication destiné aux réseaux sans fil de la famille des LR WPAN (Low Rate Wireless Personal Area Network) du fait de leur faible consommation, de leur faible portée et du faible débit des dispositifs utilisant ce protocole.

La norme 802.15.4 est utilisé par de nombreuses implémentations basées sur des protocoles propriétaires ou sur IP (Internet Protocol), comme le ZigBee et le 6LoWPAN. Les caractéristiques des LR WPAN sont :

- la formation d'un réseau de type étoile ou maillé,
- l'allocation d'une adresse de 16 bits ou de 64 bits,
- l'utilisation de CSMA/CA pour communiquer,
- la faible consommation d'énergie,
- la détection d'énergie (ED),
- l'indication de la qualité de la liaison (LQI),
- l'utilisation de :
  - 16 canaux dans la bande de fréquence de 2.4 à 2.4835 GHz,
  - 10 canaux dans la bande de fréquence de 902 à 928 MHz,
  - 1 canal dans la bande de fréquence de 868 à 868.6 MHz.

L'IEEE a défini deux types de dispositifs pouvant participer à un réseau :

- le dispositif ayant toutes les fonctions possibles (FFD : Full Function Device);
- le dispositif ayant des fonctions limitées (RFD : Reduced Function Device).

- Le FFD peut assurer trois rôles dans un réseau : coordinateur PAN, routeur ou dispositif relié à un capteur (plus petite fonction possible, appelé dispositif de fin : End Device).

- Le RFD est prévu pour des applications simples (signaler l'état d'un capteur, contrôler l'activation d'un actionneur). Il est considéré comme un dispositif "d'extrémité" (End Device), dans le sens où il n'est pas essentiel au réseau.

Pour communiquer sur un même réseau, un FFD (au moins) et des RFD doivent utiliser le même canal physique parmi ceux définis selon la bande de fréquence choisie. Le FFD peut dialoguer avec des RFD et des FFD, tandis que le RFD dialogue avec un FFD uniquement.

Chapitre II : Conception du systèmede mesure

II.1.Introduction :

Dans ce chapitre nous allons présenter les différents aspects liés à la conception de notre système de mesure. Nous présenterons les choix réalisés et les solutions retenues pour satisfaire le cahier de charges ainsi que les différents outils utilisés dans le développement du système.

II.2.Cahier de charge :

Le système de mesure à réaliser est un système d'acquisition de données analogiques et numériques destinés pour des applications sportives diverses dont le but d'acquérir des mesures de vitesses, d'accélérations, d'inclinaisons des différents segments du corps de l'athlète (avant bras, bras, cuisse, tibia, pied, tête,...etc) afin d'étudier et d'analyser le geste technique du sportif en vue de le quantifier, le modéliser et de l'améliorer. Ainsi divers types de capteurs peuvent être utilisés avec notre système tels que :

- les accéléromètres pour la mesure de l'accélération, la vitesse, la position et l'inclinaison ;
- les gyromètres pour la mesure des vitesses angulaires ;
- les goniomètres pour la mesurer des angles de rotation des articulations du corps ;
- les compas électroniques pour la mesure de l'orientation ;
- les FSR pour la mesure de la force et de la pression; ...etc.

Ces capteurs sont le plus souvent des capteurs MEMS de taille miniature qui peuvent être facilement fixés sur le corps de l'athlète.

Les exigences des tests et des mesures dans le domaine sportif peuvent être résumées par un besoin d'un système permettant l'acquisition, le prétraitement des mesures et la transmission par voie Radio Fréquence de ces mesures vers un système de traitement et de stockage plus puissant tel qu'un PC. Le système doit s'adapter avec diverses situations de mesures, doit être autonome, permettant l'interfaçage d'un ensemble de capteurs à sortie analogique ou numérique (I2C, ...) avec une portée de transmission de quelques dizaines de mètres.

II.3.La solution adoptée :

Nous avons opté pour la solution d'un petit réseau de capteur sans fil comme le montre le schéma de principe de la figure II-1 puisqu'il s'adapte bien avec les applications visées dans le domaine sportif.

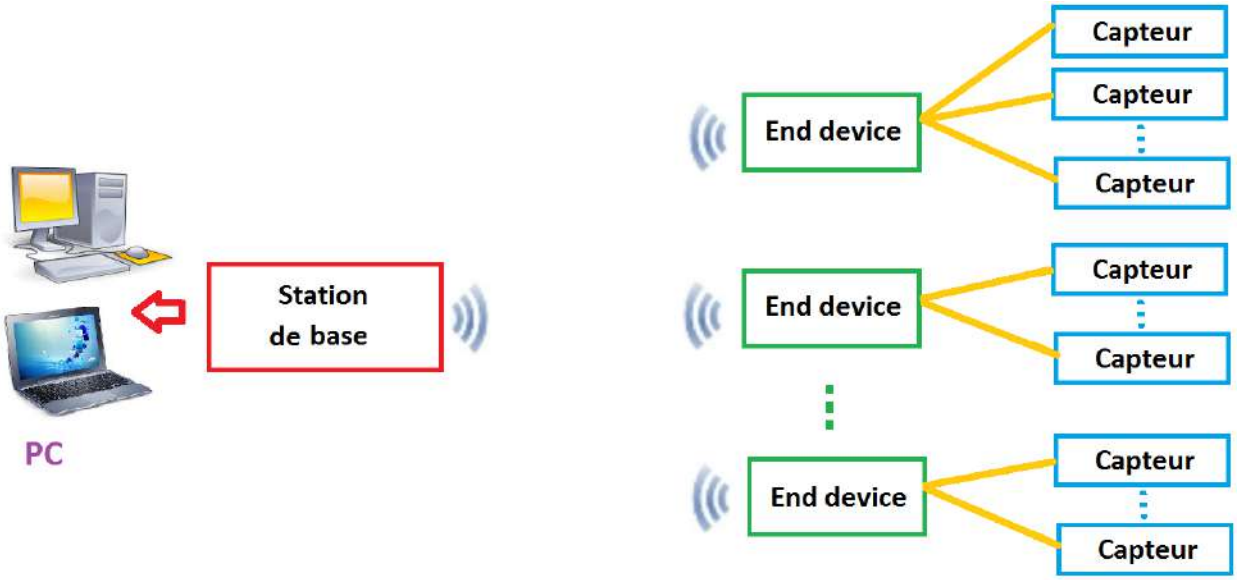


Figure II-1: Schéma de principe du système à concevoir

Ainsi le système qu'on se propose de réaliser dans le cadre de ce travail est constitué d'un ensemble de nœuds dont chacun est lié à un ou plusieurs capteurs, le nœud pourra être fixé sur un segment du corps du sportif, l'ensemble des mesures collectées par les nœuds après un prétraitement (conversion analogique numérique, moyennage, ...) sont envoyées vers une station de base qui est reliée à un PC de bureau ou à PC portable via liaison filaire (USB, Série, ...) pour assurer la visualisation, le traitement et le stockage des mesures reçues.

Pour la réalisation du réseau de capteur sans fil (WSN) de notre application nous avons opté pour l'utilisation d'un kit de capteur sans fil (WSN) parmi ceux proposés par les constructeurs de ce type de systèmes.

Cette solution nous a permis de contourner beaucoup de problèmes (matériels et techniques) liés à la réalisation des circuits hardware (station de base et nœuds de capteurs) et nous a permet d'économiser sur le temps du développement en plus des couts assez faible de certains kits proposés dans le commerce.

Ainsi la plus grande tache du travail sera orienter vers la partie soft (programmation) du système à réalisé puisque le hardware est disponible.

II.4.Choix du kit WSN :

Dans le commerce il existe plusieurs plateformes hardware pour réseaux de capteurs sans fil qu'on peut adaptés pour les applications visées, ces dispositifs se présentent sous forme kits complets pré à utilisé (figure II-2), composés en général par une station de base et plusieurs nœuds capteurs (End device). Parmi ces kits nous citons à titre indicatif et non limitatif :

- le kit NI Wireless Sensor Networks (WSNs) de National instruments ;
- le kit eZ430-RF2500 de Texas instruments ;
- le kit CC2530EM wireless de Texas instruments ;
- kit de développement WSN de freescale utilisant le chip MC13213; .....etc.



Figure II-2: Différents kits pour réseaux de capteurs sans fil

Pour des raisons de cout et de disponibilité nous n'avons pas eu beaucoup de choix pour cela nous avons utilisé le kit de Texas Instruments l'eZ430-RF2500 réalisé autour des microcontrôleurs MSP430. Ces derniers sont des microcontrôleurs de la marque Texas Instruments, construit autour d'un CPU 16 bits. Le MSP430 a été conçu pour des applications embarquées à basse consommation et à faible coût. Il est particulièrement adapté aux applications sans-fil fonctionnant sur batteries.

Le composant se décline en une série de configurations comprenant les périphériques usuels :

- convertisseurs A/D 10/12/14/16 bits ;
- convertisseurs D/A 12 bits ;
- comparateurs ;
- interfaces de communications : USART, SPI, I2C,
- Interface de pilotage LCD,
- Temporisateur chien de garde (Watchdog);
- multiplicateur hardware,
- DMA, oscillateur interne, ...etc.

Tous les puces MSP430 sont programmables par JTAG ou BSL à part certains modèles anciens à EEPROM (MSP430C3xx) ou certaines versions grandes volumes à ROM.

Le MSP430 est une option très répandue pour les appareils de mesure à basse consommation alimentés par batterie. Son mode de fonctionnement en attente consomme moins de 1 mA, il fonctionne à des fréquences de l'ordre de 8 et 16 MHz pour les modèles les plus récents.

Il a toutefois des limitations qui l'empêchent d'être utilisé dans des systèmes embarqués plus complexes. Par exemple, il n'a pas de bus mémoire externe (qui permettrait d'accéder efficacement à des mémoires RAM / ROM externes) et sa capacité de mémoire (8K de RAM à 1 Mo de flash pour les mieux dotés) peut se révéler insuffisante pour des applications qui demandent de grandes tables de données.

II.5.Présentation du kit eZ430 RF2500 [12]*:*

Ce kit a une taille d'une clé USB, c'est un outil de développement complet avec une interface USB permettant de programmer le microcontrôleur MSP430 ainsi que l'émetteur-récepteur sans fil CC2500 2.4-GHz.

La carte électronique est équipée en plus du MSP430 qui peut fonctionner à 16 MHz et de la puce RF CC2500. Il à 18 E/S configurables, deux Leds, un bouton poussoir et une antenne intégrée. L'eZ430-RF2500 utilise l'environnemnet de développement IAR Embedded Workbench Integrated Development Environment (IDE) ou le Code Composer Essentials (CCE) pour écrire, télécharger, et corriger une application.

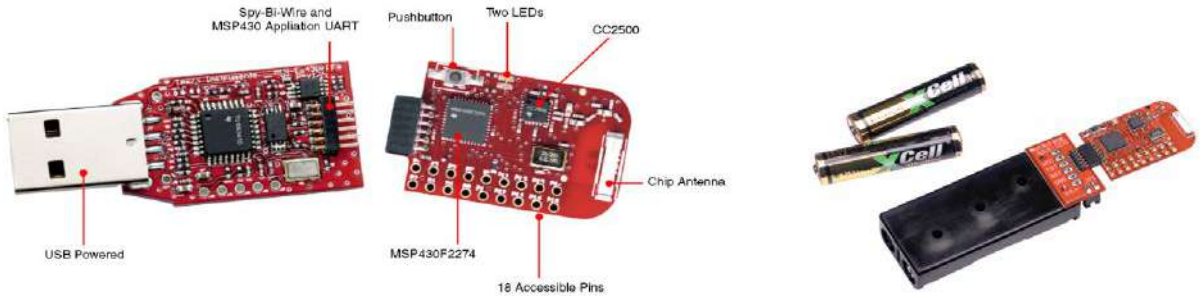


Figure II-3. Le kit eZ430-RF2500 de Texas Instruments

II.5.1. Contenu de kit eZ430-RF2500 [12]:

Le kit eZ430-RF2500 peut être employé comme instrument de développement autonome. En plus, le nœud (PCB) peut également être isolé de l'interface de téléchargement et intégrer dans une autre conception en enlevant la clôture en plastique.

Le PCB d'un nœud comporte un microcontrôleur MSP430F2274 et la plupart de ses E/S sont facilement accessibles. Les lignes d'E/S sont montrées sur la Figure II-4 et décrites dans le tableau II-1 et le tableau II-2

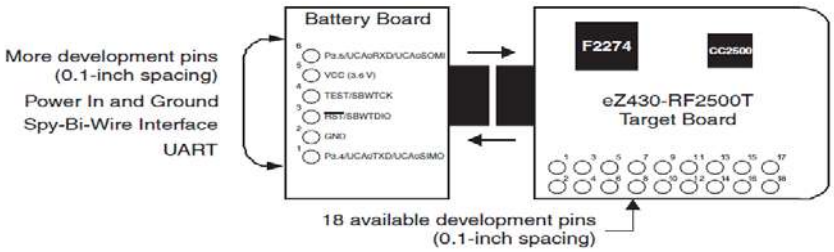


Figure II-4. Les entrées/sorties du support batterie et du nœud eZ430-RF2500

Le tableau suivant décrit le brochage du connecteur alimentation d'un nœud:

Pin	Fonction	Description
1	P3.4/UCA0TXD/ UCA0SIMO	E/S numérique d'usage général / Sortie de transmission du module UART / Entrée de communication SPI en mode esclave
2	Gnd	Référence de la tension.
3	~RST / SBWTDIO	Entrée de remise à zéro / lignes E/S de données pendant la programmation et le test de l'interface Spy-Bi-Wire
4	TEST / SBWTCK	Entrée de sélection du mode Test pour l'interface JTAG/ ligne d'horloge pendant la programmation et le test de l'interface Spy-Bi-Wire
5	Vcc (3.6 V)	Tension d'alimentation.
6	P3.5/UCA0RXD/ UCA0SOMI	E/S numérique d'usage général / Entrée de réception du module UART / sortie de communication SPI en mode esclave

Tableau II-1. Les entrées/sorties du support batterie

Par contre le tableau suivant décrit le brochage du connecteur d'un nœud :

Broche	Fonction	Description
1	GND	référence de masse
2	VCC	tension d'alimentation
3	P2.0/ACLK/A0/OA0I0	Usage général numérique E/S broche / ACLK sortie / ADC10, entrée analogique A0
4	P2.1 / TAINCLK / SMCLK / A1 / A0O	Broche Polyvalent numérique / ADC10 E /S, entrée analogique A1 Minuterie A, signal d'horloge à ACLK, sortie signal SMCLK
5	P2.2 / TA0 / A2 / OA0I1	Broche polyvalent numérique / ADC10 I / O, entrée analogique A2 Minuterie A, capture: CCI0B entrée / BSL recevoir, comparer: OUT0 sortie
6	P2.3 / TA1 / A3 / VREF– / VeREF–/ OA1I1 / OA1O	Polyvalent numérique broches d'E/S / Timer_A, capture: entrée CI1B,comparer: OUT1 Sortie / ADC10, entrée analogique A3 / négatif référence sortie / entrée de tension
7	P2.4/TA2/A4/VREF+/ VeREF+ / OA1I0	Polyvalent numérique broches d'E / S / Timer_A, comparer: sortie OUT2 / ADC10, entrée analogique A4 / positif référence sortie / d'entrée de tension
8	P4.3 / TB0 / A12 / OA0O	Polyvalent numérique broches d'E / S / ADC10 entrée analogique A12 / Minuterie A, capture: entrée CCI0B, comparer: OUT0 sortie
9	P4.4 / TB1 / A13 / OA1O	Entrée analogique polyvalente numérique pin / ADC10 I / O A13 / Minuterie A, capture: entrée CCI1B, comparer: sortie OUT1

10	P4.5 / TB2 / A14 / OA0I3	Polyvalent numérique broches d'E / S / ADC10 entrée analogique A14 /Minuterie A, comparer: sortie OUT2
11	P4.6 / TBOUTH / A15 / OA1I3	Polyvalent numérique broches d'E / S / ADC10 entrée analogique A15 / Minuterie A, passer tous TB0 à TB3 sorties à haute impedance
12	GND	Référence de masse
13	P2.6 / XIN (GDO0)	Polyvalent numérique broches d'E / S / borne d'entrée de l'oscillateur
14	P2.7 / XOUT (GDO2)	Polyvalent numérique broches d'E / S / borne de sortie de l'oscillateur
15	P3.2 / UCB0SOMI / UCB0SCL	Polyvalent numérique broches d'E / S USCI_B0 esclave out / maître en mode SPI, I2C SCL horloge en mode I2C
16	P3.3 / UCB0CLK / UCA0STE	Polyvalent numérique broches d'E / S USCI_B0 horloge d'entrée / sortie / USCI_A0 esclave transmettre permettre
17	P3.0 / UCB0STE / UCA0CLK / A5	Polyvalent numérique broches d'E / S / USCI_B0 esclave transmettent activer / USCI_A0 horloge entrée / sortie / ADC10, entrée analogique A5
18	P3.1 / UCB0SIMO / UCB0SDA	Polyvalent numérique broches d'E / S / USCI_B0 esclave / maître dans le mode SPI, I2C SDA des données en mode I2C

Tableau II-2. Les entrées/sorties du connecteur d'un Noeud eZ430-RF2500

II.5.2. Caractéristiques des composants de bases [12]:

Chaque nœud est composé d'un microcontrôleur MSP430F2274 et d'une puce RF CC2500 qui ont les caractéristiques suivantes :

MSP430F2274 :

- Puissance de traitement de 16MIPS ;

- Convertisseur A/N 10 bits de type SAR avec fréquence max de 200 ks/s ;
- Deux amplificateurs opérationnels intégrés ;
- Horloge de surveillance, Timer_A3 de 16 bits et Timer_B3
- Module de communication USCI soutenant : UART/LIN, (2) SPI, I2C, ou IrDA
- Cinq modes de basse consommation de puissance jusqu'a 700 nA.

CC2500

- émetteur récepteur (RF) radiofréquence à 2.4-GHz ;
- Débit programmable jusqu'à 500 Kbps ;
- Faible consommation.

La série MSP430 de Texas Instruments comme nous l'avons déja évoqué est une série de microcontrôleurs 16 bits à très basse consommation fonctionnant à une fréquence max de 16 MHz. Trois signaux d'horloge sont générés : l'horloge maîtresse (MCK) qui est utilisée par le CPU et qui est alimentée par un oscillateur haute vitesse DC0, l'horloge auxiliaire (ACLK) et la Sub-main clock (SMCLK) qui est utilisée par certains périphériques.

Il existe cinq modes de fonctionnement basse consommation (LPM0, LPM2, LPM3, LPM4) obtenus en désactivant sélectivement certaines horloges. La figure II-5 représente la consommation de courant pour chaque mode. Pour notre application nous utiliserons le mode LPM3, qui représente un mode où les interruptions sont actives.

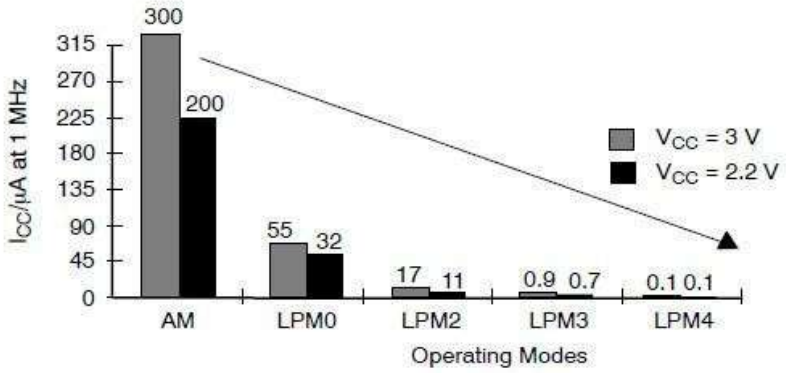


Figure II-5. La consommation du courant pour chaque mode

La figure suivante montre le brochage du MSP430F2274 :

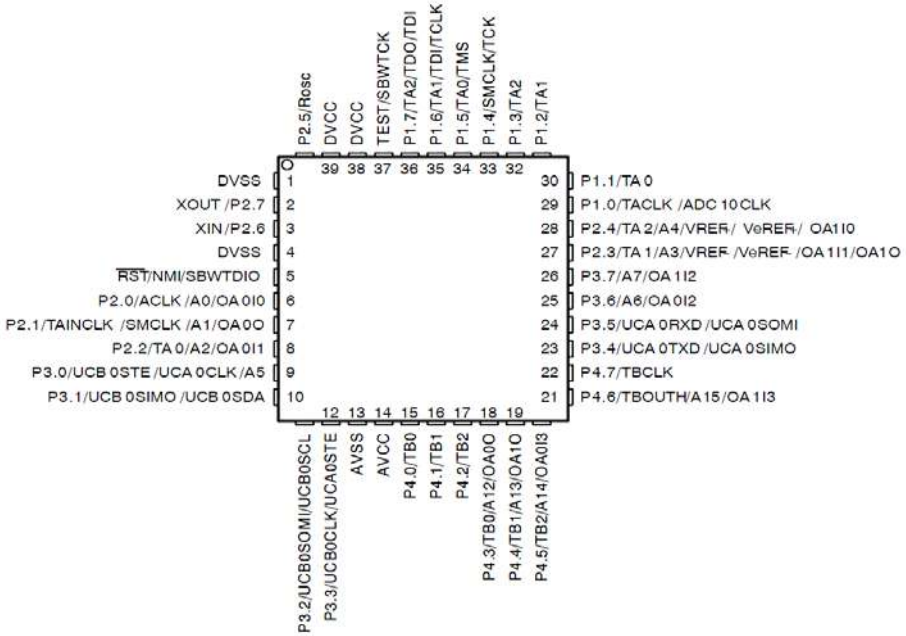


Figure II-6. Brochage du MSP430x22x4[12]

La figure suivante présente le schéma fonctionnel des microcontrôleurs de la famille MSP430x22x4 :

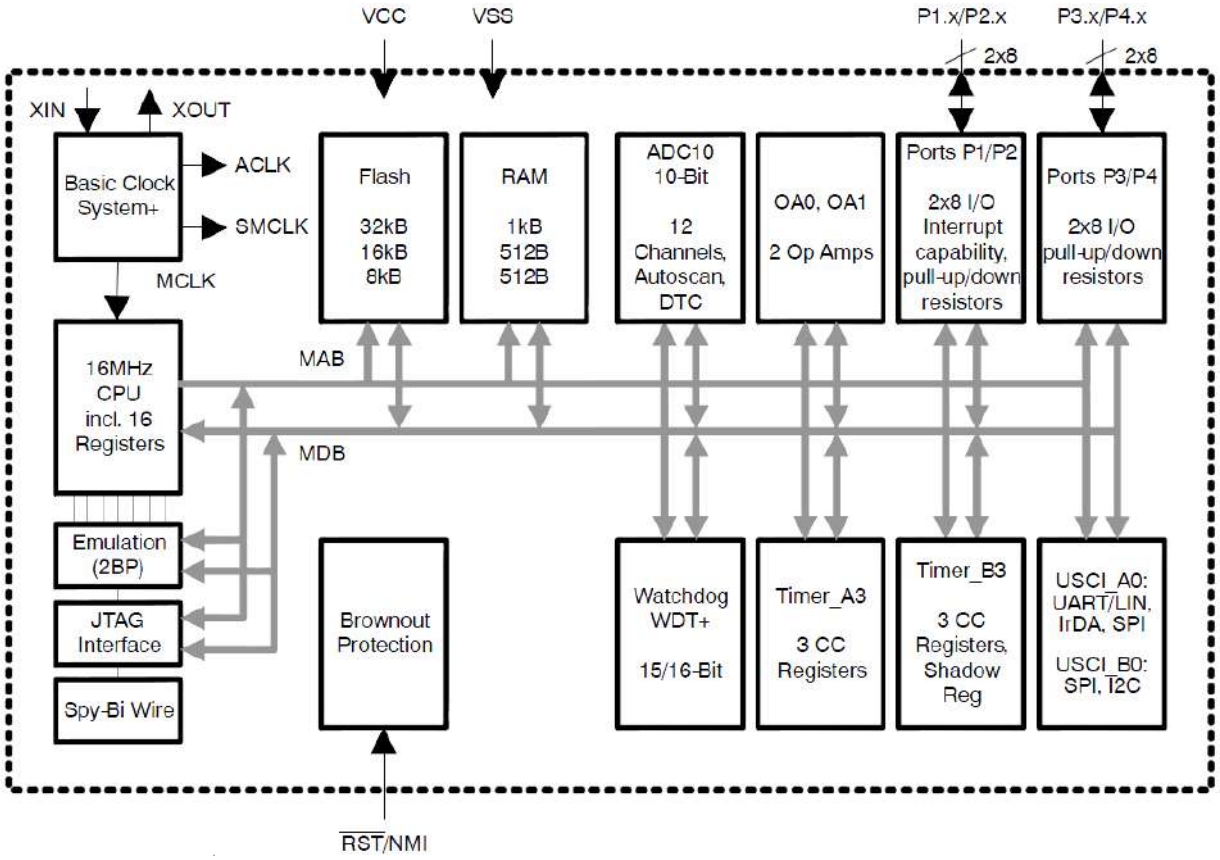


Figure II-7. Schéma fonctionnel de MSP430x22x4[12]

Il faut noter pour ce microcontrôleur la présence des éléments suivants :

- 2 bus séparés (MAB et MDB) relative à l'architecture Harvard ;
- Convertisseur A/N à 12 canaux ;
- Une mémoire flash de 32 Kbits + 256 bits (pour le cas du MSP430F2274) ;
- Une RAM de 1Kb (MSP430F2274) ;
- 2 Timers 16 bits ;
- Module de communication série universel améliorer (USCI) supportant les interfaces : série, LIN, I2C, SPI, IrDA ;
- 4 ports d'E/S avec résistance Pull Down/Pull Up;
- 2 Ampli OP intégrés; ...etc.

La figure suivante montre le schéma électrique d'un nœud eZ430-RF2500 :

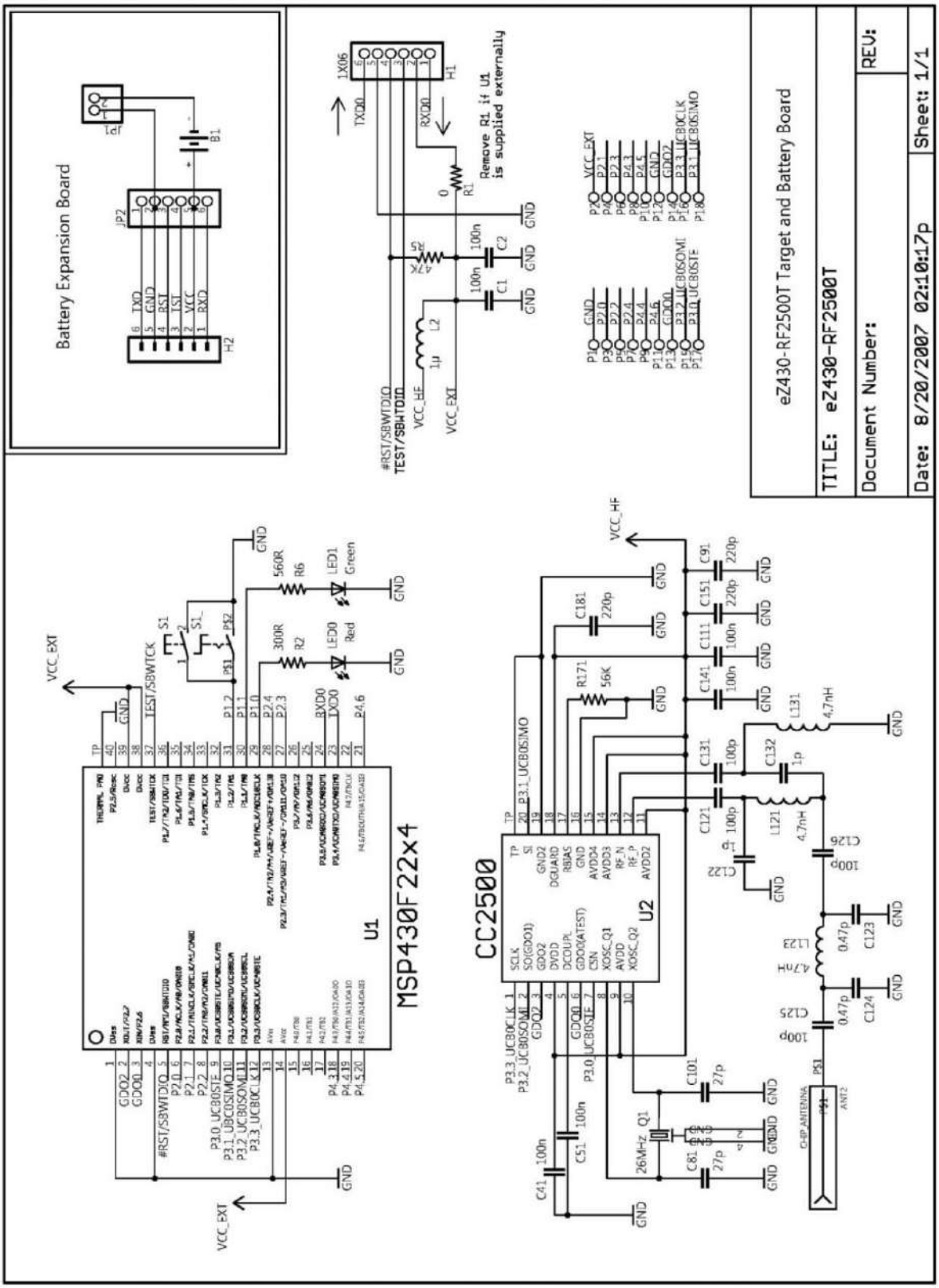


Figure II-8. Schémas électriques de l'eZ430-RF2500[12]

II.6. Le protocole SimpliciTI [5]*:*

SimpliciTI est un protocole de communication RF de faible puissance destiné pour les petits réseaux sans fil. Ce protocole open source est un excellent outil pour la construction d'un réseau avec des dispositifs fonctionnant sur batterie en utilisant des faibles puissances il est pris en charge par les microcontrôleurs MSP430. SimpliciTI a été conçu pour la mise en œuvre facile et déploiement out-of-the-box sur plusieurs plateformes de produits RF (CC1XXX/CC25XX) de Texas Instruments.

La figure suivante montre l'organisation en couches de ce protocole :

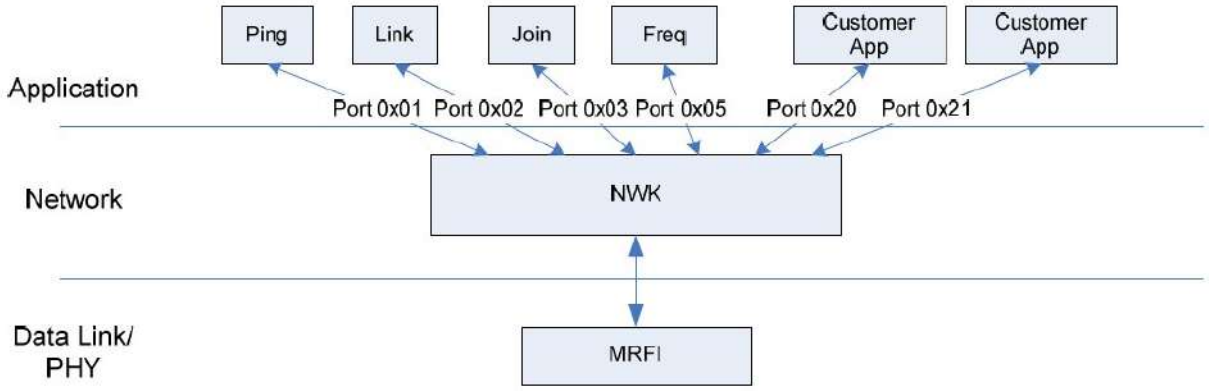


Figure II-9. Les couches logiques du protocole SimpliciTI.

Il est constitué de 3 couches :

- ***La couche application :*** qui représente la partie du soft à implémenter par l'utilisateur pour manipuler les capteurs et assurer la transmission entre End Device et Station de base.
- ***La couche réseau :*** cette couche permet de gérer les queues de transmission et de réception (TX et RX) et permet d'acheminer les paquets à leurs destinations qui est toujours une application désignée par un numéro de Port.
- ***La couche physique :*** peut être subdivisée en deux sous-couches :
    - BSP (Board Support Package): qui représente un ensemble d'instructions et de fonctions manipulant les périphériques de communications comme l'interface SPI, I2C, ...
    - MRFI (Minimal RF Interface) : qui représente le hardware minimal nécessaire à la communication.

II.7. L'outil de développement utilisé :

Pour travailler avec le kit eZ430-RF2500 de Texas Instruments on dispose de deux logiciels de développement pour les familles des microcontrôleurs MSP430 :

- l'IAR Embedded Workbench for TI MSP430
- et Code Composer Studio CCS for MSP430

Dans notre travail nous avons choisi de travailler avec le logiciel IAR qui est plus simple et nous permet de faire des simulations avec Proteus lors du développement.

IAR Embedded Workbench :

IAR Embedded Workbench pour MSP430 est un environnement de développement intégré (IDE) pour la génération et le débogage des applications embarquées pour les microcontrôleurs MSP430. L'IDE comprend un compilateur de code C/C++, un assembleur, un débogueur et un simulateur. La version light offre la possibilité d'écrire un code de taille limitée (la limitation de la taille du code du compilateur C/C++ est de 4 Ko pour les dispositifs MSP430 traditionnels de 8 Ko pour les MSP430X de 60 KB Flash).

II.8. Schéma synoptique du réseau de capteurs élaboré :

La plate-forme qu'on se propose de réaliser est très simple (Figure II-10), elle est constituée dans un premier lieu d'une station de base et de 3 nœuds de capteurs (End device) :

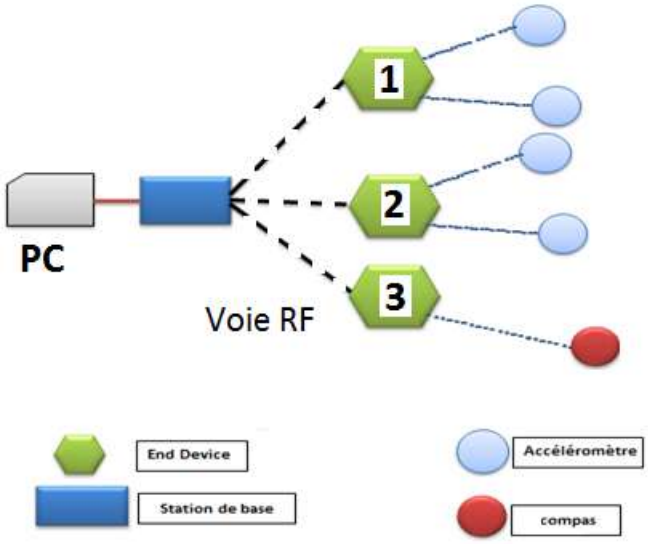


Figure II-10. Shémas Synoptique du WSN de notre application

- Les nœuds 1 et 2 sont interfacés à des accéléromètres analogiques permettant la mesure des accélérations et des inclinaisons des segments du corps.

- Le nœud 3 est relié à un compas électronique qui permet de mesurer l'orientation d'un segment donné, la donnée est récupérée via une liaison I2C.

Les mesures sont transmises des nœuds capteurs vers une station de base (Acces Point) relié à un PC via le port USB pour les afficher à l'aide d'une interface graphique.

Ces choix se sont traduits par le développement de trois programmes :

- Deux programmes destinés pour les nœuds de capteurs :
    - Le programme des nœuds 1 et 2 permet l'acquisition des signaux analogiques des capteurs accéléromètres, sa tâche principale consiste à convertir les différentes entrées analogiques reliées au CAN 10 bits intégré et de transmettre les mesure vers la station de base.
    - Le programme du nœud 3 permet la lecture de l'information d'orientation sous forme d'un ensemble d'octets via la liaison I2C et transmettre la valeur lue vers la station de base.
- Un programme destiné pour le nœud principal constituant la station de base qui permet de gérer les nœuds de capteurs et récupère les mesures transmises par ces derniers et les acheminées vers le PC via liaison USB.

II.8.1. Le module ADC10 [12]:

Le module de conversions analogique-numérique (ADC) est de type convertisseur a approximations successives, il a une résolution de 10 bits. Ce module est associé à : un dispositif d'échantillonnage, un générateur de tension de référence, et un contrôleur de transfert de données (DTC).

Le DTC permet aux échantillons du module ADC10 d'être convertis et stockés n'importe où dans la mémoire sans l'intervention du CPU. Le module peut être configuré avec le logiciel d'utilisateur pour réaliser une série d'applications.

L'ADC10 du MSP430F2274 est caractérisé par :

- Un convertisseur A/D10 (10 bits) à approximations successives ;
- Jusqu'à 200 ks/s : 200,000 échantillons par seconde, ou une période de 5 µs;
- Contrôle du démarrage des conversions par software ou par Timer;
- Plusieurs canaux d'entrée configurable individuellement (10 pour l'eZ430-RF2500) ;

- 4 canaux internes supplémentaires (Température interne, V_{CC}, Ve_{REF+}, Ve_{REF-});
- Contrôle des séquences d'acquisition (simple, répétitive, unique, séquentielle);
- Contrôle de mise en veille ;
- Buffer de stockage des entrées analogiques ;

La figure suivante présente le schéma fonctionnel du module ADC10 du MSP430-RF2500 :

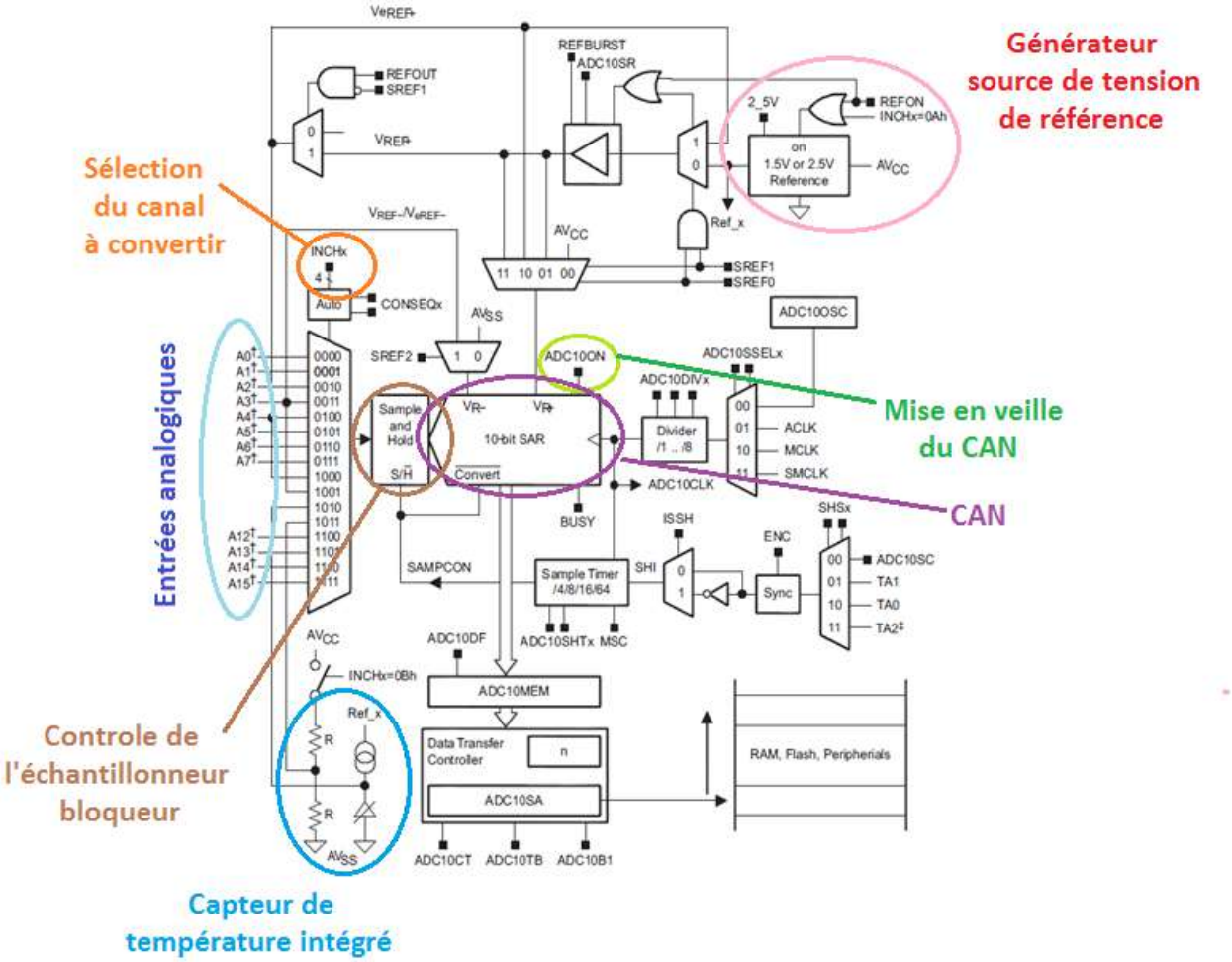


Figure II-11. Schéma fonctionnel d'ADC10 [12]

II.8.1.1. Les entrées analogiques :

Dans le schéma global de l'ADC10, il y'a 16 entrées analogiques 12 d'entre eux sont accessibles par l'utilisateur et 4 reliés aux périphériques intégrés (capteur de température, Tension d'alimentation, Ve_{REF+} et Ve_{REF-}).

Le microcontrôleur que nous utiliserons sur les nœuds de l'eZ430-RF2500 dispose de 15 lignes d'E/S 10 lignes d'entre eux peuvent acceptées des entrées analogiques donc on dispose en tous de 10 canaux analogiques (A0 à A5 + A12 à A15) pour chaque nœud.

Deux des canaux analogiques sont multiplexées avec les entrées de tensions de références externes. Ve_{REF+} est multiplexée avec l'entrée analogique A4, par contre Ve_{REF-} est multiplexée à l'entrée analogique A3. Une seule fonction est choisie à la fois (tension de référence ou entrée analogique) grâce à des registres spéciaux.

II.8.1.2. Le convertisseur A/N:

D'après le datasheet du constructeur le résultat de la conversion N_{ADC} d'un signal V_{in} (figure II-12) est donné par la relation :

$$N_{ADC} = (2^{10} - 1)\frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}} \qquad (1)$$

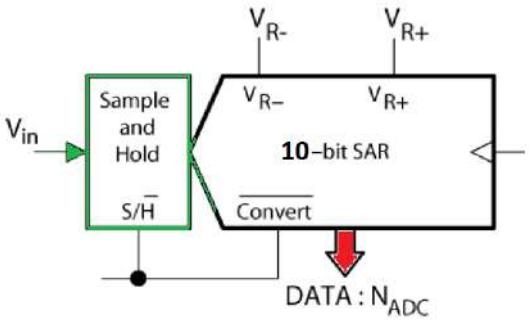


Figure II-12. Le convertisseur A/N

Lors de l'utilisation de l'ADC10 du MSP430, il y a un ensemble de registres à manipulés pour programmer les différentes taches de la numérisation, ces registres sont représentés sur la figure suivante avec leurs abréviation et adresse mémoire :

ADC10	ADC data transfer start address	ADC10SA	1BCh
	ADC memory	ADC10MEM	1B4h
	ADC control register 1	ADC10CTL1	1B2h
	ADC control register 0	ADC10CTL0	1B0h
	ADC analog enable 0	ADC10AE0	04Ah
	ADC analog enable 1	ADC10AE1	04Bh
	ADC data transfer control register 1	ADC10DTC1	049h
	ADC data transfer control register 0	ADC10DTC0	048h

Figure II-13. Les registres de controle de l'ADC10

Dans ce qui suit nous présentons d'une manière brève les registres de contrôle les plus importants que nous avons utilisés :

- ***Les registres ADC10CTL0 et ADC10CTL1:***

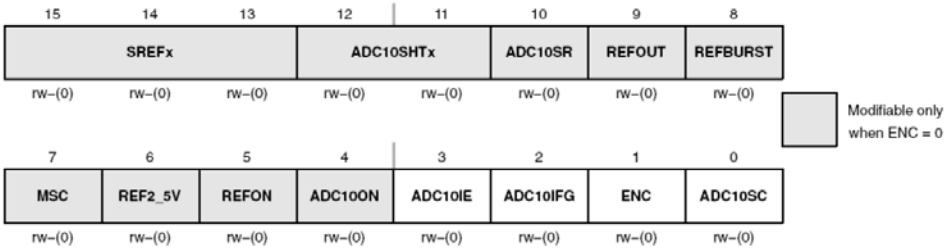


Figure II-14. Le registre ADC10CTL0

Ces registres permettent de contrôler divers paramètres liés au convertisseur, pour le cas du premier registre les bits de ce dernier sont :

- SREF_X : Choix des références de tension V_{R+} et V_{R-}
  - 000 : $V_{R+}=AV_{CC}$ et $V_{R-}=AV_{SS}$
  - 001 : $V_{R+}=V_{REF+}$ et $V_{R-}=AV_{SS}$
  - 01Φ : $V_{R+}=V_{eREF+}$ et $V_{R-}=AV_{SS}$
  - 100 : $V_{R+}=AV_{CC}$ et $V_{R-}=V_{REF-}$ / V_{eREF-}
  - 101 : $V_{R+}=V_{REF+}$ et $V_{R-}=V_{REF-}$ / V_{eREF-}
  - 11Φ : $V_{R+}=V_{eREF+}$ et $V_{R-}=V_{REF-}$ / V_{eREF-}
- ADC10SHT : Durée d'acquisition (nombre de cycles d'horloge ADC10CLK pour la durée de la phase d'acquisition et du maintien ;
- ADC10SR : Fréquence d'échantillonnage ;
- REFOUT : Sortie référence on/off ;
- REFBURST : «Reference burst»
- MSC : Conversion multiple (mode séquentielle ou / et répétitif)
  - 0 : Le « Sample Timer » est déclenché par un flanc montant de SHI
  - 1 : Le premier flanc montant de SHI déclenche le « Sample Timer » les autres conversions se font automatiquement
- REF2_5V : Référence 2,5V (sinon 1,5V); Choix de la référence de tension (REFON doit être mis à 1)
  - 0 : Référence 1.5V,
  - 1 : Référence 2.5.
- REFON : Référence active (ON);
  - 0 : Référence OFF (désactivé)
  - 1 : Référence ON (activé)
- 1 : Référence ON (activé)
- ADC10ON : Ecriture overflow enable ;
- ADC10IE : Interruption activée ;
- ADC10IFG : Flag d'interruption ;
- ENC : Autorise conversion ;
- ADC10SC : Démarre conversion.

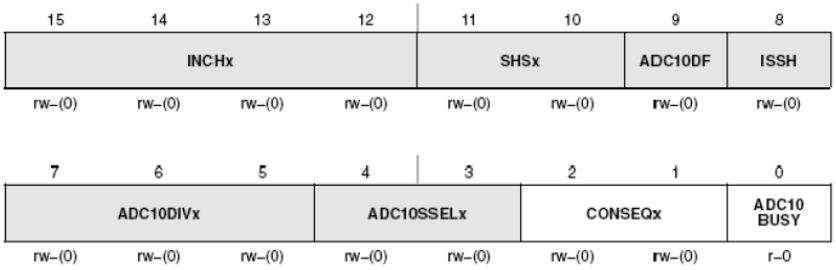


Figure II-15. Le registre ADC10CTL1

Les bits du registre ADC10CTL1 sont :

- INCH : sélection de la Voie (Channel) à convertir ;
  - 00 : ADC10SC (commande software, registre de contrôle)
  - 01 : Timer_A.OUT1
  - 10 : Timer_A.OUT0
  - 11 : Timer_B.OUT1
- SHS_X : Source du démarrage de la conversion ;
- ADC10DF : Format (binaire simple ou complément à 2) ;
- ISSH : Signal inversé ;
- ADC10DIV_X : Diviseur de l'horloge ;
  - 000 : /1
  - 001 : /2
  - 010 : /3
  - 011 : /4
  - 100 : /5
  - 101 : /6
  - 110 : /7
  - 111 : /8
- ADC10SSEL_X : Source de l'horloge ;
  - 00 : ADC10 OSC
  - 01 : ACLK
  - 10 : MCLK
  - 11 : SMCLK
- CONSEQ_X : Mode de conversion ;
  - 00 : conversion simple d'un seul canal
  - 01 : conversion séquentielle de plusieurs canaux
  - 10 : conversion répétitive d'un seul canal
  - 11 : conversion séquentielle répétitive de plusieurs canaux
- ADC10BUSY : indication que l'ADC10 est occupé.

Pour les autres registres :

- ADC10AEx : permet d'activer le CAN « Analog enable » ;
- ADC10MEM : Buffer du résultat ;
- ADC10DTCx : permet le Transfer direct ;
- ADC10SA : spécifie l'adresse initiale pour le module DTC.

La programmation de l'ADC suit le modèle général suivant :

- Choisir une fréquence d'échantillonnage par rapport à une fréquence de référence ;
- Choisir un canal à numériser ;
- Attendre la stabilisation du signal à convertir ;
- Demandé la lecture du convertisseur ;
- Attendre la validation ;
- Possibilité de recevoir une interruption qui signale la fin de conversion si besoin.

L'organigramme de la figure suivante présente d'une manière générale les différentes étapes du programme de conversion des signaux analogiques des capteurs interfacés avec les nœuds 1 et 2 :

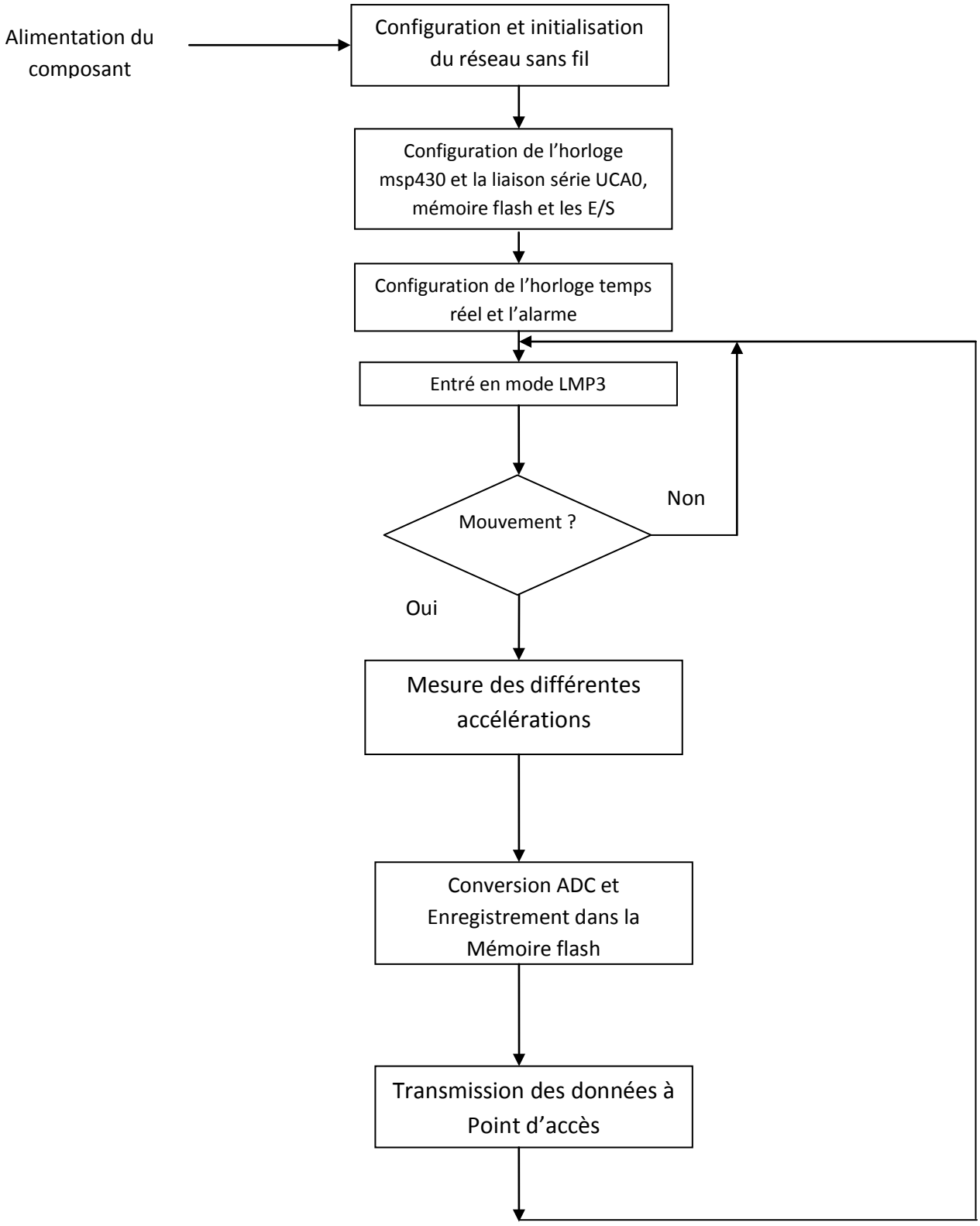


Figure II-16. Organigramme de L'End Device 1 et 2

A la première mise sous tension l'End Device 1 ou 2 s'occupera de la configuration du réseau sans fil et de son horloge temps réel, ensuite l'MSP430 entrer en mode veille pour réduire sa consommation électrique en attendant une interruption qui sera générée par le mouvement du segment sur lequel les capteurs sont fixés. En effet une fois l'interruption détectée la routine d'interruption va lancer la procédure de mesure de l'acceleration à travers le convertisseur analogique numérique, ainsi les valeurs des entrées analogiques sont lues, par la suite la donnée est mise en forme et stocker dans la mémoire flash, enfin le contenu sera envoyé à l'Access Point. Sinon le microcontrôleur se remettra en mode basse consommation et attendra un nouveau réveil.

II.8.2. Le module USCI_B0 [12]:

La Communication I2C nécessaire pour le noeud de capteur n° 3 avec le MSP430 exploite le module de communication USCI_B0 du microcontrôleur.

II.8.2.1. Présentation :

Le bus I2C est caractérisé par une liaison en mode série réalisée à l'aide de 2 fils. C'est la société Philips qui en à créer le concept au début des années 80. Son succès est lié à sa simplicité. La figure II-17 présente l'architecture type d'un bus I2C, on remarque que plusieurs composants viennent se "greffer" sur le même bus. Les données transitent par les lignes :

- SDA : signal de donnée, généré par le Maitre ou l'Esclave.
- SCL : signal d'horloge généré par le Maitre.

La communication sur le bus est orchestrée de la manière suivante :

Le Maitre envoie sur le bus l'adresse du composant avec qui il souhaite communiquer, chacun des esclaves ayant une adresse fixe, l'esclave qui se reconnaît répond à son tour par un signal de confirmation, puis le Maitre continue la procédure de communication... .(écriture/lecture). Dans tous les cas, les transactions seront confirmées par un ACK

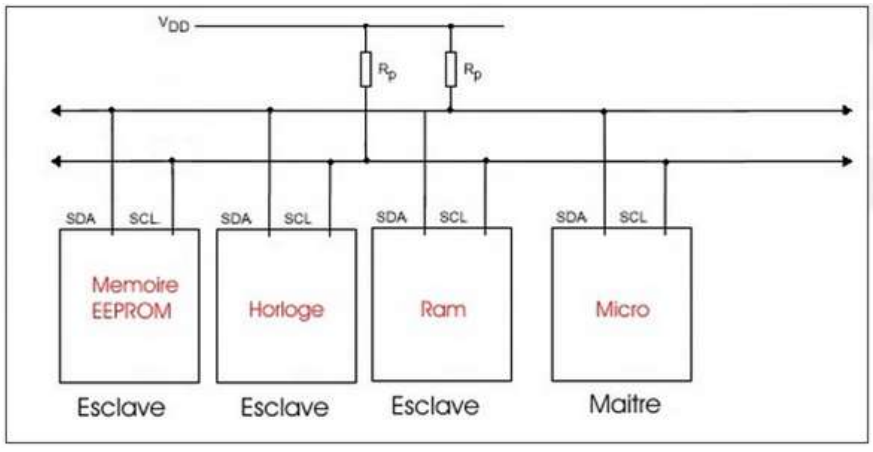


Figure II-17. La communication sur le bus I2C

II.8.2.2. Caractéristiques électrique :

- Fréquence maximale de fonctionnement **:** 400 KHz (variable suivant les composants connectés au bus, voir datasheet du composant) ;
- Etat logique Haut **: de 3 à 5 Volts ;**
- Etat logique Bas **: de 0 à 1.5 Volts ;**
- Capacité maximum admissible sur le bus **: 400 pF ;**
- Temps de monté des signaux **: inférieur à 1 µs ;**
- Résistance de rappel **: à calculer en fonction de l'impédance du Bus (valeurs pratiques constatées de 1,5 K à 3,5 K) ;**
- Distance de communication **: de quelque dizaines de centimètres à plusieurs mètres avec un tampon.**

II.8.2.3. Les bases du protocole :

La condition de repos : Condition dans la quel aucun composants ne communiquent, les lignes de bus sont à l'état Haut

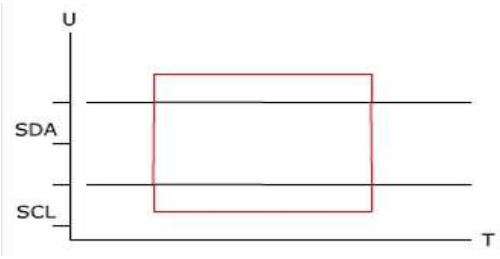


Figure II-18. L'état de repos du bus I2C

La condition de départ : Le Maitre force la ligne SDA au niveau bas pour "réveiller" les Esclaves quand SCL est au niveau haut

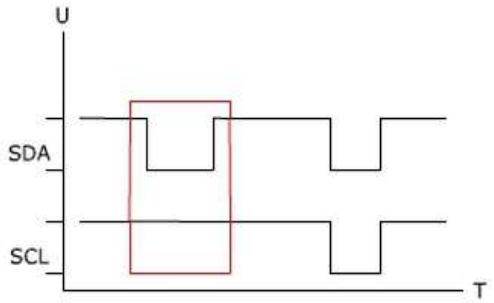


Figure II-19 La condition de départ du bus I2C

La condition d'arrêt : Les lignes SDA et SCL sont au niveau bas, puis quand SCL passe au niveau haut, le Maitre libère la ligne SDA au niveau haut. A cet instant le bus est considéré comme disponible

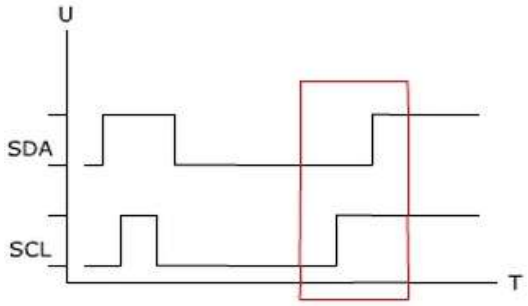


Figure II-20 La condition d'arret du bus I2C

L'acquittement : Une fois les données transmises, un acquittement est opéré par celui qui reçoit les données. Cette procédure est réalisée en fin de trame de transmission, soit à la neuvième impulsion. Le récepteur maintien la ligne SDA au niveau bas pendant le front d'horloge :

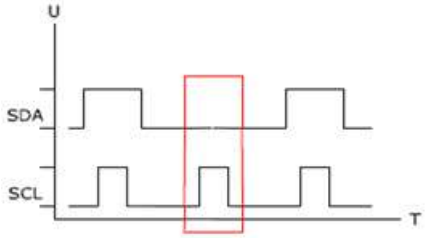


Figure II-21 L'acquitement sur le bus I2C

II.8.2.4. L'adressage :

Pour pouvoir communiquer avec les différents composants raccordés au bus, le Maitre indique avec quel Esclave il souhaite se connecter, pour cela il enverra sur le bus l'adresse de l'Esclave composé de 7 bits, puis un dernier indiquant le sens du transfert : Lecture (1) ou Écriture (0). L'esclave reconnaissant sont adresse validera par un acquittement.

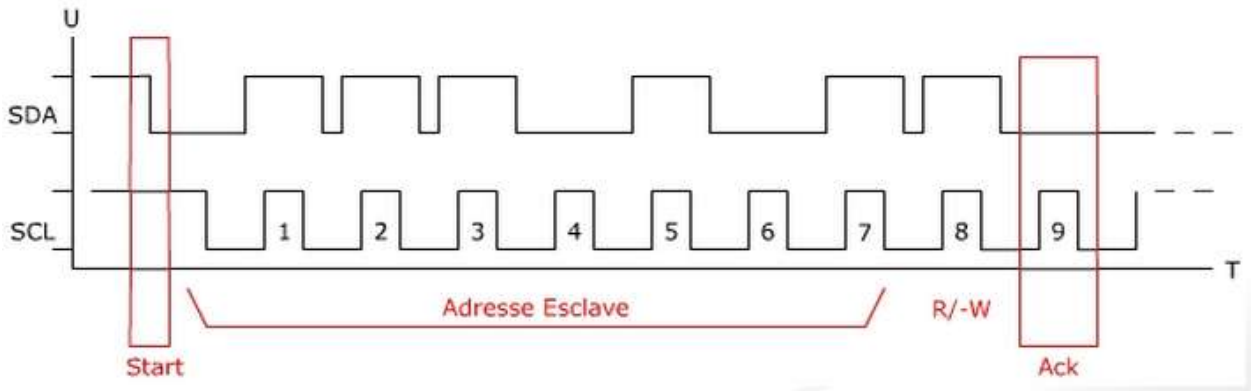


Figure II-22. L'adressage sur le bus I2C

II.8.2.5. Trames de Communications :

Nous avons vu précédemment les opérations de base effectuées sur le Bus I2C, voyons maintenant comment assembler ces "morceaux" afin de constituer une trame complète. Dans un premier temps, il faut réveiller le bus en réalisant une condition de départ, ensuite viennent s'intercaler les trames de données. L'Esclave ayant reçu les informations acquittera le Maitre pour lui faire savoir qu'il à bien reçu la trame de données, dans le cas contraire, c'est au maitre de recommencer les transactions depuis le départ. Pour finir le signal de Stop mettra fin à toute communication.

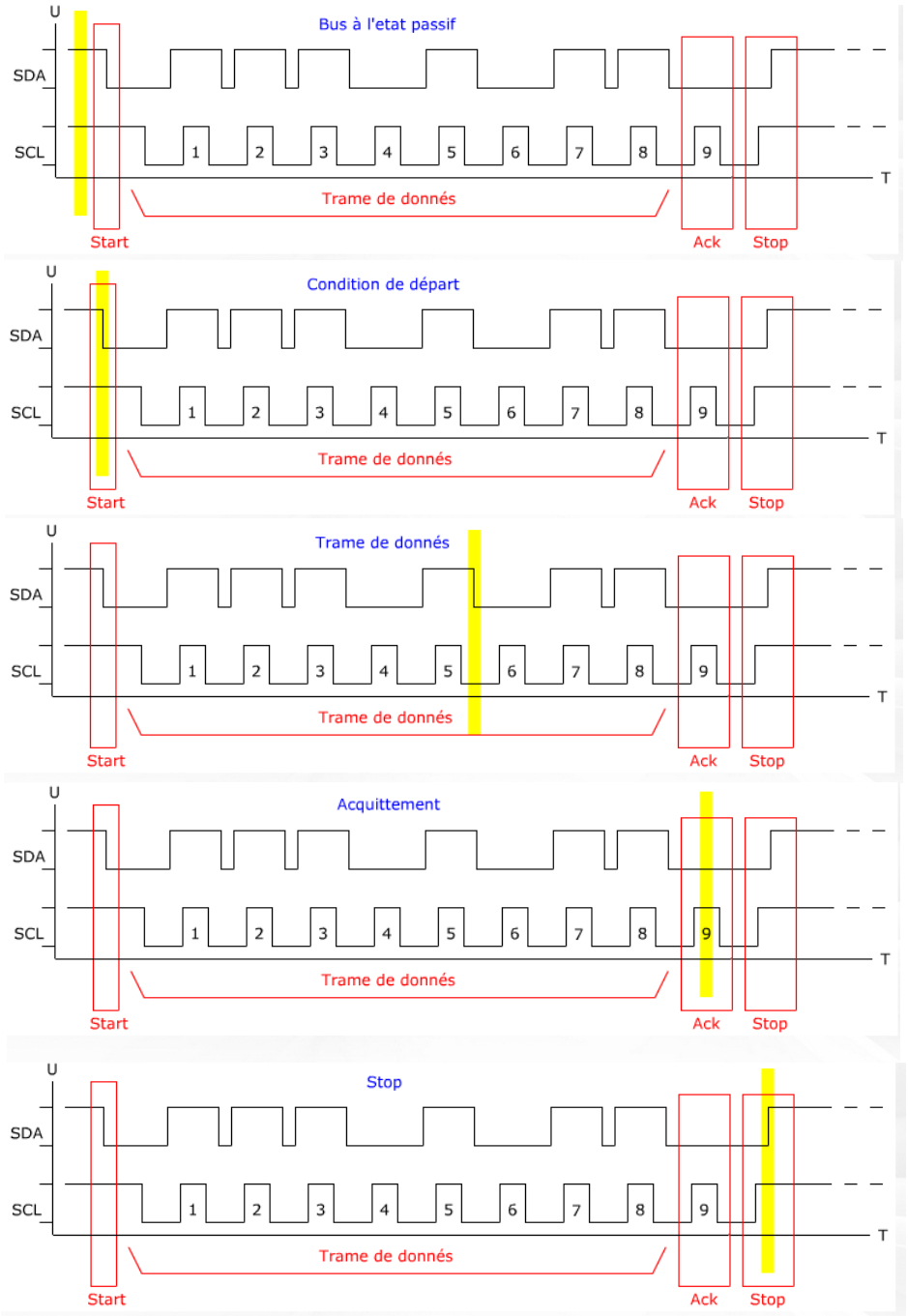


Figure II-23. Structure d'une trame de communication

II.8.2.6. La communication par octet :

Nous allons prendre comme exemples les cas suivants afin de visualiser à quoi peut ressembler un échange complet sur le bus entre Maitre et Esclave.

Exemple : Écriture de donnés dans une mémoire EEPROM I2C :

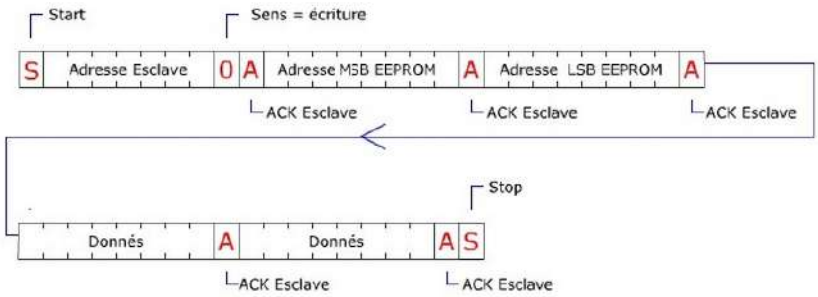


Figure II-24. Écriture de donnés dans une mémoire EEPROM I2C

La séquence se décompose ainsi :

- Condition de départ;
- Emission de l'adresse esclave, soit de L'EEPROM;
- Acquittement de la part de l'esclave;
- Emission par le maitre de l'adresse mémoire à laquelle il souhaite inscrire les futures données, acquittement successif de l'esclave;
- Transmission des données à écrire, acquittement à chaque reception de paquet par l'esclave;
- Condition de stop;

On remarque que le maitre, après avoir émis l'adresse de stockage des donnés, n'a pas besoin d'interrompre la communication à chaque octet et d'écrire à nouveau l'adresse de pointage. En règle générale les composants I2C sont à 'auto-chargement', c'est à dire que leur pointeur d'adresse s'incrémente automatiquement en fin d'émission et ce pour la valeur de 8 octets.

II.8.2.7. Les registres de contrôle de la communication I2C [12] *:*

Pour notre microcontrôleur MSP430 nous disposons d'un ensemble de registre pour implémenter une communication I2C (figure II-25):

No	Registre	Nom court	Bits à configurer
1	Port 3 select register	P3SEL	Bits 1 et 2
2	USCI_B1 control register 1	UCB0CTL1	UCSWRST, UCSSELx
3	USCI_B1 control register 0	UCB0CTL0	UCMST + UCMODE_3 + UCSYNC
4	USCI_B0 bit rate control register 1 & 0	UCB0BR1, UCB0BR0	tous
5	USCI_B0 I2C slave address register	UCB0I2CSA	tous

Registres à initialiser

No	Registre	Nom court	Bits à utiliser
6	USCI_B1 transmit buffer register	UCB0TXBUF	tous
2	USCI_B1 control register 1	UCB0CTL1	UCTR, UCTXSTT, UCTXSTP
7	SFR interrupt enable register 2	IE2	UCB0TXIE

Registres à utiliser lors de communication

Figure II-25. Les registres de controle de la communication I2C

Il faut toujours garder à l'esprit que le bus I2C est un bus multi-master, et ***que*** les esclaves peuvent mettre plus ou moins de temps à réagir. Il faut donc:

- Attendre que le bus soit libre avant de communiquer ;
- Que l'esclave acquitte les communications.

Voici en résume les étapes d'une communication I2C avec le MSP430 :

1. Dans le début du programme : Configuration des pins I/O utlisées pour la communication I2C.
2. Configuration des registres UCB0CTL0 et UCB0CTL1 du MSP430 régissant la communication I2C par le module USCI. Il faut mettre le module USCI_B0 en reset pendant sa configuration à l'aide du bit UCSWRST. Enlever le reset à la fin de la configuration.
3. Configuration de l'adresse de l'esclave par le registre UCB0I2CSA.
4. Le maître initie la communication par une «start condition».
5. Le maître envoie l'adresse du périphérique esclave avec qui il veut communiquer (7 bits), ainsi qu'un bit R/W indiquant le sens des données à suivre.
6. Les données sont transmises. Il peut y avoir plusieurs octets.
8. Le maître termine la communication par une « stop condition ».

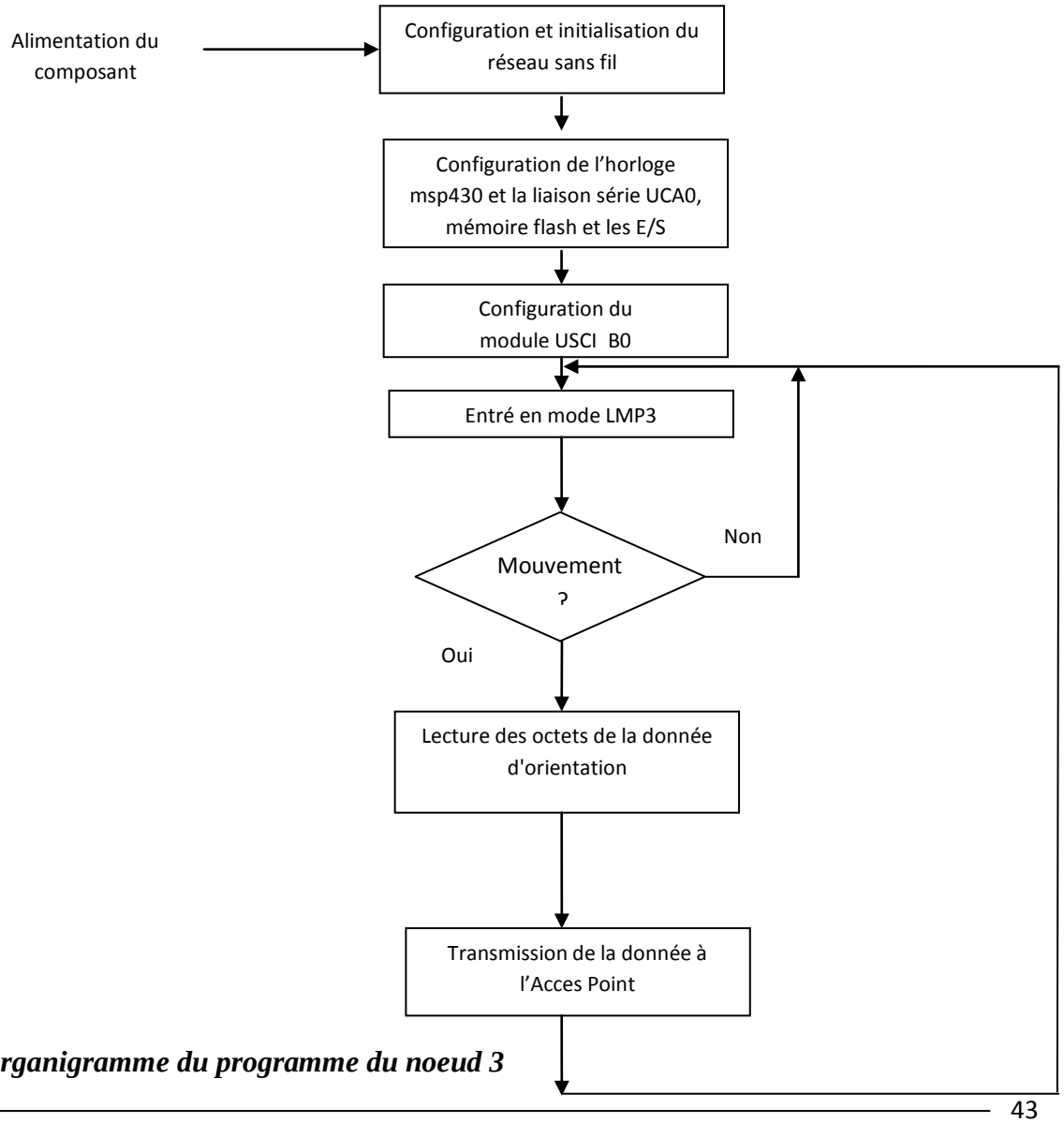


Figure II-26. Organigramme du programme du noeud 3

II.8.3. Le programme du nœud station de base (Acces point) :

Le rôle d'un Access Point et de récupérer les données transmises par les différents End Device (Noeuds capteurs) et de les envoyer par liaison série au PC qui s'occupera de la collecte des informations. L'organigramme de la figure suivante résume les étapes de base du programme implémenté sur la station de base:

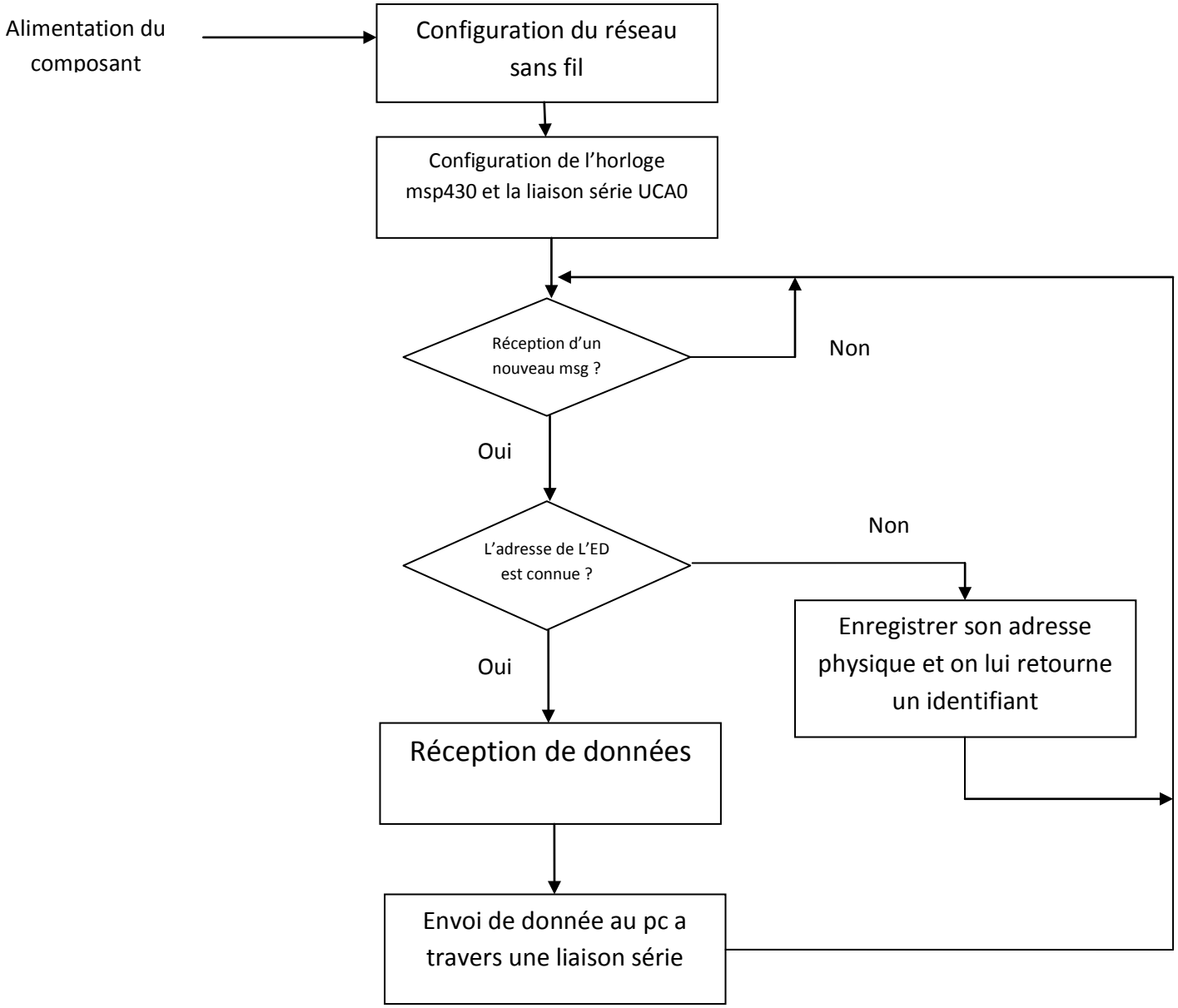


Figure II-27 Organigramme du programme de Point d'Accés

II.9. Conclusion :

Les données échangées à travers le réseau de capteur sans fil de notre application utilise le protocole SimpliciTI, la communication se fait sous forme de paquets, contenant en plus du message à transmettre un nombre d'information concernant l'émetteur, le destinataire et le contenu du message.

Chapitre III :

Mise en marcheet Tests

III .1 Introduction :

Dans ce qui suit nous présentons les étapes qui nous ont permis de développer la partie soft de notre application ainsi que la démarche suivie pour tester les programmes élaborés.

III.2 La simulation:

En premier lieu nous avons voulus commencer le développement sur PC à l'aide d'un logiciel de simulation pour apprendre la programmation des MSP430 dans un temps assez court, pour cela nous avons choisie le programme ISIS Proteus de "Labcenter Electronic". Malgré que ce logiciel permet de simuler les microcontrôleurs MSP430, le modèle F2274 n'est pas pris en charge par Proteus, pour contourner ce problème nous avons choisie le modèle F2272 qui ressemble à notre microcontrôleur (MSP430F2274), la seule différence réside dans la taille de l'espace mémoire et la présence des AOP sur le F2274, ce qui n'a pas une grande influence sur nos essais.

- **Proteus** est une suite de logicielle permettant la Conception Assitée par Ordinateur de circuits électroniques analogique et numérique, il prend en charge un grand nombre de compsants programmable (microcontroleurs, microprocesseurs, mémoires,...). Cette suite logicielle éditée par la société Labcenter Electronics est composé de deux programmes principaux : ISIS, permettant entre autres la création de schémas et la simulation électrique, et ARES, dédié à la création de circuits imprimés.

Grâce à des modules additionnels, ISIS est également capable de simuler le comportement d'un microcontrôleur (PIC, Atmel, 8051, ARM, HC11, MSP430...) en exécutant son programme embarqué et simule son interaction avec les composants qui l'entourent.

Nos premiers tests sur le MSP430F2272 ont été orientés pour faire fonctionner le convertisseur analogique numérique ADC intégré sur le MSp430. Nous avons implémenter au début un simple programme permettant la manipulation des registres de l'ADC10. Le code écrit en langage C sous l'IDE IAR nous a permet de convertir un signal analogique généré par une source de tension en un signal numérique qui a été envoyer sur le Port P.1 relié à un ensemble de pour vérifier l'état de fonctionnement du programme.

La figure suivante présente le code simulé sur Proteus :

```
#include "msp430x22x2.h"
int A;
void main(void)
{
  WDTCTL = WDTPW + WDTHOLD;                  // Arrêter WDT
  ADC10CTL0 = ADC10SHT_2 + ADC10ON + ADC10IE; // ADC10ON, interruption permise
  ADC10AE0 |= 0xFF;                          // P2.0 ADC Option choisie
  P1DIR = 0xFF;                              // déclarer tout P1 en sorties

  for (;;)
  {
    ADC10CTL0 |= ENC + ADC10SC;              // Echantillonnage et début de conversion
    __bis_SR_register(CPUOFF + GIE);         // LPM0, ADC10_ISR forcera la sortie

    if (ADC10MEM < 0x1FF)
    A = ADC10MEM;
    P1OUT = A;

  }
}

// ADC10 routine de service d'interruption
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void)
{
  __bic_SR_register_on_exit(CPUOFF);         // Effacer CPUOFF bit depuis 0(SR)
```

Figure III-1. Programme de conversion d'un canal analogique avec MSP430F2272

Après compilation du programme sous IAR, nous nous somme heurté à un problème de simulation, ce problème est lié au fichier de sortie du compilateur C de l'IDE IAR, car ISIS travail avec les formats : ***.hex*** et ***.cof*** qui ne sont pas les extensions par défaut de IAR. La solution a consisté à apporter des modifications sur les options de compilations dans l'IDE IAR, cette procédure est décrite sur les figures III-1 et III-2 :

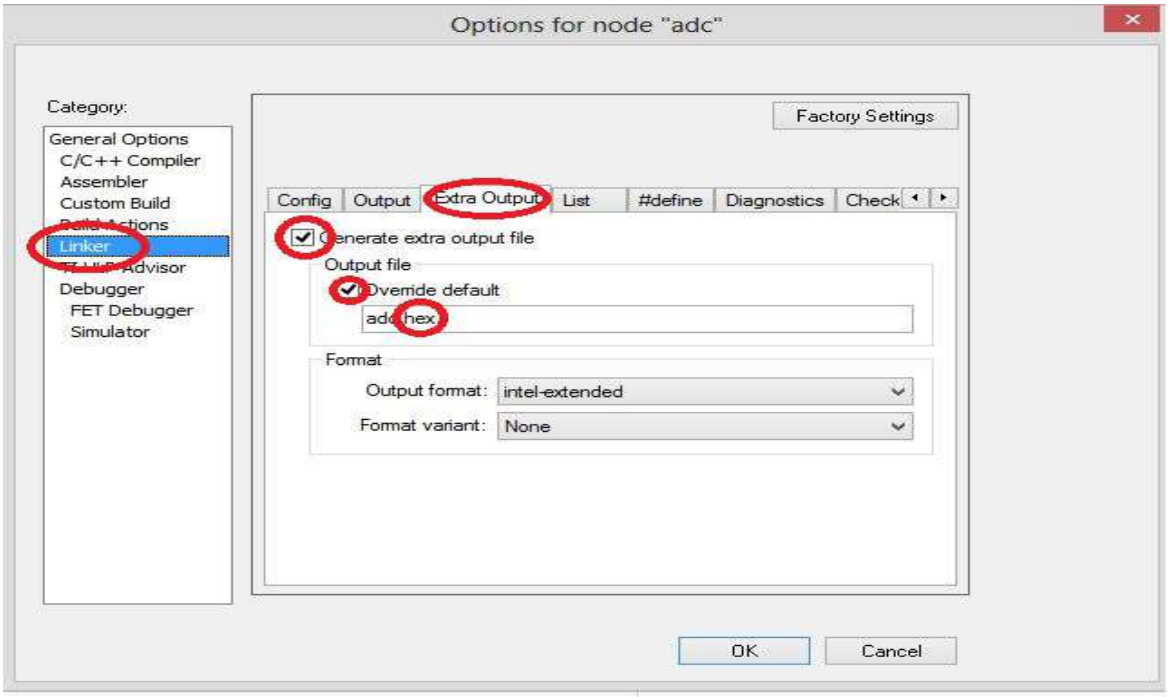


Figure III-2. Modification du format de fichier de sortie IAR

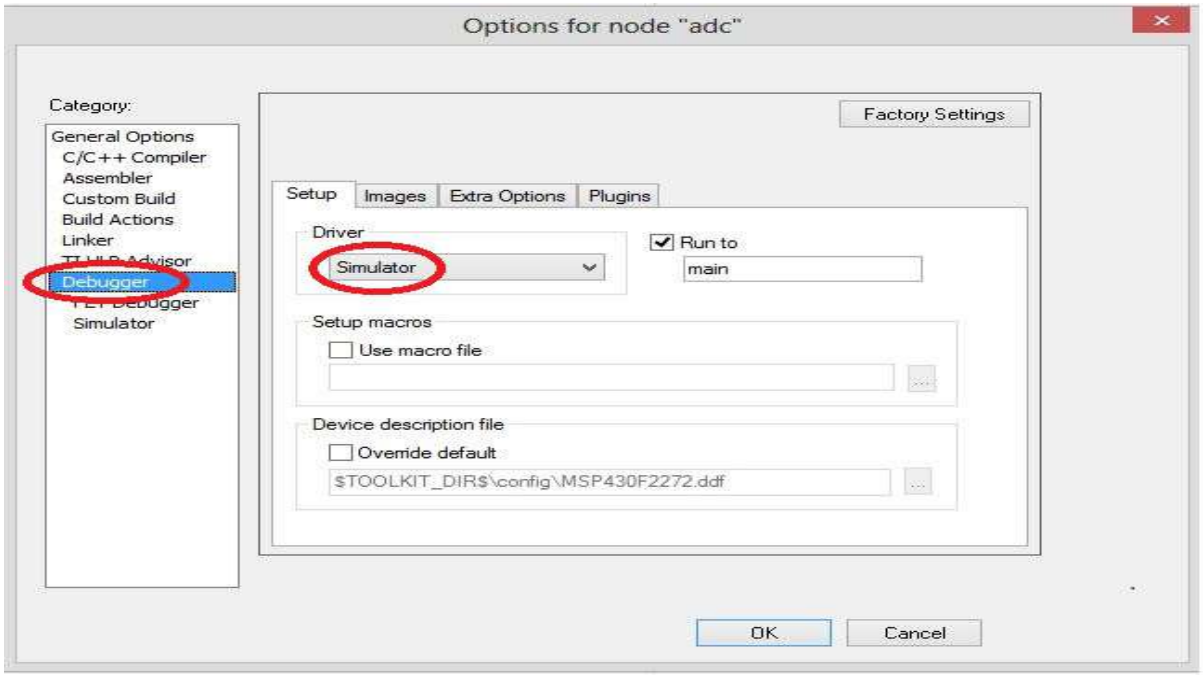


Figure III-3. Configuration des options de Débougage

Après exécution des modifications on a recompilé le programme et nous avons procédé à la simulation du programme sur ISIS, les Figures III-4 et III-5 montre le circuit de simulation utilisé sur ISIS ainsi que les résultats après variation de l'amplitude du signal d'entrée :

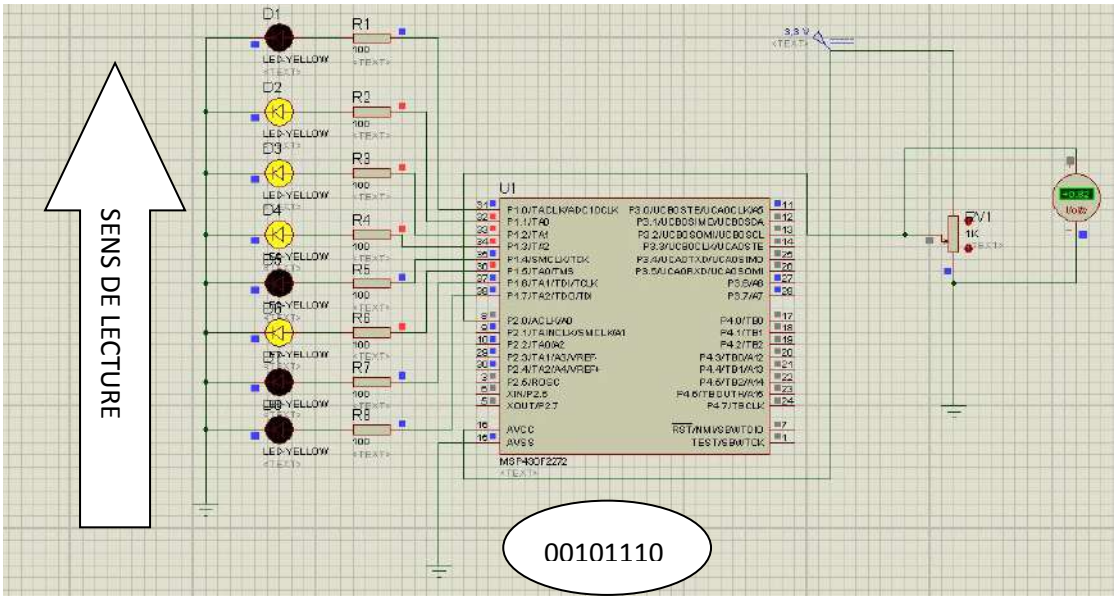


Figure III-4. Numérisation d'un signal continue d'amplitude 0.8 V

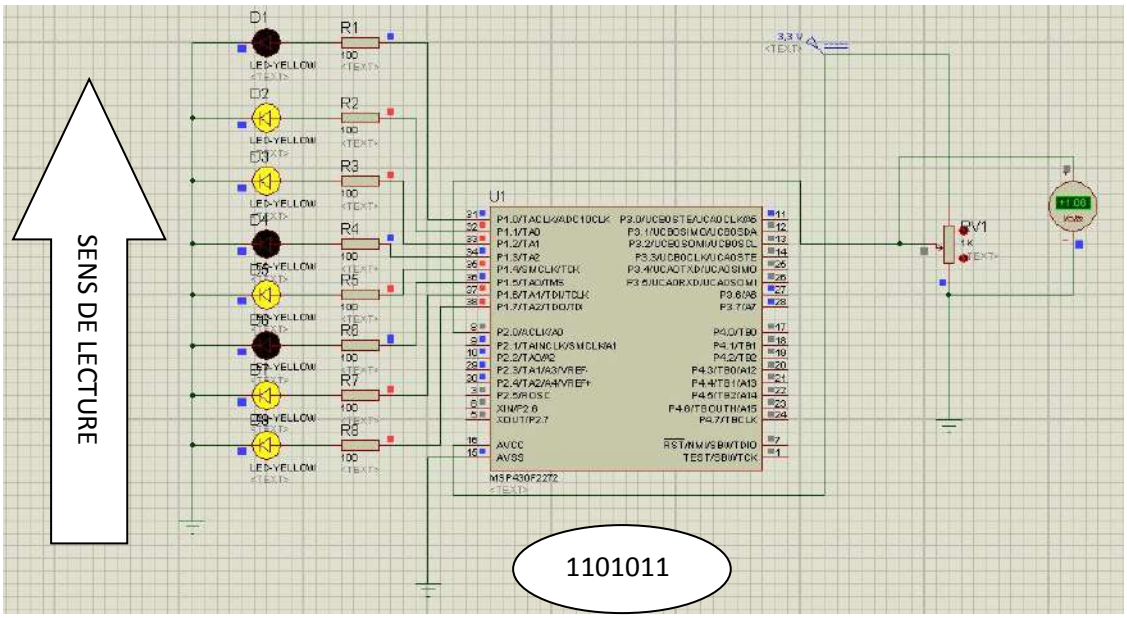


Figure III-5. Numérisation d'un signal continue d'amplitude 1.06 V

Pour vérifier si le programme de conversion A/N assure une numérisation correcte nous avons utilisé la formule suivante pour comparer les résultats avec les amplitudes affichées par le voltmètre sur le simulateur :

$$\text{Valeur}_{\text{Digital}} = 1023*(\text{Vin analog} - V_{ref-}) / (V_{ref+} - V_{ref-})$$

Dans nos tests nous avons choisie $V_{ref+} = V_{CC}$ (3.3 V) et $V_{ref-} = V_{SS}$ (0 V)

III-3 Les tests pratiques:

Lors de la deuxième phase de tests nous nous somme orienté vers des tests pratiques, pour cela il fallait choisir des capteurs à sorties analogiques et un compas à interface I2C pour pouvoir effectué les tests.

III.3.1. Les accéléromètres utilisés :

Ce type de capteurs est le plus utilisé pour la mesure des signaux dynamiques liés au mouvement, à la détection des chutes et des chocs chez l'être humain.

Un accéléromètre est un capteur qui, fixé à un mobile, permet de mesurer l'accélération de ce dernier. Le principe de tous les accéléromètres est basé sur la loi fondamentale de la dynamique : F = M x a (F : force, M : masse et a : l'accélération).

Plus précisément, il consiste en l'égalité entre la force d'inertie de la masse sismique du capteur et une force de rappel appliquée à cette masse. On distingue deux grandes familles d'accéléromètres : les accéléromètres non asservis et les accéléromètres à asservissement.

Les accéléromètres permettent de mesurer l'accélération d'un point particulier dans un système en cours de test. Cependant, ce signal correctement intégré par rapport au temps permet également de quantifier une vitesse et un déplacement.

Les accélérations sont généralement exprimés en unités du Système International (m/s^2). La constante gravitationnelle "**g**" est souvent utilisée en industrie pour exprimer une accélération, mais cette dernière ne fait pas partie des Unités du Système International. Heureusement, la conversion en USI peut facilement s'effectuer avec une erreur inférieure à 2% puisque : 1,0 g = 9,81 m/s^2 . Les accéléromètres simples (1D) mesurent la composante de l'accélération dans une direction particulière. Si trois de ces accéléromètres sont combinés dans un unique capteur (accéléromètre 3D), l'accélération peut être mesurée simultanément dans les trois directions orthogonales.

III3.1.1.Fonctionnement de l'accéléromètre :

La plupart des accéléromètres se fondent sur l'utilisation de l'effet piézoélectrique. Dans le cas des accéléromètres à compression, l'accélération de la structure d'essai est transmise à une masse séismique qui produit une force proportionnelle sur le cristal piézoélectrique. Cet effort externe sur le cristal produit alors une charge électrique proportionnelle à la force appliquée et donc à l'accélération. Une autre conception répandue est le mode cisaillement, qui est moins sensible aux effets de la température (la charge est prise dans une direction perpendiculaire à la direction de polarisation).

L'accéléromètre piézoélectrique présente certaines caractéristiques importantes :

• Très large gamme de fréquences et d'accélérations et bonne linéarité sur sa gamme de fonctionnement

• Robustesse et fiabilité dues à l'absence de pièces mobiles

• Stabilité dans le temps

• Alimentation externe inutile

Certains accéléromètres utilisent une électronique intégrée (ICP) et nécessitent une source de courant constant pour alimenter l'amplificateur de charge intégré au capteur.

Dans nos tests pour des raisons de disponibilités nous avons choisie d'utiliser un accéléromètre MEMS 3 axes de très faibles dimensions qui est le MMA7361 de Freescale.

C'est un excellent capteur avec une interface analogique très simple. Il s'alimente en 3.3V et délivre en sortie une tension analogique sur chacune des 3 sorties. Cette tension est directement proportionnelle au rapport de l'accélération mesurée divisée par la tension d'alimentation (rapport métrique). Sa sensibilité est réglable par sélection des entrées adéquats.

Spécifications:

- Tension d'alimentation : 3.3 à 8V;
- Sensibilité réglable : ±15g / 6g;
- Consommation d'énergie très Basse : 500µA en mode mesure, 3µA en mode standby;
- Haute sensibilité : 800 mV/g pour la sensibilité 1.5g ;
- Filtre passe bas intégré;
- Taille : 23 x26 mm;
- Poids : 5 g;

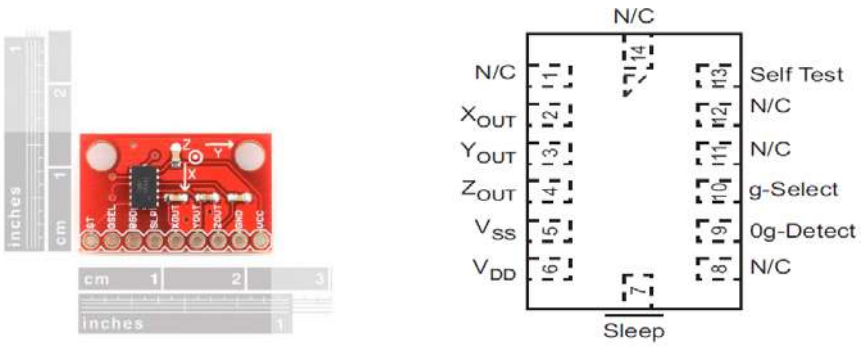


Figure III-6. L'accéléromètre MMA7361 monté sur un PCB & brochage du capteur

Applications typiques: ce capteur est très utilisé on le trouve dans :

• Les Jeu 3D : pour la mesure des Inclinaison et mouvement, enregistreur d'événement;

• Les HDD MP3 Player : Détection de chute libre;

• Les PC d'ordinateur portable : Détection de chute libre, antivol;

• Les Téléphone portable : Stabilité d'image, mouvement composant, E-Boussole

• Les Pédomètres : Détection de mouvement;

• Les PDA : dispositif de roulement des textes;

• Robotique : Détection de mouvement; ...etc.

La figure suivante montre le schéma fonctionnel de ce capteur:

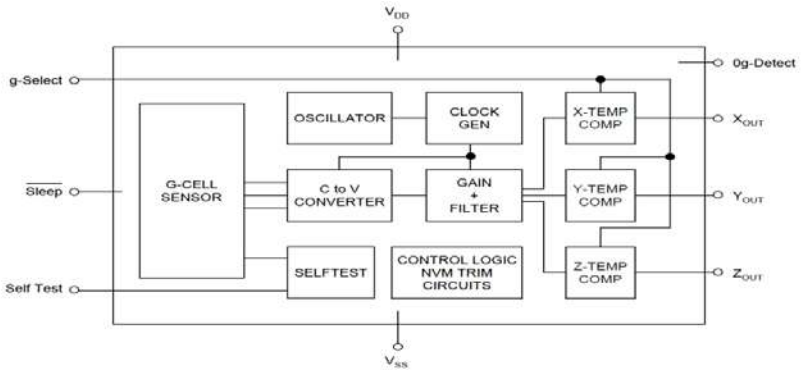


Figure III-7. Schéma fonctionnel simplifié d'accéléromètre

Sélection de la gamme de mesure :

Le dispositif tient compte du choix entre deux sensibilités. Selon l'entrée logique placée sur la borne 10, le gain interne du dispositif sera changé en lui permettant de fonctionner avec une sensibilité de 1.5 g ou 6 g (tableau suivant), ce dispositif est idéal aux applications qui exigent deux différentes sensibilités. La sensibilité peut être changée à tout moment pendant le fonctionnement, la ligne de sélection g-select peut être laissé non lié pour des applications qui exige seulement une sensibilité de 1.5 g.

g-Select	g-Range	Sensitivity
0	1.5g	800 mV/g
1	6g	206 mV/g

Tableau III-1. Sélection de la gamme de mésure

III-3-2. Le compas électronique :

Le compas choisie est module CMPS10 qui est une boussole compensée en inclinaison. Employant un magnétomètre 3 axes, un accéléromètre à 3 axes et un puissant processeur 16-bit, le CMPS10 a été conçu pour éliminer les erreurs causées par l'inclinaison du compass par rapport à l'horizontal lors des mesure de l'orientation..

Le CMPS10 produit un résultat de 0 à 3599 représentant les angles de 0 à 359,9 ° ou de 0 à 255. Ce module nécessite une alimentation de 3,3 à 5 V et consomme un courant nominale de 25mA. Il y a trois façons d'obtenir la mesure du cap, par interface série, par interface I2C ou par sortie PWM.

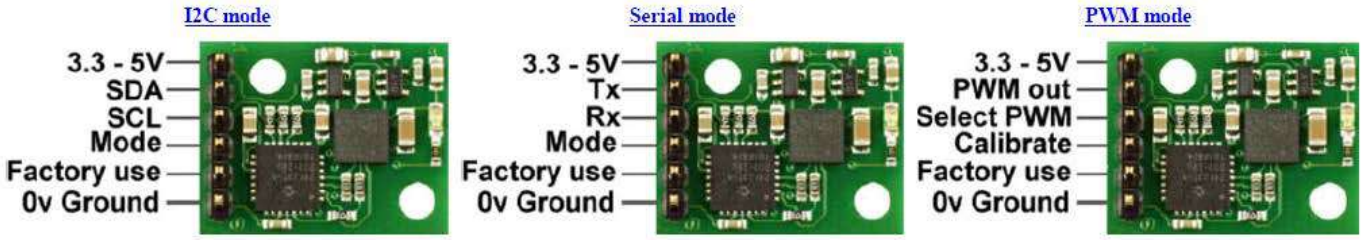


Figure III-8. les modes de selections

III.4. L'interface de visualisation sur PC:

Afin de voir l'évolution des signaux mesurés nous avons réalisé une interface très simple sous Matlab pour dessiner les courbes des accélérations mesurées, La figure suivante montre l'aspect de cette interface:

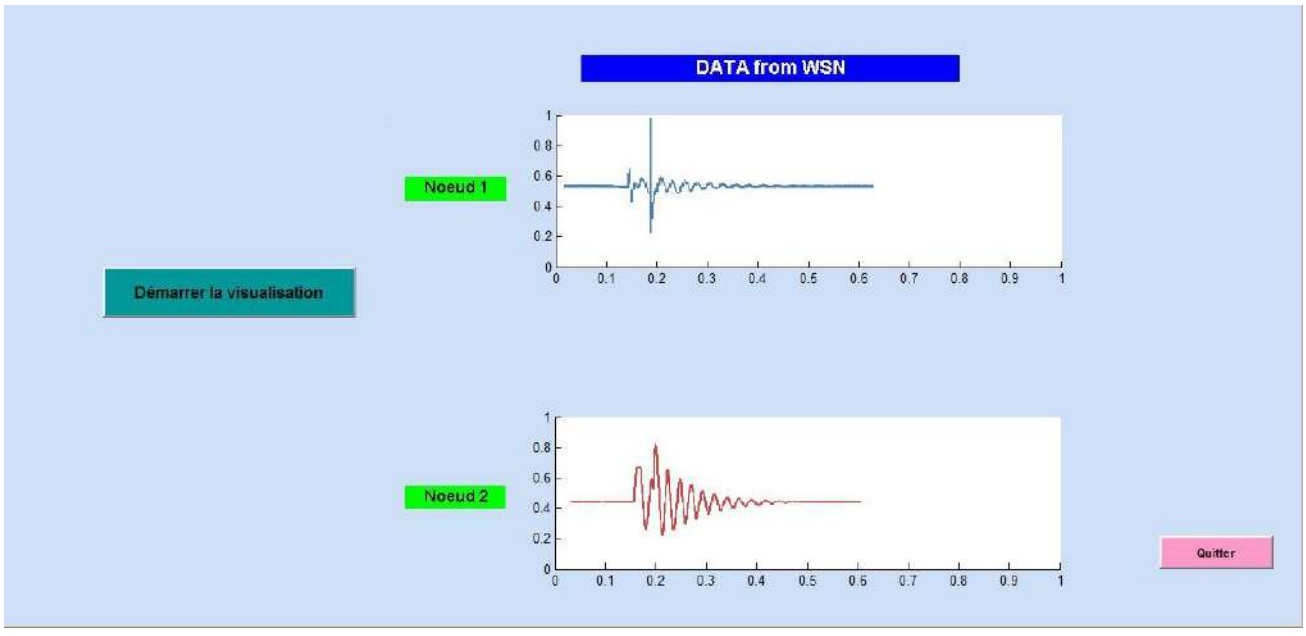


Figure III-9. L'interface de visualisation des courbes sous Matlab

Cette interface est basée sur l'organigramme suivant:

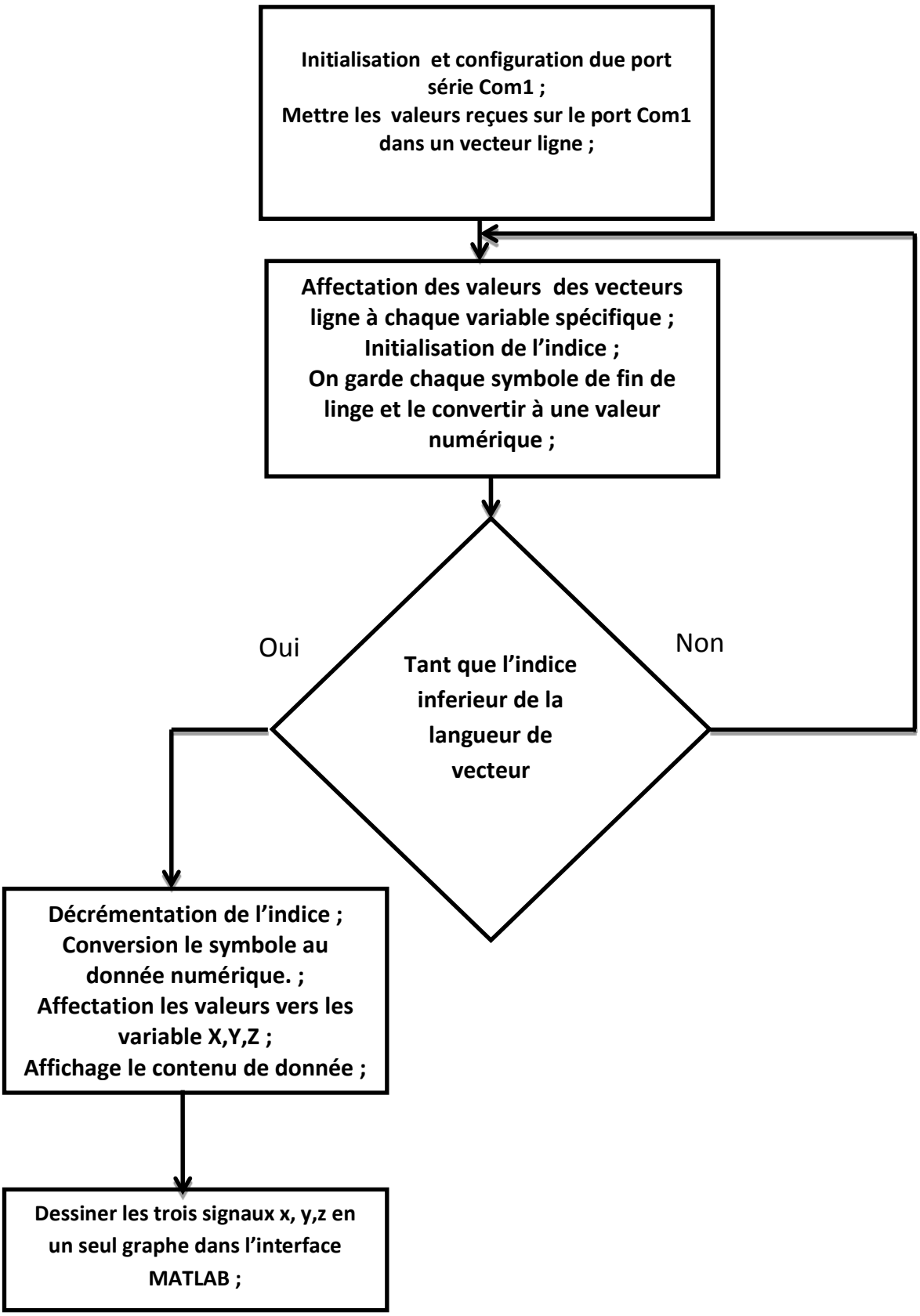


Figure III-10. Lorganigramme de l'interface de visualisation

Le code suivant permet de créer l'interface de visualisation des courbes

```
%Réception depuis le port série (COM1) et représentation des données

%partie1 : Déclaration et initialisation

clear all
s = serial('COM1');
set(s,'BaudRate',9600,'DataBits',8,'Parity','none');%,'InputBufferSize',28);
fopen(s);
s.ByteOrder = 'bigEndian';
l = 200;
i = 2;
t = [];
x = [];
y = [];
z = [];
t(1) = 0;


%partie2 : Prendre les derniers data depuis le vecteur S

data = fgets(s);
data = fgets(s);
tic
data2 = str2num(data); % conversion de symbole à des valeurs numerique
x(1) = data2(1);
y(1) = data2(2);
z(1) = data2(3);

%partie3 : Affectation des vecteur X,Y et Z depuis le vecteur S

while(i < (l + 2))
data = fgets(s);
data = fgets(s);
t(i) = t(i-1) + toc;
tic
data2 = str2num(data);
x(i) = data2(1);
y(i) = data2(2);
z(i) = data2(3);
i = i + 1;
disp(data)
end

%partie4 : Ploter tous les vecteur sur le même graphe

n = toc;
fclose(s);delete(s);clear s
x = (x - 474)./85.3;
y = (y - 481)./85.3;
z = (z - 478)./85.3;
plot(t,x,t,y,t,z)
axis([0 t(l) -6 6])
```

Figure III-11. Le code Matlab de l'interface de visualisation

III.5 - Conclusion :

Les tests éffectué avec un noeud et la station de base ont été réalisés avec succés pour les capteurs accéléromètres, mais pour le cas du compas les tests n'ont pas été satisfaisantes malgré tous les efforts apportés, reste à rétudié la communication série par liaison I2C pour trouver les bugs du programmes du noeud de capteur n°3.

Conclusion générale

Conclusion Générale

Les réseaux de capteurs sans fil (RCSF) sont une nouvelle technologie qui a surgit après les grands progrès technologie concernant le développement des capteurs intelligents, des processeurs puissants et des protocoles de communications sans fil. Ce type de réseau composé de centaines ou de milliers d'éléments, a pour but la collecte de données de l'environnement, leur traitement et leur dissémination vers le monde extérieur.

Les applications des réseaux de capteurs sans fil sont nombreuses. Elles comprennent différents domaines : agricole, militaire,... Pour que ces réseaux puissent mener à bien leurs missions ils doivent assurer un certain niveau de sécurité qui différent selon l'application déployée.

Dans ce travail nous avons présenté la mise au point d'une plate-forme de réseau de capteur sans fil à base du Kit eZ430 RF2500 de Texas Instruments qui permet d'interfacer des capteurs accéléromètres, un compas à interface I2C.

Il est important de maintenir les éléments du système physiquement petits pour s'assurer qu'il peut être utilisé dans la plupart des situations où l'accélération devrait être mesurée. Dans cette application l'accéléromètre est très petit, léger, et mesure jusqu'à 6g.

Le réseau de capteurs accéléromètres sans fil pourrait être un circuit qui pourrait être utilisé pour beaucoup d'applications dans le domaine du sport.

Nous espérons que ce travail apportera une aide aux étudiants d'instrumentation et de télécoms et qu'il soit compléter par la mise en marche de l'interface I2C et de l'interface SPI puisque un grand nombre de capteur modernes sont ineterfaçable par le bais de ces liaisons.

Annexe

Programme d'AP :

```
//Fichier de Point d'accès


    #include "bsp.h"
    #include "mrfi.h"
    #include "bsp_leds.h"
    #include "bsp_buttons.h"
    #include "nwk_types.h"
    #include "nwk_api.h"
    #include "nwk_frame.h"
    #include "nwk.h"
    #include "msp430x22x4.h"
    #include "vlo_rand.h"

    #define MESSAGE_LENGTH 6
    void TXString( char* string, int length );
    void MCU_Init(void);
    void transmitData(char msg[MESSAGE_LENGTH] );
    void transmitDataString(char msg[MESSAGE_LENGTH]);
    void createRandomAddress();




    //Données pour la sortie du terminal
    const char splash[] = {"\r\n-------------------------------------------------\r\n     ****\r\n
    ****           eZ430-RF2500\r\n     ******o****    //Temperature Sensor
    Network\r\n********_///_****   Copyright 2007\r\n ******/_//_/*****  Texas Instruments
    Incorporated\r\n  ** ***(__/*****   All rights reserved.\r\n      *********     Version 1.02\r\n
    *****\r\n        ***\r\n-------------------------------------------------\r\n"};



    __no_init volatile char Flash_Addr[4] @ 0x10F0 ; // Adresse flash est réglé aléatoirement
    //Cela signifierait que le compilateur ne doit pas initialiser la variable, et serait utilisé
    quand une variable est placé à
    // un emplacement de registre du matériel et vous ne voulez pas le démarrage du programme à
    définir le registre à 0

    // réserver de l'espace pour le maximum possible ID Link pairs
    static linkID_t sLID[NUM_CONNECTIONS];  //
    static uint8_t sNumCurrentPeers;

    // gestionnaire de rappel
    static uint8_t sCB(linkID_t); //type de données d'entier sans signe qui a seulement 8 bits
    // sémaphores de boucle de travail
    static uint8_t sPeerFrameSem;
    static uint8_t sJoinSem;
    //sSelfMeasureSem statique d'uint8_t;
    int run;

    void main (void)
    {
    addr_t lAddr;
    bspIState_t intState;

    WDTCTL = WDTPW + WDTHOLD;  // Arrêter WDT
```

```
// la boucle de retard pour assurer le démarrage approprié avant SimpliciTI augmente DCO
// Ceci est typiquement conçu en fonction l'alimentation d'énergie utilisée, et dans ce cas-ci
// est excessif pour la sécurité en raison de large diffusion.
volatile int i;
for(i = 0; i < 0xFFFF; i++){}
if( CALBC1_8MHZ == 0xFF )  // N'executez pas si les valeurs d'appels sont effacées
{
volatile int i;
P1DIR |= 0x03;
BSP_TURN_ON_LED1();
BSP_TURN_OFF_LED2();
while(1)
{
for(i = 0; i < 0x5FFF; i++){}
BSP_TOGGLE_LED2();
BSP_TOGGLE_LED1();
}
}
BSP_Init();
if( Flash_Addr[0] == 0xFF &&
Flash_Addr[1] == 0xFF &&
Flash_Addr[2] == 0xFF &&
Flash_Addr[3] == 0xFF )
{
createRandomAddress();  // adresse de dispositif aléatoire réglée au démarrage initial
}
lAddr.addr[0]=Flash_Addr[0];
lAddr.addr[1]=Flash_Addr[1];
lAddr.addr[2]=Flash_Addr[2];
lAddr.addr[3]=Flash_Addr[3];
SMPL_Ioctl(IOCTL_OBJ_ADDR, IOCTL_ACT_SET, &lAddr);
MCU_Init();
//Transmettre l'écran de démarrage et d'initialisation de notification réseau
TXString( (char*)splash, sizeof splash);
TXString( "\r\nInitializing Network....", 26 );
SMPL_Init(sCB);
// réseau initialisée
TXString( "Done\r\n", 6);
//Certains de nos code original commence ici
BSP_TOGGLE_LED2();
// boucle de travail principal
while(1)
{
// Attendez que le sémaphore Rejoignez être fixé par la réception d'une trame Joignez-vous à
partir d'une
// Interface qui prend en charge et de fin de périphérique ED  .
if(sJoinSem && (sNumCurrentPeers < NUM_CONNECTIONS))
{
// Attendez pour une nouvelle connexion
SMPL_LinkListen(&sLID[sNumCurrentPeers]);
sNumCurrentPeers++;
BSP_ENTER_CRITICAL_SECTION(intState);
if(sJoinSem)
{
sJoinSem--;
}
BSP_EXIT_CRITICAL_SECTION(intState);
}
// Avons-nous reçu une trame sur l'un des liens ED?
// Aucune section critique - il n'a pas vraiment beaucoup d'importance si nous manquons un
sondage
if(sPeerFrameSem)
{
uint8_t msg[MAX_APP_PAYLOAD], len, i;
// Traiter toutes les trames d'attente
for(i=0; i<sNumCurrentPeers; ++i)
{
if(SMPL_Receive(sLID[i], msg, &len) == SMPL_SUCCESS)
{
ioctlRadioSiginfo_t sigInfo;
sigInfo.lid = sLID[i];
SMPL_Ioctl(IOCTL_OBJ_RADIO, IOCTL_ACT_RADIO_SIGINFO, (void *)&sigInfo);
transmitData( (char*)msg );
BSP_TOGGLE_LED2();
BSP_ENTER_CRITICAL_SECTION(intState);
sPeerFrameSem--;
BSP_EXIT_CRITICAL_SECTION(intState);
}
}
```

```
}
}
}

/*-------------------------------------------------------------------------------
*
-------------------------------------------------------------------------------*/
void createRandomAddress()
{
unsigned int rand, rand2;
do
{
rand = TI_getRandomIntegerFromVLO();  // le premier octet ne peut pas être 0x00 de 0xFF
}
while( (rand & 0xFF00)==0xFF00 || (rand & 0xFF00)==0x0000 );
rand2 = TI_getRandomIntegerFromVLO();
BCSCTL1 = CALBC1_1MHZ;          // Placer DCO à 1MHz
DCOCTL = CALDCO_1MHZ;
FCTL2 = FWKEY + FSSEL0 + FN1;   // MCLK/3 pour le générateur de synchronisation instantanée
FCTL3 = FWKEY + LOCKA;          // Effacer LOCK & LOCKA bits
FCTL1 = FWKEY + WRT;            // Réglez le bit de WRT pour l'opération d'écriture
Flash_Addr[0]=(rand>>8) & 0xFF;
Flash_Addr[1]=rand & 0xFF;
Flash_Addr[2]=(rand2>>8) & 0xFF;
Flash_Addr[3]=rand2 & 0xFF;
FCTL1 = FWKEY;  // Effacer le bit de WRT
FCTL3 = FWKEY + LOCKA + LOCK;  // Set LOCK & LOCKA bit
}
/*-------------------------------------------------------------------------------
*
-------------------------------------------------------------------------------*/
void transmitData(char msg[MESSAGE_LENGTH] )
{

transmitDataString( msg );
}
/*-------------------------------------------------------------------------------
*
-------------------------------------------------------------------------------*/
void transmitDataString(char msg[MESSAGE_LENGTH] )
{
char x_string[] = {"XXXX"};
char y_string[] = {"XXXX"};
char z_string[] = {"XXXX"};

char x1_string[] = {"XXXX"};
char y1_string[] = {"XXXX"};
char z1_string[] = {"XXXX"};

int tempx = msg[0] + (msg[1]<<8);
int tempx1 = msg[0] + (msg[1]<<8);

x_string[3] = '0'+ (tempx%10);
x_string[2] = '0'+ ((tempx/10)%10);
x_string[1] = '0'+ ((tempx/100)%10);
x_string[0] = '0'+ ((tempx/1000)%10);

x1_string[3] = '0'+ (tempx1%10);
x1_string[2] = '0'+ ((tempx1/10)%10);
x1_string[1] = '0'+ ((tempx1/100)%10);
x1_string[0] = '0'+ ((tempx1/1000)%10);

int tempy = msg[2] + (msg[3]<<8);
int tempy1 = msg[2] + (msg[3]<<8);

y_string[3] = '0'+ (tempy%10);
y_string[2] = '0'+ ((tempy/10)%10);
y_string[1] = '0'+ ((tempy/100)%10);
y_string[0] = '0'+ ((tempy/1000)%10);

y1_string[3] = '0'+ (tempy1%10);
y1_string[2] = '0'+ ((tempy1/10)%10);
y1_string[1] = '0'+ ((tempy1/100)%10);
y1_string[0] = '0'+ ((tempy1/1000)%10);


int tempz = msg[4] + (msg[5]<<8);
int tempz1 = msg[4] + (msg[5]<<8);
```

```
z_string[3] = '0'+ (tempz%10);
z_string[2] = '0'+ ((tempz/10)%10);
z_string[1] = '0'+ ((tempz/100)%10);
z_string[0] = '0'+ ((tempz/1000)%10);

z1_string[3] = '0'+ (tempz1%10);
z1_string[2] = '0'+ ((tempz1/10)%10);
z1_string[1] = '0'+ ((tempz1/100)%10);
z1_string[0] = '0'+ ((tempz1/1000)%10);

//char output[] = {"\r\nX: XXXX, Y: XXXX, Z: XXXX"};
char output[] = {"\r\nXXXX XXXX XXXX"};
output[2] = x_string[0];
output[3] = x_string[1];
output[4] = x_string[2];
output[5] = x_string[3];
output[7] = y_string[0];
output[8] = y_string[1];
output[9] = y_string[2];
output[10] = y_string[3];
output[12] = z_string[0];
output[13] = z_string[1];
output[14] = z_string[2];
output[15] = z_string[3];

output[2] = x1_string[0];
output[3] = x1_string[1];
output[4] = x1_string[2];
output[5] = x1_string[3];
output[7] = y1_string[0];
output[8] = y1_string[1];
output[9] = y1_string[2];
output[10] = y1_string[3];
output[12] = z1_string[0];
output[13] = z1_string[1];
output[14] = z1_string[2];
output[15] = z1_string[3];
TXString(output, sizeof output );
}
/*-------------------------------------------------------------------------------
*
-------------------------------------------------------------------------------*/
void TXString( char* string, int length )
{
int pointer;
for( pointer = 0; pointer < length; pointer++)
{
volatile int i;
UCA0TXBUF = string[pointer];
while(!(IFG2&UCA0TXIFG));  // USCI_A0 TX buffer Prét?
}
}
/*-------------------------------------------------------------------------------
*
-------------------------------------------------------------------------------*/
void MCU_Init()
{
BCSCTL1 = CALBC1_8MHZ;  // configurer DCO
DCOCTL = CALDCO_8MHZ;
BCSCTL3 |= LFXT1S_2;  // LFXT1 = VLO
TACCTL0 = CCIE;  // TACCR0 interruption permise
TACCR0 = 2000;  // ~1/6 en second lieu
TACTL = TASSEL_1 + MC_1;  // ACLK, upmode
P3SEL |= 0x30;  // P3.4,5 = USCI_A0 TXD/RXD
UCA0CTL1 = UCSSEL_2;  // SMCLK
UCA0BR0 = 0x41;  // 9600 depuis 8Mhz
UCA0BR1 = 0x3;
UCA0MCTL = UCBRS_2;
UCA0CTL1 &= ~UCSWRST;  // **Iinitialiser la machine d'état d'USCI**
IE2 |= UCA0RXIE;  // Permettre l'interruption d'USCI_A0 RX
__enable_interrupt();
}
/*-------------------------------------------------------------------------------
* Court dans le contexte d'ISR. La lecture de l'armature devrait être faite dans
* le fil d'application pas dans le fil d'ISR.
 -------------------------------------------------------------------------------*/
static uint8_t sCB(linkID_t lid)
{
if(lid)
```

```
{
sPeerFrameSem++;
}
else
{
sJoinSem++;
}
// laisser l'armature à lire par application.
return 0;
}
/*-------------------------------------------------------------------------
* Routine de service de l'interruption ADC10
-------------------------------------------------------------------------*/
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void)
{
__bic_SR_register_on_exit(CPUOFF);  // effacé CPUOFF bit depuis 0(SR)
}
/*-------------------------------------------------------------------------
* Routine de service d'interruption du temporisateur A0
-------------------------------------------------------------------------*/
#pragma vector=TIMERA0_VECTOR
__interrupt void Timer_A (void)
{
run = 1;}
```

Programme d’ED :

```
//Le Fichier End Device

#include "bsp.h"
#include "mrfi.h"
#include "nwk_types.h"
#include "nwk_api.h"
#include "bsp_leds.h"
#include "bsp_buttons.h"
#include "vlo_rand.h"
void linkTo(void);
void MCU_Init(void);
__no_init volatile char Flash_Addr[4] @ 0x10F0; // Adresse flash est réglé aléatoire
void createRandomAddress();
void main (void)
{
addr_t lAddr;
WDTCTL = WDTPW + WDTHOLD; // arreter le chien de garde(WDT)
{
// retarder la boucle pour assurer le démarrage correct avant le protocole de SimpliciTI augmente
DCO
// Ceci est typiquement adapté à l'alimentation utilisée, et dans ce cas
// est excessif pour la sécurité en raison de large diffusion.
volatile int i;
for(i = 0; i < 0xFFFF; i++){}
}
if( CALBC1_8MHZ == 0xFF ) // N'execute pas si les valeurs d'appels sont effacées
{
volatile int i;
P1DIR |= 0x03;
BSP_TURN_ON_LED1();
BSP_TURN_OFF_LED2();
while(1)
{
for(i = 0; i < 0x5FFF; i++){}
BSP_TOGGLE_LED2();
BSP_TOGGLE_LED1();
}
}
// SimpliciTI va modifier les paramètres de broches du port
P1DIR = 0xFF;
P1OUT = 0x00;
P2DIR = 0x20;
//P2OUT = 0x00;
P3DIR = 0xC0;
P3OUT = 0x00;
P4DIR = 0xFF;
P4OUT = 0x00;
//P2SEL = 0x07;
P4OUT |= 0x08; //Définition du Broche 8 (P4.3) en active , Activation de l'accéléromètre, mis en
bas pour mettre à dormir
BSP_Init();
if( Flash_Addr[0] == 0xFF &&
Flash_Addr[1] == 0xFF &&
Flash_Addr[2] == 0xFF &&
Flash_Addr[3] == 0xFF )
{
createRandomAddress(); // définir l'adresse de l'interface aléatoirement au démarrage initialisé
}
lAddr.addr[0]=Flash_Addr[0];
lAddr.addr[1]=Flash_Addr[1];
lAddr.addr[2]=Flash_Addr[2];
lAddr.addr[3]=Flash_Addr[3];
SMPL_Ioctl(IOCTL_OBJ_ADDR, IOCTL_ACT_SET, &lAddr);
BCSCTL1 = CALBC1_8MHZ; // Réglage  DCO après la fonction aléatoire
DCOCTL = CALDCO_8MHZ;
BCSCTL3 |= LFXT1S_2; // LFXT1 = VLO
// continuer à essayer de joindre jusqu'à ce qu'elle réussisse. basculer LED pour indiquer que
// la jonction n'a pas eu lieu. LED 3 est rouge, mais étiquetés LED 4 sur l'expiration
// la carte de circuit imprimé. LED1 is green.
while (SMPL_NO_JOIN == SMPL_Init((uint8_t (*)(linkID_t))0))
{
BSP_TOGGLE_LED1();
BSP_TOGGLE_LED2();;
__bis_SR_register(LPM3_bits + GIE); // selection le mode LPM3 avec les interruptions activées.
}
```

```
// lien inconditionnel du point d'accée qui est à l'écoute doit rejoindre avec succès.
linkTo();
}
void createRandomAddress()
{
unsigned int rand, rand2;
do
{
rand = TI_getRandomIntegerFromVLO(); // le premier octet ne peut pas être 0x00 0xFF
}
while( (rand & 0xFF00)==0xFF00 || (rand & 0xFF00)==0x0000 );
rand2 = TI_getRandomIntegerFromVLO();
BCSCTL1 = CALBC1_1MHZ; // Réglage de l'ADC à 1MHz
DCOCTL = CALDCO_1MHZ;
FCTL2 = FWKEY + FSSEL0 + FN1; // MCLK / 3 pour le générateur de synchronisation du registre flash
FCTL3 = FWKEY + LOCKA; // Effacer les bits des registre LOCK et LOCKA
FCTL1 = FWKEY + WRT; // Réglage les bits de registre WRT pour l'opération d'écriture
Flash_Addr[0]=(rand>>8) & 0xFF;
Flash_Addr[1]=rand & 0xFF;
Flash_Addr[2]=(rand2>>8) & 0xFF;
Flash_Addr[3]=rand2 & 0xFF;
FCTL1 = FWKEY; // Effacement les bits de registre WRT
FCTL3 = FWKEY + LOCKA + LOCK; // reglage les bits des registres LOCK et LOCKA
}
void linkTo()
{
linkID_t linkID1;
uint8_t msg[6];
// continuer à tenter de lier...
while (SMPL_SUCCESS != SMPL_Link(&linkID1))
{
__bis_SR_register(LPM3_bits + GIE); // selection le mode LPM3 avec les interruptions activées.
BSP_TOGGLE_LED1();
BSP_TOGGLE_LED2();
}
// Eteignez tous les LEDs
if (BSP_LED1_IS_ON())
{
BSP_TOGGLE_LED1();
}
if (BSP_LED2_IS_ON())
{
BSP_TOGGLE_LED2();
}
while (1)
{
char msg[6];
int data_x,data_y,data_z;
volatile long temp;
int results[3];
SMPL_Ioctl( IOCTL_OBJ_RADIO, IOCTL_ACT_RADIO_SLEEP, "" );
//__bis_SR_register(LPM3_bits+GIE); // selection le mode LPM3 avec les interruptions activées.
SMPL_Ioctl( IOCTL_OBJ_RADIO, IOCTL_ACT_RADIO_AWAKE, "" );
//lire x: x est A0 qui est P2.0
BSP_TOGGLE_LED2();
ADC10CTL1 = INCH_0 + ADC10DIV_2; //ADC mis à A0 avec la devision de l'horloge sur 2, on peut ne
pas avoir besoin de la dision de l'horloge
ADC10CTL0 = SREF_0 + ADC10SHT_3 + ADC10ON + ADC10IE; //on utilise Vcc et GND pour référence, 16
CLKS / échantillon, tournez AC en marche et activer les interruptions
//for( int a = 100; a > 0; a-- ); // retarder pour permettre référence à s'installer, ne pas être
nécessaire
ADC10CTL0 |= ENC + ADC10SC; // Echantillonnage et début de conversion
__bis_SR_register(CPUOFF + GIE); // LPM0 avec les interruptions activées, ce qui signifie qu'il
pénètre faible puissance jusqu'à la conversion ADC est effectué résultats [0] = ADC10MEM; / /
copie résultats ADC aux données x
ADC10CTL0 &= ~ENC; //Réinitialiser le bit d'encodeur
//lire y: y est A1 qui est P2.1
ADC10CTL1 = INCH_1 + ADC10DIV_2; //ADC mis à A0 avec la devision de l'horloge sur 2, on peut ne
pas avoir besoin de la dision de l'horloge
ADC10CTL0 = SREF_0 + ADC10SHT_3 + ADC10ON + ADC10IE; //on utilise Vcc et GND pour référence, 16
CLKS / échantillon, tournez AC en marche et activer les interruptions
//for( int a = 100; a > 0; a-- ); // retarder pour permettre référence à s'installer, pourrait ne
pas être nécessaire
ADC10CTL0 |= ENC + ADC10SC; // Echantillonnage et début de conversion
__bis_SR_register(CPUOFF + GIE); // LPM0 avec les interruptions activées, ce qui signifie qu'il
pénètre faible puissance jusqu'à la conversion ADC est fait
results[1] = ADC10MEM; //copie ADC résultats aux données y
ADC10CTL0 &= ~ENC; //réinitialiser les bits de l'encodeur
//lire z: z est A2 qui est P2.2
```

```
    ADC10CTL1 = INCH_2 + ADC10DIV_2; //ADC mis à A2 avec la devision de l'horloge sur 2, on peut ne
    pas avoir besoin de la dision de l'horloge
    ADC10CTL0 = SREF_0 + ADC10SHT_3 + ADC10ON + ADC10IE; //utilisant Vcc et GND pour référence, 16
    CLKS / échantillon, tournez AC en marche et activer les interruptions
    //for( int a = 100; a > 0; a-- ); // retarder pour permettre référence à s'installer, pourrait ne
    pas être nécessaire
    ADC10CTL0 |= ENC + ADC10SC; // Echantillonnage et début de conversion
    __bis_SR_register(CPUOFF + GIE); // LPM0 avec les interruptions activées, ce qui signifie qu'il
    pénètre faible puissance jusqu'à la conversion ADC est fait
    results[2] = ADC10MEM; //copie ADC résultats aux données z
    ADC10CTL0 &= ~ENC; //Réinitialiser le bit d'encodeur
    ADC10CTL0 &= ~(REFON + ADC10ON); // éteindre A / D pour économiser l'énergie
    /*
    Les données analogiques sont converties en utilisant la formule suivante:
    Digital = 1023*(Vanalog - Vref-) / (Vref+ - Vref-)
    Le 1023 vient du bit ADC 10 (2 ^ 10)
    Pour notre cas, car c'est l'installation, Vref-= GND et Vref + = Vcc
    une tension d'entrée de Vcc / 2 correspond à une accélération nulle si l'accélération est nulle
    512 en numérique (1023/2 par approximation)
    */
    data_x = results[0];
    data_y = results[1];
    data_z = results[2];
    msg[0] = data_x&0xFF;
    msg[1] = (data_x>>8)&0x03;
    msg[2] = data_y&0xFF;
    msg[3] = (data_y>>8)&0xFF;
    msg[4] = data_z&0xFF;
    msg[5] = (data_z>>8)&0xFF;
    BSP_TOGGLE_LED1();
    if (SMPL_SUCCESS == SMPL_Send(linkID1, msg, sizeof(msg)))
    {
    BSP_TOGGLE_LED2();
    }
    else
    {
    BSP_TOGGLE_LED2();
    BSP_TOGGLE_LED1();
    }
    }
    }
    /*-------------------------------------------------------------------------------
    * ADC10 routine de service d'interruption
    -------------------------------------------------------------------------------*/
    #pragma vector=ADC10_VECTOR
    __interrupt void ADC10_ISR(void)
    {
    __bic_SR_register_on_exit(CPUOFF); // Effacer les bits de registre CPUOFF  de registre 0(SR)

}
```

Bibliographie

Mémoires :

[1] : « Routage hiérarchique sur les réseaux de capteurs sans fil » par Khaled BOUCHAKOUR
[2] :« Réseaux de Capteurs Sans Fil : Routage et Sécurité »
[3] :« Implantation de capteurs communicants enfouis dans le béton à l'aide de l'outil de développement eZ430-RF2500 de Texas Instruments» par Hamza AYZI et Ismaïl ELHASNAOUI
[4] :« Wireless Accéléromètre » par Clay Cochran, Tim Duly, Brandon Ervin, & Mike Jender

Site Web:

[5] :fr.wikipedia.org
[6] : « Les Capteurs » gte.univ-littoral.fr
[7] :www.transport-intelligent.net/technologies/capteurs
[8] :www.techno-science.net
[9] :« Texas Instruments » ; www.ti.com
[10] :«SPI and I2C communication with CMA3000-D01 using MSP430 ultra low-power microcontroller» ; www.muratamems.fi
[11] :« Beginning Microcontrollers MSP430 Tutorial» www.glitovsky.com
[12] :site web : « MSP430 TP» www.datasheets.org

Thèse de Doctorat :

[13] : « Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique» par LEHSAINI Mohamed.

Livre :

[14] :« Analog and Digital Circuits for Electronic Control System Applications» par Jerry Luecke.

[15] : PDF : Synthese_capteur, gilles.berthome.free.fr

[16] : PDF : Réseaux de capteurs, www.iutbayonne.univ-pau.fr

Glossaire

A

ADC: Analog-to-digital converter
ACK: Acknowledge
AP: Access Point

C

CPU: Central processing unit
CCE: Code Composer Essentials

D

DMA: Direct Memory Access
DTC: Data Transfer Controller

E

EM: Etendue de Mesure
EEPROM: Electrically-erasable programmable read-only memory
ED: End Device

G

GSM: Global System for Mobile Communications

I

IEEE; Institute of Electrical and Electronics Engineers
I2C: Inter Integrated Circuit
IrDA: Infrared Data Association
IDE : Environnement de Développement intégré

J

JTAG: Joint Test Action Group

L

LCD; Liquid-crystal display
LPM: Low Power Magnetometer

M
MEMS : Microsystème électromécanique
MAC : Media Access Control
MIP : Million d'instructions par seconde
MCK : l’horloge maîtresse

N

NL; Niveau logique
NACK: Negative-acknowledge

P

PWM: Pulse-width modulation

R

RCSF : Réseau de Capteurs Sans Fil.
RAM: Random-access memory
RF: Radio frequency
ROM: Read-Only Memory

S

SL ; signal logique
SPI : Serial Peripheral Interface
SDA: signal data
SCL: signal clock
SMCLK: Sub-main clock

T

TOR; Tout Ou Rien
TI: Texas instruments

U

USB; Universal Serial Bus
USART: Universal Synchronous/Asynchronous Receiver/Transmitter
UART: Universal Asynchronous Receiver Transmitter

W

WSN; Wireless sensor network
WD: Watchdog