

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
**UNIVERSITE ABOU BEKR BELKAID TLEMCCEN**  
**FACULTE DE TECHNOLOGIE**

**DEPARTEMENT DE TELECOMMUNICATIONS**



# MEMOIRE

Pour l'obtention du diplôme de

## MASTER

**Réseaux Mobiles et Services de Télécommunications**

Réalisé par

**BENCHAA Souleyman**

**MAMOUNI Omar**

THEME

***Développement d'un RCSF pour le Contrôle de la Température,  
l'Humidité, CO et CO2***

**Soutenu le 27 MAI 2015 devant le Jury :**

MERZOUGUI Rachid

M.C.A à l'Université de Tlemcen

Président

MOUSSAOUI Djilali

M.A.A à l'Université de Tlemcen

Examineur

KADRI Benamar

M.C.A à l'Université de Tlemcen

Encadreur

Année universitaire : 2014 - 2015

## **DEDICACE**

Je dédie ce modeste travail :

A mes très chers parents

A mon frère et sœur

A mon binôme Omar

Et toute personne que je connais et qui me sont chers et tous ceux qui m'aiment.

**BENCHAA Souleyman**

## **Dédicace**

Je dédie ce modeste travail :

A mes très chers parents

A mes frères Mohammed, Abdelssamed, Nouh

A mes très chers amis Khir-Eddine, Abdelkader, Djamel

A toute ma grande famille

A toute mes amis

A mon très cher binôme Souleyman

**MAMOUNI Omar**

## REMERCIEMENTS

Nous remercions notre Dieu le tout puissant qui nous a accordé la volonté, la patience, et surtout la santé durant tout notre cursus.

Nous remercions tout d'abord notre encadreur Mr KADRI Benamar de nous avoir apporté ses valeureux conseils et soutiens durant la réalisation de ce mémoire.

Nous remercions également Mr MERZOUGUI Rachid pour avoir accepté de présider le jury de notre mémoire et nous tenons à remercier Mr MOUSSAOUI Djilali pour l'honneur qu'il nous a fait d'accepter de juger ce travail.

Nous tenons à remercier le corps professoral et administratif de la faculté de Technologie pour la richesse et la qualité de leurs enseignements et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.

A tous nos collègues de promotion RMST que nous avons eu le plaisir de côtoyer pendant cette période de formation. Une pensée va particulièrement à tous ceux d'entre nous qui n'ont pas eu la possibilité d'aller jusqu'au bout de leur formation.

Nos profonds remerciements à nos parents qui nous ont encouragés, qui nous ont appris à travailler honnêtement, et nous ont réalisé tous les moyens afin d'apprendre. Sans leur soutien moral, ce mémoire n'aurait pu commencer.

# TABLE DES MATIERES

Dédicace .....	I
Remerciement.....	III
Table des matières.....	IV
Liste des figures .....	VII
Liste des tableaux.....	IX
Glossaire.....	X

<b>Introduction générale.....</b>	<b>1</b>
-----------------------------------	----------

## *Chapitre 1 : Généralités sur les réseaux sans fil*

---

1. Introduction .....	3
2. Réseaux sans fil .....	3
3. Catégories des réseaux sans fil .....	4
3.1 WBAN ( Wireless Body Area Network ) .....	4
3.2 WPAN ( Wireless Personal Area Network ) .....	4
3.3 WLAN (Wireless Local Area Network ) .....	5
3.4 WMAN (Wireless Metropolitan Area Network) .....	5
3.5 WWAN (Wireless Wide Area Network) .....	6
3.6 WRAN (Wireless Regional Area Network) .....	8
4. Les réseaux Ad-hoc .....	8
4.1 Caractéristique des Réseaux Ad-hoc .....	8
4.2 Problématiques des réseaux Ad-hoc .....	9
4.3 Typologie des Réseaux Ad-hoc .....	11
5. Réseaux de capteurs sans fil.....	11
6. Applications des RCSFs.....	12
7. Conclusion.....	14

## *Chapitre 2 : Réseaux de capteurs sans fil*

---

1. Introduction.....	15
2. Les capteurs sans fil .....	15
3. Caractéristiques des réseaux de capteurs sans fil .....	16
4. Les handicaps des réseaux de capteur sans fil .....	17
5. Architecture de capteur sans fil .....	20

6.	Architecture protocolaire .....	21
7.	Architecture logicielle.....	23
8.	Autres systèmes d'exploitation pour les RCSF .....	23
9.	Les standards de communication pour les RCSFs.....	23
10.	La transmission de l'information dans les réseaux de capteurs .....	26
11.	Protocoles de routage dans les réseaux de capteurs.....	27
11.1	Routage à plat .....	27
11.2	Routage large échelle .....	28
12.	La tolérance aux pannes dans les réseaux de capteurs.....	28
13.	Conclusion.....	30

### ***Chapitre 3 : Architecture Logicielle***

---

1.	Introduction .....	31
2.	TinyOS .....	31
2.1	Propriétés de TinyOS .....	31
2.2	Modèle d'exécution de TinyOS .....	32
2.2.1	Programmation par évènement .....	32
2.2.2	Les Tâches .....	33
2.2.3	L'ordonnanceur TinyOS.....	33
3.	NesC.....	34
3.1	Les Principales caractéristiques de NesC .....	34
3.2	Les fichiers dans NesC .....	35
3.3	Concepts principaux dans NesC.....	36
3.3.1	Composants.....	36
3.3.2	Implémentations .....	36
3.3.3	Configurations.....	37
3.3.4	Module.....	37
3.4	Types des données.....	38
3.5	Types de fonctions en NesC .....	39
4.	Conclusion.....	40

## ***Chapitre 4 : Conception et réalisation de l'application***

---

1.	Introduction .....	41
2.	Objectif d'application .....	41
3.	Environnement de travail .....	42
3.1	Architecture matérielle .....	42
3.2	Architecture logicielle .....	42
4.	Fonctionnement de notre système .....	43
4.1	Installation logicielle .....	43
4.2	Installation matérielle .....	43
5.	Description de la plateforme WEB .....	44
5.1	Wampserver .....	44
5.2	PHPmyadmin .....	44
5.3	Notepad++ .....	44
5.4	Les serveurs .....	44
5.4.1	Serveur Apache .....	44
5.4.2	Serveur MySQL .....	45
5.5	Création des tables.....	45
6.	Les langages de programmation utilisés .....	46
6.1	Le langage HTML & CSS .....	46
6.2	Le langage de requête SQL.....	46
6.3	Le langage de script PHP.....	46
7.	Les différentes parties de notre projet.....	47
7.1	Partie I : Le site web.....	47
7.2	Partie II : Module capture.....	49
8.	Déploiement .....	51
8.1	Déploiement des capteurs .....	52
8.2	Déploiement de Station de base .....	52
9.	Conclusion.....	54
	<b>Conclusion Générale.....</b>	<b>55</b>
	Bibliographie.....	56
	Annexes	
	Résumé	

## LISTE DES FIGURES

<b>Figure 1.1</b>	Catégories des réseaux sans fil .....	3
<b>Figure 1.2</b>	Model WLAN.....	5
<b>Figure 1.3</b>	Réseaux de capteurs.....	11
<b>Figure 1.4</b>	Applications des RCSF.. ..	13
<b>Figure 2.1</b>	Capteur Tmote en recto.....	15
<b>Figure 2.2</b>	Capteur Tmote en verso.....	16
<b>Figure 2.3</b>	Architecture de capteurs sans fil.....	21
<b>Figure 2.4</b>	Architecture protocolaire.....	21
<b>Figure 2.5</b>	Energie consommée par le sous-système d'un capteur.....	22
<b>Figure 2.6</b>	Topologie IEEE 802.15.4.....	25
<b>Figure 2.7</b>	Envoi direct.....	26
<b>Figure 2.8</b>	Envoi par routage Ad-hoc.....	26
<b>Figure 2.9</b>	Protocoles de routage.....	27
<b>Figure 2.10</b>	Procédure de tolérance aux pannes.....	29
<b>Figure 3.1</b>	Logo TinyOS.....	31
<b>Figure 3.2</b>	Architecture d'une application NesC .....	35
<b>Figure 3.3</b>	Processus de compilation .....	35
<b>Figure 4.1</b>	Page d'accueil du site web.....	41
<b>Figure 4.2</b>	Schéma de communication dans la plateforme.....	43
<b>Figure 4.3</b>	Aperçu de table « Capture ».....	45
<b>Figure 4.4</b>	Aperçu de table « GPS ».....	45
<b>Figure 4.5</b>	Onglet Insertion du site Web.....	47
<b>Figure 4.6</b>	Onglet Modifier du site Web.....	47
<b>Figure 4.7</b>	Onglet Afficher liste des villes du site Web.....	48
<b>Figure 4.8</b>	Onglet Statistiques du site Web.....	48
<b>Figure 4.9</b>	Onglet Maps du site Web.....	49
<b>Figure 4.10</b>	Capteurs MTM CM5000.....	50
<b>Figure 4.11</b>	Capteur MTM CM3000.....	50
<b>Figure 4.12</b>	Sensor Board DS1000.....	51
<b>Figure 4.13</b>	Sensor Board DS1000 attaché au MTM CM3000.....	51
<b>Figure 4.14</b>	Déploiement des capteurs MTM CM5000 et CM3000.....	52
<b>Figure 4.15</b>	Raspberry Pi avec capteur CM5000 et Clef 3G.....	53



**Figure 4.16** Station de base.....53

# LISTE DES TABLEAUX

<b>Tableau 2.1</b> Comparaison entre Zigbee , Bluetooth et Wifi .....	25
---	----

# ***Glossaire***

---

## GLOSSAIRE

### A

**AODV:** Ad hoc **O**n **D**emand **D**istance **V**ector **R**outing.

### B

**BPSK:** **B**inary **P**hase **S**hift **K**ey.

### C

**CPU:** **C**entral **P**rocessing **U**nit.

**CGSR:** **C**lusterhead **G**ateway **S**witch **R**outing .

### D

**Diff Serv:** **D**ifferentiated **S**ervices.

**DSSS:** **D**irect **S**equence **S**pread **S**pectrum.

**DSR :** **D**ynamic **S**ource **R**outing.

### E

**EDGE:** **E**nhanced **D**ata rates for **G**SM **E**volution.

### F

**FIFO:** **F**irst **I**n **F**irst **O**ut.

**FFD:** **F**ull **F**unction **D**evelopments.

**FSR:** **F**isheye **S**tate **R**outing.

### G

**GSM:** **G**lobal **S**ystem for **M**obile **C**ommunications.

**GPRS:** **G**eneral **P**acket **R**adio **S**ervice.

**GPS :** **G**lobal **P**osition **S**ystem.

**GPSR:** **G**reedy **P**erimeter **S**tateless **R**outing .

**GRA:** **G**eographical **R**outing **A**lgorithm .

### H

**HSDPA:** High Speed Downlink Packet Access.

**HSUPA:** High Speed Uplink Packet Access.

**HSR :** Hierarchical State Routing .

### I

**ISM:** Industrial, Scientific and Medical.

**IEEE:** Institute of Electrical and Electronics Engineers.

**IntServ:** Integrated Services.

**IP:** Internet Protocol.

### L

**LTE:** Long Term Evolution.

**LED:** Light-Emitting Diode.

### M

**MAC:** Medium Access Control.

**MANET:** Mobile Ad-Hoc Networks.

### N

**NGN:** Next Generation Network.

### O

**OSI:** Open Systems Interconnection.

**OS:** Operating System.

**OQPSK:** Offset Quadrature Phase Shift Keying.

**OLSR:** Optimized Link State Routing.

### P

**PDA:** Personal Digital Assistant.

**PC:** Personal Computer.

**PSK:** Phase-Shift Keying.

## **R**

**RCSF:** Réseau de Capteurs Sans Fil.

**RFD:** Reduced Function Devices.

## **T**

**TDMA:** Time Division Multiple Access.

**TCP:** Transmission Control Protocol.

## **U**

**USB:** Universal Serial Bus.

**UWB:** Ultra Wide Band.

**UMTS:** Universal Mobile Telecommunications System.

## **V**

**VANET:** Vehicular Ad-Hoc Network.

## **W**

**WIFI:** Wireless Fidelity.

**WIMAX:** Worldwide Interoperability for Microwave Access.

**W-CDMA:** Wideband Code Division Multiple Access.

**WSN:** Wireless Sensor Networks.

# ***Introduction générale***

---

### Introduction générale

La téléphonie sans fil et mobile a connu l'essor fulgurant que l'on sait ces dernières années. Or le domaine des réseaux mobiles ne se limite pas à la voix et aux réseaux cellulaires comme le GSM.

Le progrès de l'électronique et de la technologie de communication sans fil a conduit le développement des réseaux de capteurs sans fil de faible puissance afin de détecter des phénomènes physiques provenant de l'environnement comme la température, pression, et l'humidité...etc). Des efforts importants ont été portés sur la miniaturisation de ces équipements mobiles qui peuvent atteindre des tailles minuscules de quelques millimètre qui sont diversifiés dans plusieurs domaines d'applications militaire, environnemental, médical...etc.

Des nœuds de capteurs se composent de matériel de détection, processeur, mémoire, alimentation, et l'émetteur-récepteur. Ces nœuds de capteurs ont l'énergie, la mémoire et la puissance de calcul limitée, Ces nœuds communiquent entre eux selon une certaine topologie du réseau afin d'acheminer les informations à un centre de contrôle distant de la zone de leur déploiement.

Pour éclairer cette technologie, ce mémoire propose une mise en œuvre d'un réseau de capteur sans fil capable de recueillir les informations provenant de l'environnement comme la température, l'humidité, CO et CO2 déployé dans une zone d'intérêt .

Afin d'aborder tous les aspects ayant trait au fonctionnement de ces réseaux, notre mémoire est organisé comme suit :

**Chapitre 1** : Nous verrons des concepts généraux sur les réseaux sans fil et leurs catégories. Nous allons voir aussi les réseaux Ad-hoc et leurs caractéristiques ainsi des différents typologies et les problématiques qui peuvent perturber le bon fonctionnement de ces réseaux, et enfin on va jeter un clin d'œil sur les réseaux de capteurs sans fil et leurs applications dans différents domaines.

**Chapitre 2** : Nous décrivons uniquement les réseaux de capteurs sans fil et leurs caractéristiques ainsi que les contraintes influençant l'architecture de ces réseaux.



**Chapitre 3** : présente le système d'exploitation TinyOS, avec ses propriétés, caractéristiques, sa structure logicielle, ainsi le système d'exploitations NesC.

**Chapitre 4** : constitue le cœur de notre travail pour la conception de l'application « Développement d'un réseau de capteurs pour le contrôle de la température, l'humidité et le niveau de CO et CO2 ».

Enfin, nous terminons cette thèse par une conclusion générale dans laquelle nous présentons ce qu'on a appris à travers ce projet, suivi de quelques perspectives .

# ***Chapitre 1 :***

## ***Généralités sur les réseaux sans fil***

---

*On ne connaît pas complètement une science tant qu'on n'en sait pas l'histoire.*

**Auguste COMTE**

## 1. Introduction

L'essor des technologies sans fil offre aujourd'hui de nouvelles perspectives dans le domaine des télécommunications. L'absence du câble dans les réseaux sans fil c'est l'atout majeur dont l'installation de tels réseaux ne demande pas de lourds aménagements, des infrastructures existantes comme c'est le cas avec les réseaux filaires (creusement des tranchées pour acheminer les câbles, équipements des bâtiments en câblage, goulottes et connecteurs). Ces technologies ont permis aux acteurs de l'organisation de communiquer en temps réel avec les systèmes d'informations indépendamment de leur position géographique qui pourront être divisés en différentes catégories suivant leur étendu.

## 2. Réseaux sans fil

Un réseau sans fil indique un réseau dans lequel au moins deux terminaux (ordinateur portable, PDA, etc.) peuvent communiquer sans liaison filaire, la transmission s'effectue dans l'espace libre. Les réseaux sans fil sont divisés en plusieurs catégories [1].

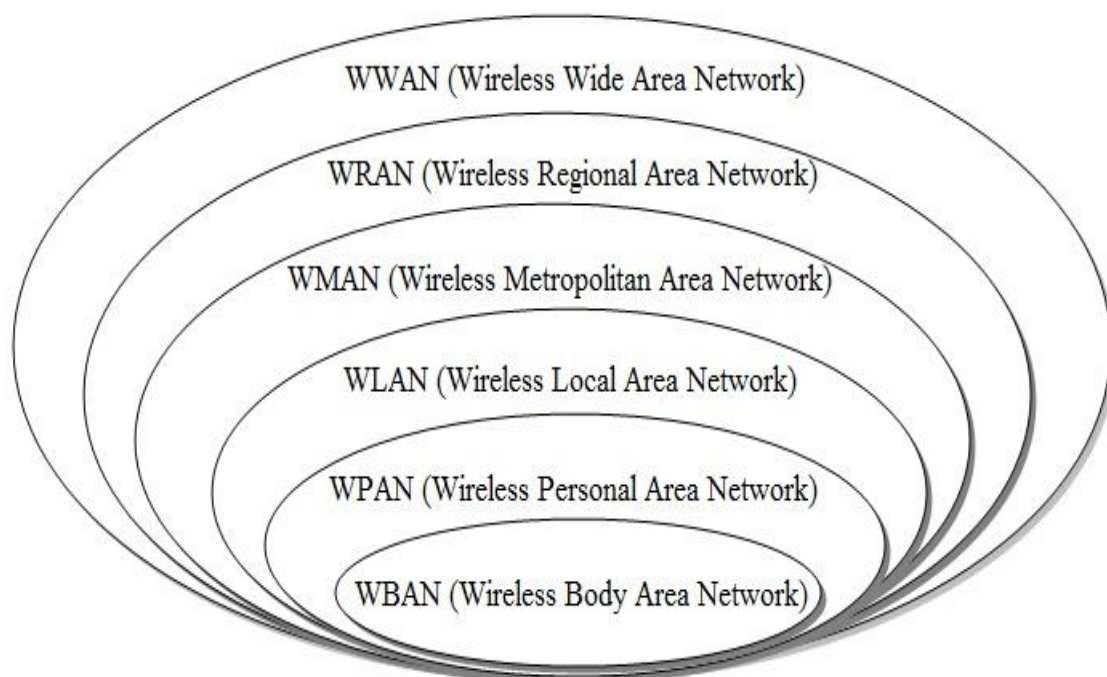


Figure 1.1 Catégories des réseaux sans fil

### 3. Catégories des réseaux sans fil

#### 3.1 WBAN ( Wireless Body Area Network )

Le Body Area Network est une topologie qui consiste à interconnecter sur, autour ou dans le corps humain de minuscules dispositifs pouvant effectuer des mesures (capteurs) ou agir de façon active (actionneurs).

Ce type de réseau utilise une architecture Ad-hoc pour la communication et envoyer les mesures à une station de base.

#### 3.2 WPAN ( Wireless Personal Area Network )

Il définit la norme IEEE 802.15, elle est destinée pour les réseaux personnels d'une dizaine de mètres de portée avec une faible consommation d'énergie. Cette norme est divisée en plusieurs catégories.

➤ **IEEE 802.15.1** : La plus connue qui prends la charge de la norme Bluetooth , lancée par Ericsson en 1994, proposant un débit théorique de 1 Mbps pour une portée maximale d'une trentaine de mètres. Cette technique utilisée dans les téléphones mobiles, périphériques informatiques et autres appareils portables comme les assistants personnels (PDA) [2].

➤ **IEEE 802.15.4** : Prends la charge de la norme Zigbee, a pour objectif de consommer peu d'énergie, de telle sorte qu'une petite batterie puisse tenir presque toute la durée de vie de l'interface, mais à un débit faible. Les nœuds sont conçus pour fonctionner plusieurs mois (jusqu'à 2 ans) en autonomie complète grâce à une simple pile alcaline de 1,5 V [3].

➤ **IEEE 802.15.3** : Prends la charge de la norme UWB (*Ultra Wide Band*) connaît actuellement un essor spectaculaire. Elle est destinée à la transmission de données à très haut débit 480 Mbits/s sur une distance de 10 mètres au maximum , envisageable pour la liaison entre PC et ses périphériques ( USB, Clavier, Souris...etc) [1].

➤ **Infrarouge** : Cette technologie est beaucoup plus sensible que Bluetooth aux perturbations lumineuses et nécessite une vision directe entre les éléments souhaitant communiquer, Cette technologie est largement utilisée pour la domotique (télécommandes).

### 3.3 WLAN (Wireless Local Area Network )

Il définit la norme IEEE 802.11, un réseau permettant de couvrir l'équivalent d'un réseau local d'entreprise, soit une portée d'environ une centaine de mètres, ainsi ils autorisent des débits allant de 2 à 54 Mbit/s dont les produits sont nommés Wi-Fi (*Wireless Fidelity*) qui fonctionne sur la bande de fréquence ISM (*Industrial, Scientific and Medical*), il existe plusieurs propositions de cette norme ceux conforme à la norme IEEE 802.11a, ainsi IEEE 802.11g avec un débit de 54 Mbits/s sur la bande de fréquence 5GHz et 2,4 GHz respectivement , et IEEE 802.11b avec un débit de 11 Mbits/s sur la bande de fréquence 2,4 GHz [4].

Les réseaux WLAN peuvent être bâtis selon deux modèles :

- **Ad-hoc** : c'est-à-dire en mode point à point entre équipements (par exemple, PDA, PC-téléphone, PC-imprimante) ;
- à **Infrastructure** : dans lequel le réseau est géré par des points d'accès (appelés bases ou encore bornes d'accès) [5].

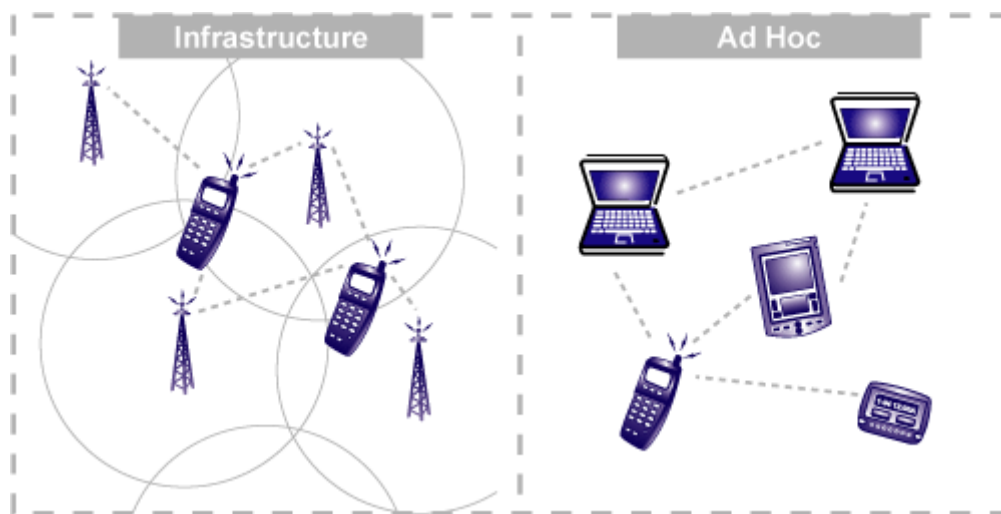


Figure 1.2 Model WLAN

### 3.4 WMAN (Wireless Metropolitan Area Network)

Utilisés pour l'accès aux réseaux d'infrastructure (boucle locale), ils offrent des débits de plusieurs dizaines de Mbit/s et capable d'émettre dans un rayon de 50 km, dont le réseau **WIMAX** (*Worldwide Interoperability for Microwave Access*) le plus connu qu'il utilise des bandes de très hautes fréquences, situées entre 2 et 66 GHz.

### 3.5 WWAN (Wireless Wide Area Network)

Recouvrent essentiellement les réseaux voix avec ses extensions données dont les débits sont relativement faibles de quelques dizaines de kbit/s (10 à 384 kbit/s). Plusieurs évolutions connus comme (GSM et UMTS, 4G) et évoluer vers des générations intermédiaires comme le GPRS, HSDPA, HSUPA qui seront présentés dans le paragraphe suivant :

#### ➤ GSM

Le réseau GSM (*Global System for Mobile Communications*) constitué au début du 21ème siècle le standard de téléphonie mobile le plus utilisé en Europe. Il s'agit d'un standard de téléphonie dit « de seconde génération » (**2G**), qui permet la transmission de la voix et des données. Il utilise une liaison radiotéléphonique entre le terminal et le réseau.

#### ➤ GPRS

Le GPRS (*General Packet Radio Service*) est la version intermédiaire s'agit d'un standard de téléphonie dit (2,5 G), elle est basée de la même infrastructure du GSM mais avec double réseaux cœur, l'un pour commutation des circuits et l'autre pour commutation des paquets qui autorisent un débit de 14,4 Kbits/s.

#### ➤ EDGE

Le EDGE (*Enhanced Data Rates for GSM Evolution*) s'agit d'une simple évolution de la technologie GSM/GPRS et du système TDMA (*Time division Multiple Access*) permettant d'obtenir un débit qui peut aller jusqu'à 384 Kbps. Ce standard utilise une nouvelle modulation du signal « 8PSK » (*Phase-Shift Keying*) qui permet d'améliorer l'efficacité spectrale et la capacité du réseau. Un terminal mobile dans un réseau EDGE est capable de transmettre et de recevoir sur plusieurs intervalles de temps (IT). Cette technique permet de définir des circuits allant jusqu'à 6 IT alloués aux mobiles par trame TDMA [6].

#### ➤ UMTS (3G)

La norme UMTS (*Universal Mobile Telecommunications System*) est une évolution de la deuxième génération à la troisième génération (3G).

Les bandes de fréquences allouées à l'UMTS se situent autour de 2 GHz. L'accès au canal utilise la technique du **W-CDMA** (*Wideband Code Division Multiple Access*) [1].

Les débits offerts à une station sont directement en relation avec la taille d'une cellule:

- de 144 à 384 kbit/s en zone rurale pour des mobiles dont la vitesse de déplacement est inférieure à 500 Km/h (TGV) .
- de 384 à 512 kbit/s en zone urbaine pour des vitesses de déplacement inférieures à 120 Km/h .
- jusqu'à 2 Mbit/s à moins de 10 Km de la base radio.

Aussi cette technologie a donné la naissance de plusieurs successeurs parmi les :

✓ **HSDPA (3.5G)**

HSDPA (*High Speed Downlink Packet Access*), cette technologie est une génération nommée 3.5 G correspond aux hauts débits de données, c'est-à-dire de plus de 1 Mbit/s. possède un lien descendant du réseau vers le terminal en mode paquet en augmentation forte par rapport à l'UMTS.

✓ **HSUPA (3.7 G)**

Le HSUPA (*High Speed Uplink Packet Access*) est une technologie s'intéresse à la voie montante du terminal vers le réseau, qui devrait atteindre à terme 5,76 Mbit/s. L'intérêt de cette technologie est d'offrir la possibilité d'émettre vers d'autres mobiles toutes sortes des documents électroniques (images, vidéos...etc) favorisant ,ainsi le développement de l'Internet haut débit mobile.

➤ **LTE (4G)**

LTE l'abréviation de (*Long Term Evolution*), constitue un système dit quatrième génération, il permet une transmission de très haut débit mobile .

- 100 Mbit/s en voie descendante pour une largeur de bande allouée de 20 MHz, soit une efficacité spectrale crête de 5 Bit/s/Hz .
- 50 Mbit/s en voie montante pour une largeur de bande allouée de 20 MHz, soit une efficacité spectrale crête de 2,5 Bit/s/Hz.

Son objectif d'avoir un cœur de réseau basé sur IP (*Internet Protocol*) et de ne plus offrir de mode commuté (établissement d'un circuit pour transmettre un appel "voix") [7].

### ➤ NGN

NGN (*Next Generation Network*) ou les futures générations de réseau reposent sur l'intégration de plusieurs systèmes et technologies d'accès sans fil. Ce système de télécommunication représente la convergence entre les 3ème et 4ème générations et les diverses technologies radio complémentaires. Cela permet de réduire les coûts de déploiement et d'augmenter la couverture à moindre frais.

Les terminaux de cette technologie sont capables de sélectionner à chaque instant la meilleure solution pour accéder à un réseau donné. L'objectif ici, est de fournir aux utilisateurs mobiles des services rapides et sans interruption. Cette génération offre des taux de données supérieurs à 100 Mbps [8].

### 3.6 WRAN (Wireless Regional Area Network)

Il définit la norme IEEE 802.22. Cette technologie a démarrée son activité en 2004 dans l'objectif de réaliser un réseau sans fil régional.

La portée d'un tel réseau atteindra jusqu'à une centaine de Km autour de l'émetteur, ce qui permettrait de couvrir des zones faiblement peuplées telles que les zones rurales [1].

## 4. Les réseaux Ad-hoc

Ces réseaux nécessitent aucune infrastructure fixe, sont capable de s'auto-initialiser et d'auto-configurer à l'issue de mouvement des mobiles sans aucune intervention humaine.

Les avantages des réseaux Ad-hoc sont leurs extensions très simples, leur couverture physique et leur coût [1][9].

### 4.1 Caractéristique des Réseaux Ad-hoc [10]

#### ➤ Mobilité

La mobilité des nœuds constitue à l'évidence une caractéristique très spécifique des réseaux Ad-hoc. Cette mobilité est intrinsèque au fonctionnement du réseau. Dans un réseau Ad-hoc, la topologie du réseau peut changer rapidement, de façon aléatoire et non prédictible et les techniques de routage des réseaux classiques, basées sur des routes préétablies, ne peuvent plus fonctionner correctement.



➤ **Equivalence des nœuds des réseaux**

Dans un réseau classique, il existe une distinction nette entre les nœuds terminaux (stations, hôtes) qui supportent les applications et les nœuds internes (routeurs par exemple) du réseau, en charge de l'acheminement des données. Cette différence n'existe pas dans les réseaux Ad-hoc car tous les nœuds peuvent être amenés à assurer des fonctions de routage.

➤ **Liaison sans fil (bande passante limitée)**

Les technologies de communication sans fil sont indispensables à la mise en place d'un réseau Ad-hoc. Malgré des progrès très importants, leurs performances restent et resteront en dessous de celles des technologies des réseaux filaires. La bande passante est moins importante, alors que le routage et la gestion de la mobilité génèrent d'avantage de flux de contrôle et de signalisation que dans une architecture de réseau filaire. Ces flux doivent être traités de façon prioritaire pour prendre en compte rapidement les modifications de topologie.

➤ **Autonomie des nœuds (Contrainte d'énergie)**

La consommation d'énergie constitue un problème important pour des équipements fonctionnant grâce à une alimentation électrique autonome. Ces équipements intègrent des modes de gestion d'énergie et il est important que les protocoles mis en place dans les réseaux ad hoc prennent en compte ce problème.

➤ **Vulnérabilité (une sécurité physique limitée)**

Les réseaux sans fil sont par nature plus sensibles aux problèmes de sécurité. Pour les réseaux Ad-hoc, le principal problème ne se situe pas au niveau du support physique mais principalement dans le fait que tous les nœuds sont équivalents et potentiellement nécessaires au fonctionnement du réseau.

## **4.2 Problématiques des réseaux Ad-hoc**

Comme les réseaux Ad-hoc ont beaucoup d'avantages et caractéristiques, peuvent aussi avoir des difficultés ou des problèmes, nous allons les présenter dans ce qui suit :

- **Atténuation** : le signal s'attenu rapidement en fonction de la distance (bien plus rapide que sur un câble) induit l'impossibilité pour un émetteur de détecter une collision au moment même où il transmet. Dans un réseau filaire, un émetteur sait

qu'il y a collision quand le signal qu'il lit sur le câble est différent de celui qu'il cherche à émettre.

Dans un réseau radio, un signal venant d'un autre nœud est tellement atténué par la distance qu'il ne provoquera que des perturbations négligeables par rapport au signal émis localement [11].

- **Sécurité** : un problème reste très difficile à contrôler, notamment le signal d'information qui se propage dans l'interface air est relativement vulnérable, une écoute clandestine est simple à réaliser qui permettent à un adversaire d'avoir accès à des informations secrètes, et des attaques actives qui peuvent détruire, injecter, modifier des messages et usurper l'identité d'un nœud.
- **Energie limitée** : les applications relatives aux réseaux sans fil ont en général un caractère nomade et tirent leur autonomie de batterie. En effet, l'émission ou la réception d'un signal d'information consomme l'autonomie d'énergie .
- **La puissance du signal** : ce facteur dépend de la portée et également limité par des réglementations très strictes ainsi que les stations mobiles doivent être légères, de petite taille et surtout doivent être capable de fonctionner de manière autonome (batterie). La prise en compte de tous ces éléments participe à la faible puissance de l'électronique embarquée.
- **La topologie dynamique** : chaque terminal mobile est amené à se déplacer dont la topologie des réseaux change fréquemment. Le voisinage d'un nœud peut varier continuellement : à tout moment des stations peuvent rejoindre ou quitter le réseau.
- **Qualité de service** : de nombreuses applications multimédia ou temps réel ont besoin de certaines garanties relatives liées au débit, au délai ou encore à la gigue. Dans un réseau Ad-hoc , il est très difficile d'obtenir ces garanties. Ceci est du à la nature du canal radio (interférences et taux d'erreur élevés) ainsi que les modèles IntServ (*Integrated Services* ), DiffServ (*Differentiated Services*) sont utilisés dans les réseaux filaire reste mal adaptés aux réseaux Ad-hoc à cause de leur topologie dynamique et leurs ressources limitées en terme de bande passante et d'énergie.

### 4.3 Typologie des Réseaux Ad-hoc

Les réseaux Ad-hoc peuvent être décomposés en trois catégories :

- **MANET** : est un système autonome composé de stations mobiles interconnectées par des liens sans fil dont chaque station participe à la constitution du réseau et joue le rôle d'un routeur.
- **VANET** : c'est un réseau Ad-hoc inter-véhiculaire qui vise certaines applications pour la sécurité routière, les systèmes d'aide de conduite et le confort des conducteurs et des passagers.
- **WSN** : est un réseau Ad-hoc avec un grand nombre de nœuds, sont des micro-capteurs capables de récolter et de transmettre des données environnementales d'une manière autonome.

## 5. Réseaux de capteurs sans fil

Le progrès réalisé ces dernières années dans les domaines des réseaux sans fil a donné la naissance d'une nouvelle technologie nommée les réseaux de capteurs sans fil (RCSF) qui a pour objectif principal de collecter une ou plusieurs informations.

Pour cela, le réseau est constitué de deux types de structures:

- Les capteurs qui envoient l'information.
- Les puits (passerelle) qui récoltent les informations.

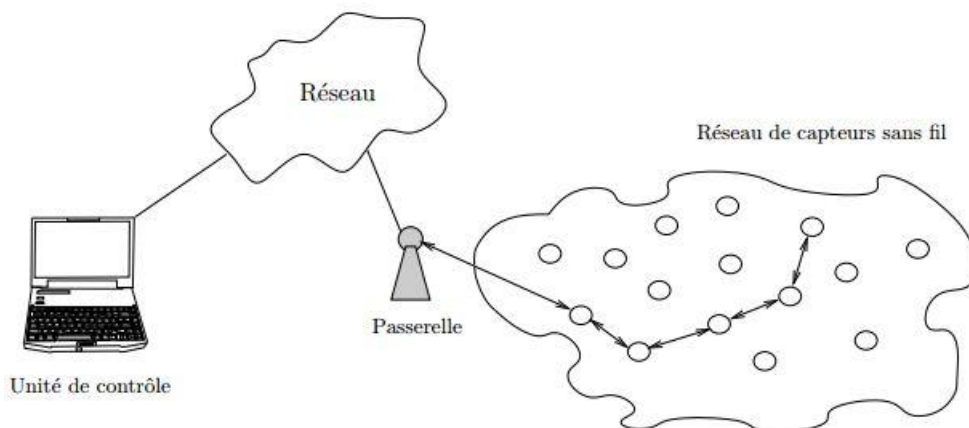


Figure 1.3 Réseaux de capteurs

## 6. Applications des RCSF

La miniaturisation, l'adaptabilité, le faible coût et la communication sans fil promettent plusieurs applications envisagées dans les réseaux de capteurs, font toujours l'objet d'une recherche et d'un développement actifs universitaires ou industriels dont le but est la réalisation de plusieurs types d'applications. Ces dernières peuvent être subdivisées en plusieurs domaines militaire, environnement, médical...etc. Nous décrivons ici brièvement certains domaines pour donner une idée de cette étendue.

### ➤ Applications militaires

Les réseaux de capteurs sans fil peuvent constituer des modules intégraux dans les systèmes militaires de commandes, contrôle, communication, calcul, intelligence, surveillance, reconnaissance et ciblage afin de surveiller toutes les activités des forces ennemies ou d'analyser le terrain avant d'envoyer des troupes [12].

### ➤ Applications environnementales

Les applications environnementales des réseaux de capteurs peuvent permettre de détecter des incendies, surveiller des catastrophes naturelles (séismes, inondations...), surveiller des phénomènes météorologiques, de détecter de la pollution [13].

### ➤ Applications médicales

Les capteurs peuvent être implantés dans le corps humain pour contrôler les problèmes médicaux comme le cancer dont le but est de surveiller les patients et l'avancement de leurs états dans un hôpital, comme aussi ils peuvent détecter les battements de cœur et la pression du sang...etc.

### ➤ Applications commerciales

Les capteurs possèdent également d'autres applications dans le domaine commerciale parmi lesquelles on peut énumérer, contrôler la qualité des produits, aussi la détection et surveillance des vols de voitures... etc.

➤ Application agricoles

Dans le domaine de l'agriculture, les capteurs peuvent être semés avec les graines, ce qui permet la détection des zones sèches et faciliter le processus d'irrigations dans un champ agricole.

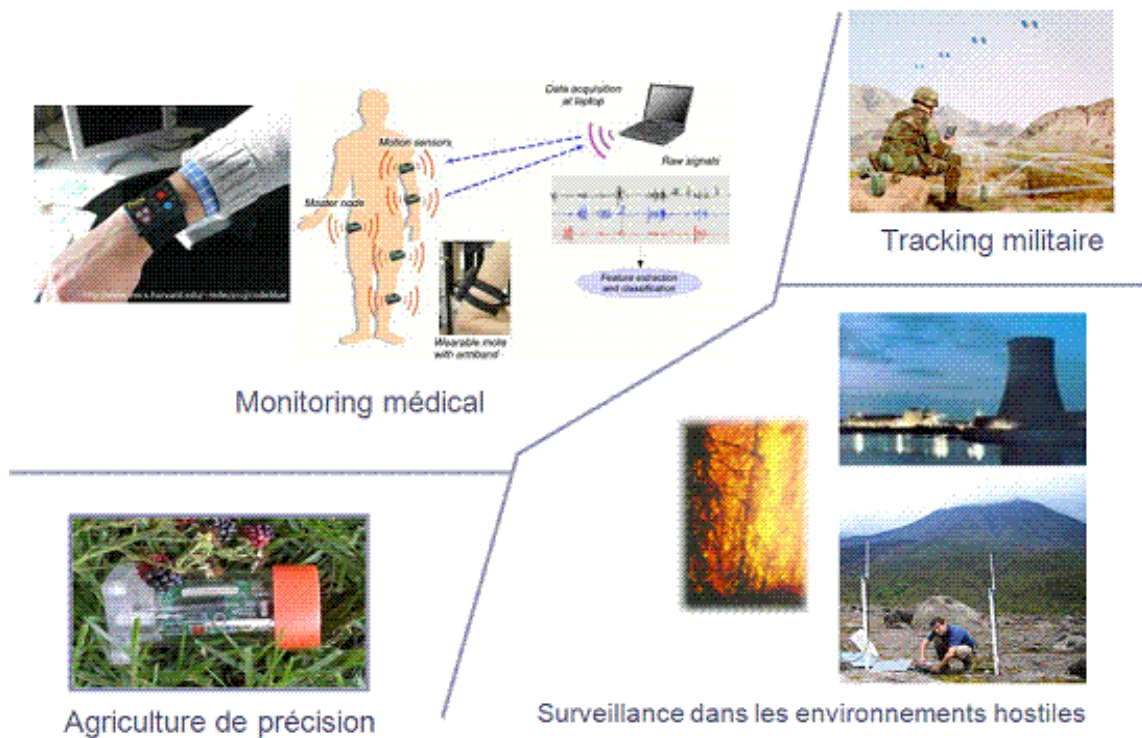


Figure 1.4 Applications des RCSF

## **7. Conclusion**

Dans ce chapitre nous avons présenté les réseaux sans fil et leurs catégories, ainsi plus de détails sur les réseaux Ad-hoc, leurs caractéristiques et leurs typologies, aussi leurs problématiques qui peuvent présenter un obstacle pour le bon fonctionnement des réseaux Ad-hoc.

Nous intéressons aux réseaux de capteurs sans fil qui possèdent des caractéristiques particulières qui les différencient des autres types de réseaux. Dans le prochain chapitre on va traiter uniquement les RCSF et leurs architectures logiciels et matériels et le routage pour l'acheminement des paquets vers le nœud destiné.

# ***Chapitre 2 :***

## ***Réseaux de capteurs sans fil***

---

## 1. Introduction

Depuis quelques décennies le développement technologique des réseaux de communication sans fils, a connu un grand essor qui a engendré les Réseaux de Capteurs Sans-fil (RCSF) ou en anglais Wireless Sensor Networks (WSN) . Cette évolution ne cesse pas à se développer au sein des communautés scientifiques et industrielles.

Ces réseaux facilitent le suivi et le contrôle à distance de l'environnement physique. Ils ont un intérêt particulier pour les applications militaires, environnementales ,médicales...etc.

## 2. Les capteurs sans fil

Un capteur est un petit appareil capable d'effectuer des mesures sur son environnement, par exemple le contrôle de la température, l'humidité et le niveaux de CO et CO2...etc. Chaque capteur assure trois fonctions principales la collecte, le traitement et la communication de l'information vers un ou plusieurs points de collecte appelés station de base sur une distance limité à quelques mètres.

Il existe plusieurs capteurs sans fil qui sont commercialisés tels MicaZ, Mica2, Telosb..

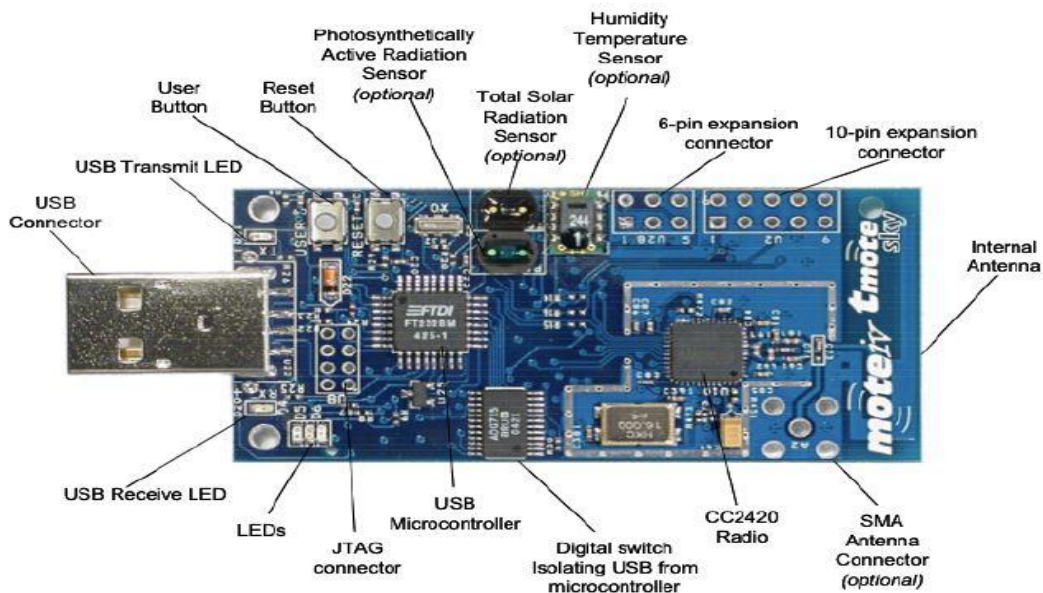


Figure 2.1 Capteur Tmote en recto



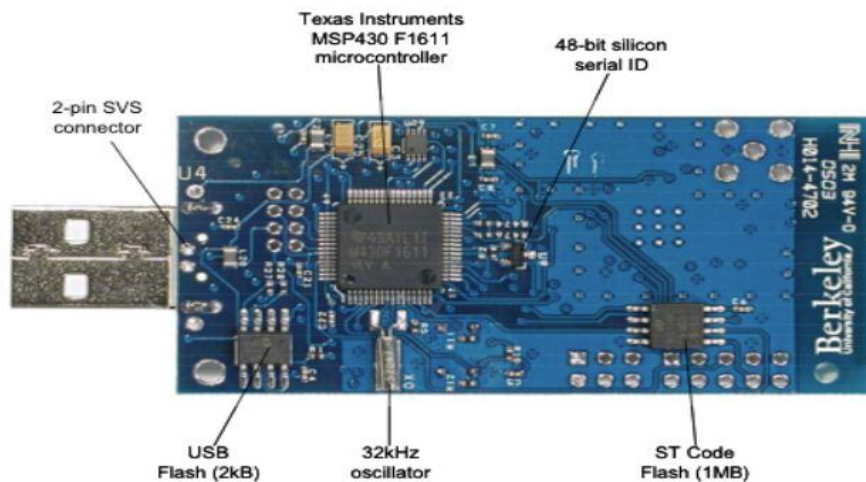


Figure 2.2 Capteur Tmote en verso.

### 3. Caractéristiques des réseaux de capteurs sans fil

Les réseaux de capteurs sans fil présentent des caractéristiques particulières par rapport aux réseaux sans fil. Parmi les principales caractéristiques nous décrivons [14]:

- **Absence d'infrastructure:** Les réseaux de capteurs se distinguent des autres réseaux mobiles par la propriété d'absence d'infrastructure préexistante et de tout genre d'administration centralisée. Lors de la détection de l'information par un capteur, cette dernière sera communiquée aux voisins de proche en proche en utilisant un mode multi-sauts jusqu'à l'aboutissement à la station de base.
- **Déployés en grand nombre:** Les RCSF sont généralement déployés en grand nombre pour assurer la couverture de la zone d'intérêt d'une part et d'autre part pour faire face à la tolérance aux pannes puisque les capteurs sont sujets à des pannes telles que l'épuisement de leur énergie ou écrasement par des animaux. Nous pouvons avoir dans de telles applications, des réseaux de capteurs comprenant des milliers voire des millions de capteurs.
- **Contrainte d'énergie:** Le principal facteur limitant la durée de vie d'un réseau de capteurs est l'énergie. Dans plusieurs applications, les nœuds capteurs sont placés dans des zones hostiles là où l'accès de l'homme est difficile voire impossible. De ce fait, le rechargement et le remplacement des batteries deviennent des tâches difficiles.

D'où, nous devons mettre en place un mécanisme pour préserver l'énergie et par suite garantir une longue durée de vie pour ces réseaux.

- **Topologie dynamique:** Les capteurs peuvent être attachés à des objets mobiles qui se déplacent d'une façon libre et arbitraire rendant ainsi, la topologie du réseau fréquemment changeante. Par exemple, des capteurs supportés par des animaux pour les surveiller.
- **Auto-organisation du réseau:** Cette caractéristique peut être nécessaire dans plusieurs cas. Par exemple, un réseau comportant un grand nombre de nœuds placés dans des endroits hostiles où la configuration manuelle n'est pas faisable, doit être capable de s'auto-organiser. Un autre cas est celui où un nœud est inséré ou retiré (à cause d'un manque d'énergie ou de destruction physique). Ainsi le réseau doit être capable de se reconfigurer pour continuer sa fonction.
- **Sécurité physique limitée:** Les RCSF sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui font le contrôle des données transférées doit être minimisé.

#### 4. Les handicaps des réseaux de capteurs sans fil

Les différents nœuds de capteurs sans fil peuvent être achevés par nombreux obstacles comme la tolérance aux pannes, les coûts de production, la consommation d'énergie, l'environnement ou la topologie du réseau qui seront détaillés dans ce qui suit :

- **Tolérance aux pannes**

Les nœuds peuvent être sujets à des pannes dues à leur fabrication (ce sont des produits de série bon marché, il peut donc y avoir des capteurs défectueux) ou plus fréquemment à un manque d'énergie. Les interactions externes (chocs, interférences) peuvent aussi être la cause des dysfonctionnements. Afin que les pannes n'affectent pas la tâche première du réseau, il faut évaluer la capacité du réseau à fonctionner sans interruption [15].

- **Coût de production**

Le coût de production d'un seul micro capteur est très important pour l'évaluation du coût globale du réseau, si ce dernier est supérieur à celui nécessaire pour le déploiement des capteurs classiques . L'utilisation de cette nouvelle technologie ne serait pas rentable . Par conséquent réduire le coût de production jusqu'à moins de 1 Dollar par nœud est un objectif important pour la faisabilité de la solution de capteur sans fil [16].

- **Passage à l'échelle**

Le nombre de nœuds capteurs déployés dans un réseau peut être à l'ordre des centaines voire des milliers. Pour certaines applications, il peut atteindre quelques millions. Afin de garantir le bon fonctionnement du réseau, les nouveaux schémas de déploiement doivent être capables de travailler avec ce grand nombre de nœuds. Par ailleurs, ils doivent utiliser la propriété de haute densité dans les réseaux de capteurs ; et donc pouvoir déployer un grand nombre de nœuds dans une petite surface [17].

- **Contraintes matérielles**

Un nœud doit être placé dans une petite surface n'excédant pas, généralement, un centimètre cube (1cm<sup>3</sup>). En outre de cette contrainte de surface, un ensemble de conditions doit être satisfait. Un nœud capteur doit [17] :

- Consommer le strict minimum d'énergie .
- Fonctionner dans de fortes densités .
- Avoir un faible coût de fabrication .
- Être autonome .
- S'adapter à l'environnement .

- **Topologie dynamique**

En raison de leur forte densité dans la zone à observer, il faut que les nœuds-capteurs soient capable d'adapter leur fonctionnement afin de maintenir la topologie souhaitée.

On distingue généralement trois phases dans la mise en place et l'évolution d'un réseau [15] :

- Déploiement : Les nœuds sont soit répartis de manière prédéfinie soit de manière aléatoire ( lancés en masse depuis un avion). Il faut alors que ceux-ci s'organisent de manière autonome.
- Post Déploiement – Exploitation : Durant la phase d'exploitation, la topologie du réseau peut être soumise à des changements dus à des modifications de la position des nœuds ou bien à des pannes.
- Redéploiement : L'ajout de nouveaux capteurs dans un réseaux existant implique aussi une remise à jour de la topologie.

- **Environnement**

Les capteurs sont souvent déployés en masse dans des endroits hostiles. Ils sont soumis à différentes conditions d'environnement; ils peuvent fonctionner sous haute pression au fond de l'océan, dans un environnement dur tel que les champs de bataille, dans des champs biologiquement ou chimiquement souillés ou même dans des milieux extrêmement froids. Par conséquent, ils doivent pouvoir fonctionner sans surveillance dans des régions géographiquement éloignées ou inaccessibles [17].

- **Agrégation des données**

Dans les réseaux de capteurs, les données produites par les nœuds capteurs voisins sont très corrélées spatialement et temporellement. Ceci peut engendrer la réception par la station de base d'informations redondantes. Réduire la quantité d'informations redondantes transmises par les capteurs permet de réduire la consommation d'énergie dans le réseau et ainsi d'améliorer sa durée de vie. L'une des techniques utilisée pour réduire la transmission d'informations redondantes est l'agrégation des données.

Avec cette technique, les nœuds intermédiaires agrègent l'information reçue de plusieurs sources. Cette technique est connue aussi sous le nom de fusion de données [18].

- **Consommation d'énergie**

L'économie d'énergie est l'une des problématiques majeures dans les réseaux de capteurs. En effet, la recharge des sources d'énergie est souvent trop coûteuse et parfois impossible. Il faut donc que les capteurs économisent au maximum l'énergie afin de pouvoir fonctionner. En effet, un réseau de capteurs ne peut pas survivre si la perte de nœuds est très importante car ceci engendre des pertes de communications dues à une très

grande distance entre les nœuds restants. Les réseaux de capteurs fonctionnant selon un mode de routage par saut, chaque nœud du réseau joue un rôle important dans la transmission de données. Le mauvais fonctionnement d'un nœud implique un changement dans la topologie et impose une réorganisation du réseau [17].

- **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.

- **Qualité de service**

Des protocoles au niveau de la couche MAC (*Medium Access Control*) 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.

## 5. Architecture de capteurs sans fil

Il existe une multitude de capteurs sans fil sur le marché afin de prélever différentes grandeurs à partir de l'environnement. Cependant, malgré cette diversité, ils restent dotés d'une architecture similaire.

Un nœud capteur est constitué de quatre unités principales [19][20] :

➤ **L'unité de captage** : Chargée de mesurer divers paramètres environnementaux (Température, pression, humidité) et d'un Convertisseur Analogique/Numérique (CAN) qui va convertir l'information relevée et la transmettre à l'unité de traitement.

➤ **L'unité de traitement** : Est généralement constituée de deux parties, une interface pour l'unité d'acquisition et une interface pour l'unité de transmission qui doit effectuer la transmission de l'information au réseau.

➤ **L'unité de transmission** : Est responsable de toutes les émissions et réceptions de données via un support de communication radio.

➤ **Unité d'énergie** : Un capteur est muni d'une ressource d'énergie généralement une batterie pour alimenter ses composants.

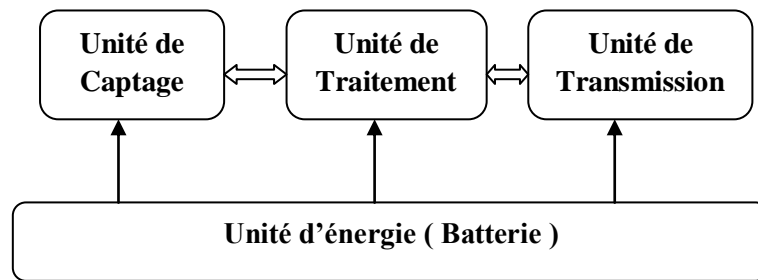


Figure 2.3 Architecture de capteurs sans fil.

## 6. Architecture protocolaire

Les réseaux de capteurs sans fil utilisent une architecture de communication en couche qui comprend cinq couches du model OSI (*Open Systems Interconnexion*) composé d'une couche application, une couche transport, une couche réseaux, une couche liaison de données, une couche physique et chaque couche a son rôle, ainsi que trois niveaux qui sont : ( un plan de gestion d'énergie, un plan de gestion de mobilité, un plan de gestions des tâches ) [21] .

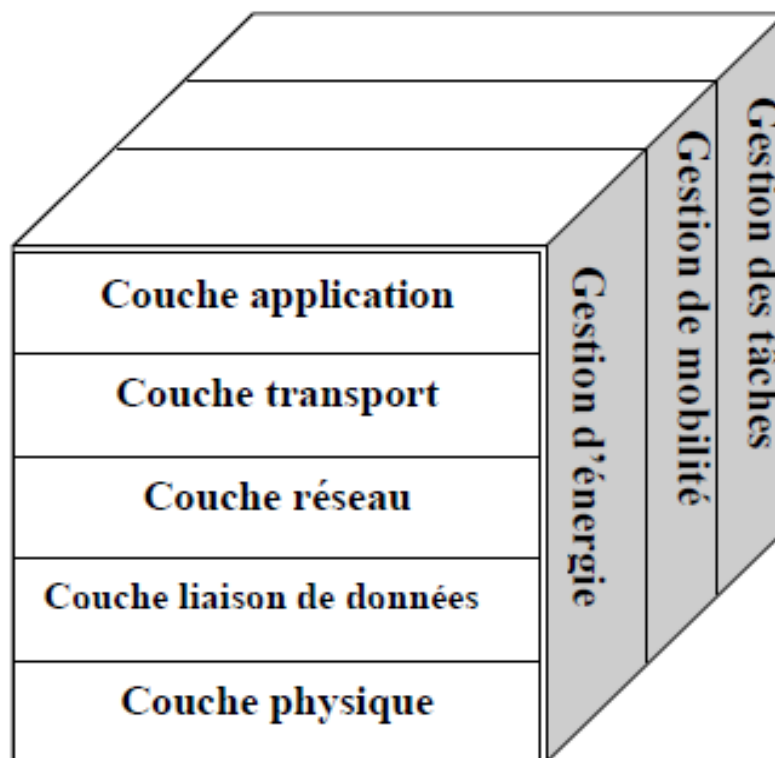


Figure 2.4 Architecture protocolaire.

- **La couche physique :** Est responsable de choix de la fréquence, de la détection du signal, de la modulation.
- **La couche liaison de données :** Généralement cette couche est chargée du multiplexage du flux de données, de la détection et verrouillage de la trame de données, et de contrôle d'accès aux médium MAC, ainsi que la plupart de cette dernière tente d'éliminer entièrement les collisions.
- **Couche réseau :** Le rôle principal de cette couche est de trouver une route et une transmission fiable de données à travers le réseau entre un nœud de capteur source et la destination.
- **Couche transport :** Cette couche est chargée de transport de données, elle manipule la segmentation des paquets, du contrôle du flux.
- **Couche applications :** Selon les tâches de capture, il existe différents types de logiciels qui peuvent être installés et employés pour la couche application.
- **Le plan de gestion d'énergie :** Un nœud de capteur sans fil nécessite une source d'énergie limitée compris entre ( $< 0,5\text{Ah}$ ,  $1,2\text{ V}$ ).

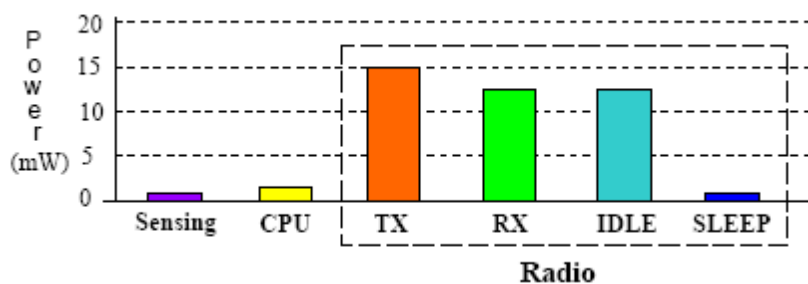


Figure 2.5 Energie consommée par le sous-système d'un capteur

- **Le plan de gestion de mobilité :** Un système de mobilité doit exister puisque les nœuds de capteurs sans fil peuvent être mobiles et qui doit détecter et enregistrer le mouvement (mobilité) de l'ensemble des nœuds.

- **Le plan de gestion des tâches :** Planifie l'équilibrage et le traitement des tâches ,les nœuds capteurs qui appartiennent à la même zone de capture ne sont pas obligés d'effectuer les tâches de capture en même temps. Selon leur niveau d'énergie dont ils disposent.

## 7. Architecture logicielle

La contrainte énergétique des capteurs exige l'utilisation des systèmes d'exploitations légers tels que TinyOS ou Contiki , Cependant, TinyOS reste toujours le plus utilisé et le plus populaire dans le domaine des RCSF. Il est libre et il est utilisé par une large communauté de scientifiques dans des simulations pour le développement, le test des algorithmes et protocoles réseaux.

## 8. Autres systèmes d'exploitation pour les RCSF

Il existe d'autres systèmes d'exploitation dédiés au réseau de capteurs sans fil , nous décrivons les plus répandus [22] :

- Contiki: c'est un système d'exploitation open-source multitâche, développé pour les systèmes embarqués avec contraintes de mémoire.
- SOS: système d'exploitation développé par l'université de Los Angeles en Californie écrit en langage C et qui reprend l'aspect événementiel de TinyOS.
- Mantis OS: système d'exploitation dédié aux réseaux de capteurs, développé par l'université du Colorado (USA) et écrit en langage C. Contrairement à TinyOS qui est basé sur un modèle de programmation événementielle, Mantis OS s'articule autour d'un modèle commandé par l'exécution de processus.
- Nut/OS: système d'exploitation multitâche pour les systèmes embarqués avec une pile TCP/IP.

## 9. Les standards de communication pour les RCSF

Comme nous l'avons cité dans le chapitre 1 la norme IEEE 802.15.4 Zigbee reste largement utilisé dans les réseaux de capteurs sans fil grâce à sa forte économie d'énergie et le nombre des nœuds. Elle s'appuie sur la couche physique et la sous couche MAC [23].



Au niveau de la couche physique, la norme 802.15.4 offre deux options. Les deux couches physiques sont basées sur les techniques d'étalement de spectre par séquence directe DSSS (*Direct Sequence Spread Spectrum*).

La première couche physique opère dans la bande 2,4 GHz et offre un débit de 250 kbit/s avec une modulation du type OQPSK (*Offset Quadrature Phase Shift Keying*).

La seconde opère dans les bandes 868/915 MHz et offre un débit de 20 kbit/s pour la bande 868 MHz et 40 kbit/s pour la bande 915 MHz, la modulation utilisée ici est la BPSK (*Binary Phase Shift Key*).

La norme IEEE 802.15.4 définit deux types d'entités : RFD (*Reduced Function Devices*) et les FFD (*Full Function Devices*) .

- Les RFD sont des nœuds aux fonctionnalités réduites, très économes en énergie, et pouvant être connectés à des capteurs ou à des actionneurs.
- Les FFD sont des nœuds ayant l'ensemble des fonctionnalités et la capacité à router ou à coordonner le réseau.

Aussi IEEE 802.15.4 définit deux topologies : la topologie en étoile et la topologie pair à pair.

- **La topologie en étoile :** Est composée d'un nœud central appelé le coordinateur, et de nœuds feuilles, ces derniers communiquent entre elles en passant systématiquement par le coordinateur.
- **La topologie pair à pair :** Cette topologie diffère de la topologie en étoile est composée de trois types de nœuds : le coordinateur du PAN, des coordinateurs et des feuille. Chaque entité sauf les feuilles est capable de communiquer avec les nœuds qui sont à sa portée.

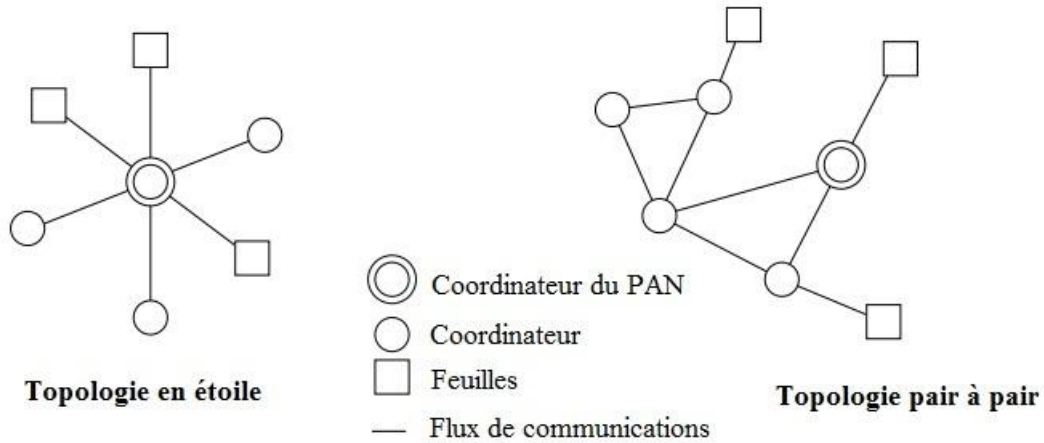


Figure 2.6 Topologie IEEE 802.15.4

- **La couche physique** : gère principalement l’activation ou la désactivation du module radio, la transmission des trames et leur réception et le test de médium.
- **La couche liaison de données** : gère l’accès au médium, l’acquiescement des trames, l’association des nœuds et la dés-association des nœuds.

Le standard définit deux modes : le mode avec suivi de balises lorsque le réseau nécessite une synchronisation ou de faibles latences et le mode sans suivi de balises.

Parmi les standards les plus aptes à être exploités dans les réseaux de capteurs sans fil.

Protocole	Zigbee	Bleutooth	Wifi
IEEE	802.15.4	802.15.1	802.11
Besoin mémoire	4 – 32 Kb	+ 250 Kb	+ 1 Mb
Durée de vie	Années	Jours	Heures
Nombres de nœuds	+ 65000	7	32
Vitesse de transfert	250 Kb/s	1 Mb/s	11 – 54 Mb/s
Portée	100 m	10 m	300 m

Tableau 2.1 Comparaison entre Zigbee , Bluetooth et Wifi [23]

## 10. La transmission de l'information dans les réseaux de capteurs

La transmission de l'information peut se faire en deux façons [24] :

- **Envoi direct** : chaque nœud envoie l'information au nœud puit sans aucun intermédiaire, ce dernier joue la passerelle qui permet l'interconnexion avec d'autres réseaux (satellites, internet,...) et la récupération des données comme le cas illustré dans la figure ci-dessous.

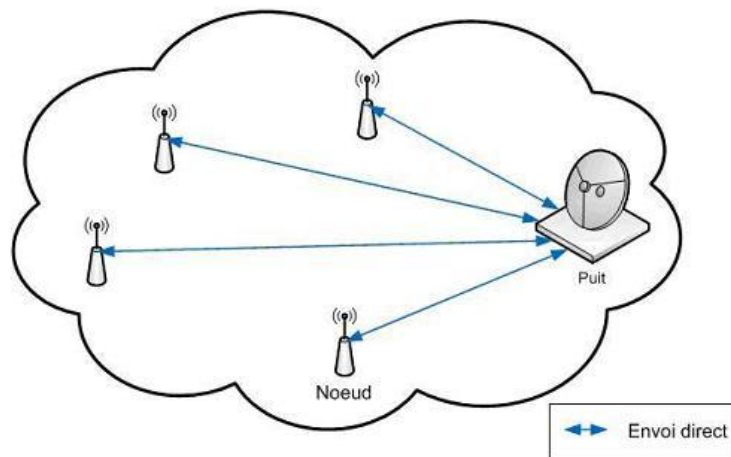


Figure 2.7 Envoi direct.

- **L'envoi par routage Ad-hoc** : Lorsque des nœuds ne sont pas reliés à l'unité de collecte, l'envoi direct n'est pas possible, c'est le cas d'appliquer des protocoles de routages de l'information qui peuvent être divisés en trois types : réactifs, proactifs ou hybrides.

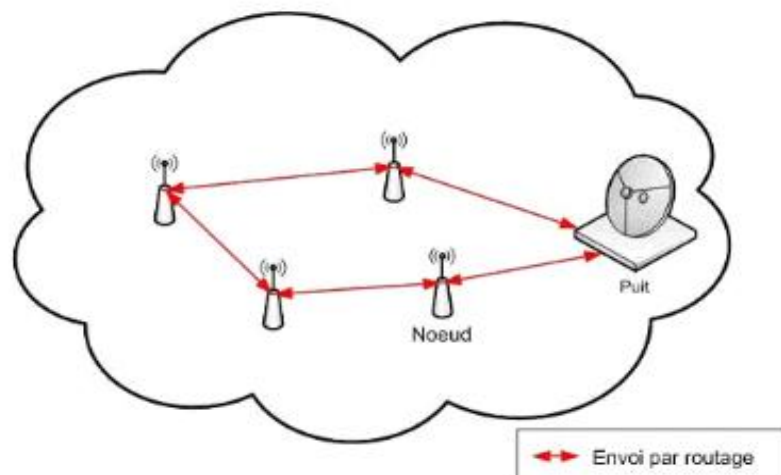


Figure 2.8 Envoi par routage Ad-hoc.

## 11. Protocoles de routage dans les réseaux de capteurs

Le routage est une méthode d'acheminement des informations à la destination appropriée à travers un réseau de connexion donné, aussi pour un but de découvrir les chemins qui existent entre les nœuds.

Un nœud A du réseau Ad-hoc souhaite envoyer un message à un nœud B de ce même réseau, ; comment détermine-t-il l'ensemble des nœuds du réseau par lequel le message doit être relié de proche en proche via le médium radio pour atteindre B [25].

Tout cela, c'est grâce aux protocoles de routage qui seront présentés dans ce qui suit :

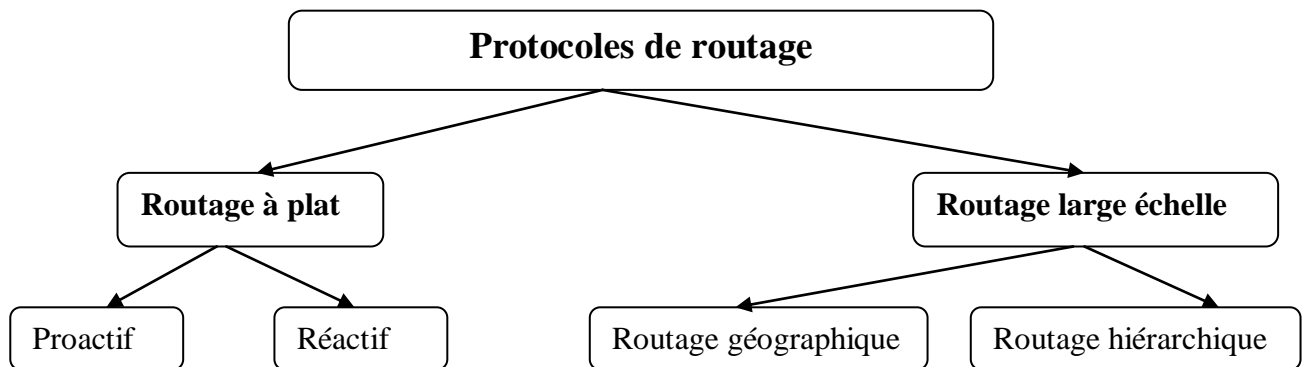


Figure 2.9 Protocoles de routage.

### 11.1 Routage à plat

On distingue deux protocoles de routage proactifs et réactifs.

- **Les protocoles proactifs** : Les routes dans ce type de routage sont calculées à l'avance. Chaque nœud met à jour plusieurs tables de routage par échange de paquets de contrôle entre voisins [26].

On peut citer par exemple les protocoles FSR (*Fisheye State Routing*) ou OLSR (*Optimized Link State Routing*).

- **Les protocoles réactifs :** Contrairement aux protocoles proactifs, les protocoles réactifs ne calculent la route que sur demande, il faut donc un laps de temps nécessaire à la recherche du chemin [26].

On peut citer par exemple les protocoles DSR (*Dynamic Source Routing*) et AODV (*Ad hoc On demand Distance Vector routing*).

## 11.2 Routage large échelle

On distingue deux types de routage géographique et hiérarchique.

- **Routage géographique :** il s'agit d'un routage prenant en compte la situation physique des nœuds. Les prés-requis pour effectuer un routage géographique dans un réseau ad hoc sont :
  - Tous les nœuds possèdent un moyen de localisation, soit un système natif comme le GPS (*Global Position System*), soit un système logiciel comme un protocole de localisation.
  - Un nœud source connaît toujours la position du nœud destinataire. Pour ce faire, soit tous les nœuds connaissent les positions initiales de tous les nœuds, soit un service de localisation doit être utilisé.

On peut citer par exemple les protocoles GPSR (*Greedy Perimeter Stateless Routing*) et GRA (*Geographical Routing Algorithm*).

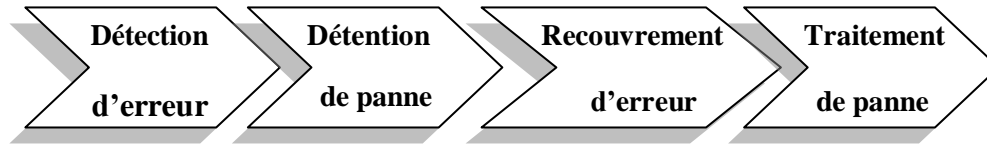
- **Routage hiérarchique :** Le réseau est constitué d'un ensemble de groupe de capteurs (cluster), Le nœud représentant le cluster, appelé cluster-head, a la responsabilité de transmettre les données à la station de base.

On peut citer par exemple les protocoles HSR (*Hierarchical State Routing*) et CGSR (*Clusterhead Gateway Switch Routing*).

## 12. La tolérance aux pannes dans les réseaux de capteurs

La limitation d'énergie dans les capteurs sans fil, et les environnements hostiles dans lesquels ils pourraient être déployés, sont des facteurs qui rendent ce type de réseaux très vulnérables.

La panne d'un nœud capteur ne doit pas affecter le fonctionnement global de son réseau. C'est le problème de fiabilité ou de tolérance aux pannes qui a donné une conception d'une procédure qui dépend de l'architecture et des fonctionnalités du système [27].



**Figure 2.10** Procédure de tolérance aux pannes.

➤ **Détection d'erreur**

C'est la première phase dans chaque schéma de tolérance aux pannes, dans laquelle on reconnaît qu'un événement inattendu s'est produit. Les techniques de détection de pannes sont généralement classifiées en deux catégories : en ligne et autonome (offline). La détection offline est souvent réalisée à l'aide de programmes de diagnostic qui s'exécutent quand le système est inactif. La détection en ligne vise l'identification de pannes en temps réel et est effectuée simultanément avec l'activité du système.

➤ **Détention de panne**

Cette phase établit des limites des effets de la panne sur une zone particulière afin d'empêcher la contamination des autres régions. En cas de détection d'intrusion par exemple, l'isolation des composants compromis minimise le risque d'attaque des composants encore fonctionnels.

➤ **Recouvrement d'erreur**

C'est la phase dans laquelle on effectue des opérations d'élimination des effets de pannes. Les deux techniques les plus utilisées sont « masquage de panne » et « répétition », dont la première utilise l'information redondante correcte pour éliminer l'impact de l'information erronée et la deuxième après que la panne soit détectée, on effectue un nouvel essai pour exécuter une partie du programme, dans l'espoir que la panne soit transitoire.

➤ **Traitement de panne**

Dans cette phase, la réparation du composant en panne isolé est effectuée. La procédure de réparation dépend du type de la panne. Les pannes permanentes exigent une substitution du composant avec un autre composant fonctionnel. Le système doit contenir un ensemble d'éléments redondants (ou en état standby) qui servent à remplacer les nœuds en panne.

### 13. Conclusion

Dans ce chapitre nous nous sommes concentrés sur la présentation de réseaux de capteurs sans fil qui possèdent des caractéristiques particulières et les différencient des autres types des réseaux sans fil, ces spécificités telles que la consommation d'énergie réduite. En outre, nous avons décrit les principales caractéristiques des RCSFs et leurs architectures matérielles et logicielles .

Dans la suite de ce mémoire (chapitre III), nous présenterons le système d'exploitation TinyOS, avec ses propriétés, caractéristiques et sa structure logicielle qui va nous permettre la conception de notre application au but de la mise en œuvre d'un capteur sans fil pour la détection des grandeurs physiques qui provient de l'environnement .

# ***Chapitre 3 :***

## ***Architecture Logicielle***

---



## 1. Introduction

Le Tinyos est un système d'exploitation embarqué le plus célèbre développé par Berkeley qui cible les réseaux de capteurs sans fil, il présente un environnement de calcul particulier soumis à plusieurs contraintes tel que l'énergie, l'espace mémoire...etc.

Dans ce chapitre, nous introduirons le mode de fonctionnement de la plateforme TinyOS et le langage NesC

## 2. TinyOS

**TinyOS** est un système d'exploitation open-source destiné pour des réseaux de capteurs sans fil conçu par l'Université Américaine de BERKELEY.

Il respecte une architecture basée sur une association de composants, réduisant la taille du code nécessaire à sa mise en place. Cela s'inscrit dans le respect des contraintes de mémoires qu'observent les réseaux de capteurs [28].



Figure 3.1 Logo TinyOS

### 2.1 Propriétés de TinyOS

TinyOs est basé sur quatre propriétés qui vont lui permettre de s'adapter aux systèmes à faible ressources [29].

➤ **Temps réel** : Lorsqu'un système est dit « temps réel » celui-ci gère des niveaux de priorité dans ses tâches permettant de respecter des échéances données par son environnement. Dans le cas d'un système strict, aucune échéance ne tolère de dépassement contrairement à un système temps réel. TinyOS se situe au-delà de ce second type car il n'est pas prévu pour avoir un fonctionnement temps réel.

➤ **Non préemptif** : Le caractère préemptif d'un système d'exploitation précise si celui-ci permet l'interruption d'une tâche en cours. TinyOS ne gère pas ce mécanisme de préemption entre les tâches mais donne la priorité aux interruptions matérielles. Ainsi, les tâches entre elles ne s'interrompent pas mais une interruption peut stopper l'exécution d'une tâche.

➤ **Évènementiel** : Le fonctionnement d'un système basé sur TinyOS s'appuie sur la gestion des évènements se produisant. Ainsi, l'activation d'une tâche, leur interruption ou encore la mise en veille du capteur s'effectue à l'apparition d'évènements, ceux-ci ayant la plus forte priorité. Ce fonctionnement évènementiel *eventdriven* s'oppose au fonctionnement dit temporel ou *timedriven* où les actions du système sont gérées par une horloge donnée.

➤ **Consommation** : TinyOS a été conçu pour réduire au maximum la consommation en énergie du capteur. Ainsi, lorsqu'aucune tâche n'est active, il se met automatiquement en veille.

## 2.2 Modèle d'exécution de TinyOS

Pour maintenir une grande efficacité requise par les réseaux de capteurs, TinyOS utilise une programmation par évènement. Ce modèle permet un très haut niveau de concurrence pour un espace très réduit de mémoire.

L'implémentation des composants de TinyOS s'effectue en déclarant des tâches, des commandes ou des évènements [30].

### 2.2.1 Programmation par évènement

Dans un système basé sur la programmation par évènement, chaque exécution est partagée entre les différentes tâches de traitement. Dans TinyOS, chaque module est désigné pour fonctionner en attendant continuellement de répondre aux évènements inattendus.

Quand un évènement est signalé, le traitement correspondant est exécuté. Une fois totalement terminé, la main est redonnée au système pour continuer sa tâche antérieure.

En plus de l'efficacité de l'allocation du CPU, la programmation par évènement permet d'obtenir des opérations économiques en énergie. Il est très important aussi pour la consommation d'énergie que les applications signalent la fin de leurs évènements et l'utilisation du CPU.

Dans TinyOS, les tâches associées avec un évènement sont exécutées très rapidement après leurs signalisations. Une fois terminé, le CPU entre en veille en attendant une nouvelle réception d'évènement [30].

### 2.2.2 Les Tâches

Un des facteurs limitant la programmation par évènement est la longue exécution des tâches qui peut interrompre d'autres programmes importants. Si l'exécution d'un évènement ne finit jamais, toutes les autres fonctions vont être interrompues.

Pour éviter ce problème, TinyOS fourni un mécanisme d'exécution appelé tâche. Une tâche est un bout de programme qui s'exécute jusqu'à la fin sans interférer avec les autres évènements. Les tâches sont utilisées pour effectuer la plupart des blocs d'instruction d'une application.

A l'appel d'une tâche, celle-ci va prendre place dans une file d'attente de type FIFO (*First In First Out*), mais elle ne sera exécuté que lorsque il n'y a plus d'évènements. En plus les tâches peuvent être interrompues à tout moment par des évènements.

D'autre part, il n'y a pas de mécanisme de préemption entre les tâches et une tâche activée s'exécute en entier. Ce mode de fonctionnement permet de bannir les opérations pouvant bloquer le système (inter blocage, famine, ...).

Par ailleurs, lorsque la file d'attente des tâches est vide, le système d'exploitation met en veille le dispositif jusqu'au lancement de la prochaine interruption (on retrouve le fonctionnement Event-driven [30]).

### 2.2.3 L'ordonnanceur TinyOS

La gestion des tâches et des évènements est vitale pour TinyOS. Le choix d'un ordonnanceur déterminera le fonctionnement global du système et le dotera de propriétés précises telles que la capacité à fonctionner en temps réel.

L'ordonnanceur TinyOS c'est :

- Deux niveaux de priorité ( bas pour les tâches, haut pour les évènements) ;
- Un file d'attente FIFO ( disposant d'une capacité de sept).

Par ailleurs, entre les tâches, un niveau de priorité est défini permettant de classer les tâches, tout en respectant la priorité des interruptions (ou évènements). Lors de l'arrivée d'une nouvelle tâche, celle-ci sera placée dans la file d'attente en fonction de sa priorité (plus elle est grande, plus le placement est proche de la sortie). Dans le cas ou la file d'attente est pleine, la tâche dont la priorité est la plus faible est enlevée de la FIFO [30].

### 3. NesC

Le système d'exploitation TinyOS s'appuie sur le langage NesC. Celui-ci propose une architecture basée sur des composants, permettant de réduire considérablement la taille mémoire du système et de ses applications. Chaque composant correspond à un élément matériel (LEDs, Timer, ADC ...) et peut être réutilisé dans différentes applications.

Ces applications sont des ensembles de composants associés dans un but précis. Les composants peuvent être des concepts abstraits ou bien des interfaces logicielles aux entrées sorties matérielles de la cible étudiée (carte ou dispositif électronique).

Les composants NesC présentent des similarités avec des objets. Les états sont encapsulés et on peut y accéder par des interfaces. En NesC, l'ensemble des composants et leurs interactions sont fixés à la compilation pour plus d'efficacité. Ce type de compilation permet d'optimiser l'application pour une exécution plus performante. En langage objet, cette phase est réalisée lors de l'exécution ce qui rend celle ci plus lente.

#### 3.1 Les Principales caractéristiques de NesC

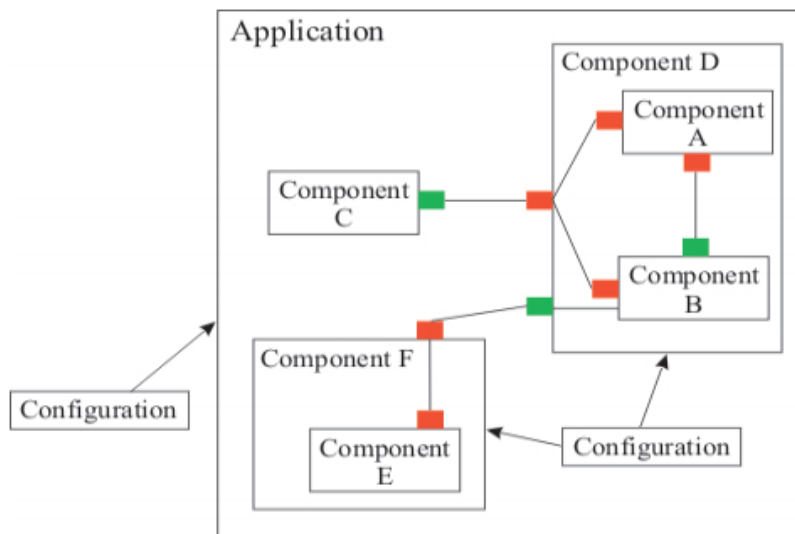
NesC est constitué d'interfaces et de composants. Une interface peut être utilisée ou peut être fournie.

Les composants sont des modules ou des configurations. Une application est représentée comme un ensemble de composants, regroupés et rattachés entre eux (figure 3.2).

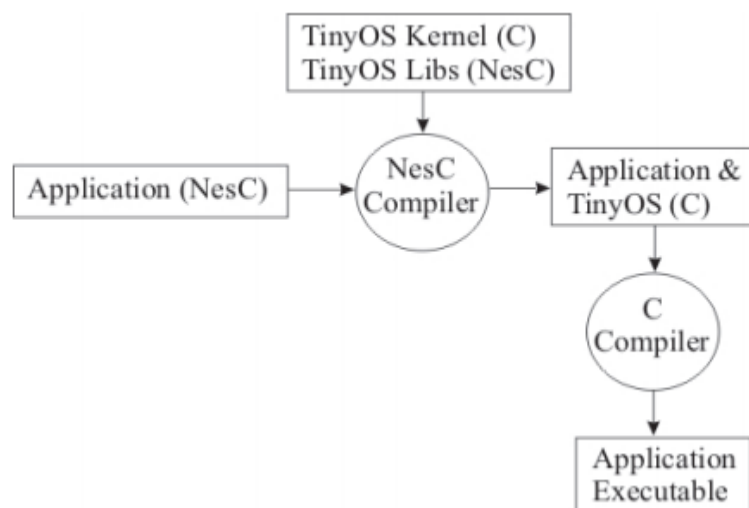
Les interfaces sont utilisées pour les opérations qui décrivent l'interaction bidirectionnelle. Le fournisseur de l'interface doit mettre en application des commandes, alors que l'utilisateur de l'interface doit mettre en application des événements [31].

Deux types de composants existent : Les modules, qui mettent en application des spécifications d'un composant. Les configurations, qui se chargeront d'unir différents composants en fonction des interfaces (commandes ou événements).

La figure 3.3 montre un diagramme de blocs dans lequel est décrit le processus de compilation pour une application TinyOS écrite en NesC.



**Figure 3.2** Architecture d'une application NesC



**Figure 3.3** Processus de compilation

### 3.2 Les fichiers dans NesC

Les fichiers de NesC sont classés en trois types : Interfaces, modules et configurations.

- **Interface** : Ce type de fichier déclare les services fournis et les services qui seront utilisés. Ils se trouvent dans le répertoire /tos/interface. (Exemple : StdControl.nc).
- **Module** : Le type Module contient le code de l'application, en mettant en œuvre une ou plusieurs interfaces. (Exemple : BlinkM.nc)

- **Configuration** : Dans ces fichiers on déclare la manière d'unir les différents composants et comment effectuer le contrôle des flux. (Exemple : Blink.nc).

### 3.3 Concepts principaux dans NesC

#### 3.3.1 Composants

TinyOS définit un nombre important de concepts qui sont exprimés dans NesC. D'abord, les applications NesC sont construites par des composants avec des interfaces bidirectionnelles définies. Aussi, NesC définit un modèle basé sur les tâches et les captures d'événements matériels, et détecte des éclatements d'information pendant la compilation.

Un composant, du point de vue de la programmation, est composé de plusieurs sections et l'ensemble de toutes ces sections donne lieu à la création de ce composant.

#### 3.3.2 Implémentations

Cette section définit les connections entre les différents composants qu'utilise l'application. Dans cette section « implémentation » sont donc principalement définis. Quels sont les composants qui fournissent les interfaces à notre application (ils seront généralement des composants primitifs).

Généralement nous devons utiliser les interfaces que nous fournissent d'autres composants primitifs ou non primitifs, et définitivement pour chacune de ces interfaces, que nous utiliserons dans la création de notre composant, on doit obligatoirement définir des relations avec les composants qui fournissent ces interfaces [31].

Le processus définissant ces relations s'appelle « wiring »

**Exemple :**

```
implementation {
  components Main , MonAppliM ;
  Main . StdControl -> MonAppliM .
  StdControl ;
}
```

### 3.3.3 Configurations

C'est à cet endroit que l'on déclare les autres composants dont se servira l'application. Cette possibilité offerte par le langage permet de faire de la programmation modulaire et de réutiliser des composants préalablement définis.

La structure de la partie Configurations est la même que celle de la partie Implémentation.

**Exemple :**

```
Configuration MonAppli {
  implementation {
    Components <mettre les composant
nécessaire> }}
```

### 3.3.4 Modules

Cette partie du code est généralement plus étendue et c'est dans celle-ci que l'on programme réellement le comportement qu'on souhaite voir réaliser par l'application.

Cette partie l a et à son tour divisée en trois sous-sections : « Uses, Provides, Implementation ».

**Exemple :**

```
module MonAppliM {
  provides {... }
  uses {... }
}
implementation {... }
```

La première sous-section, « provides », indique au compilateur les interfaces que va fournir notre composant. Par exemple, si notre composant est une application, on doit fournir au moins l'interface « StdControl ».

```
provides {
  interface
interfacequenousfournissons ;
}
```

La sous-section « uses » informe le compilateur que nous allons faire usage d'une interface (on pourra donc effectuer des appels aux méthodes de cette interface).

Pour faire cela, on a besoin de respecter quelques règles : si nous utilisons une interface, il nous faut avoir dans la section « implementation » un lien « wiring » reliant cette interface avec un composant qui la fournit.

Finalement, utiliser une interface oblige implicitement à gérer les événements pouvant se produire du fait d'avoir utilisé cette interface précise.

```
uses {  
  interface  
  interfaceque nous utilisons ;  
}
```

La sous-section « implementation », est celle qui contiendra toutes les méthodes nécessaires pour fournir le comportement souhaité à notre composant ou à notre application. Cette sous-section doit contenir au moins :

- Les variables globales qui vont utiliser notre application.
- Les fonctions qu'elle doit mettre en œuvre pour les interfaces que nous fournissons.
- Les événements qu'elle doit mettre en œuvre venant des interfaces que nous utilisons.

### 3.4 Types des données

Les types de données qui peuvent être utilisés en NesC sont tous ceux que fournit le langage C standard plus quelques autres qui n'apportent pas de puissance de calcul mais qui sont très utiles pour la construction de paquets puisqu'ils fournissent à l'utilisateur le nombre de bits qu'ils occupent (ceci est important au moment de la transmission des informations par l'intermédiaire des ondes radio).

Ces types additionnels sont :

- `uint16_t` : entier non signé sur 16 bits.
- `uint8_t` : entier non signé sur 8 bits.
- `result_t` : utilisé pour savoir si une fonction a été exécuté avec succès ou non, c'est comme un booléen mais avec les valeurs SUCCESS et FAIL. (retour de fonction)
- `bool` : valeur booléenne qui peut être TRUE ou FALSE.



En NesC, il est possible de faire une utilisation dynamique de la mémoire mais ce n'est pas très recommandé (à moins que cela ne soit absolument nécessaire). Pour pouvoir l'utiliser il existe un composant spécial appelé MemAlloc qui permet une gestion dynamique de la mémoire.

### 3.5 Types de fonctions en NesC

En NesC les fonctions peuvent être de types très variés.

Il y a d'abord les fonctions classiques avec la même sémantique qu'en C et la façon de les invoquer est aussi la même qu'en C. Il y a aussi des types supplémentaires de fonctions : task, event, command.

Les fonctions « command » sont principalement des fonctions qui sont exécutées de manière synchrone, [31] c'est-à-dire que lorsqu'elles sont appelées, elles sont exécutées immédiatement. La manière d'appeler ce genre de fonction est :

```
call interface . nomFonction
```

Les fonctions « task » sont des fonctions qui sont exécutées dans l'application, utilisant la même philosophie que les fils ou « threads », c'est principalement une fonction normale qui est invoquée de la manière suivante :

```
post interface . nomTache
```

Immédiatement après son invocation, l'exécution du programme qui l'a invoqué se poursuit. Les fonctions « event » sont des fonctions qui sont appelées quand on relèvera un signal dans le système, elles possèdent principalement la même philosophie que la programmation orientée événements, de sorte que lorsque le composant reçoit un événement, on effectue l'invocation de cette fonction. Il existe une méthode pour pouvoir invoquer manuellement ce type de fonctions :

```
signal interface . nomEvenement
```

#### **4. Conclusion**

Dans ce chapitre, nous avons présenté le système d'exploitation TinyOS, avec ses propriétés, caractéristiques et sa structure logicielle. Nous avons parlé un peu sur l'allocation de mémoire et de ressources par la plateforme TinyOS.

Nous avons également présenté le langage de programmation NesC qui est créé pour la conception des applications embarquées et plus précisément pour les réseaux de capteurs sans fil .

Dans le prochain chapitre , nous présenterons les détails de notre application et de parcourir les étapes de conception qui nous seront nécessaires pour la mise en œuvre d'un capteur sans fil pour la détection de la température, l'humidité et le niveau de CO et CO2.

# ***Chapitre 4 :***

## ***Conception et réalisation de l'application***

---

*Le projet est le brouillon de l'avenir.  
Parfois il faut à l'avenir des  
centaines de brouillons.*

**Jules RENARD**

## 1. Introduction

Les réseaux de capteurs sont généralement déployés pour observer un phénomène particulier et leur collaboration permet de collecter des données et d'envoyer des informations sur le phénomène supervisé. Ces informations peuvent être exploitées directement ou stockées dans une base de données pour effectuer des éventuelles statistiques.

## 2. Objectif d'application

L'application développée consiste à faire des prélèvements de températures, humidité, CO et CO2 dans plusieurs endroits d'une zone d'intérêt et les envoyer à un site WEB qu'il les analyse et les stocke dans une base de données.

Le processus d'exécution de notre application suit les démarches suivantes:

- Détecter les paramètres de l'environnement (température, humidité, CO et CO2) grâce à un nœud capteur.
- Envoyer les données captées vers la station de base via l'interface radio.
- La station de base est responsable de la collecte de toutes les données issues des nœuds de capteurs dans laquelle est connectée à un PC afin de transmettre les données collectées à notre Site Web.
- Enfin, le site Web est réalisé pour la visualisation des données captées et faire des statistiques sur les mesures effectuées suivant la date ou les emplacements dans lesquelles sont déployés les capteurs.

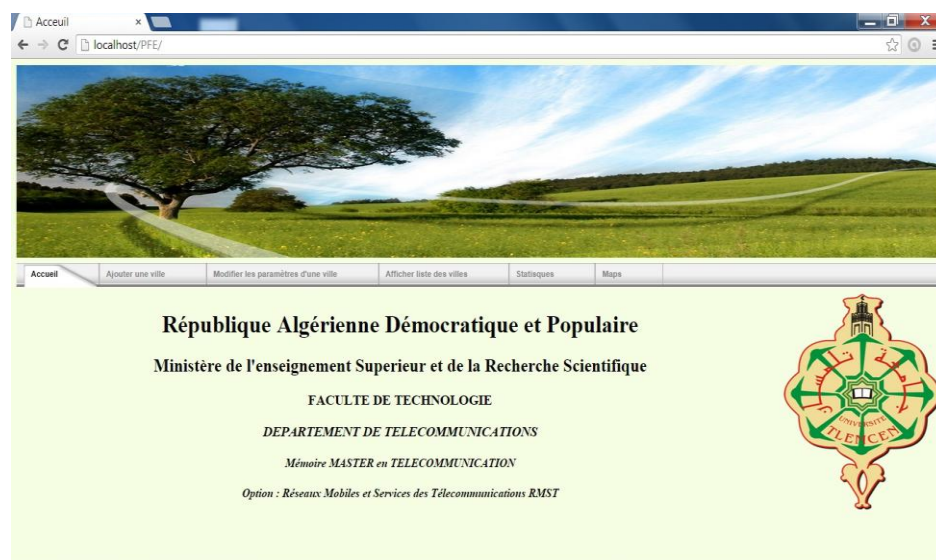


Figure 4.1 Page d'accueil du site web

### 3. Environnement de travail

Dans cette section, nous présentons l'environnement de travail dans lequel nous avons développé notre application et les différents outils utilisés.

#### 3.1 Architecture matérielle

Nous avons développé notre application en utilisant :

- **Un PC portable** : ayant les caractéristiques suivantes:
  - Type : HP 650
  - Processeur : Intel Core i3-2328M CPU 2.20 Ghz
  - Disque dur de capacité : 300 Go
  - Système d'exploitation Microsoft Windows 7 avec une machine virtuelle pour exécuter linux qui contient les outils de développement TinyOS.
- **Capteurs** : on a utilisé deux types de capteurs :
  - MTM- CM5000- MSP « MAXFOR TECHNOLOGY ING »
  - MTM- CM3000- MSP avec Sensor Board DS1000.

#### 3.2 Architecture logicielle

Suite à l'incapacité de trouver des plateformes de développement pour réseaux de capteurs sous Windows, on a eu tendance à installer une machine virtuelle VirtualBox permettant d'exécuter d'autres systèmes d'exploitations sous Windows. En utilisant les ressources matérielles de l'ordinateur (système hôte) , il permet la création d'un ou de plusieurs ordinateurs virtuel dans lesquels s'installent d'autres systèmes d'exploitation (systèmes invités).

Les systèmes invités fonctionnent en même temps que le système hôte, mais seul ce dernier a accès directement au véritable matériel de l'ordinateur.

Les systèmes invités exploitent le matériel générique, simulé par un « faux ordinateur » (Machine Virtuel) créé par VirtualBox [32].

## 4. Fonctionnement de notre système

### 4.1 Installation logicielle

La première étape a été de prendre en main les capteurs, le langage NesC et le système d'exploitation TinyOs. On a utilisé l'environnement Linux (Ubuntu) pour la mise en place de notre application.

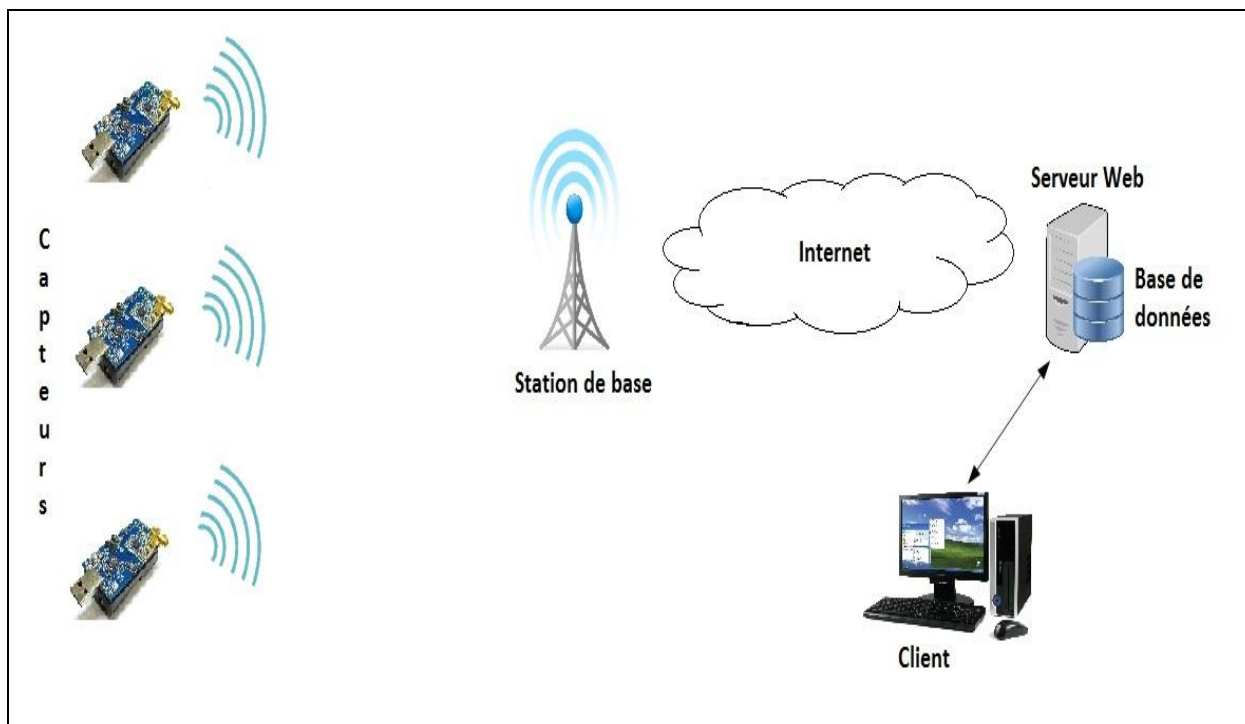
La phase d'installation a été la plus délicate. En effet, on a commencé par installer TinyOS2.1.1, pour les capteurs de type TelosB.

### 4.2 Installation matérielle

Une fois l'installation logicielle est terminée, il a fallu installer le matériel : une station de base reliée à l'ordinateur via un port USB, différents capteurs TelosB (Senders).

Chacun des capteurs Senders mesure les paramètres suivants : température, humidité, CO et CO2 et les communique à la station de base via une liaison sans fil, et la station de base qui communique ces mesures vers le site web hébergé sur internet.

La figure 4.2 présente un schéma de communication dans la plateforme qu'on a développée au cours de notre application.



**Figure 4.2** Schéma de communication dans la plateforme.

## 5. Description de la plateforme WEB

Afin de mettre en place cette plateforme et tester son fonctionnement on a utilisé les outils suivants :

### 5.1 Wampserver

WAMP signifie Windows + Apache + MySQL + PHP, est un paquetage regroupant un serveur Apache, une base de données MySQL et un interpréteur de script (PHP). Il permet d'installer automatiquement et facilement une plateforme permettant l'exploitation d'un site web en PHP qui éventuellement aurait besoin d'un accès à une base de données.

### 5.2 PHPmyadmin

PhpMyAdmin est une interface d'administration pour le SGBD MySQL, cette interface écrite en PHP et s'appuie sur le serveur HTTP Apache, elle permet également de donner à l'utilisateur un accès à ses propres bases de données.

### 5.3 Notepad++

Un éditeur de texte gratuit qui colore automatiquement le code que l'on entre en fonction de la logique de votre langage informatique, ce qui vous permettra une relecture facile. Cette aide indispensable s'appelle la coloration syntaxique de code source.

### 5.4 Les serveurs

#### 5.4.1 Serveur Apache

C'est le serveur le plus répandu sur Internet, permettant la configuration de l'environnement d'exécution des pages web. Il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type Unix, mais il a désormais été porté sur de nombreux systèmes, dont Microsoft Windows grâce à sa conception modulaire (morceaux de code) qui correspond à différents aspects ou fonctions du serveur.

Cette conception autorise le développeur à choisir quelles fonctionnalités seront incluses dans le serveur en sélectionnant les modules à charger soit à la compilation, soit à l'exécution.

Elle lui permet aussi d'écrire son propre morceau de code qui pourra ensuite être facilement intégré dans le serveur Web Apache.

### 5.4.2 Serveur MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR) libre fonctionnant sous diverses plates-formes telles que UNIX, Linux et Windows.

Pour faire simple, son rôle est d'enregistrer des données de manière organisée afin de vous aider à les retrouver facilement plus tard.

## 5.5 Création des tables

Dans l'application qu'on a développée, on a créé une base de données nommé « Database ». Celle-ci contiendra deux tables nommées « Capture » « GPS ».

La première table « **Capture** » est utilisée pour conserver les données envoyées par la station de base vers notre site web (Id, Emplacement , Température, Humidité, Co, Co2 , Date).

Capture	Type
<i>id</i>	<i>Int</i>
<i>plac</i>	<i>Varchar (50)</i>
<i>temp</i>	<i>Varchar (50)</i>
<i>humid</i>	<i>Varchar (50)</i>
<i>co</i>	<i>Varchar (50)</i>
<i>co2</i>	<i>Varchar (50)</i>
<i>dte</i>	<i>Date</i>

Figure 4.3 Aperçu de table « Capture »

La deuxième table « **GPS** » est utilisée pour conserver les emplacements dans lesquelles sont déployés les capteurs afin de les visualiser sur une carte à l'aide d'un site de géolocalisation tel que Google Map Api. Cette table comprend les champs suivants (Id, Emplacement, Altitude, Longitude).

GPS	Type
<i>id</i>	<i>int</i>
<i>plac</i>	<i>Varchar (50)</i>
<i>x</i>	<i>Varchar (50)</i>
<i>y</i>	<i>Varchar (50)</i>

Figure 4.4 Aperçu de table « GPS »



## 6. Les langages de programmation utilisés

### 6.1 Le langage HTML & CSS

HTML (HyperText Mark up Langage) : est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques.

CSS (Cascading Style Sheets): Le langage CSS permet de définir des feuilles de style qui peuvent être appliquées à un site Web. Il permet la manipulation des styles appliqués à chaque balise HTML, *via* un langage de script [24].

### 6.2 Le langage de requête SQL

Pour communiquer avec une base de données, on a besoin de lui envoyer des commandes ou instructions appelées requêtes. Que ce soit pour la création, la suppression d'une table, la modification, l'insertion ou la sélection de données, le langage standard de requêtes est SQL.

SQL ou (*Standard Query Language*) est un langage permettant d'interroger les bases de données de manière simple. Il est doté d'une syntaxe particulière que l'on doit respecter pour que la communication avec la base se passe au mieux.

SQL couvre les trois fonctions indispensables à la mise en œuvre et à l'exploitation de bases de données relationnelles :

- La création des données.
- La manipulation des données.
- Le contrôle des données.

### 6.3 Le langage de script PHP

Le « *PHP: Hypertext Preprocessor* », plus connu sous son sigle *PHP*, c'est un langage que seuls les serveurs comprennent et qui permet de rendre un site dynamique.

Un script PHP est exécuté par un interpréteur situé du côté serveur. Quand un fichier avec une extension *.PHP* est demandé au serveur web, ce dernier le charge en mémoire et y cherche tous les scripts PHP, qu'il transmet à l'interpréteur. L'interpréteur exécute le script, ce

qui permet de produire du code HTML qui vient remplacer le script PHP dans le document finalement fourni au navigateur. Ce dernier reçoit donc du HTML pur.

## 7. Les différentes parties de notre projet

Notre application est composée de deux parties :

### 7.1 Partie I : Le site web

Dans cette partie nous allons exposer les différentes pages Web de notre site:

- **Insertion** : Cette page permet d'insérer les coordonnées GPS « Altitude et Longitude » d'une ville ou un emplacement d'intérêt dans lequel on a déployé nos capteurs .

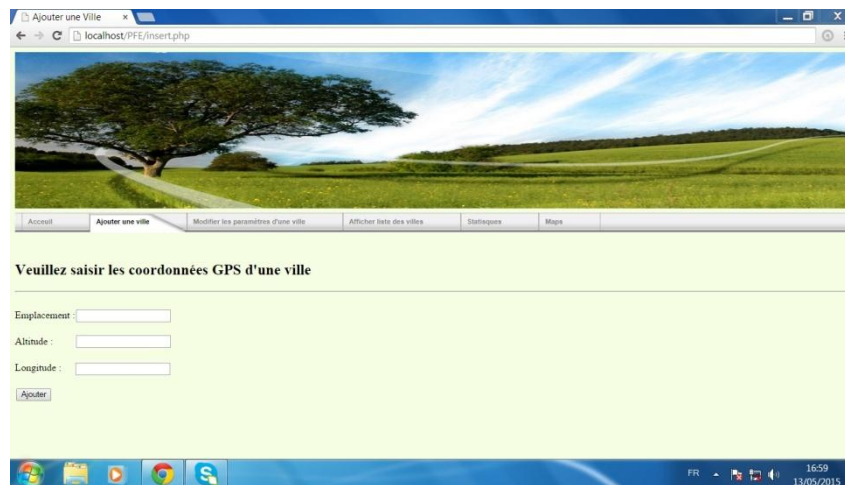


Figure 4.5 Onglet Insertion du site Web

- **Modification** : Cette page permet également de supprimer un emplacement ou de modifier les coordonnées GPS de cet emplacement.

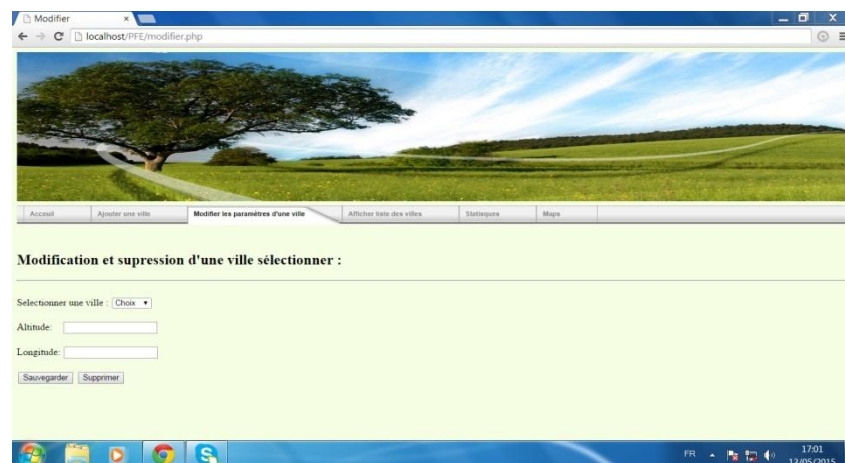


Figure 4.6 Onglet Modifier du site Web

- **Afficher liste des villes** : cet onglet permet d'afficher les valeurs actuelles de la température en degré Celsius (°C) et l'humidité en pourcentage (%), ainsi les valeurs du mono carbone et carbone en partie par million (ppm) dans les différents emplacements .

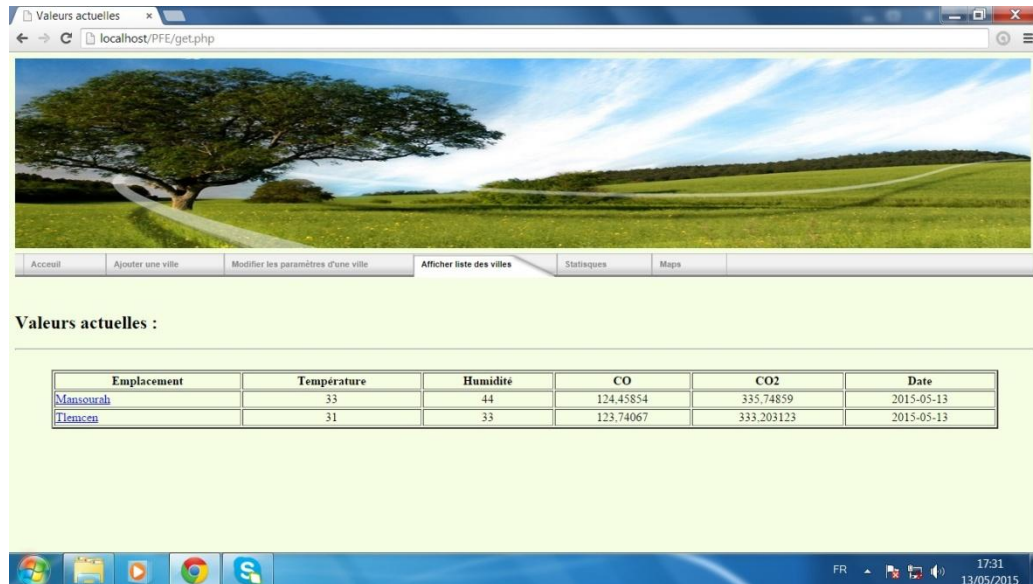


Figure 4.7 Onglet Afficher liste des villes du site Web

- **Statistiques** : cet onglet du site web permet de visualiser des statistiques sur les données capturées et sauvegarder dans la base de données afin de faciliter la tâche des écologistes et des observateurs utilisant cette plateforme

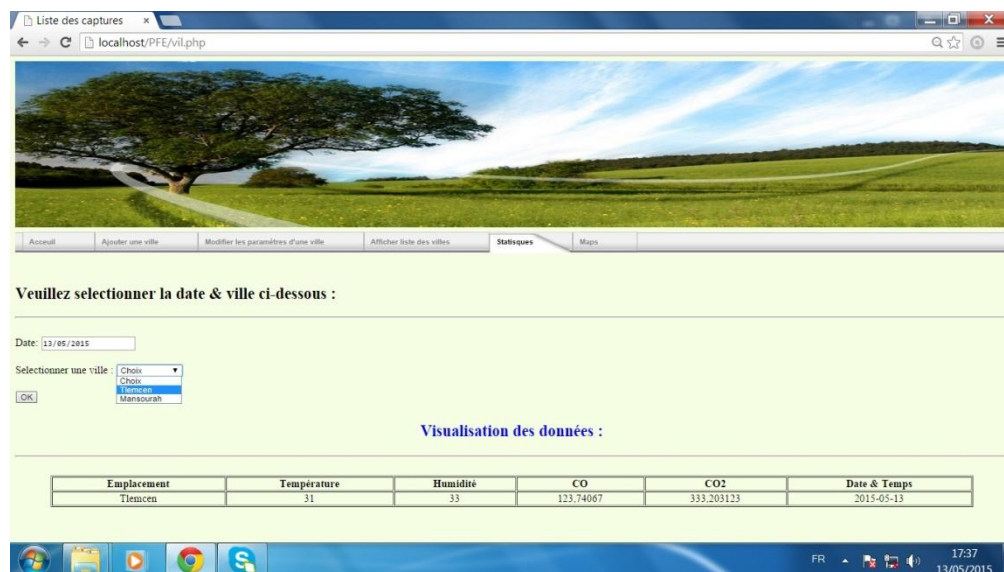
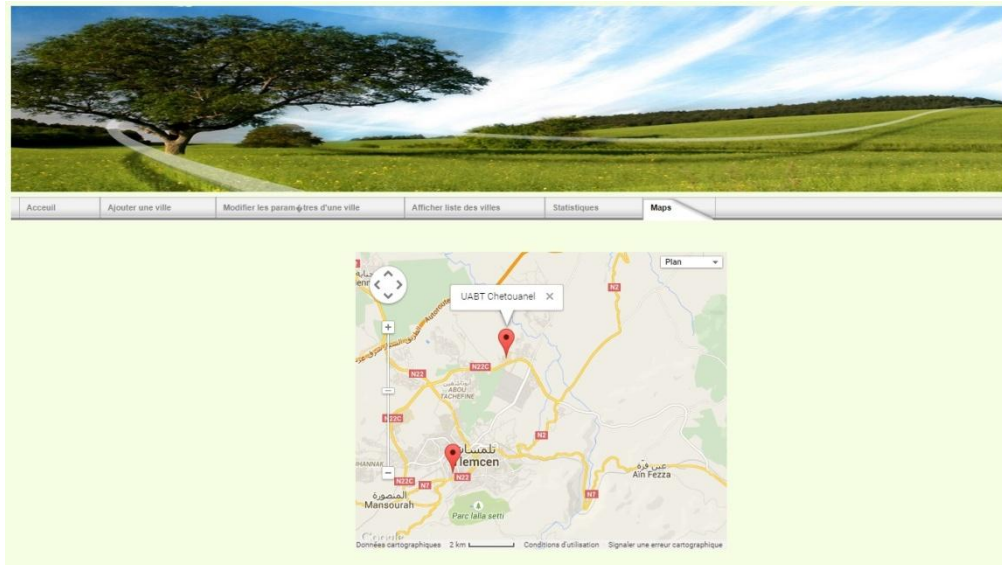


Figure 4.8 Onglet Statistiques du site Web

- **Maps** : cette onglet exploite Google Maps afin de visualiser les emplacements dans lesquels sont déployés les capteurs sur une carte géographique permettant ainsi de faciliter la localisation des capteurs pour les utilisateurs de l'application.



**Figure 4.9** Onglet Maps du site Web

## 7.2 Partie II : Module capture

Cette partie permet de décrire les différentes étapes de l'installation et le déploiement des capteurs.

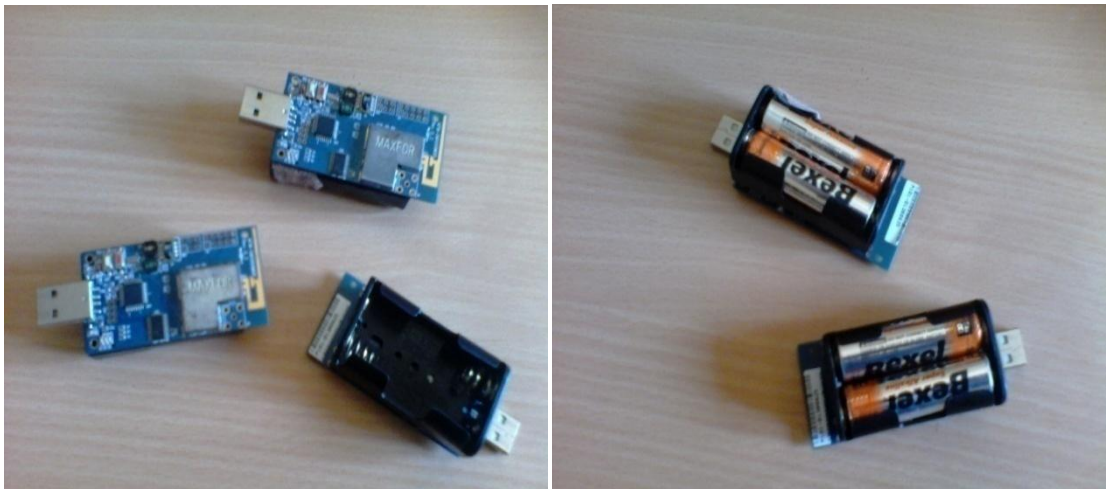
Au cours de notre application, nous avons utilisé deux types de capteurs, l'un MTM CM5000 pour la détection de la température et l'humidité, et CM 3000 avec un Sensor Board de type MTS DS 1000 pour la détection du niveau de CO et CO<sub>2</sub>.

- **MTM CM5000**

Conçus par l'université de Californie à Berkeley, ce mote est de type TelosB et se prénomme MTM-CM5000-MSP. Il est composé d'un microcontrôleur MSP430, d'un émetteur-récepteur, d'un capteur de température et d'humidité, d'un capteur de luminosité pour le domaine du visible, d'un autre pour l'infrarouge, d'un « bouton utilisateur », d'un bouton reset, de trois LEDs, d'un port USB....etc. Certains motes possèdent même un connecteur SMA (SubMiniature version A) qui permet d'ajouter une antenne supplémentaire, dans le but d'améliorer la portée du nœud, sa durée de vie est donc limitée par le voltage des



piles. En outre, il possède 10kB de mémoire vive et seulement 48kB de mémoire flash, ce qui est relativement peu.



**Figure 4.10** Capteurs MTM CM5000

➤ **MTM CM 3000**

Conçus par l'université de Californie à Berkeley, ce capteur est de type TelosB et se prénomme MTM-CM3000-MSP, il est composé d'un microcontrôleur MSP430F1611, d'une mémoire vive RAM de 10kB et 48kB mémoire flash .....etc.

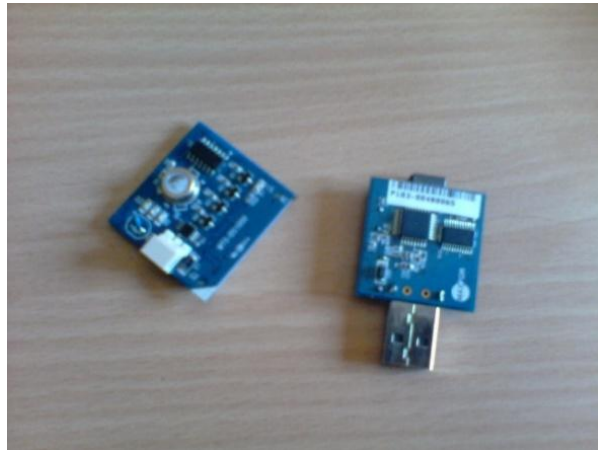
Il possède un connecteur SMA (SubMiniature version A) qui permet d'ajouter une antenne supplémentaire : dipôle de 5dBi, la durée de capteur est aussi limitée par le voltage des piles.



**Figure 4.11** Capteur MTM CM3000

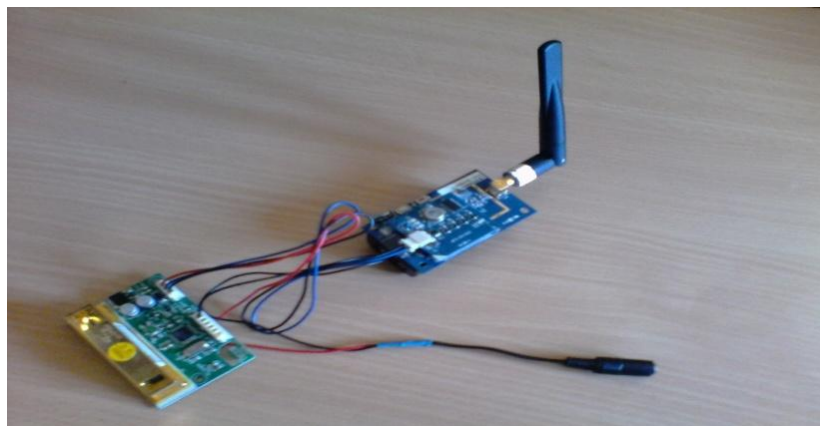
➤ **MTS DS 1000**

C'est un Sensor Board attachable, compatible avec toute la série CMXXXX de capteur WSN qui ont les 51-PIN connecteur, il inclut les sonde CO,CO2, Température.



**Figure 4.12** Sensor Board DS1000

Dans la figure suivante nous vous présentons l'état final de Sensor Board DS1000 Attaché au capteur MTM CM3000.



**Figure 4.13** Sensor Board DS1000 attaché au MTM CM 3000

## 8. Déploiement

Nous sommes en mesure de réaliser notre expérience à l'université AbouBekr Belkaid, Faculté de Technologie à CHETOUANE.

Pour cette expérience, notre application s'articule d'un kit de quatre capteurs TelosB, soit 3 nœuds capteur et une station de base pour faire les mesures de la température, l'humidité, CO et CO2 qui provient de l'environnement.

### 8.1 Déploiement des capteurs

Comme le montre la figure 4.14, on doit choisir des emplacements bien protégés pour les capteurs afin de les conserver du sabotage par des gens malintentionnés permettant de prolonger la durée de vie de notre réseau de capteurs.



Figure 4.14 Déploiement des capteurs MTM CM5000 et CM3000

### 8.2 Déploiement de station de base

La station de base est chargée de récupérer toutes les données envoyées par les autres nœuds du réseau ainsi que l'agrégation des données et envoyer ces données vers un PC ou autre plateforme comme un site web.

Comme station de base on propose d'utiliser une plateforme Raspberry Pi qui contient toutes la fonctionnalité d'un PC :

La carte Raspberry Pi (Figure 4.15), elle est très répandu pour multiples applications, par ce que sa plateforme est Linux, le système d'exploitation doit être charger dans SD carte mémoire (Raspbian « Wheezy »), le model B est équipé d'un port RJ45 pour les taches internet, deux ports USB 2.0, RAM intégrer de 512 Mo, un port HDMI pour l'affichage et 26 lignes I/O GPIO (General Purpose Input Output) on peut les utiliser comme des entrées ou des sorties du processeur ARM.



**Figure 4.15** Raspberry Pi avec capteur CM5000 et Clef 3G.

Dans l'absence du temps pour l'étude de la plateforme Raspberry Pi, on a utilisé comme station de base un PC de bureau de type SAMSUNG.



**Figure 4.16** Station de base.



## **9. Conclusion**

Dans ce chapitre, nous avons tenté de mettre en place une plateforme pour faire des mesure sur l'environnement ainsi que les démarches à suivre et les différents outils et langages utilisés pour réaliser cette application qui consiste à prélever la température, l'humidité, CO et CO2 dans différents endroits dans une ville telle que Tlemcen.

L'application est accompagnée d'un site web permettant la visualisation des données pour les écologistes et de faire des statistiques sur ces données.

# ***Conclusion générale***

---

### Conclusion générale

Les réseaux de capteurs sans fil ont un large potentiel et constituent un sujet de recherche innovant ainsi qu'un outil convoité par plusieurs domaines. Cependant, il y a encore beaucoup de problèmes qui doivent être abordés pour un fonctionnement efficace de ces réseaux dans des applications réelles, dont le routage dans les réseaux de capteurs et la gestion d'énergie sont des domaines de recherche d'une grande fertilité.

Toutefois, tout au long de notre projet, nous avons constaté que l'implémentation d'un réseau de capteurs sans-fil pour la détection de la température, l'humidité, CO et CO<sub>2</sub> pose de grands défis auxquels il faut répondre. Parmi ces défis, la maîtrise du système d'exploitation TinyOs et du langage NesC à cause de la documentation très succincte.

D'un point de vue personnel ce projet nous a permis de découvrir la programmation sur des systèmes embarqués, qui doit tenir compte aux ressources limitées, ainsi la conception d'un site web dynamique pour la visualisation des résultats obtenues qui nous a permis d'apprendre mieux manipuler les langages PHP, HTML et CSS et approfondir nos connaissances sur le langage SQL avec le MySQL.

Comme perspective de ce travail on propose de développer un kit de capteurs manipulable et facile à installer par des gens hors domaine qui contient un ensemble de capteurs et une carte Raspberry Pi accompagnée d'une clef 3G.

Nous envisageons tout cela dans nos futurs travaux de recherche qu'il soient d'ordre individuels, professionnels ou académiques.

# ***Bibliographie***

---

## BIBLIOGRAPHIE

- [1] « Les réseaux édition 2008 » Guy Pujolle **ISBN10** : 2-212-11757-4 , pp 505-507,pp478-481.
- [2] Site Web : <http://fr.wikipedia.org/wiki/Bluetooth> (Consulté le 03/03/2015).
- [3] Zigbee Standards Organizations Janvier 2008 « Zigbee Spécification » Documents 05347r17.
- [4] « WIFI Réseaux sans fil 802.11 : Technologie - Déploiement – Sécurisation » Philippe ATELIN.
- [5] « Réseaux d'entreprise par la pratique » Jean - LUC MONTAGNIER **ISBN** 2-212-11258-0, p 89.
- [6] Merzougui Rachid « CONCEPTION ET DÉVELOPPEMENT D'APPLICATIONS ET SERVICES DÉDIÉS À LA SANTÉ SUR DES TERMINAUX MOBILES » Thèse Doctorat Année 2011 UABTlemcen.
- [7] « LTE et les réseaux 4G » Yannick Bouguen ,Éric Hardouin, François-Xavier Wolff Groupe Eyrolles, 2012, **ISBN** : 978-2-212-12990-8.
- [8] A. RADU « Evaluation de Qualité de Service par l'utilisateur final dans les systèmes mobiles » Thèse Doctorat en informatique et Télécom de l'université de Mame-La-Vallée, France, Mars, 2004.
- [9] « Réseaux sans fil et mobiles » Khaldoun al Agha **ISBN** 2-7462-0974-8.
- [10] Messaoud BELLOULA « La géolocalisation dans les réseaux de capteurs sans fils » Mémoire Magister Université Hadj Lakhdar Batna.
- [11] YAHIA TENE Youcef « Traffic encryption keys distribution models in mobile adhoc networks » Mémoire Magister Université Mohammed Bougara Boumerdes.
- [12] S. Sentilles « Architecture logicielle pour capteurs sans fils en réseaux » Rapport de Recherche, Université de Pau et des pays de l'Adour, Juin 2006.

- [13] YACINNE Younes « minimisation d'énergie dans un réseau de capteurs » Mémoire Magistère 2012 Université Mouloud Mammeri de Tizi Ouzou.
- [14] Badaoui Abdellatif & Mostefaoui Med Amine « Acquisition de données à distance dans les réseaux de capteurs sans fil » Mémoire Master 2013 Université de Tlemcen.
- [15] Rahim KACIMI « Techniques de conservation d'énergie pour les réseaux de capteurs sans fil » Thèse Doctorat Institut National Polytechnique de Toulouse.
- [16] akyildiz ,I.F. Su, Y. Sankarasubramaniam, E. Cavirci « A survey on Sensor Networks » IEEE communications Magazine, august 102-114 (2002).
- [17] LABRAOUI Nabila « LA SÉCURITÉ DANS LES RÉSEAUX SANS FIL AD HOC » Thèse Doctorat 2012 Université de Tlemcen.
- [18] Sofiane MOAD « La consommation d'énergie dans les réseaux de capteurs sans fil » Master de recherche 2 informatique Année 2007 IFSIC -Rennes 1 .
- [19] I. Akyildiz, W. Su, E. Cayirci, Y. Sankarasubramaniam. « A survey on sensor networks », IEEE Communications Magazine, vol.40, no. 8, pp. 102-114, Georgia Institute of Technology, Atlanta, USA. Aout 2002.
- [20] B. Khalifa. « La sécurité dans les réseaux de capteurs sans fil », Conférence à l'université de Bechar. Printemps 2006.
- [21] Kazem Sohraby, Daniel Minoli, Taieb Znati «Wireless Sensor Networks : Technology, Protocols, and Application » ISBN 978-0-471-74300-2 , pp 20-22.
- [22] Maarouf Samia, Ouadah Souhila « Implémentation et évaluation des schémas de routage sur une plateforme réelle de réseaux de capteurs sans fil » Thèse Master Année 2014 Université de Tlemcen.
- [23] Shuang Hua Yang « Wireless Sensor Networks: Principles, Design and Applications » ISBN 978-1-4471-5504-1.
- [24] Systèmes embarqués « réseaux de capteurs et informatique ubiquitaire » ENSET - Master 2 2013.

[25] Houda Labiod « Réseaux mobiles ad hoc et réseaux de capteurs sans fil » ISBN 978-2746212923, pp 33-60.

[26] Kamal BEYDOUN « CONCEPTION D'UN PROTOCOLE DE ROUTAGE HIERARCHIQUE POUR LES RESEAUX DE CAPTEURS » Thèse Doctorat 2009 L'U.F.R des Sciences et Techniques de L'Université de Franche-Comte.

[27] Yacine CHALLAL « Réseaux de capteurs sans fils Version 1 » <http://moodle.utc.fr/file.php/498/support-SIT60.pdf> (Consulté le 25/03/2015).

[28] Ihsen Mallouli « Implémentation d'une bibliothèque de fonctions cryptographiques optimisées pour les réseaux de capteurs sans fil » Thèse Ingénieur 2011 L'Ecole Nationale d'Ingénieurs de Sfax.

[29] Hédi Jmed Maxime Caudron : Aurélien BRISSER - Pierre Marx GUITARD « RESEAU de capteurs sans fil » Projet avancé.

[30] Wassim ZNAIDI , « Modélisation formelle de réseaux de capteurs à partir de TinyOS » Projet Fin d'étude, école Polytechnique de Tunisie,2006.

[31] H. Alatrasta, S. Aliaga, K. Gouaich, J. Mathieu, « Implémentation de protocole sur une plateforme de réseaux de capteurs sans-fils», Thèse Master, Université de Montpellier 2, 2008.

[32] Site Web : <http://doc.ubuntu-fr.org/virtualBox> Consulté le (02/04/2015).

[24] HTML le guide complet 3em Edition ISBN : 978-2-300-019586, pp 12-19.

# ***Annexes***

---



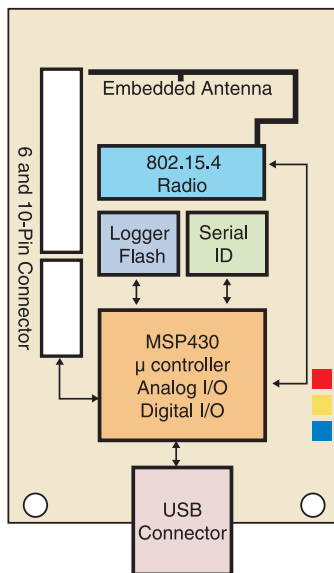
# TELOS B

## TELOS B MOTE PLATFORM

- IEEE 802.15.4 compliant
- 250 kbps, high data rate radio
- TI MSP430 microcontroller with 10kB RAM
- Integrated onboard antenna
- Data collection and programming via USB interface
- Open-source operating system
- Optional integrated temperature, light and humidity sensor

## Applications

- Platform for low power research development
- Wireless sensor network experimentation



TPR2400CA Block Diagram



## TELOS B

Crossbow's TelosB mote (TPR2400) is an open source platform designed to enable cutting-edge experimentation for the research community. The TPR2400 bundles all the essentials for lab studies into a single platform including: USB programming capability, an IEEE 802.15.4 radio with integrated antenna, a low-power MCU with extended memory and an optional sensor suite (TPR2420). TPR2400 offers many features, including:

- IEEE 802.15.4/ZigBee compliant RF transceiver
- 2.4 to 2.4835 GHz, a globally compatible ISM band
- 250 kbps data rate
- Integrated onboard antenna
- 8 MHz TI MSP430 microcontroller with 10kB RAM
- Low current consumption
- 1MB external flash for data logging
- Programming and data collection via USB
- Optional sensor suite including integrated light, temperature and humidity sensor (TPR2420)
- Runs TinyOS 1.1.10 or higher

The TelosB platform was developed and published to the research comm-

unity by UC Berkeley. This platform delivers low power consumption allowing for long battery life as well as fast wakeup from sleep state. Though the TPR2400 is an uncertified radio platform, it is fully compatible with the open-source TinyOS distribution.

TPR2400 is powered by two AA batteries. If the TPR2400 is plugged into the USB port for programming or communication, power is provided from the host computer. If the TPR2400 is always attached to the USB port no battery pack is needed.

TPR2400 provides users with the capability to interface with additional devices. The two expansion connectors and onboard jumpers may be configured to control analog sensors, digital peripherals and LCD displays.

TinyOS is a small, open-source, energy-efficient software operating system developed by UC Berkeley which supports large scale, self-configuring sensor networks. The source code software development tools are publicly available at:

<http://www.tinyos.net>

Specifications	TPR2400CA	TPR2420CA	Remarks
<b>Module</b>			
Processor Performance	16-bit RISC	16-bit RISC	
Program Flash Memory	48K bytes	48K bytes	
Measurement Serial Flash	1024K bytes	1024K bytes	
RAM	10K bytes	10K bytes	
Configuration EEPROM	16K bytes	16K bytes	
Serial Communications	UART	UART	0-3V transmission levels
Analog to Digital Converter	12 bit ADC	12 bit ADC	8 channels, 0-3V input
Digital to Analog Converter	12 bit DAC	12 bit DAC	2 ports
Other Interfaces	Digital I/O,I2C,SPI	Digital I/O,I2C,SPI	
Current Draw	1.8 mA	1.8 mA	Active mode
	5.1 $\mu$ A	5.1 $\mu$ A	Sleep mode
<b>RF Transceiver</b>			
Frequency band <sup>1</sup>	2400 MHz to 2483.5 MHz	2400 MHz to 2483.5 MHz	ISM band
Transmit (TX) data rate	250 kbps	250 kbps	
RF power	-24 dBm to 0 dBm	-24 dBm to 0 dBm	
Receive Sensitivity	-90 dBm (min), -94 dBm (typ)	-90 dBm (min), -94 dBm (typ)	
Adjacent channel rejection	47 dB	47 dB	+ 5 MHz channel spacing
	38 dB	38 dB	- 5 MHz channel spacing
Outdoor Range	75 m to 100 m	75 m to 100 m	Inverted-F antenna
Indoor Range	20 m to 30 m	20 m to 30 m	Inverted-F antenna
Current Draw	23 mA	23 mA	Receive mode
	21 $\mu$ A	21 $\mu$ A	Idle mode
	1 $\mu$ A	1 $\mu$ A	Sleep mode
<b>Sensors</b>			
Visible Light Sensor Range		320 nm to 730 nm	Hamamatsu S1087
Visible to IR Sensor Range		320 nm to 1100nm	Hamamatsu S1087-01
Humidity Sensor Range		0-100% RH	Sensirion SHT11
	Resolution		0.03% RH
	Accuracy		$\pm$ 3.5% RH
Temperature Sensor Range		-40°C to 123.8°C	Sensirion SHT11
	Resolution		0.01°C
	Accuracy		$\pm$ 0.5°C
<b>Electromechanical</b>			
Battery	2X AA batteries	2X AA batteries	Attached pack
User Interface	USB	USB	v1.1 or higher
Size (in)	2.55 x 1.24 x 0.24	2.55 x 1.24 x 0.24	Excluding battery pack
	(mm)	65 x 31 x 6	Excluding battery pack
Weight (oz)	0.8	0.8	Excluding batteries
	(grams)	23	Excluding batteries

**Notes**

<sup>1</sup>Programmable in 1 MHz steps, 5 MHz steps for compliance with IEEE 802.15.4/D18-2003.  
Specifications subject to change without notice

## Ordering Information

Model	Description
TPR2400CA	IEEE 802.15.4 TelosB Mote
TPR2420CA	IEEE 802.15.4 TelosB Mote with Sensor Suite



TPR2420 with Sensor Suite

Document Part Number: 6020-0094-01 Rev B

## Résumé

Les réseaux de capteurs sans fil connaissent dernièrement un très grand essor suite à leurs avantages telles que la facilité d'utilisation et le déploiement ainsi que leur coût minime ; pour cela plusieurs applications ont eu lieu dans différents domaines pour observer et de contrôler des phénomènes physiques telles que la température, la pression ou encore la luminosité.

En utilisant les réseaux de capteurs sans fil, nous avons développé une application de surveillance de température, humidité et CO2 dans l'environnement à base d'un réseau de capteurs sans fil via un centre de contrôle à travers le réseau Internet.

**Mots clés:** Réseau de capteurs sans fil (RCSF) , TinyOS , NesC.

## Abstract

Wireless sensor networks know a great development due to their facility of use and deployment as well as their costless given them a large field of application to control and observe physical phenomena such as the temperature, the pressure and the luminosity. For this purpose, the wireless sensor networks are technological progress made to control a geographical zone and to go back up an alarm in case of detection of a dreaded event.

Taking advantage of wireless sensor networks, we developed an application of temperature, humidity and CO2 surveillance in the environment with wireless sensor networks via control centers through Internet networks.

**Key words:** Wireless Sensor Network (WSN) , TinyOS, NesC.

## ملخص

في العقود الأخيرة شبكات الاستشعار اللاسلكية تطورا ملحوظا و ذلك للمحاسن التي تعرفها هذه الشبكات كسهولة الاستخدام و السعر المعقول و ذلك في مجالات مختلفة لمراقبة الظواهر الطبيعية و الفيزيائية مثل الحرارة والضغط والسطوع . وتحقيقا لهذه الغاية فان شبكات الاستشعار اللاسلكية هي تطور تكنولوجي هدفه رصد منطقة جغرافية والإنذار عند وقوع حدث خطير.

باستخدام شبكات الاستشعار اللاسلكية قمنا بإنشاء رصد ومراقبة درجة الحرارة ودرجة الرطوبة وكمية انبعاث غاز ثنائي أكسيد الكربون في مناطق معينة بالاعتماد على شبكة من أجهزة الاستشعار اللاسلكية من مركز التحكم وذلك مرورا بشبكة الانترنت.

**الكلمات الرئيسية:** شبكة الاستشعار اللاسلكي ، نظام التشغيل تيني أويس ، لغة البرمجة نيس سي