Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de Fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: *Réseaux et Système Distribués (R.S.D)*

Thème

Implémentation d'une application orientée surveillance pour les réseaux de capteurs

Réalisé par :

- Mr. SELATNA Abbes

Présenté le 4 Juillet 2012 devant le jury composé de MM.

- *Mme DIDI Fadoua* *(Président)*
- *Mr. LEHSAINI Mohammed* *(Encadreur)*
- *Mme LABRAOUI Nabila* *(Examinateur)*
- *Mr. BENMAMMAR Badr* *(Examinateur)*

Année universitaire : 2011-2012

Remerciement

Au terme de ce travail, je tiens à exprimer mes vifs remerciements :

A M Lehsaini pour l'encadrement technique et pour m'avoir guidé, encouragé et conseillé pendant toute la période de formation. Je tiens à mentionner le plaisir que j'ai eu à travailler avec lui.

A mes frères, mes oncles, tantes, cousins, cousines, grands parents. Et tous mes amis qui ont cru en moi, m'ont encouragé et m'on donné la force d'aller jusqu'au bout.

A tout la famille SELATNA, OULD ABBES, BELKHEROUFA, BENCHOHRA, BENAISSA, MAAMAR. Surtout la famille ZETTAM et KERZABI pour leur soutien durant ces cinq dernière années.

A tous mes collègues de promotion que j'ai 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.

A tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Enfin et surtout au ***DIEU TOUT-PUISSANT*** *qui m'a toujours soutenu.*

Dédicaces

Cette mémoire est dédiée

à mes ***chers parents****,*

qui m'ont toujours poussé et motivé dans mes études. Sans eux, je n'aurais certainement pas fait d'études supérieures. Cette mémoire représente donc l'aboutissement du soutien et des encouragements qu'ils m'ont prodigués tout au long de ma scolarité. Qu'ils en soient remerciés par cette trop modeste dédicace.

Table des matières

Liste des figures

Chapitre I : Les réseaux de capteurs

Chapitre II : Outils logiciels conçus pour RCSF

Chapitre III : Devellopement de l'application

Liste des tableaux

Chapitre I : Les réseaux de capteurs

Introduction générale

Les progrès technologiques dans les domaines de la microélectronique, des communications sans fil, couplé aux efforts de miniaturisation et de réduction des coûts de production des composants électroniques, ont permis le développement de nouvelles générations de petits appareils électroniques, autonomes, équipés de capteurs et capables de détecter, de calculer, de stocker et communiquer entre eux sans fil. Ces petits dispositifs sont appelés des nœuds capteurs ou « motes ». Ensemble, ils forment un réseau appelé réseau de capteurs sans fil (RCSF) qui est capable de superviser une région ou un phénomène dans une zone d'intérêt, de fournir des informations utiles par la combinaison des mesures prises par les différents capteurs et de les communiquées ensuite via le support sans fil à un point de collecte appelé "station de base ou puits" qui les communique aussi à un poste de contrôle distant.

Les capteurs sont dispersés aléatoirement dans une zone géographique, appelée champ de captage, qui définit le terrain d'intérêt pour le (s) phénomène (s) ciblé (s). Les données détectées sont acheminées grâce à un routage multi-saut à un nœud considéré comme un point de collecte, appelé station de base ou puits. Ce dernier peut être connecté à l'utilisateur du réseau via Internet ou un satellite. Ainsi, l'usager peut adresser des requêtes aux autres nœuds du réseau, en précisant le type de données requises et récolter les données captées telles que les données environnementales ou physiologiques par le biais de la station de base.

L'objectif de ce travail est de réaliser une application qui assure la surveillance d'une zone d'intérêt ou des points cibles dans cette zone. Par exemple, la surveillance des personnes dépendantes (âgées ou patients) à distance ou le mode de vie de quelques espèces en particulier les espèces rares. Les informations collectées par cette application peuvent être exploitées ou stockées dans une base de données pour faire des statistiques.

Ce manuscrit est organisé en trois chapitres. Dans le premier chapitre, on présente une introduction au domaine des réseaux de capteurs sans fil. On commence d'abord par décrire un nœud capteur et ses caractéristiques, ensuite on expose les réseaux de capteurs sans fil et leurs architectures. Le deuxième chapitre est une présentation des outils matériels et logiciels nécessaires à la réalisation d'une application orientée monitoring. Le troisième chapitre

constitue le cœur de notre travail. Dans ce chapitre, on présente une architecture complète permettant de réaliser une application orientée surveillance.

Enfin, on conclue notre travail en présentant les résultats obtenus et en donnant quelques perspectives.

CHAPITRE I :

Les réseaux de capteurs

Chapitre I
Les réseaux de capteurs

I.1 Introduction

Les réseaux de capteurs vont permettre de changer notre façon de vivre, de travailler et d'interagir avec l'environnement physique qui nous entoure en envahissant une gamme d'applications très variée. Ils sont composés de capteurs qui communiquent en sans fil et qui sont dotés de capacité de calcul permettant de faciliter une série d'applications irréalisables ou trop chères il y a quelques années. Aujourd'hui, des capteurs minuscules et bon marché peuvent être littéralement éparpillés sur des routes, des structures, des murs ou des machines, capables de détecter une variété de phénomènes physiques. De nombreux domaines d'application sont alors envisagés tels que la détection et la surveillance des désastres, le contrôle de l'environnement et la cartographie de la biodiversité, le bâtiment intelligent, l'agriculture de précision, la surveillance et la maintenance préventive des machines, la médecine et la santé, la logistique et les transports intelligents, etc..

L'étude et le développement de réseaux de capteurs est un sujet vaste et enrichissant. En effet, cette technologie est en pleine croissance étant donné le nombre de domaines d'utilisations qu'elle peut couvrir.

Dans ce chapitre, on présente les capteurs et leurs caractéristiques. Puis, on présente les réseaux de capteurs, leurs architectures, les contraintes sous-jacentes à ces réseaux et leurs domaines d'applications.

I.2 Présentation des capteurs

Dans cette section, on présente les capteurs, leurs caractéristiques et leurs différents modèles.

I.2.1 Modèle d'un instrument de mesure

Dans son ouvrage sur l'instrumentation industrielle, l'électronicien Georges Asch [1], a modélisé très précisément la notion d'instrument de mesure, et donc celle de capteur comme montre la Figure I.1. La grandeur physique objet de la mesure, nommée mesurande (m), est

appréhendée par diverses opérations expérimentales, que l'on regroupe sous le terme de mesurage, qui dans un grand nombre de cas produit un signal électrique (s) image de la grandeur physique et de ses variations. Le capteur est le dispositif physique qui, soumis à l'action du mesurande, non électrique, produit la caractéristique électrique: s=F(m). Or toutes les lois physiques interagissent au sein des matériaux, donc le capteur est obligatoirement sensible à d'autres grandeurs physiques, secondaires, dites grandeurs d'influence [2]. La caractéristique devient alors, en tenant compte des grandeurs d'influence g_1, g_2,… : s=F(m, g_1, g_2, ...). Les principales grandeurs d'influence sont la température, l'accélération, les vibrations, l'humidité, les champs magnétiques, etc. [2].

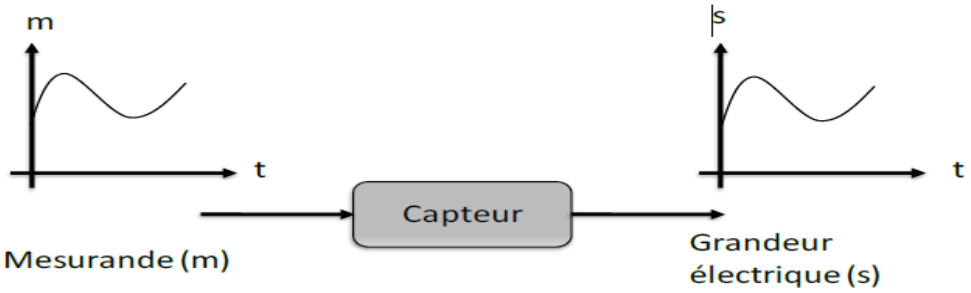

Figure I-1 : Fonctionnement d'un capteur [2]

Donc, un capteur est un dispositif équipé de fonctionnalités de sensation avancées. Il mesure ou détecte un événement réel, comme le mouvement, la chaleur ou la lumière et convertit la valeur mesurée dans une représentation analogique ou numérique. Il prélève des informations et élabore à partir d'une grandeur physique (information d'entrée), une autre grandeur physique de nature électrique.

I.2.2 Le capteur intelligent

Les capteurs intelligents "Smart Sensors" sont des dispositifs matériels dans lesquels coexistent le(s) capteur(s) et les circuits de traitement et de communication. Leurs relations avec des couches de traitement supérieures vont bien au-delà d'une simple "transduction de signal".

Les capteurs intelligents sont des "capteurs d'informations" et non pas simplement des capteurs et des circuits de traitement du signal juxtaposés. De plus, ces capteurs ne sont pas des dispositifs banalisés car chacun de leurs constituants a été conçu dans l'objectif d'une application bien spécifique [2].

I.2.3 Architecture d'un capteur intelligent

Un capteur intelligent est composé de quatre unités comme montre la Figure I.2:

- **L'unité d'acquisition** : Composée d'un ou plusieurs capteurs. Elle permet d'acquérir des mesures sur les paramètres environnementaux et les convertir en des signaux numériques à partir de signaux analogiques par le biais d'un convertisseur ADC[1] puis les transmettre à l'unité de traitement.
- **L'unité de traitement** : Composée d'un processeur et d'une mémoire intégrant un système d'exploitation spécifique comme TinyOS [3]. Cette unité possède deux interfaces, une interface avec l'unité d'acquisition et une interface avec l'unité de communication. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de communication. Cette unité est chargée aussi d'exécuter les protocoles de communications qui permettent de faire collaborer un capteur avec d'autres capteurs. Elle peut aussi analyser les données captées.
- **L'unité de communication** : Unité responsable de toutes les émissions et réceptions de données via un support de communication radio. Elle peut être de type optique comme dans les capteurs Smart Dust [4], ou de type radiofréquence comme dans les capteurs MICA2 [5].
- **Unité d'alimentation** : Un capteur est muni d'une batterie ou de piles pour alimenter tous ses composants. Cependant, à cause de sa taille réduite, la batterie dont il dispose est limitée et généralement irremplaçable. Pour cela, l'énergie est la ressource la plus précieuse puisqu'elle influe directement sur la durée de vie des capteurs.

Il existe des capteurs qui sont dotés d'autres composants additionnels comme le système de positionnement GPS (Global Positioning System) et un mobilisateur lui permettant leur déplacement. Dans le reste de notre rapport, lorsque nous parlerons de capteur nous sous-entendrons capteur "intelligent" avec un système de détection et les circuits de traitement et de communication.

[1] ADC : Analog-to-Digital Converter

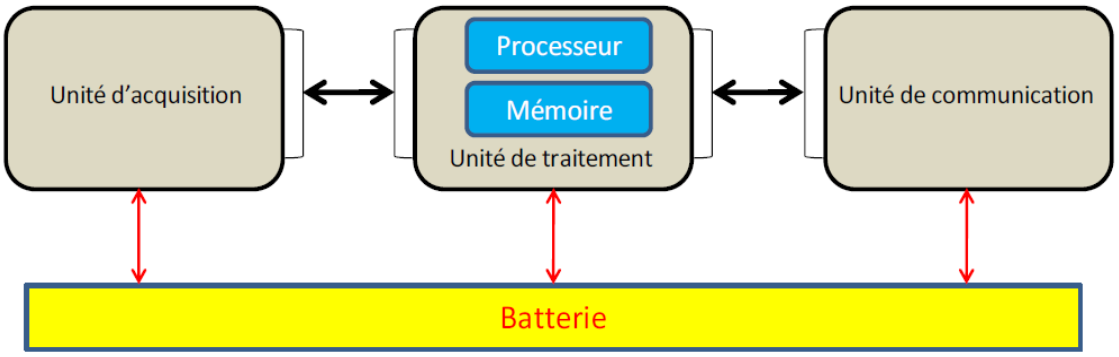

Figure I-2: Architecture d'un capteur

I.2.4 Caractéristiques principales d'un capteur

Deux entités sont fondamentales dans le fonctionnement d'un capteur: l'unité d'acquisition qui est le cœur physique permettant la prise de mesure et l'unité de communication qui réalise la transmission de celle-ci vers d'autres dispositifs électroniques. Ainsi, chaque capteur possède un rayon de communication (R_c) et un rayon de sensation (R_s). La Figure I.3 montre les zones définies par ces deux rayons pour le capteur A. La zone de communication est la zone où le capteur A peut communiquer avec les autres capteurs. Dans cet exemple, le capteur A peut communiquer avec un capteur B. D'autre part, la zone de détection est la zone où le capteur A peut capter l'événement [6,7]. Dans cet exemple, il s'agit de la zone dans laquelle se trouve le capteur C.

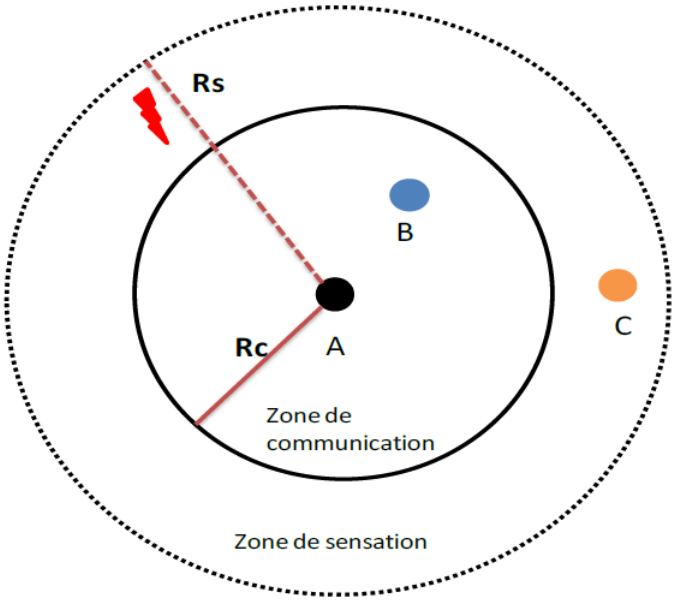

Figure I-3 : Rayons de communication et de détection d'un capteur

Par ailleurs, pour qu'un capteur ait une portée de communication suffisamment grande, il est nécessaire d'utiliser un signal assez puissant. Cependant, l'énergie consommée serait importante [8].

Il existe dans le monde plusieurs fabricants de capteurs. On cite Crossbow, Cisco, Dalsa, EuroTherm, et Sens2B. Parmi ces capteurs fabriqués, il existe quelques uns qui sont capables de varier la puissance du signal émis afin d'élargir/réduire le rayon de communication et en conséquence la zone de communication. La Figure I.4 montre un capteur intelligent qui fait partie à la famille des Telos, il s'agit d'un capteur TelosB [9].

Figure I-4 : Exemple de capteur intelligent TelosB

I.2.5 Catégories de Capteurs

Les capteurs peuvent être classés selon plusieurs critères : par type d'applications, par taille, etc. On présente dans cette sous-section quelques plateformes de réseaux capteurs:

- **Plateforme de capteurs miniaturisés :** C'est une plateforme dédiée aux capteurs de taille très réduite (quelques mm^3) et de faible bande passante (<50Kbps). Un exemple très connu de ce genre de plateforme est Spec, Ce dernier a été conçu par l'université de Berkeley et il es considéré parmi les plus petits capteurs dans le monde (sa taille est de l'ordre de 2mmx2.5mm).
- **Plateforme de capteurs généraux :** développée pour capter et router des informations du monde ambiant. Quelques plateformes de cette famille ont été développées et la plus récente est basée sur la famille des Mica telle que MicaZ qui est un capteur de taille de l'ordre de $10cm^3$ avec des protocoles de communication IEEE 802.15.4. Aujourd'hui, MicaZ devient la référence dans les travaux de recherche dans le domaine des réseaux de capteurs.
- **Plateforme de capteurs à haute bande passante :** Cette plateforme a pour but de transporter de gros volumes de données captées (la vidéo, le son, vibrations). Un

exemple typique de cette famille est Imote dont la communication se base sur la norme Bluetooth 1.1.

- **Plateforme de passerelles :** les dispositifs de cette plateforme servent à transporter les informations envoyées par le réseau de capteurs vers un réseau traditionnel (Ethernet, 802.11) dont Stargate est un exemple typique.

I.2.6 Types de capteurs

Il existe actuellement un grand nombre de capteurs, avec des fonctionnalités diverses et variées. La plupart de ces capteurs dépendent de l'application pour lesquels ils ont été conçus (capteurs aquatiques, sous-terrain, etc...).

La Figure I.5 illustre l'évolution des capteurs au cours de ces vingt dernières années. Cette représentation met en avant l'importance des travaux de recherche de l'université de Berkeley dans l'essor des réseaux de capteurs.

Figure I-5: Evolution des capteurs [10]

Les capteurs fabriqués par Crossbow au cours des dix dernières années (famille de capteurs Mica et Telos) sont les plus utilisés dans les expériences et les travaux de recherche. Ces capteurs sont capables de mesurer plusieurs métriques (température, humidité, luminosité, etc. . .) et la plupart d'entre eux s'articulent autour du Chipcon.

Nom Capteur	MCU	RAM	Flash	Stockage	Radio	Dimension
Spec Node (2003)	AVR Risk 8 bit	KB	KB	KB	RF	2 x 2.5 mm
Mica (2002)	ATMega 128	KB	KB	KB	CC1000	58 x 32 x 7 Mm
Mica2Dot (2002)	ATMega 128	KB	KB	KB	CC1000	25 x 6 Mm
MicaZ (2004)	ATMega 128	KB	KB	KB	CC2420	58 x 32 x 7 Mm
TelosA (2004)	TI MSP 430	KB	KB	KB	CC2420	
TelosB (2004)	TI MSP 430	KB	KB	KB	CC2420	65 x 31 x 6 Mm
Tmote Sky (2004)	TI MSP 430	KB	KB	KB	CC2420	3.2 x 8 x 1.3 Mm
BTnode3 (2004)	ATMega 128	KB	KB	KB	CC1000	58 .15 x 33 Mm
Imote (2003)	ARM7	KB	KB	KB	ZV4002	
Imote2 (2007)	Intel PXA271	KB	KB	KB	CC2420	36 x 48 x 9 Mm
Iris (2008)	ATmega 1281	KB	KB	KB	CC2420	58 x 32 x 7 Mm

Tableau I-1: Caractéristiques des capteurs les plus utilisés [11]

Le tableau ci-dessus Tableau I.1 reprend les principales caractéristiques des capteurs de la société Xbow(Crossbow), ainsi que les capteurs les plus utilisés dans le domaine de la recherche.

I.3 Les réseaux de capteurs sans fil (RCSF)

Les réseaux de capteurs sans fil (RCSF) sont un type particulier de réseaux Ad-hoc, dans lesquels les nœuds sont des "capteurs intelligents". Ils se composent généralement d'un grand nombre de capteurs communicants entre eux via des liens radio pour le partage d'information et le traitement coopératif.

Dans ce type de réseau, les capteurs échangent des informations par exemple sur l'environnement pour construire une vue globale de la région contrôlée, qui est rendue accessible à l'utilisateur externe par un ou plusieurs nœud(s). Les données collectées par ces capteurs sont acheminées directement ou via les autres capteurs de proche en proche à un "point de collecte", appelé station de base (ou sink). Cette dernière peut être connectée à une machine puissante via internet ou par satellite. En outre, l'utilisateur peut adresser ses requêtes aux capteurs en précisant l'information d'intérêt.

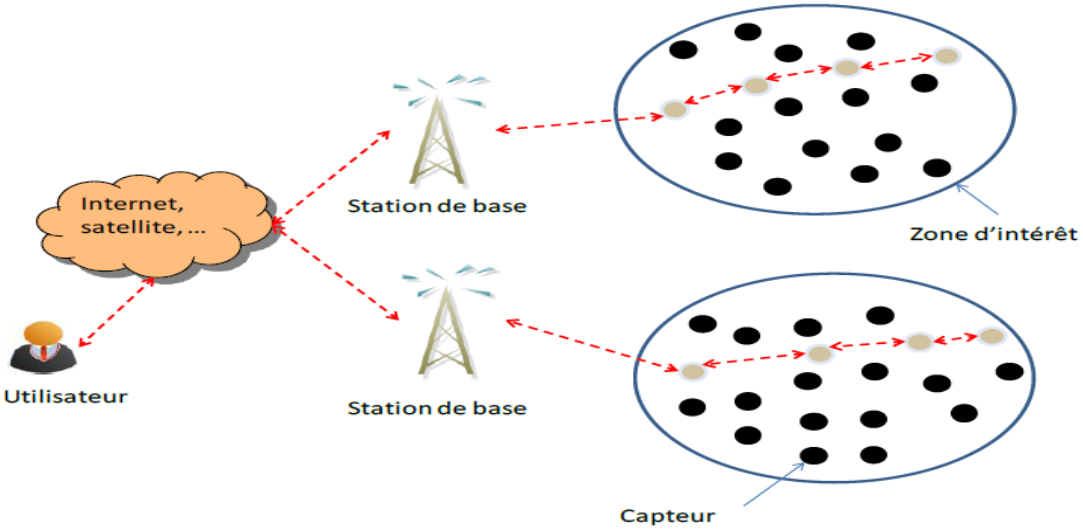

Figure I-6: Exemple de réseau de capteurs

Un exemple de réseaux de capteurs est fourni dans la Figure I-6 : les capteurs sont déployés d'une manière aléatoire dans une zone d'intérêt, et une station de base, située à l'extrémité de cette zone, est chargée de récupérer les données collectées par les capteurs. Lorsqu'un capteur détecte un événement pertinent, un message d'alerte est envoyé à la station de base par le biais d'une communication entre les capteurs. Les données collectées sont traitées et analysées par des machines puissantes.

Les réseaux de capteurs viennent en soutien de l'environnement, de la santé et de l'industrie grâce aux récents développements réalisés dans le domaine des techniques sans fil. Depuis quelques décennies, le besoin d'observer et de contrôler des phénomènes physiques tels que la température, la pression ou encore la luminosité est essentiel pour de nombreuses applications industrielles et scientifiques.

I.3.1 Domaines d'application des réseaux de capteurs

La miniaturisation des capteurs, le coût de plus en plus faible, la large gamme des types de capteurs disponibles ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs de se développer dans plusieurs domaines d'application. Ils permettent aussi d'étendre les applications existantes.

Les réseaux de capteurs peuvent se révéler très utiles dans de nombreuses applications lorsqu'il s'agit de collecter et de traiter des informations provenant de l'environnement. Parmi les domaines où ces réseaux peuvent offrir les meilleures contributions, on cite les domaines: militaire, surveillance, environnemental, médical, domestique, etc. [12].

Applications militaires

Le faible coût et le déploiement rapide sont des caractéristiques qui ont rendu les réseaux de capteurs efficaces pour les applications militaires. Plusieurs projets ont été lancés pour aider les unités militaires dans un champ de bataille et protéger les villes contre des attaques, telles que les menaces terroristes. Le projet DSN (Distributed Sensor Network) [13] au DARPA (Defense Advanced Research Projects Agency) était l'un des premiers projets dans les années 80 ayant utilisé les réseaux de capteurs pour rassembler des données distribuées. Les chercheurs du laboratoire national Lawrence Livermore ont mis en place le réseau WATS (Wide Area Tracking System) [14]. Ce réseau est composé de détecteurs des rayons gamma et des neutrons pour détecter et dépister les dispositifs nucléaires. Il est capable d'effectuer la surveillance constante d'une zone d'intérêt. Il utilise des techniques d'agrégation de données pour les rapporter à un centre intelligent. Ces chercheurs ont mis en place ensuite un autre réseau appelé JBREWS (Joint Biological Remote Early Warning System) [15] pour avertir les troupes dans le champ de bataille des attaques biologiques possibles. Un réseau de capteurs peut être déployé dans un endroit stratégique ou hostile, afin de surveiller les mouvements des forces ennemies, ou analyser le terrain avant d'y envoyer des troupes (détection des armes chimiques, biologiques ou radiations).

Applications à la surveillance

L'application des réseaux de capteurs dans le domaine de la sécurité peut diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et des êtres humains. Ainsi, l'intégration des capteurs dans de grandes structures telles que les ponts ou les bâtiments aidera à détecter les fissures et les altérations dans la structure suite à un séisme ou au vieillissement de la structure. Le déploiement d'un réseau de capteurs de détection de mouvement peut constituer un système d'alarme qui servira à détecter les intrusions dans une zone de surveillance.

Applications environnementales

Le contrôle des paramètres environnementaux par les réseaux de capteurs peut donner naissance à plusieurs applications. Par exemple, le déploiement des thermo-capteurs dans une forêt peut aider à détecter un éventuel début de feu et par suite faciliter la lutte contre les feux de forêt avant leur propagation. Le déploiement des capteurs chimiques dans les milieux urbains peut aider à détecter la pollution et analyser la qualité d'air. De même leur

déploiement dans les sites industriels empêche les risques industriels tels que la fuite de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.).

Dans le domaine de l'agriculture, les capteurs peuvent être utilisés pour réagir convenablement aux changements climatiques par exemple le processus d'irrigation lors de la détection de zones sèches dans un champ agricole. Cette expérimentation a été réalisée par Intel Research Laboratory and Agriculture and Agri-Food Canada sur une vigne à British Columbia.

Applications médicales

Dans le domaine de la médecine, les réseaux de capteurs peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humain grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau (surveillance de la glycémie, détection de cancers à l'étape précoce, etc.). Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que: la tension artérielle, battements du cœur, etc. à l'aide des capteurs ayant chacun une tâche bien particulière. Les données physiologiques collectées par les capteurs peuvent être stockées pendant une longue durée pour le suivi d'un patient [16]. D'autre part, ces réseaux peuvent détecter des comportements anormaux (chute d'un lit, choc, cri, etc.) chez les personnes dépendantes (handicapées ou âgées).

Applications domestiques

Avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs, les magnétoscopes, etc. [17]. Ces capteurs embarqués peuvent interagir entre eux et avec un réseau externe via Internet pour permettre à un utilisateur de contrôler les appareils domestiques localement ou à distance. Le déploiement des capteurs de mouvement et de température dans les futures maisons dites intelligentes permet d'automatiser plusieurs opérations domestiques telles que : la lumière s'éteint et la musique s'arrête quand la chambre est vide, la climatisation et le chauffage s'ajustent selon les points multiples de mesure, l'alarme est déclenchée par le capteur anti-intrusion quand un étranger veut pénétrer dans la maison.

Applications commerciales

Il est possible d'intégrer des capteurs au processus de stockage et de livraison dans le domaine commercial. Le réseau ainsi formé pourra être utilisé pour connaître la position, l'état et la direction d'un paquet. Il devient alors possible pour un client qui attend la réception d'un paquet, d'avoir un avis de livraison en temps réel et de connaître la localisation actuelle du paquet.

Pour les entreprises manufacturières, les réseaux de capteurs permettront de suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux réseaux de capteurs, les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts [18].

I.3.2 Caractéristiques des réseaux de capteurs sans fil

Un réseau de capteurs présente les caractéristiques suivantes :

- Absence d'infrastructure : les réseaux Ad-hoc en général, et les réseaux de capteurs en particulier se distinguent des autres réseaux par la propriété d'absence d'infrastructure préexistante et de tout genre d'administration centralisée.
- Taille importante : un réseau de capteurs peut contenir des milliers de nœuds capteurs.
- Interférences : les liens radio ne sont pas isolés, deux transmissions simultanées sur une même fréquence, ou utilisant des fréquences proches, peuvent interférer.
- 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.
- 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 que le contrôle des données transférées doit être minimisé.
- Bande passante limitée : une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé. Ce partage fait que la bande passante réservée à un nœud est limitée.
- Contrainte d'énergie, de stockage et de calcul : la caractéristique la plus critique dans les réseaux de capteurs est la modestie de ses ressources énergétiques car chaque capteur du réseau possède de faibles ressources en termes d'énergie, de calcul et de stockage. Afin de prolonger la durée de vie du réseau, une minimisation des dépenses énergétiques est exigée chez chaque nœud.

I.3.3 Ordonnancement d'activité des capteurs

Les RCSF sont généralement denses et redondants. En effet, suivant l'application, on déploiera plus ou moins des capteurs dans un souci d'allongement de la durée de vie de l'application. À tout moment, il existe donc des capteurs qui observent une même portion de la zone de déploiement. Cette redondance est surpassée par l'ordonnancement d'activité où seulement un certain nombre de nœuds est en mode actifs et les autres passent au mode veille. Ceci est dans le but d'alterner les charges de façon à épuiser les nœuds équitablement et par conséquent prolonger la longévité du réseau.

I.3.4 Spécificités des RCSF

- **Topologie :** la topologie que l'on retrouve de manière classique au sein des RCSF est représentée par la Figure I-7 où les nœuds sont déployés aléatoirement dans une zone d'intérêt. Chaque nœud peut communiquer directement avec les autres nœuds qui sont situés dans sa zone de couverture et un nœud qui souhaite communiquer avec un nœud distant doit faire transiter son message à travers des nœuds relais pour l'atteindre. On parle dans ce cas de figure de communication multi-saut ou multi-hop

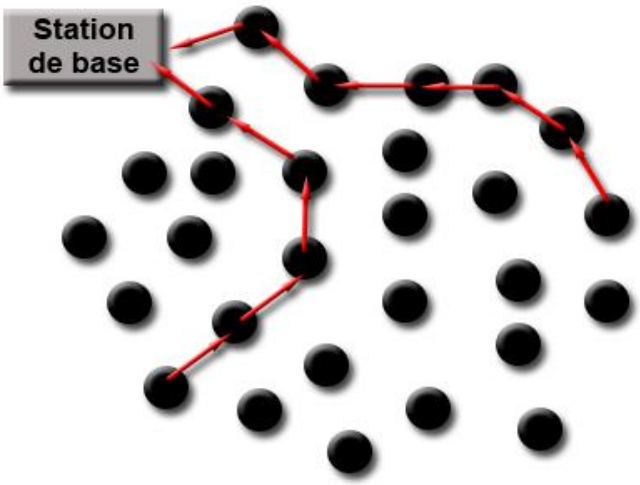

Figure I-7: Topologie classique des RCSF

Par ailleurs, les RCSF possèdent le plus souvent une, voire plusieurs stations de base. Ces stations de base ont pour mission de collecter les informations circulant sur le réseau, de les stocker ou de les transmettre directement ou périodiquement via une liaison Internet ou une liaison GSM à une autre entité. Les stations de base peuvent être un ordinateur portable ou un capteur de puissance plus grande que les autres nœuds classiques. Elles ont

le plus souvent un rôle de contrôleur du réseau et servent d'intermédiaire entre les utilisateurs du réseau et le réseau lui même.

Il existe d'autres topologies dans lesquelles les stations de base peuvent être mobiles, comme par exemple une station de base fixée à un avion ou un drone, afin de survoler la zone de couverture du réseau pour récupérer les informations des capteurs. De la même manière, les nœuds capteurs peuvent être mobiles comme dans le cas où les capteurs seraient portés par des individus ou des animaux.

- **Medium :** Le médium utilisé par les RCSF est l'onde radio. Trois grandes technologies radios ont été utilisées pour des applications à bases de réseaux de capteurs. Ces trois technologies sont :
  - IEEE 802.11x/WiFi : le protocole de communication WiFi est très certainement le protocole le plus utilisé pour toutes les applications sans fil. Il offre une large bande passante (11 à 320 Mbits/secondes) qui a permit de généraliser l'utilisation de la technologie sans fil dans les réseaux classiques WLANs[2]. Les premiers capteurs ont eu recours à ce protocole pour permettre la communication entre nœuds. Or, cette technologie exige une consommation d'énergie trop importante. Par conséquent, la durée de vie de capteurs alimentés par des piles ne dépasse que rarement quelques heures. C'est pourquoi les applications de capteurs à base de communication sans fil WiFi ont été négligées.
  - IEEE 802.15.1/Bluetooth : le standard de communication Bluetooth avait pour objectif préalable de permettre des communications pour les WPANs[3], c'est-à-dire des communications sur de courtes distances avec un débit de communication limitée (1Mbits/seconde) et une durée de vie prolongée. Ses caractéristiques ont ainsi retenu l'attention des développeurs de capteurs. Pour exemple les capteurs BtNode sont conçus pour une communication de type Bluetooth. Pour autant, le protocole Bluetooth n'est pas le protocole le plus utilisé dans les réseaux de capteurs, bien qu'il puisse répondre en partie aux problèmes de préservation de l'énergie, car il est gravement handicapé par la taille limitée du réseau qu'il peut former 80 nœuds selon une architecture Master/Slave. Ce faible nombre de nœuds est incompatible avec la volonté de former des réseaux denses de capteurs.
  - IEEE 802.15.4/Zigbee : cette norme a été développée par la communauté industrielle Zigbee Alliance, dans le but commun de définir un protocole de

[2] Wireless Local Area Networks
[3] Wireless Personal Area Networks

communication avec un faible coût énergétique et un faible coût de production des composants nécessaires pour son utilisation. Dans cette optique le protocole Zigbee est basé sur la standard IEEE 802.15.4 qui définit sa couche PHY et MAC et qui permet de prolonger théoriquement la durée de vie d'un nœud sur plusieurs années. L'autre point fort de ce protocole est qu'il propose le déploiement de réseau dense à plus de 65000 nœuds avec une portée de l'ordre de 100 mètres pour un débit de 250 Kb/s. Ces caractéristiques font aujourd'hui ZigBee le principal protocole utilisé dans les réseaux de capteurs.

- **Routage des données :** Pour limiter le nombre de communications coûteuses en énergie, les RCSF requièrent des protocoles de routage efficaces [19]. Une des solutions employées par les protocoles de routage est la clustérisation, qui divise le réseau en plusieurs clusters. Dans chacun de ces clusters, un nœud maître (cluster-head) est élu et aura pour mission de récupérer les informations des membres du cluster dont il a la charge pour les transmettre aux autres clusters et inversement. Le choix du nœud maître dans un protocole de routage est désigné selon certains critères tels que l'énergie, pour augmenter la durée de vie du réseau. La Figure I-8 représente un exemple de réseau clustérisé où les nœuds A, B et C ont été respectivement élus cluster-heads des clusters 1, 2 et 3.

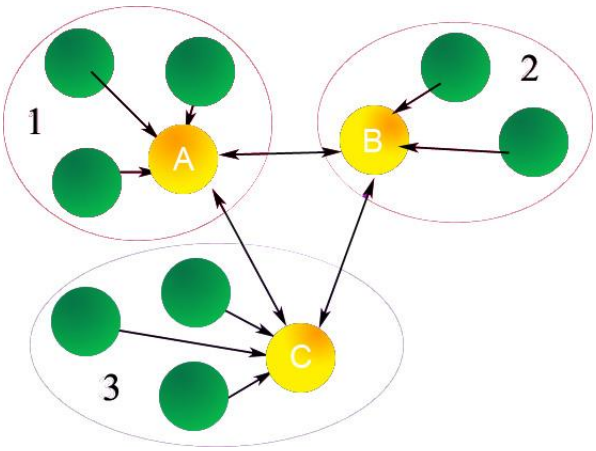

Figure I-8 : Clustérisation d'un RCSF

D'autres problèmes de routage doivent aussi être pris en compte pour limiter le nombre de communications comme les problèmes d'implosion ou de chevauchement.

Le problème d'implosion est représenté par la Figure I-9, où un nœud A va envoyer l'information à deux de ses voisins B et C, qui sans un protocole efficace vont envoyer tous deux l'information à leur nœud voisin D. Cette action aura pour effet une redondance de

l'information, une consommation double de l'énergie du réseau, voir dans un cas critique, une collision lors de l'envoi simultané des deux capteurs.

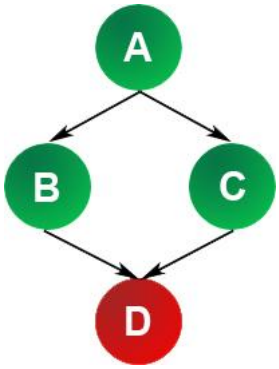

Figure I-9 : Phénomène d'implosion

Le problème du chevauchement est représenté par la Figure I-10, où deux capteurs A et B qui surveillent leur zone respective 1 et 2, et partage une zone commune 3, détectent en même temps un même événement en zone 3 et qui, sans un protocole efficace, vont envoyer chacun la même information au nœud C, d'où une redondance de l'information, une surconsommation du réseau et un risque de collision.

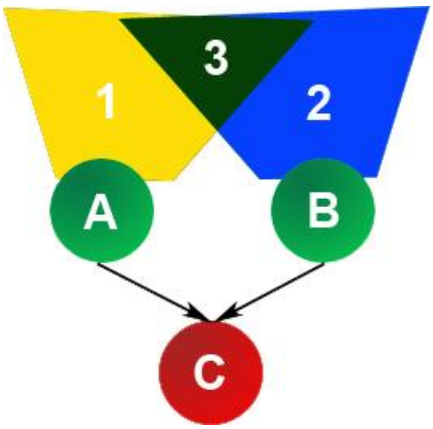

Figure I-10 : Phénomène de chevauchement

- **Tolérance aux fautes :** Dans les RCSF, un ou plusieurs capteurs peuvent ne pas fonctionner correctement, car les capteurs sont des entités sensibles aux altérations d'états comme des phénomènes climatiques (humidité, chaleur, électromagnétisme) ou du fait d'une batterie faible. Dans ces cas, le réseau doit être capable de détecter ce type d'erreur est d'y remédier, en cherchant par exemple à modifier ses tables de routages pour trouver un autre chemin afin de transmettre l'information et permettre que le réseau soit toujours opérationnel. De la même manière on doit pouvoir trouver le moyen pour détecter les capteurs défaillants qui envoient des informations erronées.

- **Mise à l'échelle :** Le nombre de capteurs utilisés dans les RCSF peut varier de quelques entités à plusieurs dizaines de milliers. C'est d'ailleurs la principale utilité des réseaux de capteurs qui doivent pouvoir s'auto-organiser à une grande échelle et être efficace quelque soit la taille du réseau. Pour cela les protocoles des RCSF doivent être capables de fonctionner et de s'adapter selon le nombre de nœuds.
- **Energie :** Les capteurs sont équipés de batteries, comme par exemple des piles LR6 dans le cas des MicaZ ou des TelosB. L'énergie de ces batteries est limitée (plusieurs jours à quelques années). De plus, les RCSF quand ils sont déployés, le sont souvent dans des zones à accès difficiles. Il devient alors inenvisageable de vouloir changer les batteries des capteurs. Si le nombre des capteurs dépasse la centaine d'entités, il est encore plus difficile d'intervenir pour trouver le capteur défaillant et changer sa batterie. La consommation de l'énergie des RCSF doit être la plus faible possible. Dans ce but, les capteurs actuels ont des périodes de veille durant leur inactivité pour préserver leur batterie. Enfin les communications sont les coûteuses en termes de consommation d'énergie. Il est donc fortement nécessaire de limiter le nombre de communications entre capteurs et si possible la quantité de calculs.
- **Puissance de calcul :** Malgré les progrès récents dans la fabrication de capteurs de plus en plus puissants, les capteurs actuels souffrent d'un manque de puissance de calcul (par exemple seulement 16 Mhz de puissance et 128Koctets de mémoire programmable pour un capteur MicaZ). Cette faible puissance ne permet pas d'utiliser des algorithmes complexes, et particulièrement des algorithmes cryptographiques gourmands en ressources CPU. De plus, la vocation des capteurs sans fil est d'être en très grand nombre et leur utilisation dans des applications avec un nombre de nœuds élevé nécessite l'utilisation de capteurs bons marchés, ce qui implique des capteurs avec une puissance de calcul très faible. La faiblesse de puissance de calcul est aussi préjudiciable pour le temps de réponse du réseau. Si l'on demande à un capteur d'effectuer de nombreux calculs, la latence va sensiblement augmenter.
- **Sécurité :** Comme évoqué précédemment les RCSF nécessitent dans de nombreuses applications des solutions qui assurent la sécurité des informations circulant sur le réseau. La sécurité des informations transitant dans les RCSF doivent répondre à plusieurs pré-requis.

I.4 Conclusion

Dans ce chapitre, on a présenté les capteurs, leurs fonctionnalités et leurs architectures. Puis, on a présenté les RCSF et leurs spécificités pour bien engendrer le problème traité dans ce projet qui consiste à développer une application orienté surveillance dans laquelle il y a l'envoi de l'information périodiquement à un poste de contrôle distant.

Dans le chapitre qui suit, on présente les outils logiciels qui permettent de développer cette application. Ces outils sont conçus spécialement aux équipements ayant des ressources limitées tels que les réseaux de capteurs.

CHAPITRE II :

Outils logiciels conçus pour RCSF

Chapitre II
Outils logiciels conçus pour RCSF

II.1 Introduction

Dans ce travail, on va réaliser une application de surveillance pour les réseaux de capteurs. Or, l'implémentation de cette application nécessite des outils logiciels bien spécifiques puisque les capteurs sont considérés comme des dispositifs à ressources limitées.

Ces outils logiciels concernent spécialement le système d'exploitation utilisé et le langage de programmation des capteurs. Dans notre contexte, on va exposer TinyOs qui est un système d'exploitation léger et qui est conçu aux réseaux de capteurs. Ce dernier est considéré parmi les systèmes d'exploitation les plus utilisés pour programmer les capteurs. En outre, pour programmer les capteurs, on va utiliser le langage NesC qui est un langage orienté composants conçus spécialement aux réseaux de capteurs. Enfin, on présente les capteurs qu'on a utilisés dans notre implémentation.

Il existe plusieurs systèmes d'exploitation pour des réseaux de capteurs tels que TinyOs, Contiki [20], SOS [21], etc. Cependant, TinyOs est considéré parmi les plus célèbres et le plus complet pour plusieurs types de capteurs à l'instar de la famille des Mica, la famille des Telos et la famille des Iris.

Dans ce chapitre, on va décrire en détails TinyOs et brièvement d'autres systèmes d'exploitation.

II.2 Système d'exploitation: TinyOs

TinyOS est un système d'exploitation Open-Source pour les réseaux des capteurs, développé par l'université de BERKELEY. Le caractère open-source permet à ce système d'être régulièrement enrichie par une multitude d'utilisateurs. Sa conception a été entièrement réalisée en NesC, langage orienté composants syntaxiquement proche du C.

TinyOs respecte une architecture basée sur une association de composants, réduisant ainsi la taille du code nécessaire à sa mise en place [26]. Cela s'inscrit dans le respect des

contraintes de mémoires qu'observent les capteurs, pourvus de ressources très limitées dues à leur miniaturisation.

Pour autant, la bibliothèque des composants de TinyOs est particulièrement complète, puisqu'on y retrouve des protocoles réseaux, des pilotes de capteurs, et des outils d'acquisition de données. Un programme s'exécutant sur TinyOs est constitué d'une sélection de composants systèmes et de composants développés spécifiquement pour l'application à laquelle il sera destiné (mesure de température, du taux d'humidité…).

TinyOs s'appuie sur un fonctionnement évènementiel, c'est à dire qu'il ne devient actif qu'à l'apparition de certains évènements, par exemple l'arrivée d'un message radio. Le reste du temps, le capteur se trouve en état de veille, garantissant une durée de vie maximale connaissant les faibles ressources énergétiques des capteurs. Ce type de fonctionnement permet une meilleure adaptation à la nature aléatoire de la communication sans fil entre capteurs.

II.2.1 Propriétés de TinyOs

TinyOs est basé sur quatre grandes propriétés qui permettent à ce système d'exploitation, de s'adapter particulièrement bien aux systèmes à ressources limitées tels les RCSF [27]:

i. **Evénementiel** : Le fonctionnement d'un système basé sur TinyOs s'appuie sur la gestion des évènements qui ce déclenche. Ainsi, l'activation de tâches, 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 (Event/Driven) s'oppose au fonctionnement dit temporel (Time/Driven), où les actions du système sont gérées par une horloge donnée.

ii. **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.

iii. **Pas de 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.

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

II.2.2 Concepts du système TinyOs

Le système TinyOs est un exo noyau qui utilise une approche composants et non objet comme beaucoup d'autres systèmes [28]. On distingue la différence entre un exo noyau et les noyaux plus classiques et les différences entre la programmation par composants et la programmation orientée objet. Dans ce qui suit, on va exposer comment TinyOs est conçu ?

- **Différents types de noyaux**

Il existe plusieurs types de noyaux qui ont été créés selon des besoins différents.

- **Les noyaux monolithiques** : Dans un noyau monolithique que ce soit modulaire ou non, les pilotes matériels et les services sont intégrés dans un même bloc. Ce noyau permet une grande rapidité d'exécution car les services étant directement intégrés à lui ne génèrent pas des appels coûteux. Cependant, un des inconvénients de cette approche est que lorsqu'une erreur survient dans un des modules ou services du noyau elle menace la stabilité du système en entier. Dans ce type de noyau, les programmes de l'espace utilisateur font appel au noyau pour accéder au matériel. C'est le premier type de noyau développé et qui est encore utilisé par les systèmes Linux, BSD et Solaris.
- **Les micros noyaux** : Pour limiter le risque d'instabilité du système lors d'erreurs dans les services des noyaux, les micros noyaux séparent les services du noyau pour les placer dans l'espace utilisateur. Si une erreur survient dans un de ces services elle est limitée à ce service et ne compromet pas tout le système. Le noyau fournit uniquement les abstractions les plus basiques nécessaires. L'avantage de cette approche est d'améliorer en théorie la fiabilité du système et de permettre au programmeur de pouvoir mieux contrôler le matériel en n'étant pas obligé d'utiliser des abstractions de haut niveau comme dans un noyau monolithique. En contrepartie, la séparation des services engendrent de nombreux appels systèmes et les mécanismes de communication entre les services deviennent complexes et lourds en

temps de traitement, il en résulte un système aux performances diminuées. C'est le type de noyau utilisé par Mac OS X et Windows XP.

- Les exo noyaux: Le principe des exo noyaux et de laisser la liberté au programmeur de se passer d'abstractions matérielles. Le système fournit uniquement des interfaces donnant au programmeur un accès direct au matériel sans rajouter de fonctionnalités. Des niveaux d'abstractions plus hauts sont disponibles mais le programmeur n'est pas forcé de les utiliser et peut très bien les remplacer par les siens. Ces abstractions étant fournies dans des librairies extérieures, le programmeur est libre de les utiliser ou non et seules les librairies vraiment nécessaires sont utilisés, contrairement aux noyaux monolithiques.

Un noyau monolithique engendrerait un programme trop lourd puisqu'il intégrerait toutes les fonctionnalités du système. Un micro noyau serait plus léger en taille mais la gestion des services trop complexe et les nombreux appels système ne sont pas adaptés à des nœuds dont la puissance est limitée et la vitesse d'exécution essentielle. La possibilité de n'utiliser que les librairies nécessaires offerte par les exo noyaux permettent aux programmes d'être plus légers et permettent une grande modularité. L'accès direct au matériel est important puisqu'il permet au programmeur d'adapter le système à ses besoins en évitant de passer par des couches d'abstraction limitant les performances du programme. C'est pourquoi les exo noyaux sont les plus adaptés pour les nœuds sans fil.

II.2.3 Tâches, événements et applications

TinyOs est basé sur la gestion de tâches et d'événements. Une tâche est un bloc d'instruction, un événement est l'équivalent logiciel d'une interruption matérielle et a priorité sur les tâches. Chaque tâche est activée ou interrompue en fonction de l'apparition d'un évènement. Cependant, les tâches ne peuvent pas s'interrompre entre elles puisque TinyOs n'étant pas préemptif, mais elles peuvent l'être par un événement.

- Gestion des tâches : Chaque tâche activée est mise en attente dans une file d'attente de type FIFO (First In First Out : première arrivée première sortie), lorsque la file des tâches est vide le système se met en veille en attendant le prochain événement. Ce mécanisme de taches a pour avantage d'empêcher une tache d'en interrompre une autre, pouvant bloquer le système, mais il a aussi pour inconvénient de ne pas permettre une gestion en temps réel.

Pour les tâches de longue durée TinyOs possède un mécanisme permettant de fragmenter l'exécution d'une tâche nommée split-phase qui permet de ne pas bloquer le système. Ce mécanisme est utilisé dans l'initialisation de composants qui demandent du temps au démarrage, comme la radio par exemple.

- **Les événements**: Lorsqu'une interruption matérielle a lieu, l'événement correspondant reçoit un signal et prend la main de manière asynchrone, c'est à dire qu'il n'attend pas la fin de la tâche courante pour s'exécuter. Des événements peuvent être signalés ne correspondant pas directement à une interruption matérielle. Il existe également des événements synchronisés: ils sont mis en attente dans la liste des tâches, avec une priorité supérieure aux tâches en attente mais n'interrompent pas la tâche courante comme dans le cas de certains Timers.

- **Les applications**: Les applications basées sur TinyOs sont formées de composants réutilisables et portables (comme les Timers, les convertisseurs de signal ou la radio) qui sont reliés entre eux. Ces composants peuvent être directement liés au matériel (composant gérant les DEL par exemple) ou un regroupement de plusieurs composants de bas niveau (composants gérant les envois de données par la radio). Les composants sont implémentés en utilisant les tâches, les événements et des commandes qui permettent de faire appel aux fonctionnalités d'autres composants auxquels ils sont liés.

- **Exemple d'application**

Une application devant mesurer la température d'une pièce régulièrement et transmettre les données à un ordinateur utilisera plusieurs composants :

- un composant de mesure de température.
- un composant qui se chargera de l'envoi des données par le port USB.
- un composant de mesure du temps.
- un composant de gestion des DEL pour afficher la fréquence des mesures.
- …

II.2.4 L'abstraction matérielle

TinyOs utilise une architecture d'abstraction du matériel (HAA : Hardware Abstraction Architecture) qui permet de satisfaire les besoins de portabilité et de réutilisabilité des applications et des composants tout en préservant l'efficacité et les performances de chaque

plateforme. L'utilisation de ce type d'abstraction dans le développement de systèmes d'exploitation a permis de simplifier grandement le développement des applications en cachant le matériel aux développeurs, mais elle comporte aussi l'inconvénient de réduire les performances du système. C'est pourquoi il faut que le choix de cette architecture soit fait de manière optimale.

L'architecture d'abstraction du matériel de TinyOs est organisée en trois couches distinctes de composants. Chaque couche a un but bien précis et est dépendante des interfaces fournies par les couches inferieures. Plus on se trouve haut dans les couches, moins le code utilisé dépend du matériel pour finalement arriver à des applications complètement indépendantes de la plateforme sur laquelle ils tournent, permettant leur portabilité.

- ***Couche HPL (Hardware Presentation Layer)*** : Cette couche est la couche de niveau le plus bas. Elle se situe juste à l'interface entre le matériel et le logiciel. Son but est de présenter les possibilités du matériel en mettant à disposition de la couche supérieure un ensemble de méthodes qui permettent de contrôler le matériel. Chaque composant de cette couche doit fournir au minimum les méthodes suivantes :
  - Des commandes d'initialisation, de démarrage et d'arrêt du composant matériel pour assurer une gestion optimale de l'énergie
  - Des méthodes d'accès aux registres du matériel (lecture, écriture)
  - Des méthodes permettant de modifier simplement les flags les plus utilisés (sans avoir à modifier tout le registre qui les contient)
  - Des commandes permettant d'activer ou de désactiver les différentes interruptions générées par le matériel
  - La gestion des interruptions matérielles pour les transmettre à la couche supérieure
- ***Couche HAL (Hardware Adaptation Layer)*** : C'est la couche centrale de l'architecture. Les composants de cette couche utilisent les interfaces fournies par la HPL pour construire des méthodes permettant d'utiliser au mieux les capacités de la plateforme pour lesquels ils sont créés. Par exemple un composant devant faire de la conversion de valeurs analogiques en valeurs digitales (ADC) ne sera pas le même en fonction de la plateforme sur laquelle on l'utilise. Certaines plateformes permettent la copie des valeurs converties par un contrôleur DMA qui accélère la copie. La couche HAL du convertisseur pour ces plateformes doit en tenir compte et utiliser ce contrôleur DMA afin de fournir un module ADC le plus performant possible.

- ***Couche HIL (Hardware Interface Layer)*** - La couche HIL met à la disposition du développeur les composants logiciels qui doivent être utilisables quelle que soit la plateforme sur laquelle s'exécute l'application. En fonction des performances des plateformes pour lesquelles ces composants sont développés, il faut faire un choix parmi les possibilités offertes par chacune et brider certaines possibilités au niveau de la HIL pour garantir l'indépendance du composant par rapport au matériel. Dans le cas d'applications destinées uniquement à une plateforme, qui demandent un grand contrôle sur le matériel ou qui doivent être optimisé au maximum, on peut se passer de la HIL.

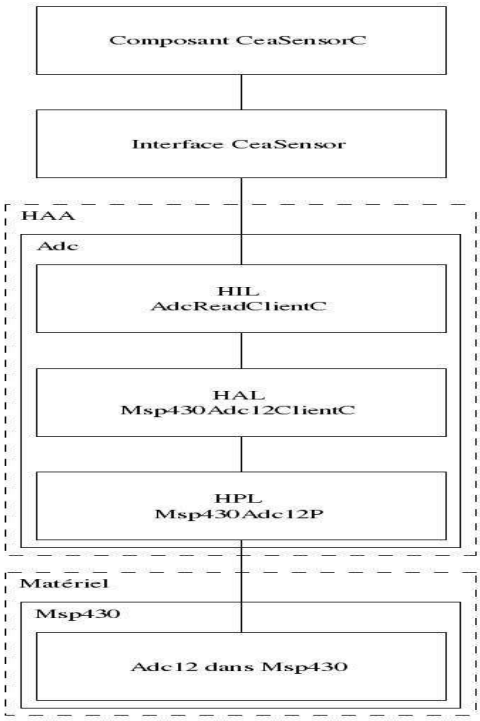

Figure II-11 : Différentes couches pour un composant utilisant l'ADC

II.3. Autres systèmes d'exploitation

D'autres systèmes d'exploitation connus pour les réseaux de capteurs sont décrits dans la liste non exhaustive suivante :

- Contiki [30] : système d'exploitation open-source multi-tâche, développé pour les systèmes embarqués avec contraintes de mémoire.
- SOS [31] : système d'exploitation développé par l'université de Los Angeles en Californie (UCLA) écrit en langage C et qui reprend le système de programmation événementielle de TinyOS.
- Mantis OS [32] : 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 [25] : Système d'exploitation multi-tâches pour systèmes embarqués avec une pile TCP/IP.

II.4 Le langage de programmation NesC

Le langage NesC a une architecture basée sur des composants. Cela permet de réduire la taille mémoire du système et de ses applications, chaque composant correspondant à un élément matériel du type LED ou Timer par exemple. Les composants peuvent être réutilisés dans n'importe quelle application. L'implémentation des composants, dans le but d'élaborer des applications, s'effectue en déclarant des tâches, des commandes ou des évènements [33].

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 pour y être exécutée. (Une instruction est une forme d'information communiquée qui est à la fois une commande et une explication pour décrire...) Lorsque la file d'attente des taches est vide (Le vide est avant tout un concept philosophique. Il désigne l'absence de matière.), le système d'exploitation met en veille le dispositif jusqu'au lancement de la prochaine interruption (on retrouve le fonctionnement event-driven).

Les évènements sont prioritaires par rapport aux tâches et peuvent interrompre la tache en cours d'exécution. Ils permettent de faire le lien entre les interruptions matérielles (pression d'un bouton, changement d'état d'une entrée, …) et les couches logicielles que constituent les tâches.

Dans la pratique, NesC permet de déclarer deux types de composants : les modules et les configurations.

- Les modules constituent les briques élémentaires de code et implémentent une ou plusieurs interfaces.
- Une application peut faire appel à des fichiers de configuration pour regrouper les fonctionnalités des modules. Un fichier top-level configuration permet de faire le lien entre tous les composants.

Les interfaces sont des fichiers décrivant les commandes et évènements proposés par le composant qui les implémente. L'utilisation des mots clefs « Use » et « Provide » au début

d'un composant permet de savoir respectivement si celui-ci fait appel à une fonction de l'interface ou redéfinit son code.

II.5 Le capteur TelosB

La plate-forme TelosB a été élaborée et publiée à la communauté scientifique par l'université Berkeley. Cette plate-forme offre une faible consommation d'énergie permettant une longue autonomie de la batterie ainsi qu'un éveil rapide de l'état de veille. Le micro-contrôleur TPR2420 utilisé dans TelosB, est compatible avec la distribution open-source de TinyOs. Ce type de nœud peut être utilisé dans les applications suivantes :

- Plate-forme à faible puissance pour le développement de la recherche.
- Expérimentation des réseaux de capteurs sans fil.

II.5.1. Principales caractéristiques de TelosB

- Micro-contrôleur MSP430 de Texas Instrument cadencé à 8MHz, avec 10Kb de RAM et 48Kb de flash.
- Radio Chipcon Wireless Transceiver CC2420 2.4GHz à 250kbps respectant la norme IEEE 802.15.4.
- Convertisseurs analogique-digital et contrôleur DMA intégrés.
- Liaison USB.
- Très faible consommation électrique.
- Temps de réveil inférieur à 6us.
- Un support d'expansion 16 ports.
- Encombrement réduit.
- Supporté par TinyOS.

II.5.2. Différents composants matériels du TelosB

Dans cette section, on détaille les différents composants matériels du capteur TelosB [30].

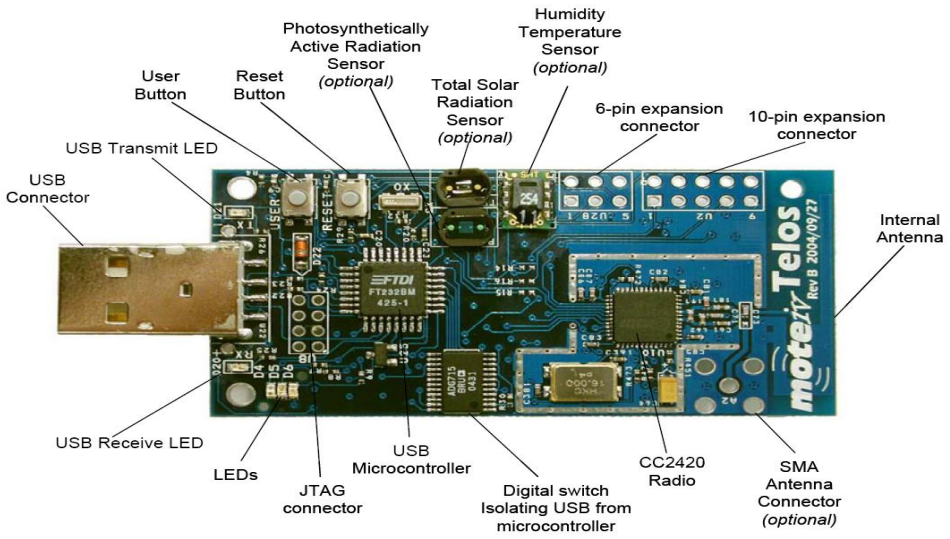

Figure II-12: Une carte TelosB

a. Le Micro-contrôleur MSP430

Le MSP430 contient un processeur RISC 16 bits et un système d'horloges flexible conçut spécialement pour une utilisation avec une batterie. Le système d'horloges met à disposition de l'utilisateur trois sources différentes en fonction des besoins en consommation et en fréquence :

- ACLK (Auxillary clock) pour les applications qui doivent consommer très peu et qui n'ont pas besoin de très hautes fréquences (32768Hz max).
- MCLK (Master clock) qui est utilisée par le CPU, utilisée pour les applications qui ont besoin de hautes fréquences (jusqu'à 8Mhz).
- SMCLK (Sub main clock) utilisable pour les modules périphériques.

Il contient aussi deux Timers internes configurables indépendamment, un module ADC 12 bits et un contrôleur DMA.

Timer : Le MSP430 possède deux Timer internes : Timer A et Timer B qui ont presque les mêmes caractéristiques, le B ayant quelques possibilités en plus. Nous allons voir le principe très simplifié de fonctionnement du Timer A pour comprendre comment le temps est géré par le nœud.

Le Timer A contient deux registres TAR (compteur) et TACCR0 (capture / comparaison) et est relié à une source d'horloge configurable. Plusieurs modes de comptage sont au choix : mode continu (CONTINUOUS) dans lequel le TAR est incrémenté jusqu'à sa valeur maximale 0xffff avant de générer une interruption, le mode montant (UP) dans lequel le TAR est incrémenté jusqu'a la valeur contenue dans TACCR0

avant de générer une interruption et le mode alterné (UP/DOWN) dans lequel TAR est incrémenté jusqu'à la valeur de TACCR0 puis décrémenté jusqu'a TACCR0 avant de générer l'interruption. L'incrémentation ou la décrémentation est déclenchée par des fronts montants sur l'entrée du registre de comparaison en fonction des cycles d'horloge. Le timer possède d'autres registres TACCR1, TACCR2 (le timer B en contient 6 au total) permettant de générer des interruptions à des intervalles différents. Plusieurs flags sont modifiables pour contrôler le timer par exemple remettre le TAR à zéro ou désactiver les interruptions.

ADC-12 : L'ADC 12 (Analog to Digital Converter 12 bits) permet la conversion de signal analogique entrant sur un ou plusieurs ports en valeur numérique de 12 bits à un débit allant jusqu'a 200 000 conversions par seconde. Plusieurs modes de conversion sont possible : Single Channel (un seul port est lu) ou Multiple Channel (plusieurs ports sont lus à la suite), on peut configurer le nombre de conversions à effectuer avant d'émettre l'interruption de fin de conversion et le nombre de cycles d'horloges avant la prochaine série de conversions. L'horloge utilisée pour le déclenchement des conversions est comme pour les Timer au choix. L'ADC contient un tampon de 16 entrées de 12 bits permettant de convertir les données et de stocker les résultats sans avoir à passer par le CPU.

DMA : Le MSP430 contient un contrôleur DMA (Direct Memory Access) qui permet de transférer des données d'une adresse mémoire à une autre sans intervention du CPU. L'utilisation de la DMA à deux avantages : les transferts sont plus rapides (ils ne nécessitent que deux cycles d'horloge) et permet de ne pas occuper le CPU et donc de pouvoir le laisser dans un mode de basse consommation électrique et surtout de le laisser libre pour d'autres taches. Le contrôleur DMA peut utiliser trois canaux de transferts indépendamment auxquels on peut assigner des priorités et qu'on peut configurer en fonction du type de transfert à effectuer. On choisit pour chaque canal la taille des données à transférer, octet (byte = 8 bits) ou mot (word = 16 bits) et le mode de transfert. Les transferts sont activés par un trigger au choix (fin de conversion ADC, Timer A ou B, fin de transmission USB...). Un flag permet d'activer la DMA en mode répétition qui permet de ne pas éteindre le contrôleur après chaque transfert qui évite des temps de réveils qui ralentissent les transferts.

Trois modes de transfert sont configurables :

- Single : le transfert est fait un octet (ou mot) à la fois.

- Block : le transfert se fait pour un bloc d'octets.
- Burst-block : le transfert se fait comme pour le mode Block mais avec deux cycles d'horloges supplémentaires tous les 4 octets exécutent par le CPU.

b. Le module radio CC2420

Un module radio CC2420, conforme à la norme IEEE 802.15.4, travaillant sur une bande de fréquences comprise entre 2,4 GHz et 2,4835 GHz. Il permet de travailler sur 16 canaux étanches différents.

c. La mémoire flash

La mémoire flash est une mémoire de masse à semi-conducteurs réinscriptible, c'est-à-dire une mémoire possédant les caractéristiques d'une mémoire vive mais dont les données ne disparaissent pas lors d'une mise hors tension. Ainsi, la mémoire flash stocke les bits de données dans des cellules de mémoire, mais les données sont conservées en mémoire lorsque l'alimentation électrique est coupée.

II.6 Conclusion

Dans ce chapitre, on a présenté les outils logiciels conçus pour les équipements à ressources limitées dont les capteurs font parties. Puis, on a décrit les compétences du capteur TelosB.

Cette étude nous a permis de tirer profit des outils logiciels tels que TinyOs et NesC qui peuvent permettre la mise en place d'un réseau de capteurs pour les applications de monitoring.

CHAPITRE III :

Développement d'une application de monitoring

Chapitre III
Développement d'une application de monitoring

III.1 Introduction

Les réseaux de capteurs ont envahi plusieurs domaines d'applications. On distingue deux types d'applications : les applications orientées événements et les applications orientées surveillance. Dans le premier type, on s'intéresse principalement aux événements pertinents qui peuvent survenir dans la zone de déploiement alors que dans le deuxième type on s'intéresse à une surveillance de longue durée d'une zone d'intérêt.

Les applications orientées surveillance consistent à envoyer des informations périodiques à un centre de contrôle distant. Ces informations peuvent être exploitées directement ou stocker dans une base de données pour effectuer des éventuelles statistiques. Par exemple, le suivi d'un patient, le suivi d'un animal, contrôle d'un bâtiment, etc.

Dans ce chapitre, on implémente une application orientée surveillance en tenant compte des contraintes matérielles des réseaux de capteurs : puissance de calcul limitée, mémoire de stockage réduite, portée de transmission courte, et une autonomie de l'énergie. Pour se faire, des outils logiciels spéciaux ont été développés pour les dispositifs à ressources limitées comme les réseaux de capteurs.

L'application développée est une application générique. Elle consiste à faire des prélèvements de températures dans plusieurs endroits d'une zone d'intérêt et les envoyer à un poste de contrôle distant qu'il les analyse et les stocke dans une base de données pour effectuer des éventuelles statistiques.

III.2 Objectif de l'application

L'étude du changement climatique se manifeste sous des formes diverses et ses conséquences sont bien plus complexes. Par exemple, le réchauffement de l'eau et la fonte des glaces peuvent éventuellement subir des dégâts graves qui menacent l'existence de l'humanité. Dans ces formes, la température est le paramètre en commun.

Dans ce contexte, on pourra tirer profit des avantages des réseaux de capteurs pour développer une application de surveillance de la température.

III.3 Les choix techniques

L'implémentation cette application à la base des réseaux de capteurs nécessite des outils matériels et des outils logiciels bien spécifiques tels que NesC et Java comme langages, TinyOs comme système d'exploitation, MySql comme base de données.

III.3.1 Choix du langage

Pour réaliser l'application, on a utilisé deux langages différents : NesC et Java. Le premier pour programmer les capteurs et le deuxième permet d'utiliser ses atouts pour bien exploiter l'application au niveau du poste de contrôle par exemple la création d'interfaces graphique grâce à leur librairie Swing et aussi la communication dans un réseau grâce au Socket. Ce langage est supporté par Windows, Unix et Mac et dispose d'outils pour communiquer et exploiter les données envoyées par les capteurs tels SerialForwarder et MIG.

Ces deux langages se valant plus ou moins, le choix du langage s'est donc fait par affinité. Java contenant les librairies nécessaires au développement d'interfaces homme/machine et étant le seul dont le code exécutable est portable, c'est ce langage qui a été choisi pour l'application. En outre, NesC est un langage orienté composant conçu pour programmer les dispositifs à ressources limitées.

III.3.2 Choix de la base de données

Dans une application de monitoring, on a besoin dans certains cas de stocker les données collectées par les capteurs durant une longue période. De ce fait, le volume de données stockées serait un peu consistant ce qui nécessite d'instaurer un moyen de stockage performant tel que les bases de données. Dans cette optique, on a utilisé le système de gestion de base de données MySQL qui est open-source, fiable, très simple d'utilisation, compatible avec Windows, Mac et Linux et possède une documentation complète.

L'accès à la base de données se fait via l'API JDBC (Java DataBase Connectivity). Cette API permet à une application Java d'accéder à une base de données si celle-ci possède des drivers JDBC (actuellement des drivers existent pour tous les SGBD). Pour se connecter à un SGBD avec cette API, il suffit de préciser le driver du SGBD qui sera utilisé. Pour notre

cas le nom de driver est "com.mysql.jdbc.Driver", et le chemin d'accès à la base est "jdbc:mysql:base_de_données ".

III.3.3 Autres outils

Il existe d'autres outils qui s'avèrent nécessaire pour la mise en place de cette application

a) NetBeans IDE 7.1.0

NetBeans est un IDE (Integrated Development Environment) modulable et open-source qui offre de nombreuses fonctionnalités :

- Il contient un éditeur avec coloration syntaxique suivant le langage choisi.
- Il permet la navigation et la gestion des différents projets et de leurs classes.
- Il contient un debugger et un compilateur.
- Il renomme automatiquement les fonctions/classes/variables dans l'ensemble du code.
- Il permet la complétion lors de la saisie du code.
- Il est compatible avec Windows, Unix et Mac.

Etant simple d'utilisation, très complète et gratuite, le choix de ce logiciel s'est donc fait naturellement.

b) MySQL 5.5

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multi-thread et multi-utilisateur.

MySQL est un logiciel libre développé sous double licence en fonction de l'utilisation qui en est faite : dans un produit libre ou dans un produit propriétaire. Dans ce dernier cas, la licence est payante, sinon c'est la licence publique générale GNU (GPL) qui s'applique

III.4 Mise en place de la plateforme

La mise en place de la plateforme s'articule autour de deux parties : l'installation logicielle et l'installation matérielle.

III.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 choisi l'environnement Linux (Ubuntu) pour la mise en place de l'application de monitoring. Il est possible de l'exploiter sous cygwin (Windows) mais il y a des soucis avec la deuxième version de TinyOs (TinyOs-2.x) sous cet environnement.

La phase d'installation a été la plus délicate. En effet, on a commencé par installer TinyOS 2.2.0, NesC et le micro-contrôleur MSP430 pour les capteurs de type TelosB. La procédure d'installation de TinyOs se déroule en plusieurs étapes et elle est décrite dans l'annexe. Puis on a installé NetBeans au niveau du poste de contrôle pour communiquer avec les capteurs et faire visualiser les valeurs des grandeurs remontées à la station de base.

III.4.2 Installation matérielle

Une fois l'installation logicielle terminée, il a fallu installer le matériel : une station de base reliée à l'ordinateur via un câble USB, différents capteurs TelosB (Sender/Receiver). Chacun des Sender mesure la température et la communique au capteur Receiver via une liaison sans fil. Un capteur Receiver joue le rôle de nœud relai et permet d'hiérarchiser la plateforme pour éviter les transmissions redondantes. Il communique avec la station de base via une liaison sans fil, et la station de base communique avec l'ordinateur via le câble USB.

La figure III.1 présente les schémas de communication dans la plateforme qu'on a développée. Dans cette architecture, on a plusieurs capteurs de type Sender qui sont déployés dans plusieurs zone et à l'extrémité de chaque il existe un capteur de type Receiver qui joue le rôle de point de collecte relatif à une zone. Chacun des ces capteurs Receiver est connecté à la station de base.

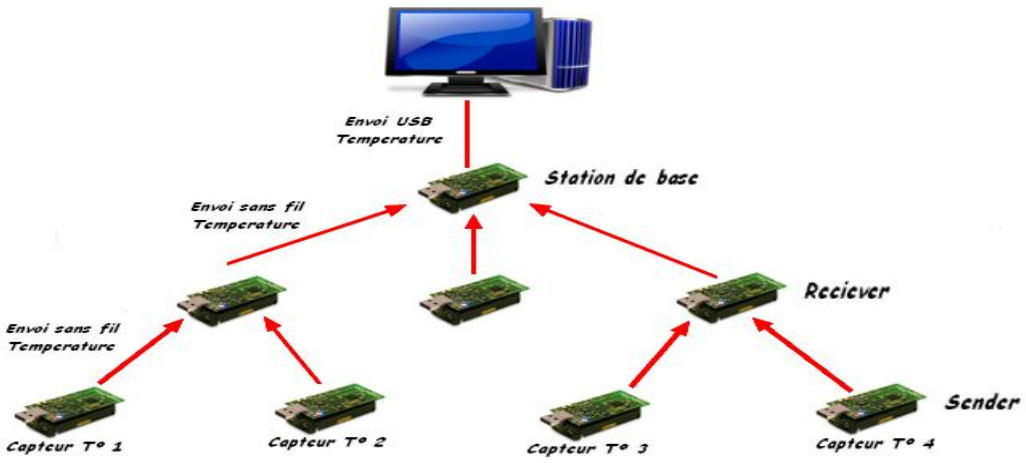

Figure III-13 : Schémas de communication dans la plateforme d'expérimentation

Pour réaliser l'application on dispose de trois capteurs de type TelosB comme montre la figure III.2 :

- Un capteur est déployé dans une zone d'intérêt et joue le rôle de Sender
- Un capteur est configuré comme Receiver
- Un capteur est configuré comme station de base et il est connecté au poste de contrôle.

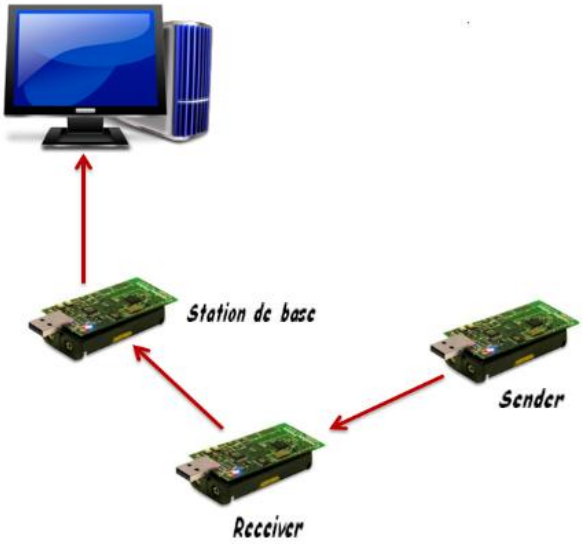

Figure III-14 : Environnement du travail

Dans cette architecture simplifiée, chaque capteur TelosB Sender envoie périodiquement la température ambiante au Receiver qui à son tour l'envoie à la station de base via une liaison sans fil. La station de base transmet ces températures via la liaison USB à un poste de contrôle distant. Ce dernier analyse et stocke ces données.

III.5 Etapes de la réalisation de la plateforme

La Figure III.3 résume l'architecture générale de la plateforme réalisée.

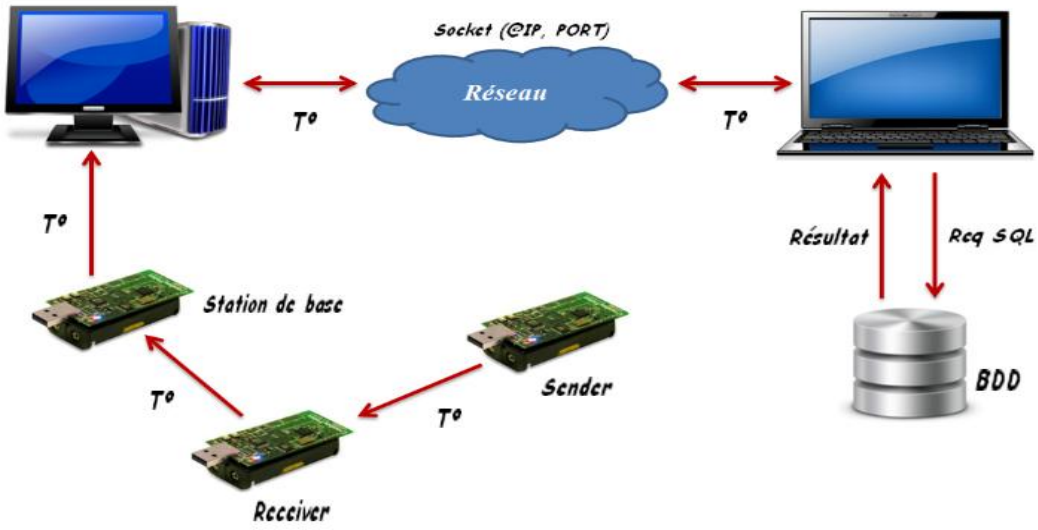

Figure III-15 : Architecture générale de la plateforme d'expérimentation

Chaque capteur TelosB Sender envoie à intervalle régulier un message contenant sa température et son identifiant à la station de base via capteur TelosB Receiver. La station de base transmet ces messages à l'application java qui va les traiter. L'application reçoit donc les températures et identifiants des capteurs, les affiche dans l'interface et les enregistre dans la base de données.

Pour communiquer avec la base de données, l'application java utilise l'API JDBC. Cette API java permet la liaison entre une base de données et une application java, l'application envoie une requête à la base de données via l'API, la base de données retourne le résultat sous forme de tableau à l'API qui le transmet alors à l'application.

III.5.1 Bases de données MySQL

Dans l'application qu'on a développée, on a besoin de créer une base de données nommé « Db_Capteur ». Celle-ci contiendra une table nommée « Table_Capteur » c'est pour conserver les données qui sont émis pas les capteurs Telosb Sender (Numéro de relevé, Température, Date, …). La Figure III.4 présente une table simplifiée qui reflète les informations remontées par les capteurs.

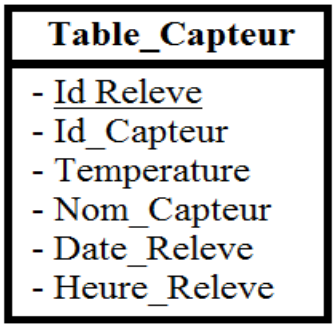

Figure III-16 : la table Table_Capteur

III.5.2 Programmes en NesC

Les programmes en NesC qu'on les met sur les capteurs s'articulent autour de ces composants :

- Main : cœur de l'application.
- ActiveMessage : permet l'accès à la liaison sans fil et l'encapsulation de messages qui pourront être ensuite envoyés via la liaison sans fil.
- Leds : permet de suivre le déroulement du programme en allumant les Leds du capteur à des moments pertinents.
- TimerMilli : permet le déclenchement d'évènements périodiques.
- SensirionSht11 : lit la température et l'humidité.
- AMSender : permet l'envoi de messages via la liaison sans fil.

Le composant TimerMilli déclenche chaque seconde un évènement. Cet événement demande au composant SensirionSht11 de lire et calculer la température puis d'encapsuler cette valeur à l'aide du composant ActiveMessage et de l'envoyer via le composant AMSender. Ces données sont ensuite reçues par la station de base, qui les envoie à l'application Java en utilisant des sockets.

III.5.3 Quelques exécutions

La Figure III.5 représente l'interface portail de l'application

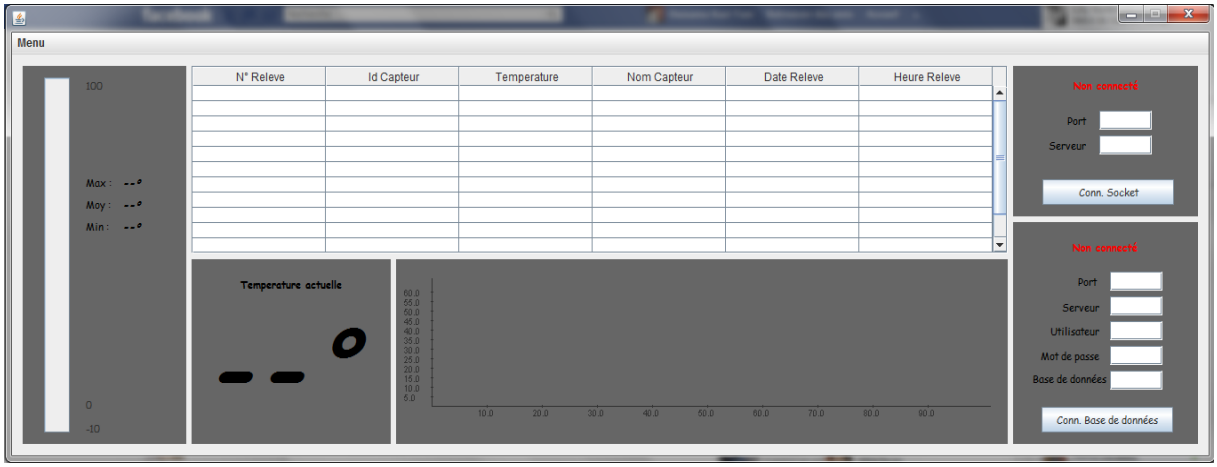

Figure III-17: Interface de l'application

L'interface principale se compose de plusieurs parties :

a) Connexion a la base de données

Après l'exécution de l'application, la première opération à effectuer c'est remplir les TextField pour établir une connexion avec la base de données selon la Figure III-7. Si cette opération se termine avec succès on continue sinon c'est qu'il y a un certain composant qui a causé cette anomalie.

Figure III-18: configuration de la base de données

b) Connexion à la station de base

Une fois la connexion à la base de données est établie, il faut remplir les TextField pour configurer une autre connexion au serveur ou à la station de base qui nous fournira les températures envoyées par les capteurs TelosB Sender

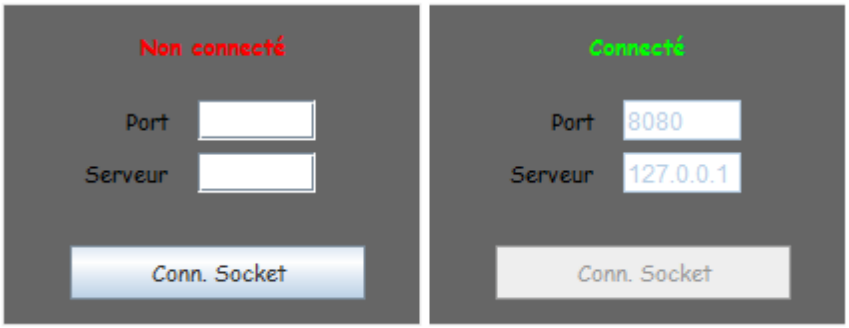

Figure III-19: **configuration de la station de base**

c) Gestion de la base de données

La partie de gestion de la base de données (Figure III-8) se compose d'un tableau qui permet de voir le contenu de la base de données en temps réel et aussi lorsque la connexion à la base de données est établie. Il affiche entre autres le numéro de relevé de température, l'identifiant du capteur, son nom et sa température à une date et heure précise. Ces capteurs peuvent être déployés dans différents endroits de la zone d'intérêt et par suite on pourra connaitre l'endroit où il y a des événements critiques, par exemple une température qui dépasse un certain seuil.

N° Releve	Id Capteur	Temperature	Nom Capteur	Date Releve	Heure Releve
306	11	10	Abbes	2012-06-06	19:04:27
307	11	11	Abbes	2012-06-06	19:04:30
308	11	12	Abbes	2012-06-06	19:04:33
309	11	13	Abbes	2012-06-06	19:04:36
310	11	14	Abbes	2012-06-06	19:04:39
311	11	15	Abbes	2012-06-06	19:04:42
312	11	16	Abbes	2012-06-06	19:04:45
313	11	17	Abbes	2012-06-06	19:04:48
314	11	18	Abbes	2012-06-06	19:04:51
315	11	19	Abbes	2012-06-06	19:04:54
316	11	20	Abbes	2012-06-06	19:04:57

Figure III-20: Gestion de la base de données

d) Courbe de températures

Une fois la phase de connexion se déroule normalement, on assiste à la phase d'envoi de données. Dans notre cas, il s'agit de l'envoi de la température d'une manière périodique et cette température s'affiche directement sur écran. La partie de courbe de température (Figure III-9) illustre l'évolution de la température au cours du temps.

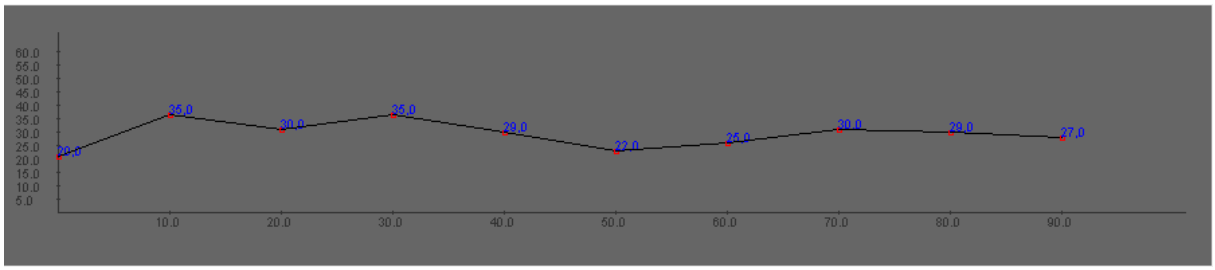

Figure III-21: courbes de températures

e) L'affichage de la température

Cette partie est consacrée à l'affichage en temps réel des données envoyées par les capteurs en deux façons. La première consiste à afficher seulement la valeur envoyée alors que dans la deuxième la température est représentée sous de thermomètre qui prend différentes couleurs en fonction de la valeur de cette température. En plus, une évaluation statistique est faite dans laquelle on trouve la valeur maximum, minimum et la moyenne durant la connexion à la station de base comme montre la Figure III-10.

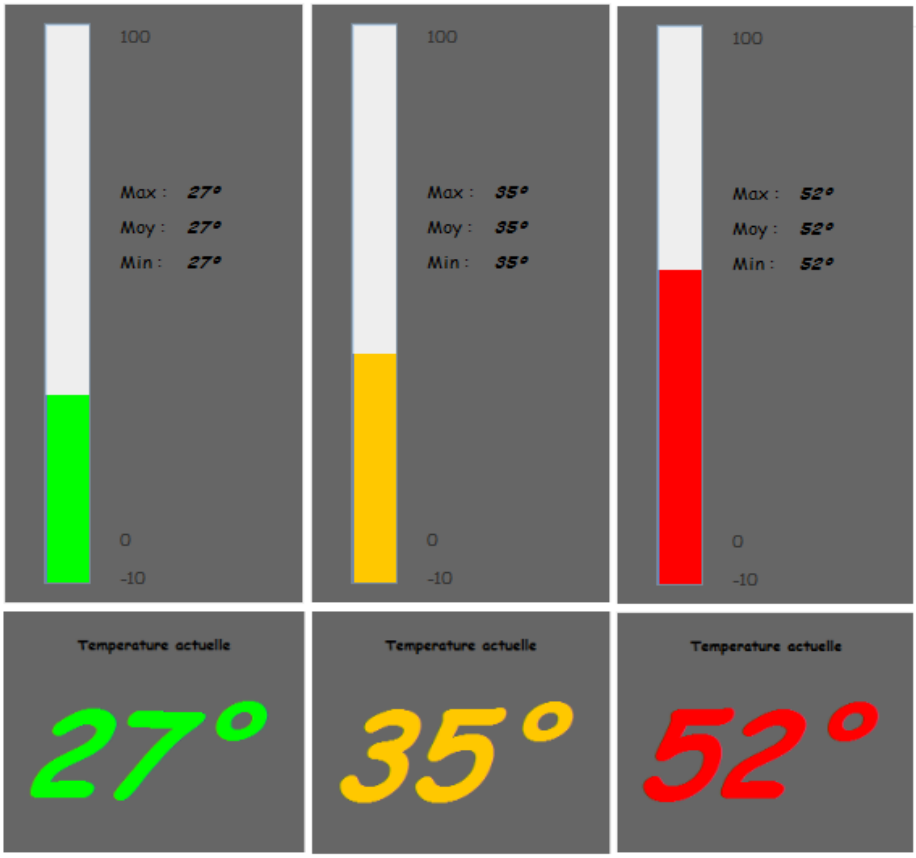

Figure III-22 : Affichage de température

III.6 Conclusion

Dans ce chapitre, on a présenté les outils logiciels et matériels ainsi que la démarche à suivre pour réaliser une application orientée surveillance.

Cette implémentation basée sur les réseaux de capteurs exige des outils bien particuliers qui sont développés pour exploiter les systèmes à ressources limitées tels que un système d'exploitation léger « TinyOs », un langage orienté composant « NesC » et une base de données pour stocker et évaluer éventuellement des statistiques sur les informations collectées.

Conclusion générale

Les réseaux de capteurs sans fil présentent des spécificités bien particulières. Ils se caractérisent par l'aspect miniaturisé et les ressources limitées en termes d'énergie, calcul et stockage. Ils traitent deux types d'applications: les applications orientées événements et les applications orientées surveillance. Dans le premier type, on s'intéresse principalement aux événements pertinents qui peuvent survenir dans la zone de déploiement alors que dans le deuxième type on s'intéresse à une surveillance de longue durée d'une zone d'intérêt.

Dans le cadre de notre projet, on s'est intéressé aux applications orientées surveillance. Ces applications consistent à envoyer des informations périodiques à un centre de contrôle distant. Ces informations peuvent être exploitées directement ou stocker dans une base de données pour effectuer des éventuelles statistiques. Par exemple, le suivi d'un patient, le suivi d'un animal, contrôle d'un bâtiment, etc.

Après avoir étudié les spécificités des réseaux de capteurs, on a implémenté une application orientée surveillance en tenant compte des contraintes matérielles des réseaux de capteurs : puissance de calcul limitée, mémoire de stockage réduite, portée de transmission courte, et une autonomie de l'énergie. Pour réaliser cette application, on a utilisé des outils logiciels spéciaux qui ont été développés pour les dispositifs à ressources limitées : TinyOs comme système d'application et NesC comme langage de programmation des capteurs.

Installing TinyOS in Ubuntu 11.10

1. To edit the sources list, open the terminal and type :

   ```
   $ sudo gedit /etc/apt/sources.list
   ```

   add this line at the end of the file opened

   ```
   deb http://tinyos.stanford.edu/tinyos/dists/ubuntu karmic main
   ```

   "karmic" for all type of ubuntu distribution.

2. Run a couple of commands to update and install tinyos

   ```
   $ sudo apt-get update
   $ sudo apt-get install tinyos-2.1.1
   ```

3. Then we need to change the environmental variables. For this we open the .bashrc file

   ```
   $ gedit ~/.bashrc
   ```

   and paste the following lines to the bashrc file.

   ```
   export TOSROOT=/opt/tinyos-2.1.1
   export TOSDIR=$TOSROOT/tos
   export CLASSPATH=$TOSROOT/support/sdk/java/tinyos.jar:.:$CLASSPATH
   export MAKERULES=$TOSROOT/support/make/Makerules
   export PATH=/opt/msp430/bin:$PATH
   #Sourcing the tinyos environment variable setup script
   source /opt/tinyos-2.1.1/tinyos.sh
   ```

4. We then change the owner for the tinyos installed on the system.

   ```
   sudo chown yourname:yourname -R /opt/tinyos-2.1.1/
   ```

5. Finally, to check if the installation was done, run the command

   ```
   $ tos-check-env
   ```

Références bibliographiques

[1] Georges, Asch. Les capteurs en instrumentation industrielle. Dunod. 1998.

[2] Noury, Norbet. Du signal à l'information : le capteur intelligent Exemples industriels et en médecine. Grenoble: TIMC-IMAG, 2002.

[3] S.J., Pister Kristofer. http://webs.cs.berkeley.edu/tos/. TinyOS. [En ligne] 2009.

[4] Kristofer, Pister. http://robotics.eecs.berkeley.edu/~pister/SmartDust/. SmartDust. [En ligne] 2001.

[5] Crossbow. MICA2 Data sheet. [En ligne] 2009. http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf.

[6] Antoine Gallais, François Ingelrest, Jean Carle, David Simplot-Ryl. Maintien de la couverture de surface dans les réseaux de capteurs avec une couche physique non idéale. CFIP, Colloque Francophone sur l'Ingénierie des Protocoles. 2006.

[7] Gallais, Antoine. Ordonnancement d'activité dans les réseaux de capteurs : l'exemple de la couverture de surface. Université des sciences et technologies de Lille. 2007. Rapport de thèse.

[8] Cheng, K. Field and Wave Electromagnetics. s.l.: Addison-Wesley, 1989. p. 639.

[9] CrossBow. Sensor Boards. [En ligne] https://www.xbow.com/Products/productdetails.aspx?sid=158.

[10] The Sensor Museum. In http://www.btnode.ethz.ch/Projects/SensorNetworkMuseum/.

[11] David Martins, "Sécurité dans les réseaux de capteurs sans fil stéganographie et réseaux de confiance", Thèse de doctorat, Université de FRANCHE-COMTÉ, 2010.

[12] I. F. Akyildiz, W. Su , Y. Sankarasubramaniam , E. Cayirci. Wireless sensor networks: a survey. 2002, Vol. 38, pp. 393-422.

[13] C.Y. Chong, S.P. Kumar. Sensor Networks: Evolution, Opportunities, and Challenges. Proceedings of the IEEE. 2003, Vol. 91, pp. 1247-1256.

[14] T.B. Gosnell, J.M. Hall, C.L. Ham, D.A. Knapp, Z.M. Koenig, S.J. Luke, B.A. Pohl, A.Schach von Wittenau, and J.K. Wolford. Gamma-Ray Identification of Nuclear Weapon Materials. Lawrence Livermore National Lab. Livermore CA, USA: s.n., February, 1997. Technical repport DE97053424.

[15] Brown, M. J. Users Guide Developed for the JBREWS Project. Los Alamos National Laboratory of California University. 1999. Technical repport LA-UR-99-4676.

[16] Andrews, P. Johnson and D.C. Remote continuous monitoring in the home. Telemedicine and Telecare. June 1996, Vol. 2, 2, pp. 107-113.

[17] E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis, and V.Z. Groza. Sensor-based information appliances. IEEE Instrumentation Measurement Magazine. December 2000, Vol. 3, 4, pp. 31-35.

[18] Michael Fitzgerald. Technnology Review : Tracking a Shopper's Habits. Technology Review. [En ligne] 04 August 2008. http://www.technologyreview.com/computing/21161/.

[20] Hung-Cuong LE. "Optimisation d'accès au médium et stockage de données distribuées dans les réseaux de capteurs". Thèse de Doctorat, Université de FRANCHE-COMTÉ.

[22] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. Contiki - a lightweight and exible LCN '04 : Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks operating system for tiny networked sensors. pp. 455-462, Washington, DC, USA, 2004. IEEE Computer Society.

[23] Chih-Chieh Han, Ram Kumar, Roy Shea, Eddie Kohler, and Mani Srivastava. A dynamic MobiSys '05 : Proceedings of the 3rd international conference on Mobile systems, applications, and services operating system for sensor nodes. pp. 163-176, New York, NY, USA, 2005. ACM.

[24] Shah Bhatti, James Carlson, Hui Dai, Jing Deng, Je Rose, Anmol Sheth, Brian Shucker, Charles Gruenwald, Adam Torgerson, and Richard Han. Mantis os : an embedded multithreaded operating system for wireless micro sensor platforms. Mob. Netw. Appl. 10(4) :563-579, 2005.

[25] Nut/OS. In http ://www.ethernut.de/en/software/index.html.

[26] Andrew Tanenbaum. Computer Networks. Prentice Hall Professional Technical Reference, 2002. 42.

[27] TinyOS. http ://www.tinyos.net/. 25, 60, 66.

[28] www.tinyos.net/tinyos-1.x/doc/.

[29] http ://www.tinyos.net/tinyos-1.x/doc/nesc/ref.pdf.

[30] http://www.contiki-os.org/.

[31] http://www.sos.enix.org/.

[32] http://www.mantisos.org/.

[33] J. Polastre, R. Szewczyk, and D. Culler. Telos : Enabling Ultra-Low Power Wireless Research. In proceedings of the 4th International Conference on Information Processing in Sensor Networks : Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN'05), pp.364-369, April 2005.

Développement d'une application orientée surveillance pour les réseaux de capteurs

Les réseaux de capteurs sans fil se caractérisent par l'aspect miniaturisé et les ressources limitées en termes d'énergie, calcul et stockage. Ils sont conçus pour plusieurs domaines d'applications.

Dans le cadre de notre projet, on a implémenté une plateforme pour les applications orientées surveillance. Ces applications consistent à envoyer des informations périodiques à un centre de contrôle distant. Ces informations peuvent être exploitées directement ou stocker dans une base de données pour effectuer des éventuelles statistiques. Pour réaliser cette plateforme, on a utilisé des outils logiciels spéciaux qui ont été développés pour les dispositifs à ressources limitées : TinyOs comme système d'application et NesC comme langage de programmation des capteurs.

Mots clés : Réseaux de capteurs, surveillance, TinyOS, NesC,

Implementation of monitoring applications for wireless sensor networks

Wireless sensor networks are characterized by the appearance miniaturized and limited resources in terms of energy, computing and storage. They are designed for several application areas.

In our project, we have implemented a platform for monitoring applications. These applications enable to send periodic information to a remote control center. These informations can be used or stored in a database to perform eventual statistics. To implement this platform, we used special software tools that were developed for devices with limited resources: TinyOS as operating system and NesC as programming sensor language.

Keywords: Wireless sensor networks, monitoring, TinyOS, NesC

تطوير برنامج مراقبة لشبكات الاستشعار

تتميزشبكات الاستشعار بطابعها المصغر و محدودية قدراتها الحسابية و الطاقوية و التخزينية. بيد أنها تستعمل في عدة مجالات .
في هذه المذكرة قمنا بإنشاء برنامج لاستغلال شبكات الإستشعار لمراقبة بعض البرامج الذين يقومون بإرسال المعلومات إلى مركز مراقبة بشكل دوي .ويمكن استخدام هذه المعلومات مباشرة أو تخزينها في قاعدة المعطيات لتنفيذ أي إحصاءات.

من أجل إنشاء هذا البرنامج لقد استعمنا البرمجيات الخاصة التي تم تطويرها لأجهزة ذات القدرات المحدودة : TinyOS كنظام تشغيل وNesC كلغة برمجة أجهزة الاستشعار.

الكلمات المفتاحية :شبكات الاستشعار, مراقبة ,TinyOS, NesC