

# Projet Tutoré

## Réseaux de capteurs sans-fil : Configuration et test de connectivité IP

LABORDE Cédric - DELGADO Frédéric  
encadrant : M. Pham

*Master Technologie de l'Internet 1ère année*

Mai 2010



# Introduction

## Objectifs du projet :

- se familiariser avec le domaine des réseaux de capteurs
- établir une connectivité IP entre des capteurs sans-fils
- réaliser une application permettant de récupérer des données
- réaliser un tutoriel afin de faciliter la poursuite de notre travail

## Avantages de IP :

- identification des capteurs par leur adresse IP
- plus facilement intègrable dans un réseau IP existant

# Plan

- 1 Présentation
- 2 Environnement
  - MicaZ
  - TinyOS
  - NesC
- 3 Implémentation
  - Fonctionnement général
  - Code embarqué
  - Client
  - Interface
- 4 Bilan
- 5 Webographie

# Présentation

# Présentation

## Capteurs et réseaux de capteurs

**définition** : Un capteur est un petit appareil autonome capable d'effectuer des mesures simples sur son environnement immédiat.

- Les réseaux de capteurs sont de plus en plus utilisés dans différents domaines :
  - la domotique
  - la sécurité
  - la médecine
  - l'industrie
  - etc...
- Le sans-fil se développe de plus en plus au détriment des liaisons filaires.

# Présentation

## Capteurs et réseaux de capteurs

- L'autonomie ainsi que la miniaturisation des capteurs impliquent de fortes contraintes matérielles et logicielles.
  - mémoire réduite
  - économie d'énergie
  - technologies adaptées
- Le réseau de capteurs sans fils est l'une des dix nouvelles technologies qui bouleverseront le monde et notre manière de vivre et de travailler d'après le magazine *Technology Review* du MIT.

# Présentation

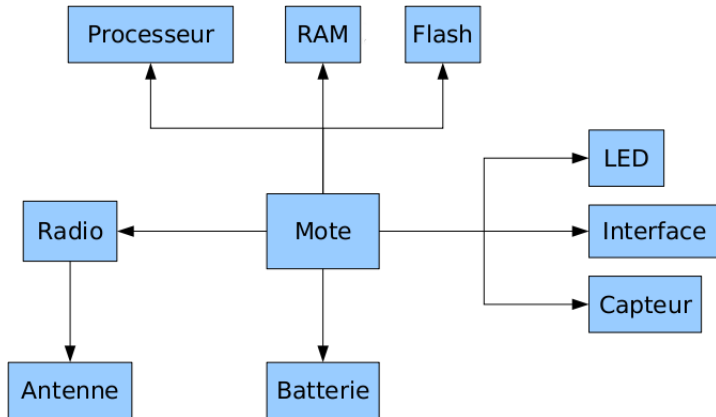


Figure: Architecture générale d'un capteur

# Environnement Technique





Le modèle de capteur utilisé est le MicaZ de la marque Crossbow.

- mémoire flash de 512 Kbytes pour stocker les mesures
- mémoire flash de 128 Kbytes pour le programme
- alimenté par deux piles AA (2,7 à 3,3 V)
- communication par ondes radio utilisant le protocole ZigBee
- portée : d'une trentaine de mètres en intérieur à une centaine en extérieur
- utilisation d'un MIB520 pour communiquer avec un PC

# MicaZ - ZigBee



ZigBee est un protocole de haut niveau permettant la communication par ondes radio

- premières ébauches dès 1998 s'inspirant du Bluetooth
- s'appuie sur le protocole 805.15.4 défini par l'IEEE
- moins gourmand que le Wifi ou le Bluetooth apparus précédemment
- protocole lent dont le rayon d'action est relativement faible
- consommation considérablement réduite

TinyOS est un système d'exploitation open-source, conçu par des chercheurs de Berkeley

- créé pour les réseaux de capteurs sans-fils
- taille du code réduite
- s'appuie sur le langage NesC
- fonctionnement événementiel
- bibliothèque de composants très complète

# TinyOS - Blip

Blip, conçu par Berkeley, est l'implémentation dans TinyOS d'un certain nombre de protocoles basés IPv6.

- Berkeley Low-power IP stack
- fournit les structures de données nécessaire à la programmation de sockets
- ne répond pas encore entièrement aux normes

# TinyOS - Blip

## Routeur de bord

Afin de pouvoir communiquer avec le réseau IP de capteurs, il est nécessaire d'installer et d'exécuter un routeur de bord sur la machine connectée au capteur.

- lien entre le réseau de la machine et le réseau de capteur.
- configuration contenu dans un fichier et chargé au démarrage
  - adresse IP
  - canal de communication

Le langage nesC est le langage utilisé par tinyOS

- Network Embedded System C
- syntaxiquement proche du C
- architecture basée sur des composants
  - LEDs
  - timer
  - carte de capture
- programmation événementielle
  - la mise en route du capteur
  - la fin d'un timer
  - la réception d'un message
  - l'envoi d'un message
  - ...

# NesC - La compilation croisée

La programmation des capteurs se faisant sur un PC ayant des caractéristiques totalement différentes des capteurs, il est nécessaire d'utiliser un compilateur croisé.

- permet de traduire un code source en code objet ayant un environnement d'exécution différent de celui où la compilation est effectuée.
- proposé par le compilateur NesC : `ncc`
- pour une application destinée au MicaZ :
  - `make micaz`
- pour une application destinée au MicaZ utilisant Blip :
  - `make blip micaz`

# Implémentation



# Fonctionnement général

**IPOscilloscope** permet de visualiser en temps réel le niveau de batterie d'un capteur, en se connectant à celui ci grâce à son adresse IP.

- composé de deux parties :
  - la partie embarquée : NesC
  - la partie cliente : Java
- La partie cliente adresse une demande de données au capteur
- le capteur effectue des mesures puis les envoie au client demandeur
- problème : pertes de paquets
- solution : utilisation d'un timer

# Fonctionnement général

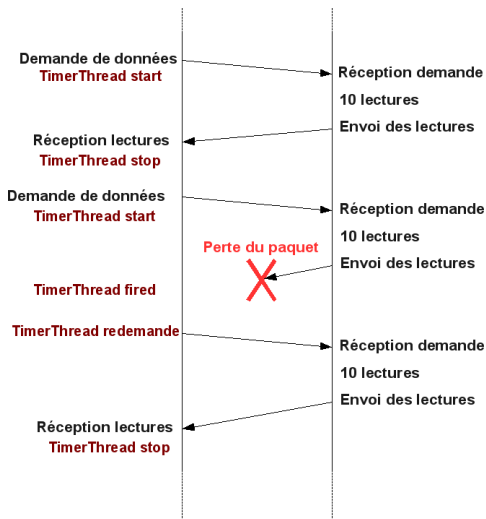


Figure: Schema du fonctionnement général de l'application

# Partie embarquée

- inclusion de la librairie Blip
  - structures IPv6
  - sockets UDP
- programmée pour réagir aux évènements :
  - démarrage du capteur
  - réception d'un message
  - fin du timer
  - envoi d'un message
- Lors de l'attente d'un évènement, le capteur est en veille

## Partie embarquée

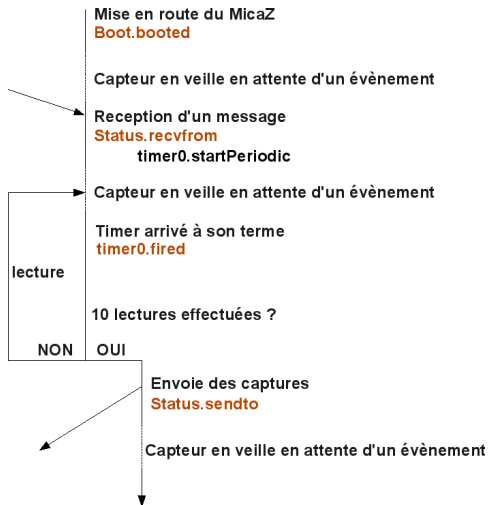


Figure: Schema du fonctionnement de la partie embarqué

## Partie Java - Client

- utilise des sockets UDP Java
- prend en paramètre l'adresse IPv6 du capteur
  - `java Client fec0::1`
- dispose du timer de gestion des pertes de paquets
  - initialise la communication en demandant des données
  - lance le timer
  - se met en attente de données
  - si les données ne sont pas reçues avant 1s, le timer redemande
  - on recommence tant que l'application est en route
- interopérabilité NesC - Java
- visualisation des données au travers d'une interface

## Partie Java - Interface

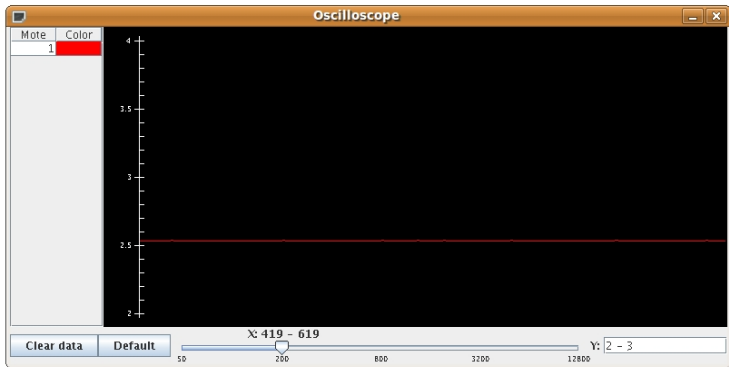


Figure: Capture d'écran de l'interface Java

# Bilan

- bénéfiques personnels
  - domaine des réseaux de capteurs en pleine expansion
  - découverte de nouvelles technologies
    - TinyOS
    - NesC
  - apprentissage de nouveaux paradigmes de programmation
    - orientée composant
    - événementielle
- possibilités d'évolution du projet
  - intégration au réseau internet
  - augmenter le nombre de capteur



# Webographie

- Site officiel de tinyos
  - <http://www.tinyos.net>
  - [http://docs.tinyos.net/index.php/Main\\_Page](http://docs.tinyos.net/index.php/Main_Page)
- Site officiel de zigbee
  - <http://www.zigbee.org>
- Tutoriel officiel blip
  - [http://docs.tinyos.net/index.php/BLIP\\_Tutorial](http://docs.tinyos.net/index.php/BLIP_Tutorial)