

Cahier des charges : Logiciel de communication capteur - base

Hexanome 4111 :

Quentin CALVEZ, Matthieu COQUET,
Jan KEROMNES, Alexandre LEFOULON,
Thaddée TYL, Xavier SAUVAGNAT,
Tuuli TYRVAINEN

Janvier 2012

Table des matières

1	Introduction	2
1.1	Présentation du projet	2
1.2	Présentation du document	2
1.3	Documents applicables / Documents de référence	2
1.4	Terminologie et abréviations	2
2	Présentation du problème	2
2.1	Buts, nature du logiciel, utilisateurs concernés	2
2.2	Formulation des besoins, exploitation et ergonomie, expérience	3
2.3	Portée, développement, mise en oeuvre, organisation de la maintenance	4
2.4	Limites	4
3	Exigences fonctionnelles	4
3.1	Fonctions de base, performances et aptitudes	4
3.2	Contraintes d'utilisation	4
3.3	Critères d'appréciation de la réalisation effective de la fonction	5
4	Exigences non fonctionnelles	5
5	Contraintes imposées, faisabilité technologique, moyens	5
5.1	Sûreté, planning, organisation, communication	5
5.2	Complexité	5
5.3	Compétences, moyens et règles	5
6	Configuration cible	5
6.1	Matériels et logiciels	5
6.2	Stabilité de la configuration	6
7	Guide de réponse au cahier des charges	6
8	Annexes	6
8.1	Protocole ZigBee	6

1 Introduction

1.1 Présentation du projet

Le COPEVUE a lancé un appel d'offre dans le cadre de la réalisation d'un système de monitoring de sites isolés. Il s'agit donc de concevoir en premier lieu une solution technique permettant de répondre au mieux aux exigences fonctionnelles et non fonctionnelles que le COPEVUE formule. De façon synthétique notre équipe va proposer une solution permettant de surveiller des sites naturels difficiles d'accès (souvent à cause des conditions environnementales) et peu peuplés. Dans ces sites isolés sont souvent regroupés des postes de travail et ces zones doivent pouvoir être surveillées en dépit de la distance qui les sépare du bureau de contrôle.

1.2 Présentation du document

Ce dossier présentent les besoins nécessaires par rapport à la solution d'acquisition des données des capteurs, et de la transmission de celles-ci avec la base de la station. Nous essayerons tout d'abord de voir quels sont les objectifs de cette solution technique, les utilisateurs concernés et les problèmes pouvant se présenter, autant dans son développement, sa mise en oeuvre, et enfin sa maintenance. Ce document est accompagné d'annexes permettant d'approfondir certains sujets plus techniques sur lesquels nous n'avons pas voulu nous attarder dans un but de clarté.

1.3 Documents applicables / Documents de référence

1.4 Terminologie et abréviations

- Base (de la station) : système informatique équipé d'un OS linux permettant de communiquer avec les capteurs et le serveur central. La base peut aussi enregistrer quelques données et permet de configurer la plupart des éléments de la station.
- Mode sommeil : mode dans lequel le microcontrôleur et le capteur ne consomme que très peu d'énergie. Aucune acquisition et aucun calcul ne doivent être fait pendant cette période. Le micro-contrôleur attend d'être réveillé par un événement matériel ou logiciel.

2 Présentation du problème

2.1 Buts, nature du logiciel, utilisateurs concernés

Le logiciel qui doit être implémenté a deux fonctions :

- L'acquisition des données des capteurs à un intervalle de temps définie préalablement,
- L'envoi de ses données à la base de la station à un intervalle de temps régulier. Pendant cette connexion, la configuration des capteurs par la base devra être possible. Ainsi, l'écart entre chaque acquisition ou transmission devra être paramétrable. Cette configuration sera faite pendant la transmission avec la base.

Ce composant logiciel devra ainsi interagir avec la base de la station, lui envoyer les données et vérifier si une reconfiguration est nécessaire. Il devra également se charger de récupérer la valeur du capteur auquel il est attaché. Ce composant logiciel devra impérativement être économes et devra passer la grande majorité du temps en mode "sommeil".

D'un point de vue plus technique, ce logiciel devra s'exécuter sur un contrôleur attaché à une puce ZigBee, au capteur et à la batterie longue durée.

Le protocole de communication utilisé par les modules de transmission avec la base sera décrit plus loin dans ce document.

Ce document va aussi porter sur le matériel exploité par le logiciel précédemment cité, à savoir le module de communication ZigBee. Les attentes par rapport à celui-ci seront également décrites.

2.2 Formulation des besoins, exploitation et ergonomie, expérience

Le système présentement présenté doit répondre à deux besoins bien distincts, et ce quelque soit la localisation géographique du système dans le monde :

Acquisition Le système doit pouvoir recuperer la valeur des capteurs de la façon suivante :

- Le micro-contrôleur sera reveillé selon la periode d'acquisition, il relevera la valeur du capteur
- Selon la configuration de la periode de transmission, soit la donnée du capteur est en enregistré temporairement, soit directement envoyé à la base
- Le micro-contrôleur devra s'adapter au capteur pour pouvoir gerer les differents types d'acquisitions
- Si besoin, le niveau de la batterie doit également être verifié
- Le module doit être capable de se reconnecter à la base en cas de déconnexion
- En cas de situation critique (batterie ou capteur), les données necessaire à l'analyse de cette état devront être transmis à la base le plus tôt possible

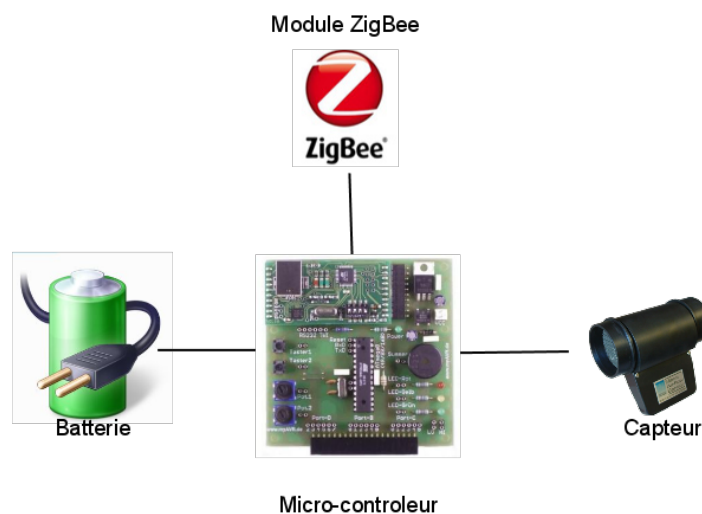
Transmission Pendant la transmission, le système doit être capable d'analyser les données et d'envoyer les informations suivantes à la base :

- Les informations nécessaire du capteur
- Au besoin et en cas de situation critique, le niveau restant de la batterie doit être envoyé
- La date du prochain reveil et eventuellement les informations utiles du micro-contrôleur

Pendant la transmission, le système doit être capable de recevoir des données correspondant à la reconfiguration du module :

- Les différents quantum de temps définissant la granularité avec laquelle le système transmet les résultats a la base, en analysant les données des capteurs.
- La configuration du capteur doit être possible.

Voici un schéma resumant les différents composants du module d'acquisition :



2.3 Portée, développement, mise en oeuvre, organisation de la maintenance

L'application devra être programmée en utilisant le langage C. Des tests unitaires et fonctionnels devront être implémentés.

Les différents éléments (programme et tests) seront documentés, afin de faciliter la maintenance du système et de permettre des futures améliorations. Cette documentation devra contenir les différentes difficultés et erreurs rencontrées lors de la phase de développement.

2.4 Limites

Le logiciel devra se limiter aux fonctions décrites dans ce document. L'autonomie, la simplicité et la fiabilité du système sont primordiales. Toute fonctionnalité non décrite sera inutile.

3 Exigences fonctionnelles

3.1 Fonctions de base, performances et aptitudes

Le module d'acquisition doit sortir du mode "sommeil" à un intervalle de temps régulier définie en configuration. Le système doit rester le moins de temps possible en mode "éveillé" pour respecter les contraintes d'autonomie. L'acquisition du capteurs se fera via une interface. L'implémentation de cette interface ne doit pas être programmée et fera l'affaire d'une autre étude. Cette interface, respectée par tous les types de capteurs, sera décrites plus loin.

L'acquisition de la batterie devra être effectuée régulièrement à un intervalle de temps définie par la configuration. Une interface similaire à celle du capteur sera utilisée pour repérer le niveau de batterie.

Après l'acquisition, une tentative de connexion devra être effectuée avec la base de la station. Si celle-ci réussit, les données des capteurs et de la batterie sont transmises. Sinon, elles sont stockées temporairement. Ce stockage doit être étudié précisément et ne doit pas être effectué si la consommation devient trop importante. Les données seront conservées dans les cas où la situation est critique.

Pendant la transmission, la configuration doit être possible. Les points suivants peuvent être modifiés :

- La valeur du capteur en cas de mauvais réglage de celui-ci,
- La période d'acquisition des données du capteur,
- La période d'acquisition des données de la batterie,
- La période de transmission, dans la plupart des cas cette période sera égale à la période d'acquisition des données du capteur.

3.2 Contraintes d'utilisation

Le système ne doit pas pouvoir être utilisé par un utilisateur humain. La sécurisation des données envoyées est à prévoir, il ne faut pas qu'un utilisateur externe puisse intercepter les données ou interagir avec le système. Cette interaction pourrait mener à des dysfonctionnements du système. Un système de cryptage, tel qu'un certificat, ou une autre méthode de sécurisation doit être mise en place. Il faut prévoir qu'une opération de maintenance sur site a un coût élevé, le système doit être réfléchi pour limiter au maximum les dysfonctionnements nécessitant un déplacement.

3.3 Critères d'appréciation de la réalisation effective de la fonction

Les différents critères seront pris en compte pour juger le système. A ces critères sont ajoutés la nécessité et le taux de difficulté estimé. Ces valeurs vont de 0 à 10, 0 étant la valeur la plus basse :

Critère	Nécessité	Difficulté
Sécurisation du protocole de transfert	7	6
Dépense énergétique	9	5
Stabilité	9	7
Facilité de la maintenance du code	8	4
Clarté de la documentation	8	4

4 Exigences non fonctionnelles

5 Contraintes imposées, faisabilité technologique, moyens

5.1 Sûreté, planning, organisation, communication

Le protocole utilisé est le ZigBee, qui à l'avantage d'être simple d'utilisation. Cependant, il est nécessaire de prévoir un temps d'apprentissage de cette technologie si nécessaire.

La communication avec la maîtrise d'ouvrage peut être envisagé et sera si nécessaire organisé par la maîtrise d'oeuvre. Le planning devra également prendre en compte une période de test.

Voici le planning prévisionnel du projet :

Phase	Durée prévisionnelle
Spécification Code et Tests	1 mois
Développement Code et Tests	3 mois
Tests intégration avec le système entier	2 mois
Finalisation et reprise des éléments	2 mois
Total	8 mois

5.2 Complexité

Le système envisagé ne demande pas une complexité importante. Le protocole ZigBee étant largement documenté, le reste du système reste simple. Le fonctionnement de ce module avec le reste du système doit être pris en compte. La principale difficulté reste la stabilité du système, celui ci doit absolument être fiable.

5.3 Compétences, moyens et règles

Les compétences techniques suivantes sont exigées :

- Expertise en système embarquée
- Programmation bas niveau
- Connaissance de protocole réseau
- Sécurisation réseau

Ce projet nécessite peu de ressource matériel, l'équipe devra être équipé de technologie pour programmer et debugger la carte choisie.

6 Configuration cible

6.1 Matériels et logiciels

Le matériel utilisé pour développer le module est le suivant :

Pour le contrôleur qui a la responsabilité de gérer le capteur, le ZigBee et le niveau de batterie, il a été décidé d'utiliser une carte FOX Board LX832.

Pour le ZigBee, une carte de type EB051 sera utilisée, la puce ZigBee est directement intégrée au circuit de la carte. Cette carte permet une utilisation simplifiée des protocoles ZigBee. Si le capteur n'en est pas doté, un micro-contrôleur sera couplé au capteur. Ce cahier des charges ne prend pas en compte cette partie. Un pilote sera développé par une autre équipe afin de coupler le capteur à la carte. L'utilisation de plusieurs cartes rend l'implémentation plus facile. Une attention au moment du développement permettra une intégration simplifiée entre les systèmes.

6.2 Stabilité de la configuration

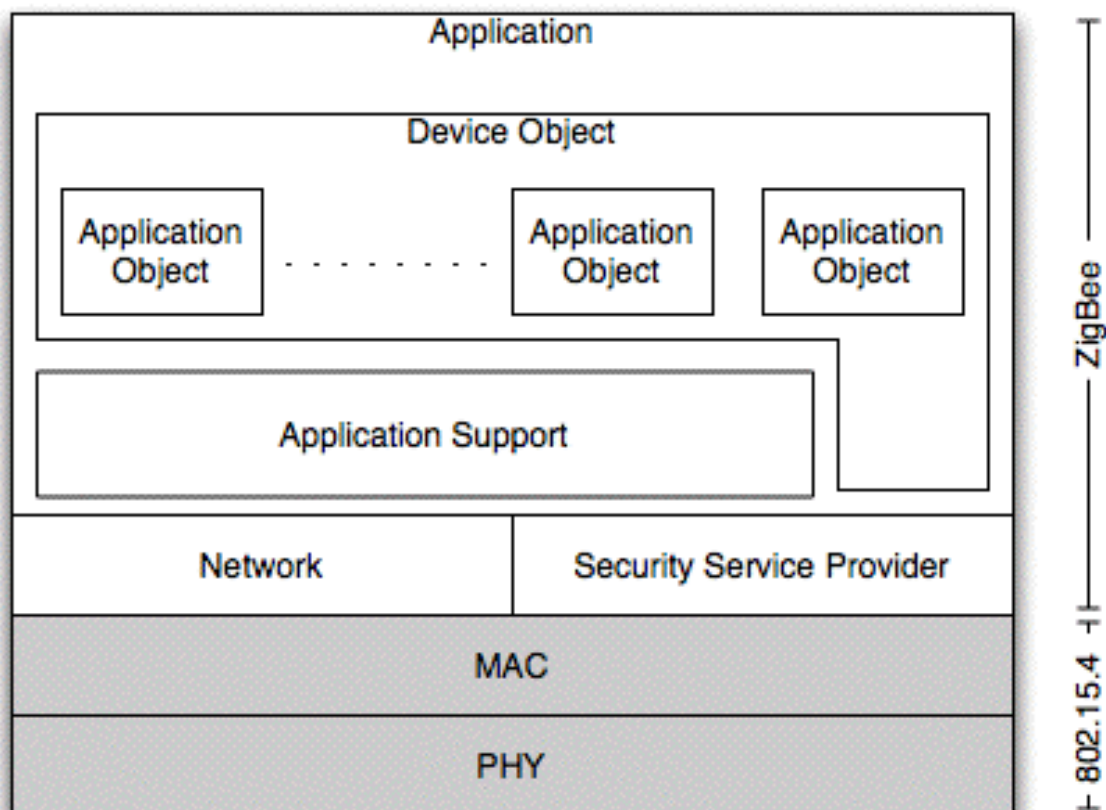
La maintenance du système étant naturellement très délicate, il sera nécessaire d'obtenir une version du logiciel très stable et ne présentant que peu de défauts. Comme précisé précédemment, une attention particulière sera donnée aux phases de tests.

7 Guide de réponse au cahier des charges

La réponse au cahier des charges se fera particulièrement sur la documentation, notamment au niveau des tests. Une première vérification sera ainsi effectuée pour confirmer l'état du projet. Une fois cette étape validée, une prise en main et vérification complète du système sera effectuée.

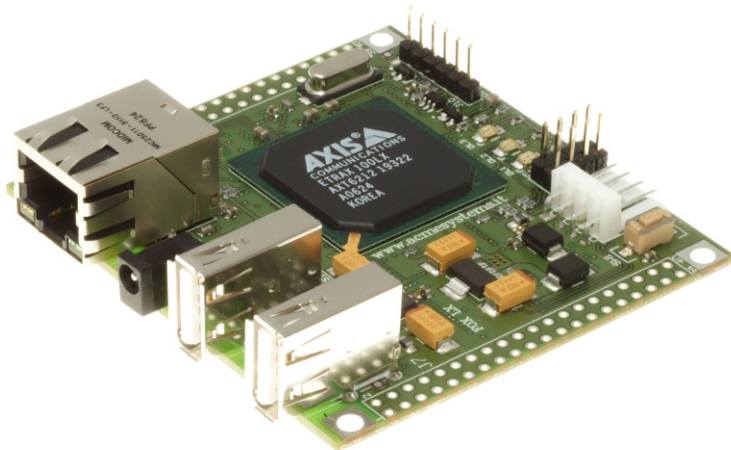
8 Annexes

8.1 Protocole ZigBee



FOX Board

A LINUX CORE ENGINE IN JUST 66x72mm



The FOX Board is an Embedded Linux Core Engine with reduced size and low power requirements especially designed to enable an easy integration of your design with standard LAN and Internet WEB protocols and applications.

Two slots for 20X2 pins strip step 0,1" (2.54mm) can be used to connect the FOX Board with your application specific board to transform it in an Embedded Linux product ready to go on Internet.

FOX Board reduces at a minimum the design effort and the time to market for devices that need to have WEB interface, multitasking, USB devices management, storage of data and all advantages that come from the use of a modern and advanced Operating System.

A completely free and Open Source Software Development Kit enables a strong customization of Linux Kernel to tailor it for the best integration with your application.

A strong international community of developers support answers quickly to any technical question and offers huge possibilities to find professional and FOX skilled designers able to help or develop for you your project based on FOX Board.



<http://www.acmesystems.it>

TECHNICAL FEATURES

Microprocessor: Axis ETRAX 100 LX
100MIPS, 32bit, RISC architecture

Memory: up to 64MB of RAM and up to 32MB of FLASH (*)

Interfaces: 1 Ethernet 10/100Mb port, 2 USB host 1.1 ports, slots for two 20x2 pin strip step 2.54 mm with up to 48 I/O lines, I2C bus, SPI, serial and parallel ports (**)

Power supply: 5 Volt, 1 Watt.

Operating System: Linux Kernel 2.4.x e 2.6.x

Development kit: Available for free under GPL license.

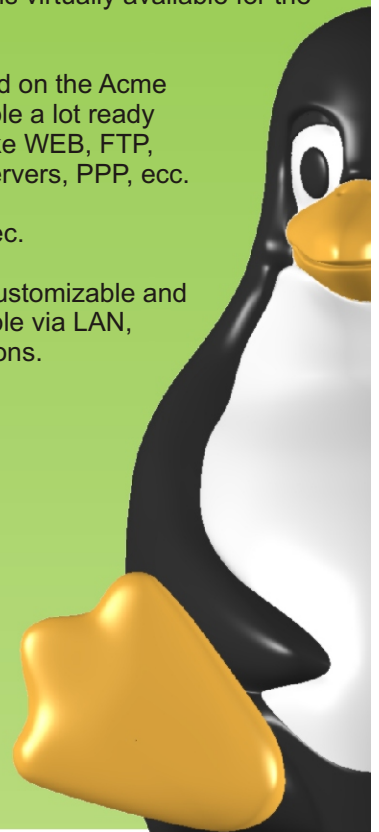
Programming languages: C, C++, Java, PHP, Python, TCL, etc.

Available software: All the Open Source software for Linux is virtually available for the FOX Board.

Inside the SDK and on the Acme Systems is available a lot ready to use software like WEB, FTP, Telnet and SSH servers, PPP, etc.

Bootstrap: <10 sec.

Firmware: Fully customizable and remotely upgradable via LAN, Web or FTP sessions.



(*) The basic memory config is 16MB of ram and 4MB of flash.

(**) Not all the lines are available at the same time.

This document is subject to change without prior notice.

The term "Linux" is a registered trademark of Linus Torvalds, the original author of the Linux kernel. Linux Penguin, "Tux", was created by Larry Ewing.