Data collection and management solution for wireless sensor networks

G. De Sousa, J.P. Chanet, A. Jacquot, D. Boffety, G. André, K.M. Hou

Data Collection and Management Solution for Wireless Sensor Networks

G. De Sousa[1], J.-P. Chanet[1], A. Jacquot[1], D. Boffety[1], G. André[1], K.-M. Hou[2]
[1]Cemagref, UR TSCF, 24 Avenue des Landais, BP 50085, 63172 Aubière cedex, France
[2]LIMOS UMR 6158 CNRS, Complexe scientifique des Cézeaux, 63173 Aubière cedex, France
Correspondent author: gil.de-sousa@cemagref.fr

Abstract
Wireless sensors networks (WSN) use can be very interesting in agricultural and environmental data collection. The first WSN generations operated in a continuous data stream mode which generates high energy consumption. This article presents a new WSN platform that limits data exchanges and has an increased lifetime. All of its components are designed in a resource aware mode in relation with energy, memory and processing. This platform is built on wireless sensors, implementing a hardware component-based concept, that allow them to be combined to form a more evolved wireless device. To manage this wireless sensor, a hybrid operating system, both multithreading and based on events, has been developed and is associated to a micro-file system. A WSN management tool is in charge of monitoring the wireless sensors and of the data collection. A first evaluation of this WSN platform has been realised in an agricultural context data collection.

Keywords: Wireless Sensor Network (WSN) platform, environmental data collection

1. Introduction

The current emergence of Wireless Sensor Networks (WSN) research topic is based on the recent advances in embedded technologies and on the various possible applications. The development of miniaturized wireless access medium has, for example, enabled the implementation of more integrated wireless sensor node. Moreover, with the availability of this kind of devices, new applications have been defined even simply imagined. WSN has then been used in military, industrial, transport and “smart care, smart home” applications (Stojmenovic 2005).

Naturally, agricultural and environmental WSN applications are also numerous and varied. With regard to this last point, rather than develop a WSN solution for a specific application, our approach aims at the design of a complete versatile and adaptive solution dedicated to agricultural and environmental data collection. Thus, this article presents a new WSN platform that integrates different components designed, based on resource-aware concept, and being able to adapt to diverse applications constraints. The following sections describe, in this order, the LiveNode (LIMOS Versatile Embedded Node) wireless sensor node, the LIMOS (LIghtweight Multithreading Operating System) kernel, the LiveFile (LIMOS Versatile Embedded File system) system and the LiveNCM (LiveNode Non invasive Context-aware and modular Management tool) tool that form the WSN platform proposed. The penultimate section shows some examples of application of this WSN platform. The last section concludes this article.

2. The LiveNode wireless sensor node

Classical wireless sensor node has the following components:

- A microcontroller.
- A wireless access medium.
- Different types of sensor(s).
- A memory or storage unit.
- A power or energy supply.

Then, in the same way, the LiveNode wireless sensor node is based on an ARM7TDMI ATMEL microcontroller AT91SAM7S256. The main characteristics of this microcontroller are a variable processor frequency from 50 Hz to 50 MHz, 256 KB of Flash and 64 KB of SRAM memories, different interfaces such as RS232, USB and I²C. This sensor node can be equipped with an IEEE 802.11 Wi-Fi or an IEEE 802.15.4 ZigBee wireless communication module, connected to the microcontroller through a RS232 serial port. The energy used to power this wireless sensor node can be provided by standard AA battery or by a photovoltaic.

The LiveNode has a modular architecture which enables to implement a hardware component-based concept. Indeed, several LiveNodes can be combined to form a more evolved wireless sensor node. For example, two LiveNodes using different wireless access medium (e.g. Wi-Fi and ZigBee) can be grouped together to realise either a wireless gateway or a multi-support module (Fig. 1). This latter is part of existing Quality of Service (QoS) methods in communication. If one of the technologies is unavailable, the other can be used in order to transmit data successfully.

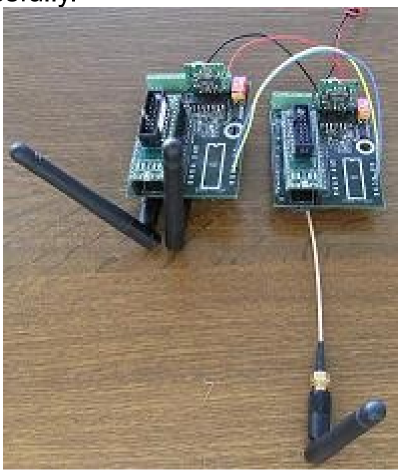

Fig 1: The combination of two LiveNode wireless sensor nodes

Like other existing wireless sensor nodes, the LiveNode has limited resources in terms of energy, memory and processing power. Moreover, these resources are closely linked. For example, energy can be saved by reducing the size of sending message which may be achieved either by storing or by compressing some part of the collected data. Then, all the software embedded in a wireless sensor node have to be designed by taking into account the constraints generated by the lack of available resources.

3. The LIMOS operating system

In wireless sensor network (WSN), there are three families of operating systems:

- Multi-threading (multitask) operating system.
- Event-driven operating system.
- Hybrid operating system.

The first one included operating system using classical multi-threading functioning mode. MANTIS (MultimodAl system for Networks of In-situ wireless Sensors) OS is part of this family (Bhatti et al. 2005).

In the second group, the operating mode is based on the event notion. The main difference between thread and event is that this latter can be pre-empted. In other words, event can be considered as a single task which runs to completion. However, both thread and event can be interrupted. TinyOS is the most representative of the existing event-driven operating system (Hill et al. 2000).
Hybrid operating systems try to combine multi-threading and event-driven properties.

Nevertheless, most of these kinds of operating systems are not natively hybrid. For example, Contiki is natively event-driven and can propose a multi-threading behaviour thanks to an optional library (Dunkels et al. 2004).

In our purpose to build a versatile WSN platform, a natively hybrid operating system called LIMOS has been designed. In this kernel, each thread is associated to an event which allows three operating modes: multi-threading, event-driven and a combination of the two others (Fig. 2). LIMOS kernel used also a two-level scheduling policy based on an event manager and a thread scheduler. In the multi-threading mode, all the threads are associated to the same event. In the event-driven mode, each event has only one thread.

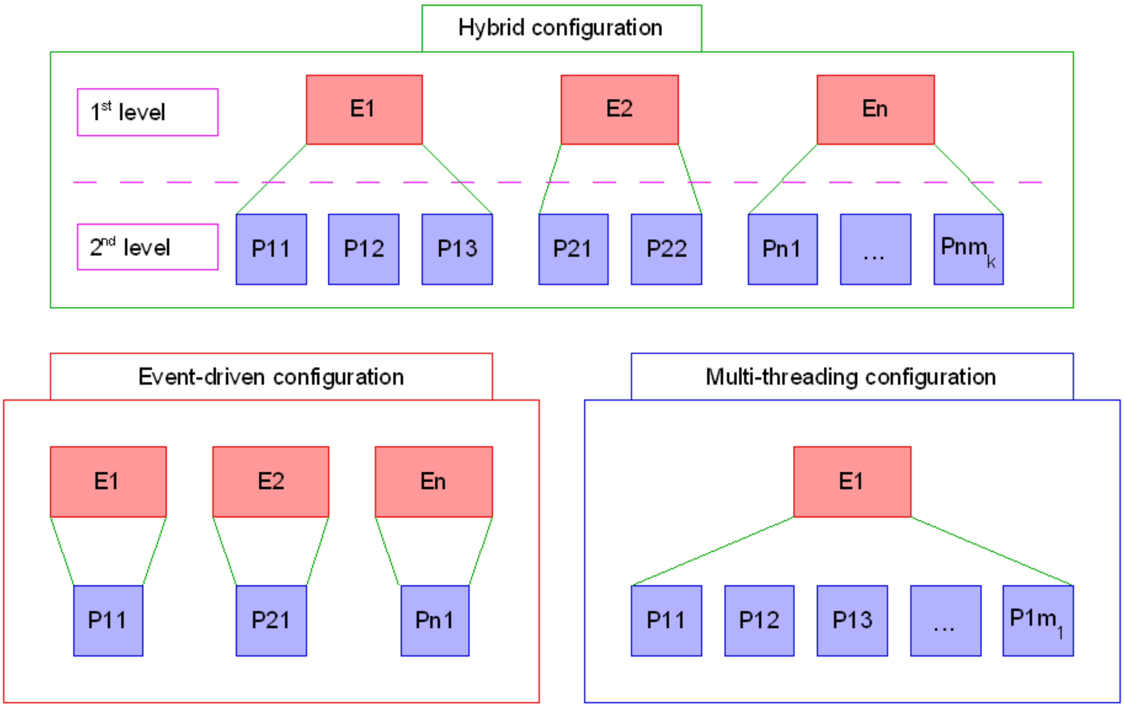

Fig. 2: LIMOS operating mode

4. The LiveFile micro-file system

The main purpose of the LiveFile system is an intelligent use of the non volatile memory embedded in a wireless node. It has been elaborated from the study of different memory flash management systems, dedicated to WSN, such as Capsule (Mathur et al. 2006). It provides different preservation mechanisms and data formats for the storage.

Flash memory is divided into pages. Each page allows a restricted and limited number of writings. This limit varies from 10,000 to 1,000,000 depending of the flash memory technology. For example, the microcontroller AT91SAM7S256 has an internal flash memory divided into 1024 pages of 256 bytes which support 10,000 writings.

Then, preservation mechanisms are essential in order to increase memory lifetime. The first mechanism found on LiveFile and also in other systems is the use of RAM Buffers in order to make page-size writings (Fig. 3). Then, effective writing only occurs when the size of the data to store is equal to a given threshold. Memory is more preserved than this threshold is closed to the page size. The second mechanism is based on the principle of balancing the distribution of the writing load between the different pages. It aims at the preservation of the most used pages and a wear evenly distributed of the memory. To implement such functionality, the number of writings of each page is necessary. Then, a header is integrated into the beginning of each memory page and contains different kind of information about it such as:

- The state.
- The writing number.
- The free space.
- The type of stored data.

About this latter attribute, to reduce data management complexity, each page only stores data of the same format. Three formats are available in the LiveFile system: “RECORD”, “FILE” and “CHECKPOINT”. The first one is highly adapted to the supported application with a definition of its size and its type according to the kind of collected data. The “File” format is used to store data stream with variable size. Finally, information useful to the recovery process after an eventual breakdown are stored thanks to the “CHECKPOINT” format.

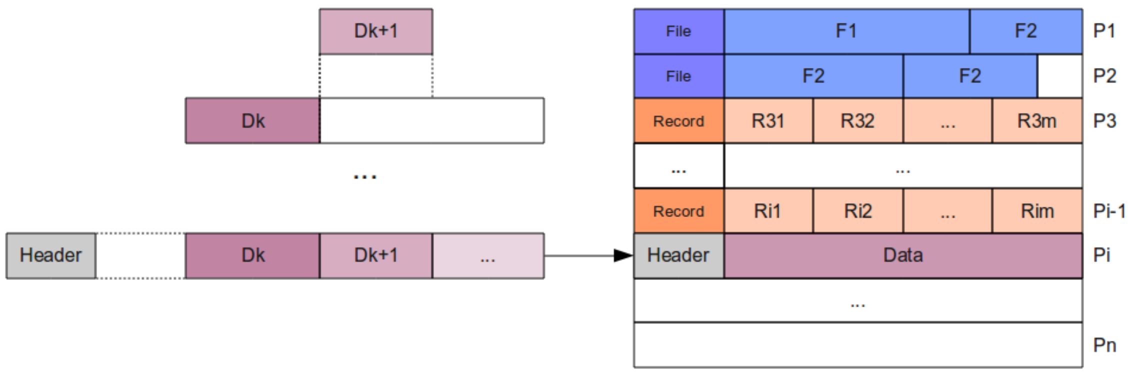

Fig 3: Memory management in the LiveFile system

5. The LiveNCM management tool

In order to manage an important number of wireless sensor node deployed during an application, a Wireless sensor network Management Tools (WMT) is required. Thus, a new WMT called LiveNCM has been designed to achieve this task in the context of the WSN platform presented in this article. The LiveNCM management tool integrates an extension of the SNMP (Simple Network Management Protocol) protocol by taking into account WSN constraints. More precisely, the LiveNCM tool consists of four main elements (Fig. 4):

- A SNMP Server with a sub-agent extension.
- A wireless gateway.
- A distributed component embedded on each LiveNode wireless sensor node.
- A supervisor.

Classical functionalities of a SNMP, though the implementation of an agent, are the storage of management data and to answer user requests. In the case of the LiveNCM tool, a sub-agent, with its own data structure called MIB (Management Information Base), has been developed to complete the SNMP agent. The SNMP Server uses a dedicated wireless gateway to communicate with a given WSN. Various WSN can be managed by the SNMP server. To answer to the requests sent by the SNMP server, LiveNodes are equipped with a specific software component. Collected data and information about the functioning state of the nodes can be consulted using a supervisor that integrates an end user interface. This supervisor is based on a database which takes its information from the SNMP MIB.

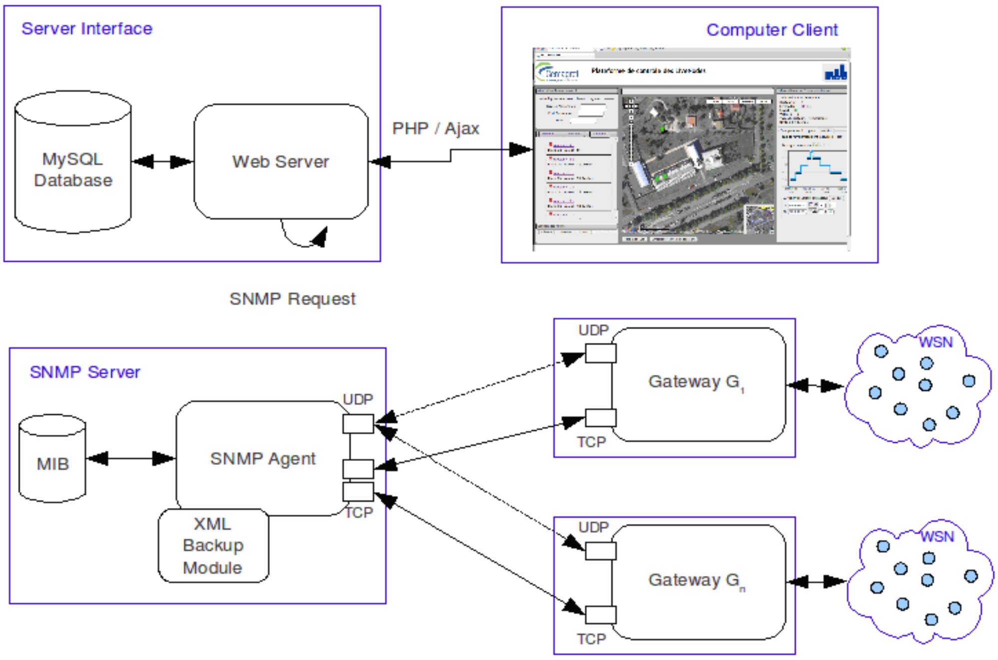

Fig 4: The LiveNCM tool

6. Applications

This WSN platform has been evaluated in different research projects. First main project was in the context of an industrial collaboration and aimed at the development of a wireless solution to collect temperature and soil moisture data. Second representative application was during a European research project called NeT-ADDeD (http://www.netadded-project.eu/) or "New Technologies to Avoid Digital Division in e-Divided areas" (Fig 5).

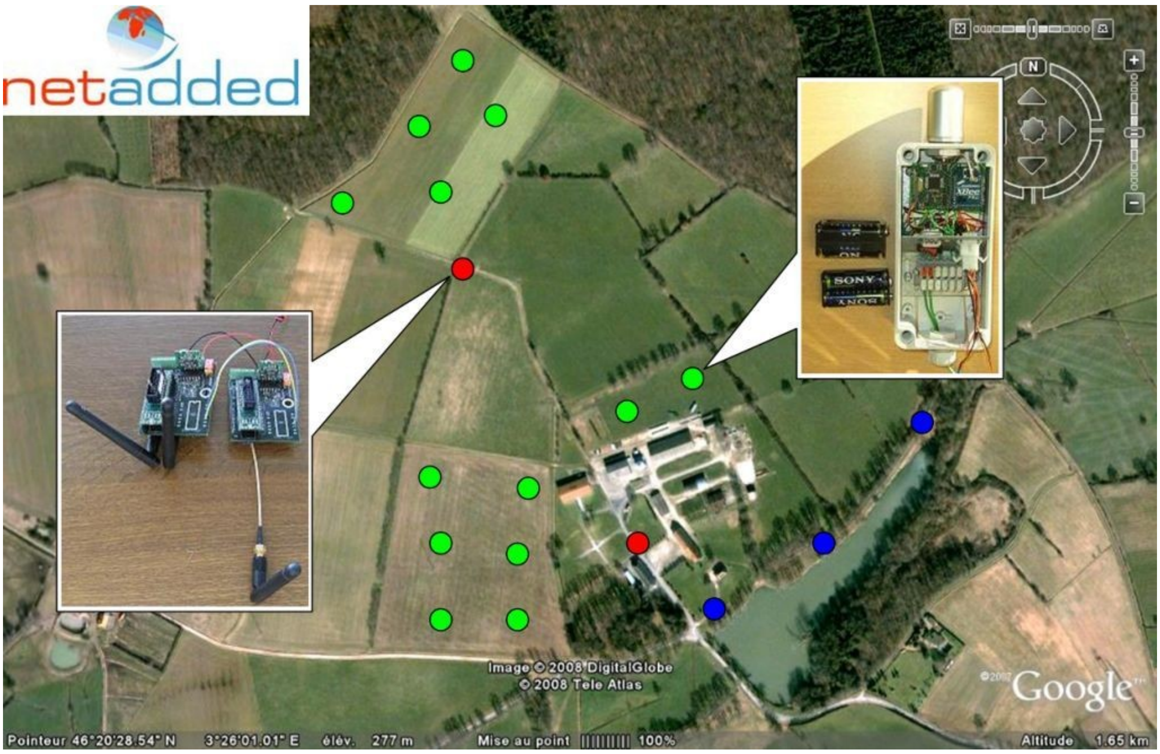

Fig 5: Deployment of the WSN platform during the NeT-ADDeD project

The purpose of this project is the development and the validation of methods or techniques allowing the fast and accurate deployment of different wireless technologies to ensure communication between isolated areas. Then, our WSN platform was integrated in a wireless communication infrastructure installed in an isolated farm.

Conclusion

This article introduces a new WSN platform and its components. These platform is based on a wireless sensor node LiveNode which resources are managed by an operating system LIMOS and a micro-file system LiveFile. This latter operates essentially on the non volatile Flash memory. Administration of the sensor nodes and collected data storage is obtained thanks to the LiveNCM management tool. This platform has been designed and implemented gradually until now. It reaches a level of performance that allows us to evaluate it in real experimentations associated to different research projects.

References

Bhatti S., Carlson J., Dai H., Deng J., Rose J., Sheth A., Shucker B., Gruenwald C., Torgerson A., Han R. (2005). MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms. *ACMKluwer Mobile Networks & Applications (MONET) Journal, Special Issue on Wireless Sensor Networks, Vol. 10, Issue 4*.

Dunkels A., Grönwall B., Voigt T., (2004). Contiki – a Lightweight and Flexible Operating System for Tiny Networked Sensors. *1st IEEE Workshop on Embedded Networked Sensors (IEEE EmNetS-I)*.

Hill J., Szewczyk R., Woo A., Hollar S., Culler D., Pister K. (2000). System Architecture Directions for Networked Sensors. *9th International Conference on Architectural Support for Programming Languages and Operating Systems, vol. 35, no. 11,* 93-104.

Mathur G., Desnoyers P., Ganesan D., Shenoy P., (2006). Capsule: An Energy-Optimized Object Storage System for Memory-Constrained Sensor Devices. *4th ACM Conference on Embedded Networked Sensor Systems (SenSys'06)*, 195-208.

Stojmenovic I. (2005). *Handbook of Sensor Networks: Algorithms and Architectures*. Wiley Series on Parallel and Distributed computing.