A WSN platform to support middleware development

André Rodrigues
Department of Informatics Engineering of the University of Coimbra
Pólo II, Pinhal de Marrocos, 3030-290, Coimbra, PORTUGAL
Tel.: +351 239790000 Fax.: +351 239701266
Email: andre@iscac.pt

ABSTRACT
According to the application domain, Wireless Sensor Networks (WSNs) differ in a number of ways (e. g., deployment strategy, node mobility, available resources, node heterogeneity, topology, sensor coverage, Quality of Service (QoS), network size, lifetime, connectivity). This diversity is an obstacle to the development of a standard platform (hardware / software) that eases the task of building sensor networks applications. In this paper we present the design principles of a middleware solution that can be adapted to different application scenarios. We also present a mobile sensor and actuator platform that has been developed at the University of Coimbra in order to evaluate the performance of middleware solutions.

Categories and Subject Descriptors
C.2.1 [Network Architecture and Design]: Wireless Sensor Networks.

General Terms
Algorithms, Performance, Design, Experimentation, Languages.

Keywords
Wireless sensor network, middleware and platform.

1. INTRODUCTION
Programming wireless sensor nodes using operating systems like TinyOS or Contiki is a difficult task that needs special expertise and requires a difficult learning curve. It would be easier if the developer had a programming environment where he would specify the application functionality / requirements and an automatic tool to generate and deploy the code for sensor nodes. This tool would select the best components that fit the application functionality / requirements and, after the deployment phase, would monitor the performance and adjust configuration parameters. In Section 2 this paper presents the features of a middleware solution that can be adapted to different application scenarios. Section 3 describes a mobile sensor and actuator platform that has been developed in order to evaluate the performance of middleware solutions. Conclusions and topics for further work are described in the last section.

CoNEXT '06, December 4–7, 2006, Lisbon, Portugal.

2. MIDDLEWARE FEATURES
There are few classifications of WSN middleware [1, 2, 3]. In [1], middleware solutions are grouped as:

Classic - solutions that offer abstractions for communication primitives, paradigms and application requirements but are somewhat limited in QoS, mobility and adaptability requirements (e.g. Impala, the middleware from the Zebranet project that studies the migration patterns of wild animals, is an example);

Data centric – the network is abstracted as a database with queries based on some form of SQL-based language enhanced to support energy efficiency and WSN specific requirements. Adaptability, reconfiguration and QoS are not well supported (TinyDB is the classic example);

Virtual machines – each node offers the services of a virtual machine that executes a high-level language. This approach is flexible, energy-efficient at the communication level and supports security at code level. However, it presents overhead in processor time and higher energy consumption due to the virtual machine execution (e.g. Maté);

Adaptable – the idea is to support adaptability. At the design phase the application requirements are stated and the system chooses the components that best fit those requirements. Then, during run-time, the middleware monitors the environment and adjusts the configuration to meet the specifications. Flexibility, cross-layer optimization and QoS support are advantages, but the complexity may not be acceptable for some environments.

The middleware solution that is being developed at Laboratory of Communications and Telematics of the University of Coimbra supports different application scenarios, such as home automation, healthcare and traffic control. In order to enable the development of applications with such diverse requirements, the proposed middleware focus on adaptability. The key concept is to support high-level abstractions based on a set of available components (rated according to their characteristics) that are selected at design phase according to the application functionality / requirements and may be changed in runtime to react to changes in the environment / nodes. Other design principles to be supported are: energy-efficiency (making use of cross-layer information), QoS awareness, security in communications, topology diversity, hardware heterogeneity, nodes and users scalability, mobility, Internet access at the network / node level probably using IPv6 (uIP and 6LowPAN are examples of projects in this area), and development environment integration (to ease the application development by non experts).

3. PLATFORM DETAILS

The motivation for developing mobile platforms was to enable the study of the main challenges in applications development for WSNs and to support the evaluation of the proposed solutions.

The challenge was to design a platform that provides mobile capabilities to existing Embedded Sensor Board (ESB) sensor nodes (at the same time being directly applicable to other architectures), allows nodes to communicate and cooperate in the study of phenomena, is autonomous from the observer, is programmer-friendly, supports debugging capabilities and, last but not least, is inexpensive.

We developed two platforms: the first one is based on a commercial robot that was modified to enhance its sensors and mobile capabilities, and to receive commands from ESB nodes. We want to study how to reuse, as much as possible, the some middleware components in both platforms. The commands to control the platform are a subset of the available Robomote [4] commands. Table 1 presents the robot hardware characteristics and Figure 1 depicts the platform hardware architecture.

Table 1. Robot hardware characteristics

Power supply	1 – 9V battery (electronics) 4 - 1,5V AA batteries (motors)
Odometry error correction	PI controller (not finished)
Traversible Terrains	Floor
Microcontroller	PIC 16F876
Memory	14KB Flash, 368B RAM, 256B EEPROM
Programming	rs232 based
Sensors[1]	Light, ultrasound, bumpers, microphone, optical encoders
Actuators	Motors, beeper, leds and claw

The second platform was based on a cheap solution (around 15€ per radio controlled (RC) vehicle). The first approach was to not use the radio electronics in the vehicle. The ESB node was connected directly to the motor drivers using the same hardware already developed for the robot. But that option was not only a waste of capabilities but also only allowed the control of the car by the sensor network and not in an autonomous way. In this scenario, some studies would not be possible (i. e. the study of the impact of car control in the WSN performance would not be possible). The chosen architecture is presented in Figure 2. In this model commands can be selected simultaneously from the car RC control or from the ESB. All the commands are recorded at the platform and they can be used for debugging or for playback purposes in a record / play scenario. The interface between the car and the ESB supports the commands *front [speed]*, *back [speed]*, *left*, *right*, *center*, *show log*, *play log* and *stop*. For now, the car RC control accepts 10 different speeds.

[1] ESB hardware not included. The ESB is based on an MSP430F149 microcontroller, a TR1001 radio and it has sensors for light, PIR, vibration, temperature and microphone.

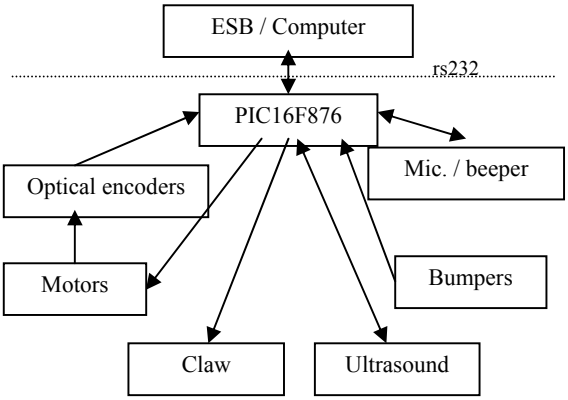

Figure 1. Robot hardware architecture

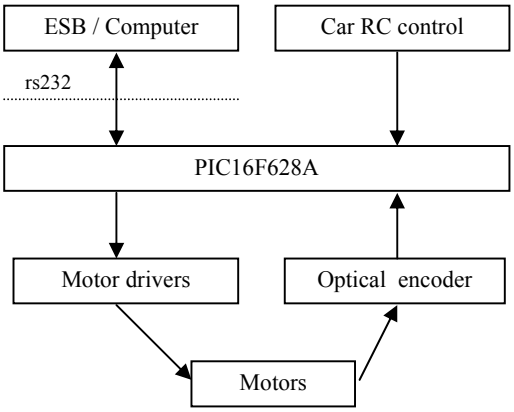

Figure 2. Car hardware architecture

4. CONCLUSIONS AND FUTURE WORK

This paper presented the characteristics that we considered in the development of our WSN middleware solution and a platform that will be used in the tests. We have already integrated the platform with the node so that they advertise themselves as a mobile sensor/actuator node. The next steps will be the extensive testing and refinement of the middleware solution.

5. REFERENCES

[1] Pedro, J. M. Middleware Approaches for Sensor Networks. *Summer School on WSN and Smart Objects* (Schloss Dagstuhl, Aug 29 – Sept 3, 2005).

[2] Kirsten, T., and Jochen, S. Driving Forces behind Middleware Concepts for Wireless Sensor Networks. In *Proceedings of the REALWSN Workshop* (2005).

[3] Kay, R. Programming Paradigms and Middleware for Sensor Networks. *GI/ITG Fachgespraech Sensornetze* (Karlsruhe, Feb 26-27, 2004).

[4] Gabriel, T. S., Mohammad H. R., and Gaurav S. S., Robomote: A Tiny Mobile Robot Platform for Large-scale Ad-hoc Sensor Networks. *In International Conference on Robotics and Automation* (2002), 1143-114