

# An Inexpensive, Off-The-Shelf Platform for Networked Embedded Robotics

M. Bordignon<sup>1</sup>, E. Pagello

Intelligent Autonomous Systems Laboratory  
Dept. of Information Engineering, University of Padova, Italy  
mirko.bordignon@iee.org epv@dei.unipd.it

A. Saffiotti

AASS Mobile Robotics Laboratory  
Dept. of Technology, University of Örebro, Sweden  
alessandro.saffiotti@aass.oru.se

**Abstract**—Recent years have witnessed the proliferation of a new class of devices, commonly referred to as Networked Embedded Devices. Their increasingly low cost and small size make them suited for large scale sensing applications. Likewise, they could be appealing as a means to embed intelligent actuation capabilities into the environment, turning simple artifacts into networked robotic appliances. The currently existing devices, however, are not suited for this development. In this paper, we present the PEIS-Mote: an open, general, small-size and inexpensive sensor-actuator node especially suited for networked robotics, and built from commonly available off-the-shelf components. This platform can run a popular operating system for sensor networks, TinyOS, which makes it interoperable with most commercially available sensor nodes.

## I. INTRODUCTION

Following a vision depicted during the 90's [1], a great deal of research has been carried out in order to enable the design and development of tiny, networked computational devices. Among the challenges to face, size, cost and energy constraints impose severe limitations to the computational capabilities of the units. Moreover, the limited reliability of each device and the desired ease of deployment call for self-organizing, self-healing networking. Research in Wireless Sensor Networks (WSNs) has undergone substantial progress, including the development of suitable operating systems, hardware platforms, and standardized low-complexity network protocols (e.g., [2], [3], [4]).

Recent research in the field (e.g., [5]) further emphasizes the need for a high degree of interoperability, both among heterogeneous sensor nodes and with other computing devices. This interoperability would allow us to extend current robotic architectures to sensor-actuator networks, thus expanding the scope of existing network and ubiquitous robotics research to include small inexpensive sensing and actuation units, which can be pervasively spread in the environment. We believe that the convergence of ubiquitous robotics and networked embedded devices is a fertile line of research. We refer to this line of research as Networked Embedded Robotics.

In this paper, we describe a generic, open sensor-actuator platform which is intended to be instrumental to perform

<sup>1</sup>Currently with the Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Denmark. Part of this work was performed while this author was a visiting student at AASS, Örebro University, Sweden. The third author was partly supported by ETRI (Korea) through the project "Embedded Component Technology and Standardization for URC (2004-2008)".



Fig. 1. The PEIS-Mote

research in Networked Embedded Robotics. It fulfills four main design desiderata: *low cost*, lower than 70USD, obtained using widely available hardware components; *small size*, comparable to that of mote-class devices; *expandability*, by providing a rich set of inputs and outputs of different kinds; and *interoperability*, achieved adopting standard network protocols and a popular open source operating system for networked embedded devices. Furthermore, our platform is *fully open*, as both its electrical design and its software are publicly available [6].

This paper proceeds as follows. First, we briefly review existing hardware and software tools and evaluate how they meet the requirements of a generic sensor-actuator platform. Then, we illustrate the design and the first prototype of our own proposal for such a platform. Finally, we show examples of embedded robotic devices we developed and integrated into an ubiquitous robotics framework, the PEIS-Ecology Project.

## II. NETWORKED EMBEDDED DEVICES: TRENDS & TOOLS

Different classes of hardware platforms target the different roles that nodes assume within the typical WSN tiered hierarchy [7]. Usually, upper-tier master nodes are 32-bit machines which have no tight constraints to satisfy, and generally run full fledged operating systems; lower-tier sensor nodes are instead heavily constrained in size and cost in order to be suitable for the intended massive deployments. A number of generic sensor nodes have been proposed to meet these challenging requirements. Usually, commercially available microcontroller, radio and storage integrated circuits (ICs) are accommodated on a single, custom designed printed circuit board (PCB). While differing with regards

to the particular choice of components, these devices share common characteristics like the small size of program and data memory, the absence of a memory management unit (MMU), and a low-power radio. Many nodes offer on-board sensors (e.g., to measure temperature and light) and limited expansion capabilities to connect extra components.

This class of devices are generally not suitable for the needs of Networked Embedded Robotics. The main limitation in this sense is that these nodes, which are mainly designed for sensing purposes, typically provide only a small number of I/O pins and do not have hardware PWM outputs. This severely restricts the number and variety of actuators that these units can control. Because of this, we decided to develop a new node platform geared towards *expandability* with respect to both general sensing and actuation needs. This effort resulted in the PEIS-Mote, a design that we describe in section III below.

In addition to the hardware design of the device itself, the above constraints heavily challenge the usual software development paradigms and practices. Though running on low-performance hardware, user programs and supporting operating systems need to take care of non trivial tasks, like fault-tolerant communication with fellow nodes. Limiting factors like the small memory size and the absence of an MMU make it difficult to implement a conventional execution model like preemptive multithreading, given the necessary overhead in terms of space and time. On the other hand, avoiding this overhead in order to keep the requirements at minimum means adhering to a paradigm conceptually closer to how hardware works, but potentially awkward for the developer. A wide range of research efforts explore this trade-off between functionalities and overhead. One of the earliest operating system for sensor networks, TinyOS [2], adopts an event-driven programming paradigm and static code linking, resulting in small code size and low RAM usage. Additional features with respect to this simple, yet effective, approach include support for loadable modules, multithreading and dual mode operation.

The prospective applications that we foresee for an embedded sensor-actuator platform are likely to take place in highly heterogeneous environments, populated by different devices possibly from different vendors. Therefore, when investigating which software platform to choose as the starting point for our custom node, important criteria included widespread usage and support for a broad range of hardware platforms. Therefore we opted for TinyOS, now probably the most popular operating system for sensor networks, whose newest features (coming with the 2.0 release) properly met our requirements. For example, platform heterogeneity is facilitated by using platform independent types [8], which allow architectures with different endianness and alignment restrictions to correctly exchange and manipulate data, thus easing cross-platform communication. Thanks to this feature, we were able to make architectures with different data alignments (i.e., Atmel Atmega128 and TI MSP430) share common data structures through messaging, without having

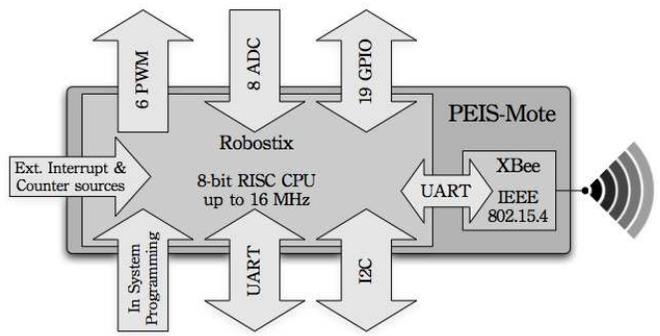


Fig. 2. Functional scheme of the PEIS-Mote

to resort to tricky and fault-prone procedures like explicit marshalling and unmarshalling.

Special emphasis to address interoperability issues was then put on standard protocols support. In particular, we chose the 802.15.4 PHY and MAC layers [4] to build our low-power experimental network upon, and no compatibility problem arose mixing hardware from different vendors (namely Telos motes [3] and XBee [9] wireless modules). The main deficiency, which we started addressing, is the lack of support for actuation hardware. This is implicit in the nature of TinyOS as an operating system for *sensor* nodes — the same reason why standard motes are not well suited to drive actuators.

### III. THE PEIS-MOTE

Low cost microcontroller boards are widespread among roboticists as platforms to experiment with when building simple robots, be it for hobby, educational or academic purposes. These kits are produced and sold in large volumes compared to the rather specialized hardware used in sensor networks research, still being marketed by few companies and used mainly in research labs. Hence, price and availability are strong points in favor of hobbyists hardware.

The main element of our custom platform is the Robostix board [10] from Gumstix Inc. It consists of an Amtel AT-Mega128 8-bit RISC microcontroller with the needed external components and circuitry, which can be powered from batteries or from a wall transformer and provides its external connections through convenient headers. Characterized by small size and low cost ( $8 \times 3.5$  cm, with a 39\$ base price), this platform is also particularly attractive for the connectors it exposes (fig. 2). These include six Pulse Width Modulated (PWM) outputs, eight analog inputs, nineteen general purpose I/O (GPIO) pins plus support for external counter and interrupts sources. In addition,  $I^2C$  and two serial interfaces are available, as well as two dedicated strips of connectors to power external elements. There is also the possibility to attach custom daughterboards via a 60 pin Hirose connector. Compared to the expansion possibilities of a WSN mote, the advantages of a breakout board like this one are clear.

For communication, we paired the board with a 19\$ MaxStream XBee module [9], a self-contained 802.15.4 wire-

less transceiver which exports its functionalities over a serial line. The resulting overall platform, which we dubbed PEIS-Mote due to its inception in the context of the PEIS-Ecology project, costs under 70\$ and in its current form is just about 3 cm wider than a Telos mote (see fig. 3).

From the software side, porting TinyOS to this platform has been relatively straightforward, due to its similarity with the Mica family of motes (which share the same microcontroller architecture of the Robostix). We then addressed the particular needs relevant to our application area by providing extra functionalities, exposing at the software layer the PWM outputs (suitable to drive servo and DC motors), and developing actuator-specific O.S. interfaces (e.g., for stepper motors and the light controller we illustrate in section IV). The first tests, carried out with standard Hitec servos (HS-422), STMicroelectronics H-bridge and stepper driver ICs (respectively the VNH2SP30 and the L297), show promising results. This suggests the possibility to standardize a TinyOS service distribution (i.e., a well-tested system API) geared towards actuation, to be then eventually ported to other platforms. The long term goal is to provide a tiny, cheap, wireless platform easy to connect to standard actuators and easy as well to operate, like WSN motes are now for sensing purposes. We started a project to contribute the code and let other researchers and practitioners experiment, in order to further refine it through usage and test. The material, contributed to the TinyOS online repository [6], comprises all the software developed so far, including support for the Robostix, the XBee and test applications.

Currently, the PEIS-Mote is a completely supported platform under TinyOS 2.0, providing standard millisecond timers, Active Message wireless and serial communication, Analog-to-Digital conversion and general purpose I/O as dictated by TinyOS standards. Extensive network tests mixing PEIS and Telos motes proved the successful interoperability, validating our design with respect to our target requirements. At the moment, the main drawback of the PEIS-Mote over other platforms is the lack of power consumption optimization. However, as actuators themselves pose even greater problems in this sense, the power optimization of the control unit can be considered a relatively minor issue, and hence will be addressed in the future.

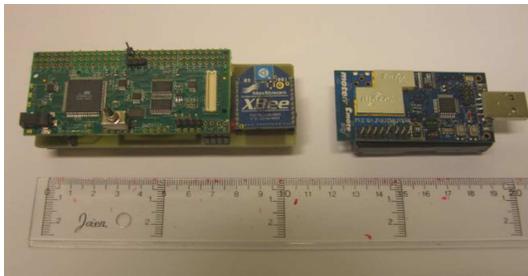


Fig. 3. The current PEIS-Mote generation next to a Telos

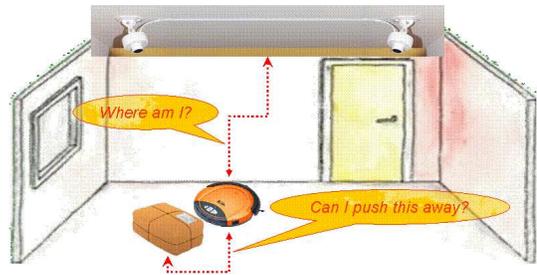


Fig. 4. An example of cooperating PEIS (from [11]).

#### IV. EVERYTHING'S A ROBOT: THE PEIS-ECOLOGY EXAMPLE

The PEIS-Ecology project [11], [12] is an ongoing research effort exploring a new approach towards the inclusion of robotic systems in common environments. Conceptually, it is centered around the close cooperation of many simple robotic entities in order to perform complex tasks through collaboration, rather than through the deployment of sophisticated (and thus complex and expensive) robots. In this context, every robot is abstracted as a PEIS (Physically Embedded Intelligent System): a device with some computational and communication capabilities embedded in the environment, and able to interact with it by means of sensors and/or actuators. This does not pose constraints to the complexity of a PEIS, which could be as simple as a toaster or as expensive as a humanoid robot. All PEIS share a *uniform communication model*, realized in a specific middleware which implements a distributed tuple space over a peer-to-peer network. In addition, a subscription mechanism, in which a subscriber element consumes tuples produced by another PEIS, implements a *uniform cooperation model* which lets functionalities to be shared within the ecology.

The simple example in fig. 4 gives an idea of the overall concept. In this scenario, an autonomous vacuum cleaner performs its cleaning task by local navigation and simple obstacle avoidance; an overhead camera system, given its position relative to the cleaner, can provide it with global localization information far more easily than simple onboard sensors can infer. Suppose then that the cleaner, during its operation, gets stuck into a parcel lying on the floor. A smart RFID contained within the parcel could make this obstacle able to communicate relevant properties upon which to decide if it is pushable or not (weight, fragility, etc), something that the cleaner alone would probably not be able to determine. This example, though simple, is representative of how non-trivial tasks could be performed by the cooperation of rather simple elements, or PEIS (see [12] for more information).

Networked, small-sized embedded devices clearly play an important role in the context of this project: we could have PEIS non-obtrusively spread into the environment, for example embedded into furniture for monitoring tasks (e.g., light and temperature sampling); in a similar manner, pervasive intelligent actuators could result useful to control the same

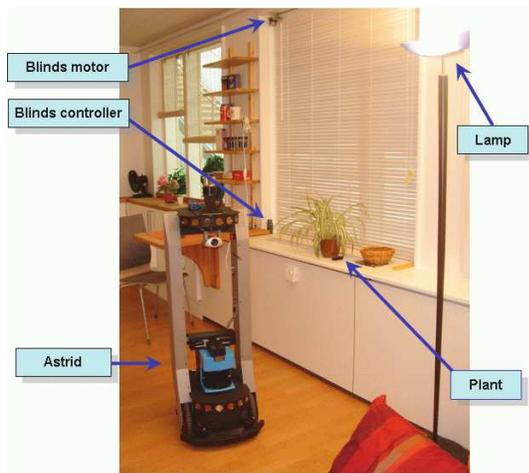


Fig. 5. The experimental testbed

environmental features (e.g., blinds, windows, artificial lights being part of the robotic ecology), or to aid during other activities (e.g., pet care, elderly assistance). As the previous example showed, even simple capabilities can lead to actual benefits for the overall ecology.

In order to integrate these devices, a stripped-down version of the PEIS-Ecology middleware has been developed [13], which implements the PEIS communication and cooperation model with reduced memory footprint. This extension of the ecology to mote-class devices is transparent, since a bridge component takes care of network translation and message format rescaling: the structure and semantics remain the same, so that PEIS as different as 64 bit PCs, standard robots, commercially available motes and our own sensor-actuator PEIS-mote are able to seamlessly communicate and cooperate.

As a proof of concept, we performed a simple experiment involving the following PEIS devices (see fig. 5).

- An apartment plant, with a monitoring Telos mote next to it. Thanks to the mote, the plant is able to sense humidity, temperature and light, and to interact with the rest of the ecology over a low-power wireless network.
- A floor lamp, controlled by another Telos mote through a commercial light dimmer.
- Window blinds, operated by a stepper motor. A PEIS-Mote, connected to the motor, can open and close them.
- An ActivMedia PeopleBot mobile robot, named “Astrid”, equipped with an on-board PC.

In our experimental run, Astrid receives the task to take a photo of the sofa in the living room, and to subsequently send it to the owner’s mobile phone. During task execution, the plant senses too much sunlight, hence it asks the window blinds to close themselves. Astrid then issues a request to the floor lamp to compensate for reduced environmental lighting by augmenting the artificial light intensity. Once this happens, the photo is taken and the task is over. The detailed events happening at the middleware level are outlined in [13].

Even a simple experiment like this demonstrates that networked embedded devices are a convenient way to spread

sensing and actuation in a distributed robotic framework, given the ease of deployment and flawless operation. To the best of our knowledge, the PEIS-Mote and the relative software provide the first standardized toolkit to tackle actuation issues in this context.

## V. CONCLUSIONS AND FUTURE WORK

Recently sensor networks research demonstrated the potential of networked embedded devices as a way to pervasively instrument the physical world. We argue that the same class of devices can be highly valuable to pursue unforeseen developments in the robotics area, especially in the networked and ubiquitous robotics subdomains. Our main contributions are a platform for Networked Embedded Robotics and its supporting software. We believe it to be the first publicly available design that is built from cheap commercial, non specialized hardware, while at the same time retaining compatibility with the majority of existing solutions like standard motes. Our initial experiments show that this platform and the accompanying tools are effective experimental tools for ubiquitous, networked and embedded robotics. To allow other researchers and practitioners to use and hopefully improve the design, source code and schematics are publicly available [6]. Further technical details can be found in [14].

## REFERENCES

- [1] M. Weiser, “The computer for the 21st century,” *Scientific American*, vol. 265, no. 3, pp. 66–75, Sep. 1991.
- [2] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, “TinyOS: An operating system for sensor networks,” in *Ambient Intelligence*, W. Weber, J. M. Rabaey, and E. Aarts, Eds. Springer Verlag, 2004.
- [3] J. Polastre, R. Szewczyk, and D. Culler, “Telos: Enabling ultra-low power wireless research,” in *Proc. Int. Conf. on Inform. Processing in Sensor Networks (IPSN/SPOTS’05)*, Los Angeles, CA, USA, Apr. 2005.
- [4] *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Std 802.15.4, 2003.
- [5] P. Costa, G. Coulson, C. Mascolo, G. P. Picco, and S. Zachariadis, “The RUNES middleware: A reconfigurable component-based approach to networked embedded systems,” in *Proc. IEEE Int. Symp. on Personal Indoor and Mobile Radio Communications (PIMRC’05)*, Berlin, Germany, Sep. 2005.
- [6] M. Bordignon. TinyBotics website. [Online]. Available: <http://www.tinybotics.net/>
- [7] J. Hill, M. Horton, R. Kling, and L. Krishnamurthy, “The platforms enabling wireless sensor networks,” *Comm. ACM, Special Issue on Wireless Sensor Networks*, vol. 47, no. 6, pp. 41–46, Jun. 2004.
- [8] K. Chang and D. Gay, “Language support for interoperable messaging in sensor networks,” in *Proc. Int. Workshop on Software and Compilers for Embedded Systems (SCOPES’05)*, Dallas, TX, USA, Sep. 2005.
- [9] “XBee OEM RF Modules - Product Manual,” MaxStream, Inc. [Online]. Available: <http://www.maxstream.net/>
- [10] Robostix documentation. Gumstix, Inc. [Online]. Available: <http://docwiki.gumstix.org/Robostix>
- [11] A. Saffiotti and M. Broxvall, “PEIS ecologies: Ambient intelligence meets autonomous robotics,” in *Proc. Int. Conf. on Smart Objects and Ambient Intelligence (sOc-EUSAI’05)*, Grenoble, France, Oct. 2005.
- [12] PEIS Ecology website. AASS Mobile Robotics Lab. [Online]. Available: <http://www.aass.oru.se/~peis/>
- [13] M. Bordignon, J. Rashid, M. Broxvall, and A. Saffiotti, “Seamless integration of robots and tiny embedded devices in a PEIS-ecology,” in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS’07)*, San Diego, CA, USA, Oct. 2007.
- [14] M. Bordignon, “An open framework for networked embedded robotics,” Master’s thesis, University of Padova, Padova, Italy, 2007.