The e-puck, a Robot Designed for Education in Engineering

13 authors, including:

Francesco Mondada
École Polytechnique Fédérale de Lausanne
257 PUBLICATIONS   **10,450** CITATIONS

Michael Bonani
École Polytechnique Fédérale de Lausanne
41 PUBLICATIONS   **1,875** CITATIONS

Adam Klaptocz
15 PUBLICATIONS   **720** CITATIONS

Some of the authors of this publication are also working on these related projects:

Odor Source Localization View project

Swarm robotics View project

The e-puck, a Robot Designed for Education in Engineering

Francesco Mondada[1], Michael Bonani[1], Xavier Raemy[2], James Pugh[2], Christopher Cianci[2],
Adam Klaptocz[3], Stéphane Magnenat[1], Jean-Christophe Zufferey[3], Dario Floreano[3], Alcherio Martinoli[2]

Abstract— **Mobile robots have the potential to become the ideal tool to teach a broad range of engineering disciplines. Indeed, mobile robots are getting increasingly complex and accessible. They embed elements from diverse fields such as mechanics, digital electronics, automatic control, signal processing, embedded programming, and energy management. Moreover, they are attractive for students which increases their motivation to learn. However, the requirements of an effective education tool bring new constraints to robotics. This article presents the e-puck robot design, which specifically targets engineering education at university level. Thanks to its particular design, the e-puck can be used in a large spectrum of teaching activities, not strictly related to robotics. Through a systematic evaluation by the students, we show that the e-puck fits this purpose and is appreciated by 90 percent of a large sample of students.**

I. Introduction

Mobile robots are both fascinating objects and the result of the fusion of multiple competences. This fascination leads to the organization of plenty of robotics contests worldwide annually [16]. From an engineering point of view, the design and control of mobile robots requires skills in many disciplines such as mechanics, electronics, energy management, computer science, signal processing, and automatic control. The combination of these two aspects (fascination and inter-disciplinarity) makes mobile robots an excellent educational platform that enables students to address a broad range of engineering fields.

This paper presents, for the first time, the design approach resulting in the *e-puck*[1], an educational desktop mobile robot developed at the École Polytechnique Fédérale de Lausanne (EPFL) for a broad exploitation in teaching activities. The main objectives of this development were:

- The use of a common platform in all EPFL courses related to mobile robotics, to replace the different robots previously in use.
- The use of a mobile robot in non-robotic courses, for instance signal processing, automatic control, and embedded programming, in order to propose more project-based exercises.
- The introduction of mobile robots earlier in the curriculum, which implies the deployment in larger classes (50 to 100 students).

II. Existing robots for education

A wide range of mobile robots are available on the market. In this section we survey the subset of them that we think to be relevant as educational platforms.

[1]e-puck: http://www.e-puck.org

Fig. 1. The e-puck robot.

The Khepera II from K-Team is a redesign of the original Khepera robot [15]. With the same size of the original Khepera, it is compatible with its extensions, software, and scientific tools. The Khepera II is interesting because its size allows to use it on a desktop. It is expensive (around 1500 € for a basic configuration) but is known for being reliable and well supported.

The Hemisson from K-Team is a cheap platform (225 €) with a diameter of 120 mm. It only provides a limited computational power and few sensors in its basic configuration, but is extensible. It is a robust platform well suited for beginners.

The IdMind's circular GT kit is a similar platform slightly cheaper (210 €) with a diameter of 150 mm. It has less standard extensions than Hemisson, but has more I/O available ports to connect self-made extensions. It is more suited for experimentation using custom self-made extensions.

Even cheaper (175 €), the platform Bot'n Roll is representative of a set of simple robots with few sensors that are excellent starting kits for beginners. This and the previous kit improve their accessibility by providing graphic programming environments.

The Lego Mindstorms RCX was the first robotic platform from the Lego company. The RCX is built around a small 8 bit processor and can manage only 3 inputs and 3 outputs (typically DC motors without encoder); but the combination with the Lego bricks makes it a fantastic tool to discover new robots shapes. The RCX has been replaced in 2006 by the Lego Mindstorms NXT. This newer version is equipped with more advanced sensors, including color and sound, and can drive motors equipped with encoders. It is a clear reference in the field because of its good computational power, its flexibility and interesting price (260 €).

The Palm Pilot Robot Kit (PPRK) is a commercially available platform[2] from Carnegie Mellon University combining a mobile base and a personnal digital assistant (PDA), originally a Palm Pilot. The PDA provides the computational power and the user interface and controls the sensors and the actuators through a serial connection with a PIC processor. The result is a compact omnidirectional platform with three distance sensors (in its basic configuration for 250 €). Furthermore, the availability and the maturity of Palm development tools makes this platform an interesting starting kit.

The Cye platform[3] is a medium-size robot (40×28×13 cm) equipped with special wheels that ensure a good odometry. Its price is around 540 €. Designed for indoor domestic environments, Cye can carry extensions such as a vacuum cleaner and can navigate in indoor environments.

The Khepera III from K-Team is a research oriented platform much larger than the Khepera II (120 mm of diameter). It is adaptable to specific research requirements through extensions, for instance the korebot board which provides an XScale processor. Flexible, efficient, and powerful with respect to its size, this robot is also quite expensive as an educational robot (around 2000 € in basic configuration).

The ER1 from Evolution Robotics[4] is a simple aluminum frame kit supporting a laptop (not included) and equipped with wheels. The laptop provides the computational hardware, which improves the performance / cost ratio of the kit (the basic configuration costs around 230 €). The motor wheels controller has some free inputs/outputs but provides limited computational power. This low-cost kit comes with a sophisticated but expensive software environment for navigation and vision.

The KHR-1 from Kondo [5] has been the first humanoid robot with good mobility capabilities (17 DOF) for a price under 1000 €.

The Pioneer 3 (P3) is the latest version of the Pioneer robot by ActivMedia. It is a large (44×38×22 cm) solid platform on which the user can install custom processors, sensors, and actuators. The AmigoBot of the same company is a cheaper version (1550 €) of the same concept.

The Garcia from Acroname[6] is a small robot frame (25×18×10 cm) designed to be controlled by a companion XScale board. The size of Garcia makes it suitable for experiments in compact environments. Its price is around 1360 € in the basic configuration.

Robotino from Festo[7] is a modern mobile robotic platform for education. Robotino runs a real-time Linux kernel and is built around industrial standards. These features make this robot powerful in term of computational power but also expensive (about 4500 €). Robotino is well suited for technical schools that want to approach technical problems using robotics.

Roomba Create from iRobot is an educational/research version of the roomba vacuum cleaner. Devoided of the

cleaning module, this platform provides low cost mobility (100 €). Its sensors, designed for vacuum cleaning tasks, offer a good support for reactive navigation. The limited internal processor, dedicated to the low-level robot control, is programmable by simple scripts. Any advanced programming or supplementary I/O requires an additional main processor.

III. ROBOT DESIGN FOR ENGINEERING EDUCATION

Most of the aforementioned products are exclusively either efficient mobile robots or good educational tools. However, being both implies the following criteria:

- *Desktop size*. A robot that can evolve on the desk near the computer improves drastically the student efficiency during experimentation. We consider that for a good mobility, the experimentation space should be 10 times the diameter of the robot. On a table, this implies a robot diameter smaller than 80 mm.
- *Wide range of possibilities from an engineering and educational point of view*. To exploit this tool in various fields of education such as signal processing, automatic control, embedded programming, or distributed intelligent systems design, the robot should provide a wide set of functionalities in its basic version.
- *User friendly.* The user interface has to be simple, efficient, and intuitive. This is an important point for the acceptance of the system by the students.
- *Low cost.* The broad introduction in engineering classes requires a large number of robots. Knowing that the budget of many schools is constant or decreasing, this is only feasible by reducing the cost of an individual robot.
- *Open information.* This robot has to be shared among professors, laboratories, schools and universities. An open source hardware/software development model is an effective way to achieve this goal.

None of the platforms available on the market is respecting these criteria. Most robots are large and thus need to operate on the floor. The smallest robots are either expensive or have limited functionalities. Very few are open source.

This motivated us to create the e-puck robot in summer 2004. We tested a first set of prototypes with students during the 2004–2005 academic year. Based on this experience, we redesigned the robot and produced the final version in summer 2005 (Fig. 1). The following sections present the robot design and a student evaluation based on its use during four semesters from 2005 to 2007.

IV. THE E-PUCK DESKTOP MOBILE ROBOT

We based the design of the robot on the first two aforementioned criteria: desktop size and flexibility. Combined, these two constraints imply the miniaturization of a complex system. To achieve a low price, we opted for the use of cheap components and mass production manufacturing techniques. We took special care to make the robot as user-friendly and interactive as possible, in order to ensure that it would be well received by students.

In this design process, a central aspect is the choice of the robot features. This particular choice is one of the

---

[2]PPRK: http://www.cs.cmu.edu/~pprk/
[3]Cye: http://www.personalrobots.com
[4]Evolution Robotics: http://www.evolution.com
[5]Kondo: http://www.kondo-robot.com
[6]Acroname: http://www.acroname.com
[7]Festo: www.festo-didactic.com

innovations of the e-puck design. The sensors, actuators, and interfaces of the e-puck are representatives of a wide range of devices one can find in several engineering sub-domains:

- sensors in different modalities: audio, visual, distances to objects, gravity,
- input devices with different bandwidths from 10 Hz to 10 MHz (Figure 2, left),
- actuators with different actions on the environment (Figure 2, right),
- wired and wireless communication devices,
- two types of processors: general purpose and DSP.

By exploiting this large set of possibilities, a teacher can present and the student can practice a broad set of engineering fields and skills.
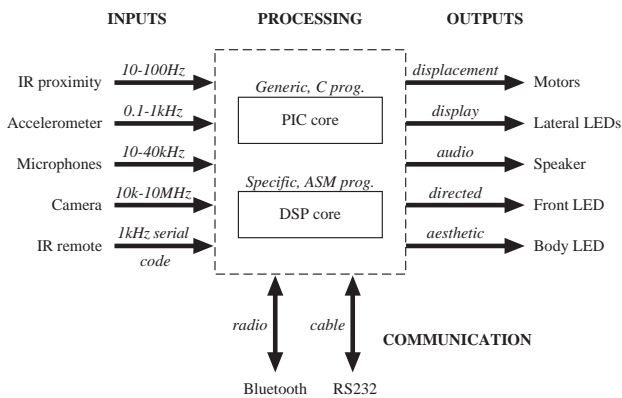
Fig. 2.   The e-puck allows exploration of different engineering topics, control options, and signal bandwidths.

For applications where the basic features are not sufficient, the e-puck can be extended with specific hardware. We provide embedded software consisting of a library and several demo applications. We also provide an open source simulator and a monitoring tool to run on a desktop computer. In this section we present the detailed hardware and software design choices.

A. e-puck hardware (basic configuration)

1) Microcontroller: The electronic structure of the e-puck (Fig. 3) is built around a Microchip dsPIC microcontroller. This microcontroller complies with the educational criteria of flexibility because it embeds both a 16 bit processor with a 16 entry register file and a digital signal processor (DSP) unit. This CPU runs at 64 MHz and provides 16 MIPS of peak processing power. The instruction set is mostly orthogonal[8] and rich; in particular, it contains multiply-acccumulate and hardware-repeat instructions suitable to drive the DSP unit, for instance to efficiently compute scalar products and fast fourier transforms. Finally, this processor is supported by a custom tailored version of the GCC C compiler. For the e-puck, we chose a microcontroller version with 8 kB of RAM and 144 kB of flash memory.

---

[8]An instruction set is *orthogonal* if any instruction can use data of any type via any addressing mode (http://en.wikipedia.org/wiki/Orthogonal_instruction_set)
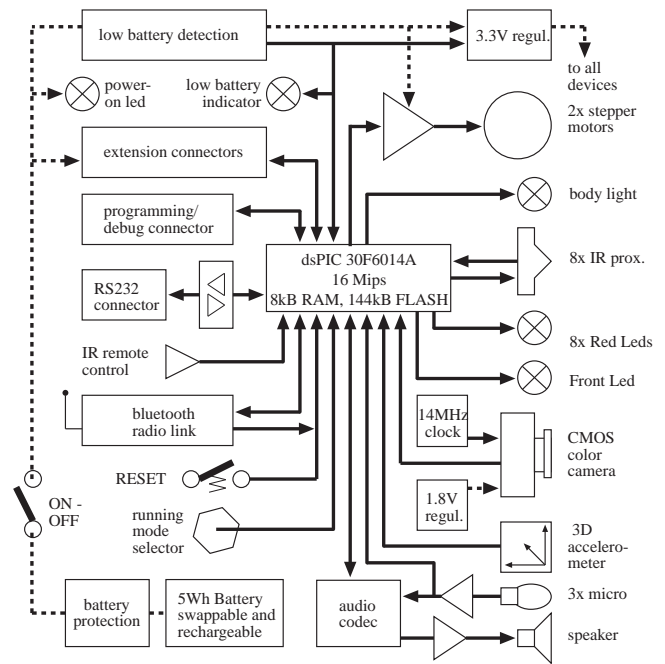
Fig. 3.   The outline of the electronic of the e-puck.

2) Sensors and actuators: To ensure a broad range of experimentation possibilities, the e-puck contains various sensors covering different modalities:

- Eight infrared (IR) proximity sensors placed around the body measure the closeness of obstacles or the intensity of the ambient infrared light. These are typical sensors for simple navigation in cluttered environments.
- A 3D accelerometer provides the acceleration vector of the e-puck. This vector can be used to measure the inclination of the e-puck and the acceleration produced by its own movement. It can also detect collisions and if the robot falls. This sensor is rarely included in miniature low-cost mobile robots. We decided to include it because it allows a rich set of experiments.
- Three microphones capture sound. Multiple microphones allow the e-puck to localize the source of the sound by triangulation. The bandwidth of this signal is much larger than the one of the accelerometer or of the infrared sensors, making the microphones, because of their larger computational demands, the ideal tools to learn how to use the DSP unit.
- A color CMOS camera with a resolution of 640×480 pixels in front of the e-puck enables experimentation in vision. Only a sub-part of the image can be grabbed: the size of acquisition is limited by the memory size of the dsPIC and the rate is limited by its processing power. Any format of sub-image is acceptable, providing these two constraints are fulfilled. For instance, the e-puck can grab a color image of 40×40 pixels at 4 frames per second; the frame rate is doubled in gray-scale. This limitation shows to the students the impact of high bandwidth sensors such as cameras.

The e-puck provides the following actuators:

- Two stepper motors. They control the movement of

the wheels with a resolution of 1000 steps per wheel revolution.

- A speaker, connected to an audio codec. Combined with the microphones, the speaker can create a communication network with the ability to detect the direction of peers. It is also an excellent output device for human interaction.
- Eight red light emitting diodes (LED) placed all around the e-puck. These LEDs are covered by a translucent plastic and the e-puck can modulate their intensities. They provide a visual interface with the user; furthermore, another e-puck can observe them with its camera which allows mutual visual interactions.
- A set of green LEDs placed in the transparent body. By lighting the body, they improve the interactions with the user.
- A red front LED placed beside the camera. This LED generates a focused beam that projects a red spot on objects in front of the e-puck. Combined with the camera, this spot allows distant measurements at longer range than the infrared proximity sensors.

3) User interface: The e-puck also contains several devices to interact with the user and to communicate with other equipments:

- Two LEDs show the status of the battery: One indicates whether the robot is powered on, while the other indicates a low battery condition.
- A connector to interface to an in-circuit debugger, to program the flash memory and to debug code.
- An infrared remote control receiver, to control the e-puck with standard television remote controls.
- A classic RS232 serial interface to communicate with a desktop computer.
- A Bluetooth radio link to connect to a desktop computer or to communicate with up to 7 other e-pucks.
- A reset button.
- A 16 positions rotary switch to specify a 4 bit number, which can be used, for instance, to select among preprogrammed behaviors or parameters.

4) Mechanics: The robot has a diameter of 75 mm and a height which depends on the connected extensions. The mechanical structure of the e-puck consists of injected plastic parts. We have chosen this manufacturing technique because it reduces the unit price of the robot for sufficient quantities. The robot structure is simple, being made of only four injected plastic parts: the main body, the light ring, and the two wheels (Fig. 4). The main body is the core of the mechanical structure and encloses the battery. The user can extract the battery from the bottom of the e-puck. The two motors are simply laterally screwed onto the main body, with the wheels directly attached to the motor axis. The main printed circuit board (PCB), containing most of the electronics, is screwed on top of the main body. A light diffusion ring and a default extension board are mounted over this main PCB; the user can replace the default extension board with application specific boards, as illustrated by some examples in the next section. All mechanical parts are transparent and allow to observe all components.
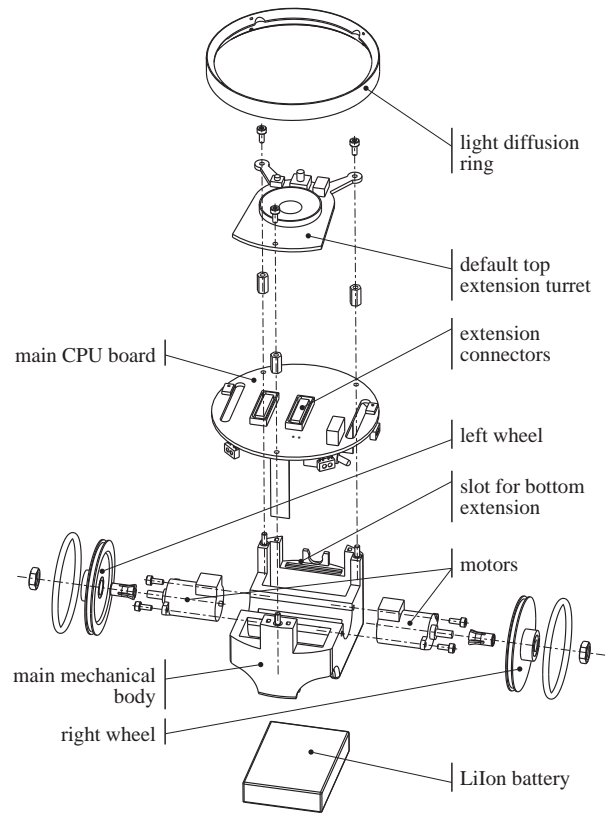

Fig. 4. The mechanical structure of the e-puck in an exploded view.

B. e-puck extensions

To address the needs of specific teaching scenarios that require different mechatronics, the user can connect physical extensions to provide additional sensors, actuators, or computational power. Extensions physically connect through an extension bus which routes a two wires interprocessor communication bus (I^2C) as well as the connections to most sensors.

There are three physically different types of extensions: "top", "bottom", and "sandwich".

Top extensions do not allow other extensions above them. The basic e-puck includes a default extension board of this type which provides the speaker, the 16 positions rotary switch, the infrared remote control receiver, the RS232 connector, and a reset button (Fig. 1, right). There are several other extensions of this type available, for instance:

- A rotating scanner. It is equipped with infrared triangulation distance sensors with a range of 40 cm (Fig. 5, left). For exercises involving robot localization, the short range (2–3 cm) of the proximity sensors available on the basic version of the e-puck is not sufficient.
- A turret with three linear cameras. It provides a very large field of view for measuring optical flow [8], [21] (Fig. 5, right) still requiring few memory.

Bottom extensions are internal PCBs, enclosed in a vertical slot in the front part of the main body. These extensions are close to the ground and are connected to the main PCB by an I^2C bus and a power connection. For instance, an extension of this type provides three analog sensors which measure the ground color (Fig. 6, left top).
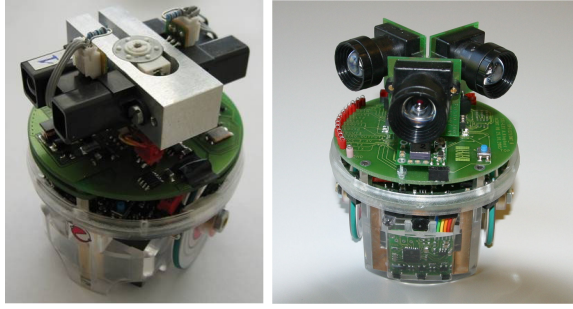
Fig. 5. Two "top" extensions for the e-puck robot: an infrared distance scanner (left) and a large field of view linear camera (right).

Fig. 6. Left top: A bottom extension to measure the ground color. Left bottom: a "sandwich" extension implementing a Zigbee radio link. Right: A complete visual communication system consisting of two "sandwich" extensions.

Extensions can also be "sandwich" boards that replicate the connector on their top, for instance:

- a Zigbee radio link with adjustable radio communication range does not require mechanical access to the top of the e-puck [4] (Fig. 6, left bottom);
- a color RGB LED ring (Fig. 6 right);
- an omnidirectional vision turret with onboard microcontroller dedicated to vision (Fig. 6 right)

Several extensions of this type can be stacked, and a top turret placed above (Fig. 6 right).

C. e-puck embedded software

To develop software for the e-puck, we provide standard components: a bootloader to program the e-puck over Bluetooth, a low-level library to drive the hardware, and a monitor to communicate with a desktop computer. Within the community of e-puck users, standard software modules are also developed. For instance a Player server is developed by VeRLab[9]. These components are all released under the e-puck open source license.

---

[9]Player driver for the e-puck robot: http://code.google.com/p/epuck-player-driver/

1) Bootloader: To program and debug a microcontroller, one typically uses an in-circuit debugger, which is a specific piece of hardware. To reduce cost and remove this requirement, the e-puck comes with a bootloader that allows the user to re-program the flash of the microcontroller through the Bluetooth or the serial port. At boot, the bootloader listens a small amount of time for activity on these ports, and if none is detected, launches the user application. During this time, the user has the opportunity to send a special command in order to flash a new program.

2) Low-level library: To facilitate the use of the e-puck hardware, which requires specific code with precise timings, we provide a collection of functions called the "Low-Level Library". This library contains functions such as "move the right motor at that speed", "read 40×40 pixel image", or "send this message through Bluetooth". It is statically linked with the user application at compile time.

3) BTcom protocol: When developing for a robot, it is often useful to be able to control it from a desktop computer. To that end, the e-puck comes with a monitor implementing a remote control protocol through the Bluetooth or the serial port, called the "BTcom protocol". This protocol provides full remote control of the e-puck, allowing the desktop computer to set the speed of the motors, read the image of the camera, specify the state of the LEDs, read the accelerometer, and so on. This allows the user to develop applications on a desktop computer, in a comfortable environment with a rich set of development tools. Moreover, with this approach, applications can exceed the computational capabilities of the dsPIC. The same strategy can be used to control the e-puck from a simulator, such as Enki or Webots.

D. e-puck simulation

Several simulators support the e-puck. Among them, we use Webots and Enki. Webots [13] is commercial and supports three-dimensional physics through the ODE[10] library. Enki[11] is open source and provides fast 2D physics, which, for instance, makes it suitable for evolutionary robotics experiments [7].

E. Availability

All our software, design, and production documents are available under an open source hardware/software license, meaning that anyone can use and modify them as long as they comply with the license.

Currently, two companies (GCtronic and AAI Japan) produce the e-puck and about ten companies distribute it under the term of the open hardware license. A growing community is using the e-puck both for research and education (see for instance [1], [20], [18], [12], [11], [5], [2], [10], [14], [9], [17]).

The production price of the e-puck basic version is around 250 €. The selling price by most companies is around 550 €.

---

[10]Open Dynamics Engine: http://www.ode.org
[11]Enki: http://home.gna.org/enki/

V. Engineering education using the e-puck robot at EPFL

At EPFL, several courses exploit the e-puck robot as experimentation platform. In particular we organize practical exercices in the following teaching areas:

- *Signal processing.* We use the e-puck to explore signal processing on sound, which is an optimal context for education because of the signal low frequency, low propagation speed, ease of generation, acquisition, and direct perception by humans. In our embedded programming course we explore signal processing applications during two sessions of five hours each. We analyze the sound using a fast fourier transform (FFT), with a close attention to its efficient implementation on the DSP unit. In each session, we first present the theory; then the students verify it on MATLAB with data acquired from the robot, and finally they implement it on the dsPIC of the e-puck.

- *Automatic control.* In our embedded programming course, we apply automatic control concepts on the e-puck. The goal of the five hours session is to build a regulator that controls the e-puck and maintains a constant distance with respect to a visual target (Fig. 7, left). The input of a proportional-integral regulator is the estimated distance to the target, extracted from the camera looking at a vertical black stripe on a white background (Fig. 7, right). The output of the regulator is assigned to the speed of both wheels, reducing the problem to one degree of freedom.
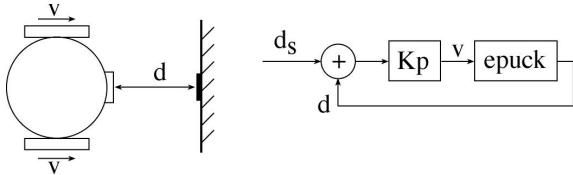

Fig. 7.   Model of the problem (left) and scheme of the regulator (right) for the e-puck automatic control practical.

- *Behaviour-based robotics.* In a robotics course, we use the e-puck to provide the students with a better understanding of the subsumption architecture [3], its advantages and its shortcomings. The goal of the exercise is to implement a behavior-based controller for a simplified industrial application consisting of the transport of goods throughout a warehouse. The e-puck (the carrier) has to follow a line painted on the ground (the track), while avoiding obstacles and going back to the line after disruption. To that end, the e-puck is equipped with the ground sensor extension (see Section IV-B and Fig. 6, left top).

- *Distributed intelligent systems.* Our course on distributed intelligent systems (formerly focusing on swarm intelligence) includes weekly laboratory exercises in which the students use a combination of real e-puck robots and realistic simulations using Webots [13] to test and verify the topics and theories presented in lectures. This help the students to assimilate theoretical concepts around multi-robot coordination and networking, and to understand the difficulties of implementing them. This also enhances their awareness of the differences between various types of implementation levels; for example realistic simulations and real experiments.

- *Position estimation and path finding of a mobile robot.* In a 10 hours practical session we use the e-puck to explore robot localization and path finding. These techniques are active fields of research and begin to see deployment in industry [6]. They are usually implemented using expensive sensors such as laser range finder [19], which, because of limited education budgets, often restricts their teaching to simulation. Yet simulation is not always sufficient to fully understand what is critical for correct functionality on physical robots. The e-puck, with the distance scanner extension (see Section IV-B and Fig. 5, left), allows students to explore these techniques in a physical setup.

VI. Evaluation of e-puck by the students

Our embedded programming course using the e-puck has been evaluated on a regular basis since the introduction of the e-puck in 2005. The goal of this course is the understanding and practice of embedded programming. Most of the course focuses on C programming, with links to assembler and some concepts of C++. Practicals explore signal processing and control problems. The students who attend this course are not specialized in robotics. All the programming exercises of this course are performed using the e-puck as a motivation tool.

Over the first three years of e-puck use, students were asked to give feedback about the use of the e-puck to illustrate the concepts of the course. Fig. 8 summarizes the results of this analysis and shows that more than 90 percent of the students agree that the e-puck is a good tool to illustrate the concepts of the course. The score has slightly improved over the years, with the fine-tuning of the course.
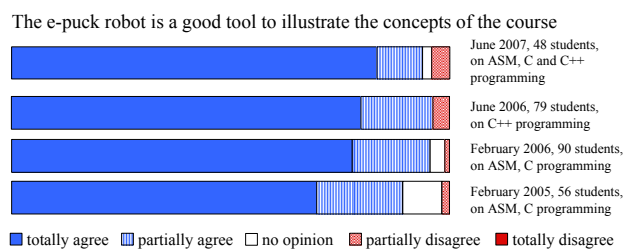

Fig. 8.   Comparison between the evaluations of the e-puck robot as teaching tool.

When asked about the quality of the e-puck, students gave the feedback shown in Fig. 9. These results show that the latest version of e-puck is considered as performing well by more than 90 percent of the students. We observe a clear improvement since the first version evaluated in February 2005.

VII. Conclusion

The e-puck robot is an innovative education tool for engineering in a broad sense. For its size and price, it is a complex systems that we exploit to teach a wide range of topics. By integrating the latest technologies in a compact

The e-puck robot is performing well

June 2007,
48 students, on V2

June 2006,
79 students, on V2

February 2006,
90 students, on V2

February 2005,
56 students, on V1

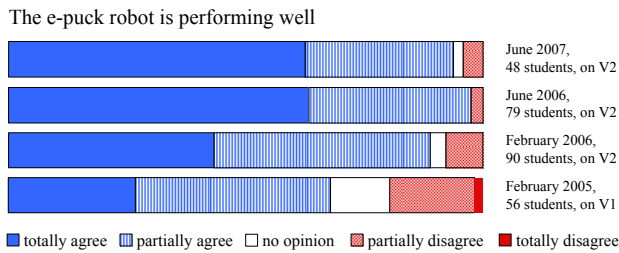■ totally agree ▥ partially agree ☐ no opinion ▨ partially disagree ■ totally disagree

Fig. 9. Comparison between the evaluation of the quality of the e-puck robot and its on-board software, following its evolution over the years and versions.

design, it repositions robotics in the engineering education landscape. More than 200 students over 3 years have validated this concept and shown their satisfaction using the e-puck. The open-source nature of this robot improves the quality of the support to the students by providing full access to knowledge at every level. For teachers this simplifies maintenance and opens new experimentation possibilities. Finally, the rapid diffusion of the e-puck in the research community shows its versatility as a scientific experimentation tool.

VIII. ACKNOWLEDGMENT

REFERENCES

[1] Alberto Acerbi, Davide Marocco, and Stefano Nolfi. Social facilitation on the development of foraging behaviors in a population of autonomous robots. *Advances in Artificial Life*, 4648:625–634, 2007.

[2] Saeed Amizadeh, Majid Nili Ahmadabadi, Babak N. Araabi, and Roland Siegwart. A bayesian approach to conceptualization using reinforcement learning. *Advanced intelligent mechatronics, 2007 ieee/asme international conference on*, pages 1–7, 2007.

[3] Rodney A. Brooks. *Cambrian Intelligence*. MIT Press, 1999.

[4] Christopher M. Cianci, Xavier Raemy, Jim Pugh, and Alcherio Martinoli. Communication in a Swarm of Miniature Robots: The e-Puck as an Educational Tool for Swarm Robotics. In *Simulation of Adaptive Behavior (SAB-2006), Swarm Robotics Workshop*, pages 103–115, 2006, Lecture Notes in Computer Science (2007), vol. 4433.

[5] Roozbeh Daneshvar, Abdolhossein Sadeghi Marascht, Hossein Aminaiee, and Caro Lucas. *A Quantitative Investigation into Distribution of Memory and Learning in Multi Agent Systems with Implicit Communications*, volume 4850, pages 124–133. Springer, 2007.

[6] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *Robotics and Automation Magazine, IEEE*, 13(2):99–110, June 2006.

[7] Dario Floreano, Sara Mitri, Stéphane Magnenat, and Laurent Keller. Evolutionary Conditions for the Emergence of Communication in Robots. *Current Biology*, 17:514–519, 2007.

[8] N. Franceschini, J. M. Pichon, C. Blanes, and J. M. Brady. From insect vision to robot vision [and discussion]. *Philosophical Transactions: Biological Sciences*, 337(1281):283–294, 1992.

[9] O. Gigliotta and S. Nolfi. Formation of spatial representations in evolving autonomous robots. *Artificial Life, 2007. ALIFE '07. IEEE Symposium on*, pages 171–178, 2007.

[10] Babak N. Araabi Hadi Firouzi, Majid Nili Ahmadabadi. A probabilistic reinforcement-based approach to conceptualization. *International Journal of Intelligent Technology*, 3:48–55, 2008.

[11] Daisuke Kurabayashi Herianto, Toshiki Sakakibara. Artificial pheromone system using rfid for navigation of autonomous robots. *Journal of Bionic Engineering*, 4:245–253, 2007.

[12] Mattias Jacobsson, Sara Ljungblad, Johan Bodin, Jeffrey Knurek, and Lars Erik Holmquist. Glowbots: robots that evolve relationships. In *SIGGRAPH '07: ACM SIGGRAPH 2007 emerging technologies*, page 7, New York, NY, USA, 2007. ACM.

[13] Olivier Michel. Webots: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems*, 1(1):39–42, 2004.

[14] Yasser F. O. Mohammad and Toyoaki Nishida. *TalkBack: Feedback from a Miniature Robot*, pages 357–366. Springer, 2007.

[15] F. Mondada, E. Franzi, and P. Ienne. Mobile robot miniaturization: A tool for investigation in control algorithms. In T. Yoshikawa and F. Miyazaki, editors, *Proceedings of the Third International Symposium on Simulation on Experimental Robotics (ISER-93)*, volume 200 of *Lecture Notes in Control and Information Sciences*, pages 501–513. Springer, 1993.

[16] R.R. Murphy. "competing" for a robotics education. *Robotics and Automation Magazine, IEEE*, 8(2):44–55, Jun 2001.

[17] F. Rastegar and M.N. Ahmadabadi. Grounding abstraction in sensory experience. *Advanced intelligent mechatronics, 2007 ieee/asme international conference on*, pages 1–8, 2007.

[18] P. Roduit, A. Martinoli, and J. Jacot. A quantitative method for comparing trajectories of mobile robots using point distribution models. *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 2441–2448, 2007.

[19] Roland Siegwart and Illah R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.

[20] Vlad M. Trifa, Christopher M. Cianci, and Dominique Guinard. Dynamic Control of a Robotic Swarm using a Service-Oriented Architecture. In *13th International Symposium on Artificial Life and Robotics (AROB 2008)*, 2008.

[21] Jean-Christophe Zufferey and Dario Floreano. Toward 30-gram Autonomous Indoor Aircraft: Vision-based Obstacle Avoidance and Altitude Control. In *IEEE International Conference on Robotics and Automation (ICRA'2005)*, pages 2594– 2599, 2005.