EmbedIT – an Open Embedded Systems Kit for Education

Dorit Assaf
Artificial Intelligence Laboratory, University of Zurich,
Zurich, Switzerland

and

Rolf Pfeifer
Artificial Intelligence Laboratory, University of Zurich,
Zurich, Switzerland

ABSTRACT

Students today are a generation of consumers of technological devices and software. They tend to consider themselves as technology savvy and digital natives. However, this pervasive use of technology seems to encourage mastering the usage of a device rather than the interest to understand the technical details behind it. This might help to explain the decreasing number of enrollments in science and technology disciplines at Universities in the USA as well as in Europe. In this paper we introduce our concept of promoting interest in science and technology through understanding technological gadgets young people are familiar with. We introduce an implementation of this concept, an open embedded systems kit for education (EmbedIT) which currently is under development. Unlike common educational robot kits EmbedIT enables students to access the technical world in a non-engineering focused way. Through a graphical user interface students can play with sensors and actuators and are able to easily implement technological objects using them. We believe that once fascination and a basic understanding of technology has been established, the barrier to learn more advanced topics such as programming and electronics is lowered. Further we describe the hardware and software of EmbedIT, the current state of implementation, and possible applications.

Keywords: Educational Technology, STEM Education, Embedded Systems, Educational Robotics, Edutainment, User /Computer Interaction.

1. INTRODUCTION

There has been a rapid development of information and communication technology in the last decades that highly influenced our society. Nowadays young people are exposed to computers, the Internet and technological gadgets such as smart phones and digital cameras. The availability and functionality of these devices are taken for granted. Informatics and engineering is seen as the sciences that drive the progress in these technologies. In order to maintain economical competitiveness, education in these disciplines is crucial. However, the number of students showing interest in pursuing a career in Science, Technology, Engineering and/or Math (STEM) is decreasing in the USA [1] as well as in Europe [2]. This phenomena has gained a lot of attention from teachers, researchers, politicians, authorities and other stakeholders. Consequently, a vast number of reform curricula and initiatives have been developed to improve the situation, but definitive solutions have not yet emerged. A popular educational approach to motivate young people in learning is based on the constructivist / constructionist paradigm, where learning through play can contribute to the construction of knowledge [3][4]. Active learning environments through the use of interactive lessons, friendly competitions, and trial and error are therefore preferred [5].

Students nowadays are used to technological devices and software and they consider themselves as technology savvy and digital natives [6]. However, even though these students are perfectly able to use these tools, they don't really understand the underlying technology. This pervasive use of technology seems to encourage mastering skills rather than understanding. Students should not only be users but rather intentional learners and co-authors of own learning tools [7][8].

In this paper we describe an open embedded systems kit (EmbedIT) that we are currently developing. The aim is to provide an open-source platform that enables teachers and students to access the world of technological gadgets from a non purely end-user perspective. Embedded systems are omnipresent, they are hidden inside TVs, media players, remote controls, coffee machines, smart phones, cars and robots. However, to work with microcontrollers, sensors and actuators, skills in programming, electronics and mechanical engineering are required. Consequently, many toolkits, mainly robotic kits, have been developed in research projects as well as in commercial companies. The field of educational robotics is growing due to the fact that robots are a proven effective learning tool [9]. A popular robot kit used in educational robotics courses is the LEGO NXT. Even though this platform is very easy to use, it focuses on a classical top-down engineering approach, where the actual robot has to be planned, built out of predefined

parts and programmed using classical programming routines. Many other educational robotic platforms require advanced programming skills in C/C++.
With EmbedIT we try to overcome the constraints of classical engineering focused toolkits. We believe since young people are used to user-friendly applications in their everyday life, we should incorporate such interfaces in our kit. The user should be able to play with sensors and actuators without having to deal with the technical details. Furthermore, he should be able to build and implement a technical object within a short time, and consequently, while testing and revising it, getting a deeper understanding about these sensors and actuators. We believe that once fascination and a basic understanding of technology has been established, motivation is increased to get familiar with more advanced topics such as programming and electronics. However, we don't solely target one group of users. EmbedIT is completely open-source and therefore technically more skilled users can reprogram and adapt the platform as they wish.

In the rest of this paper, besides the related work, the EmbedIT's hardware and software is described. This is followed by possible applications, discussion, conclusion and future work.

2. RELATED WORK

Robots have been used in the last decade to introduce kids and especially girls [10] to science and technology [11]. Class activities with robots range from kindergarten to high secondary school. A large number of robot competitions emerged such as the FIRST Lego League or RoboCupJunior, all with the aim to engage young people in these disciplines [9]. A widely used robotic platform for educational robotics is the LEGO NXT [12][13].
Other educational robotic platforms [14] use the popular Arduino boards [15]. We also used a small custom made wheeled robot based on the Arduino board to teach robotics to secondary school teachers [16]. Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. Furthermore, the Arduino project provides custom C/C++ libraries to program the on board microcontroller. Hardware design and software is open-source and a large community grew around the platform contributing libraries, tutorials, hardware designs etc. Meanwhile, several Arduino control board designs have been developed, each suited for different applications. However, even though there are plenty of well documented online tutorials, skills in electronics are needed in order to attach sensors and actuators to these boards (which have to be purchased separately). The Arduino boards can be damaged easily by connecting additional components in the wrong way.
A fairly new project, but very related to the EmbedIT platform is .NET Gadgeteer of Microsoft research [17]. It is a rapid prototyping platform for small electronic gadgets and embedded hardware devices. It combines object-oriented programming and solderless assembly of electronics using a kit of hardware modules. Unlike the Arduino platform .NET Gadgeteer provides all kinds of sensors and actuators that can be plugged into the main board. This prevents problems caused by wrong connections. Phidgets offers similar to .NET Gadgeteer a variety of sensors and motor controller modules which are all connected through an USB interface to the computer [18]. Additionally, the Phidgets libraries support the most common programming languages.
Nevertheless, Arduino, .NET Gadgeteer as well as Phidgets require advanced programming skills.

3. EmbedIT – AN OPEN EMBEDDED SYSTEMS KIT

The components of the EmbedIT platform are hardware modules consisting of small embedded systems. A Java application, the EmbedITApp, serves as the user interface to these hardware modules. To increase usefulness and usability, we are including students' feedback within the design process. Our approach is to test the prototypes with students in class at an early stage of development. Based on several studies about how e-infrastructures and technologies should be designed so that learners respond to them positively and consequently learning effectiveness is increased, we try to reflect the learning goals, the learning environment and the digital culture of the students in our system [19][20][21].
EmbedIT's hardware designs and source codes are completely open and are free available on the project website [22]. Inspired by the Arduino project, we would like to enable a community to extend and adapt the hardware and software once it is released. The main goal is to encourage users from all kinds of backgrounds to use this kit. By means of the EmbedITApp users with no background in electronics or programming are able to access and control the modules. In addition, more advanced users have the possibility to extend the source code by themselves. Since the hardware and software interfaces are open and common standards are used, it is possible to interface the modules with other kits or projects, for example with an Arduino board. By adding the required CAN components to the Arduino board and including our CAN library on the software side, the board is easily connected with EmbedIT.

EmbedIT's hardware and software

The EmbedIT kit provides hardware modules which are individual printed circuit boards equipped with a microcontroller (Atmel AVR). Each module is an independent entity that serves either a control, sensory, actuation or communication purpose. According to a module's purpose a specific control program is running on its microcontroller. Additionally, each module carries its own unique identification number. A CAN data bus (two conductor cable) is the connecting piece between the modules. Each module can easily be attached and detached from the bus. The master module, which is a control module, controls the communication traffic on the bus. In addition, the master module serves as an interface to the user. The EmbedITApp running on the user's PC connects to the master module through a communication module, that can either be a Bluetooth, USB or serial (RS232) interface. In Figure 1 the master module is attached to the CAN bus and a Bluetooth module is attached to the master module (communication modules are not attached to the bus directly, they have to be attached to a module). The EmbedITApp connects to the master module through the Bluetooth communication module. Additionally in Figure1, one actuation module (servo) and two sensor modules (accelerometer, light sensor) are connected to the bus.

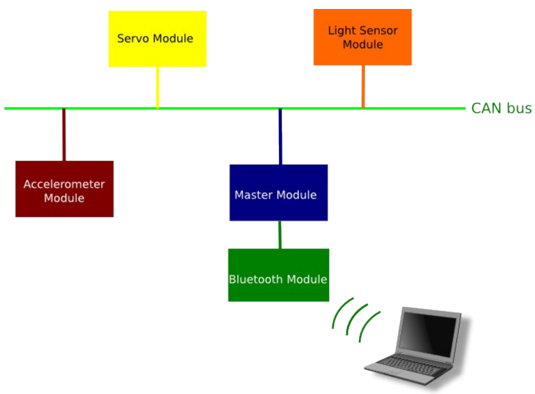

Figure 1. A master module is attached to the CAN bus and a Bluetooth module is attached to the master module. The EmbedITApp running on the PC connects to the master module through the Bluetooth communication module. Additionally, a servo module, an accelerometer module and a light sensor module are connected to the bus.

Figure 2 shows the real modules, a Bluetooth module, a master module, the CAN bus, a power supply and a servo module with one servo attached. Once a connection between the EmbedITApp and the master module has been established, the master module returns the ID and module type (sensor type, actuator type) of all hardware modules that are physically connected to the bus.

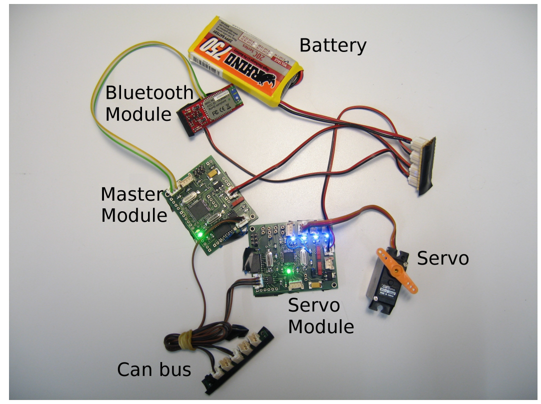

Figure 2. Some EmbedIT modules: A Bluetooth module, a master module, the CAN bus, a power supply and a servo module with one servo attached.

Figure 3 shows an example of the EmbedITApp where a servo module, a light sensor module and an accelerometer module are listed. Each available module is represented with its ID, icon and a button within the application. While clicking on a module's button, its control panel opens.

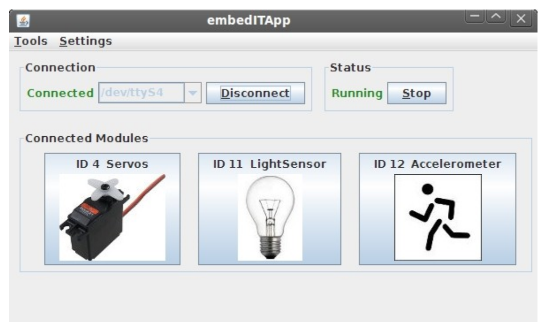

Figure 3. An example of the EmbedITApp where a servo module, a light sensor module and an accelerometer module are listed. Each available module in the application is represented with its ID, icon and a button.

Figure 4 shows the control panel of the servo module. The servo module is able to control four servos at the same time. From this control panel commands such as connect/disconnect servos, move to specific servo position can be applied. The EmbedITApp sends the commands to the master module which forwards them to the target module on the bus. Servo positions can be changed while dragging the servo position sliders. After pushing the apply button on the control panel, the servo positions are changed physically.

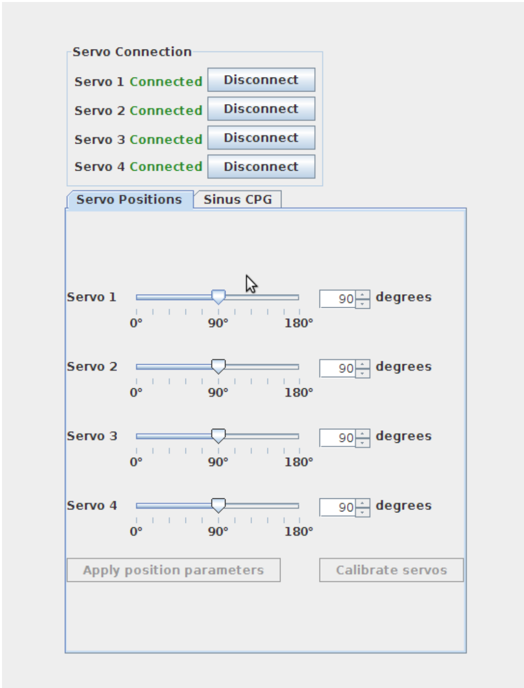

Figure 4. The control panel of the servo module. The servo module is able to control four servos at the same time. From this control panel commands such as connect/disconnect servos, move to specific servo position can be applied.

For each module on the bus a control panel can be opened where module specific commands and settings can be applied. Through the plug and play nature of the hardware modules on the bus as well as the possibility to acquire sensor data and to control modules through a GUI, usability increases significantly. With other platforms a user has to program a lot of code in order to be able to do similar tasks. At this state the user is able to remote control and monitor the hardware. By means of a graphical programming environment which is currently under development the user will be able to define the control of sensors and actuators such that the hardware will be able to run autonomously. It has to be mentioned that the preprogrammed source code of each EmbedIT module is not hidden. A more experienced user can easily change code and upload it to the modules.

4. EXAMPLE APPLICATIONS

As we are currently developing the hardware and software of EmbedIT, more modules are being added to the kit. Currently common sensor modules exist such as light sensors, accelerometers, potentiometers, gyroscopes. The goal is to have a repertoire of many different kinds of sensors and actuators, also unusual ones such as shape memory alloy, whisker sensors [23] etc. There are many possible applications for this kit and they are not limited to education only. Other uses could be e.g. a rapid prototyping tool for researchers as well as a "playground" for hobbyists.

For education it can be used for instance in a Physics class to explain how sensors work, in Math to apply functions to filter sensor data, in art to build interactive installations. Gyroscopes and accelerometers for instance are widely used for motion sensing and stabilization in devices such as mobile phones, digital cameras, and gaming consoles. This would most likely raise students interests to learn about how these sensors work and eventually lead to building own applications using them.

A skating robot controlled with EmbedIT

An example application is a skating robot that is controlled by EmbedIT (Figure 5). In the context of an informatics study program for secondary school teachers we had the opportunity to test the modules. The robot's mechanical structure has been built by a teacher (with help of a technician) and the electronics were provided by the EmbedIT kit.

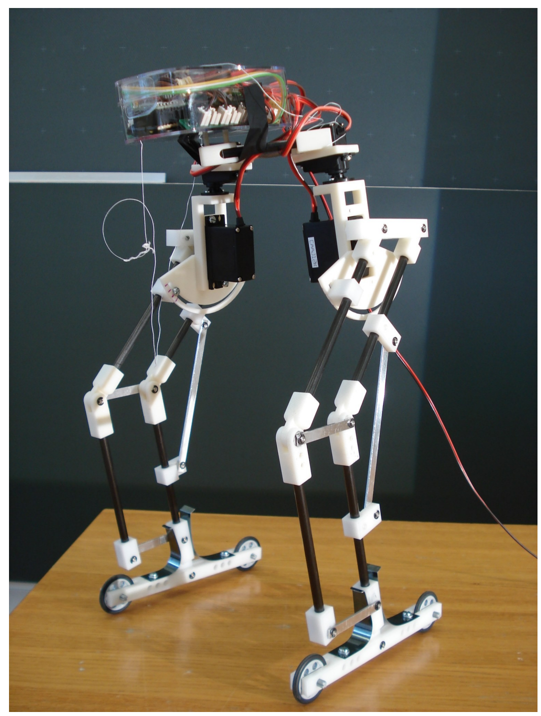

Figure 5. A skating robot controlled with the EmbedIT modules. The two servos on the hips control the bending and stretching movement of the legs and the two other servos adjust the angle between the feet.

This robot used the configuration seen in Figure 2, a servo module (four servos attached), a Bluetooth module and a master module. The servo control panel of the EmbedITApp provides next to discrete positioning of the servos simple functions such as a continuous position change in a sinusoidal way (alternating from 0° to 180°). The user has the possibility to apply different sinus parameters such as amplitude (servo angle range), offset (shift of the servo middle axis), phase lag (synchronization of all servos) and frequency. By means of this user friendly control panel the teacher was able to tune the sinus parameters and apply them immediately on the robot such that he could test and successfully proof his hypothesis: The robot should achieve a forward skating movement while alternately relocating its center of mass from one leg to the other (that, in a simplified way, is what skaters actually do). This is an example that shows how easy and with no programming knowledge one can already control a robot.

5. EVALUATION

During the ongoing development of the EmbedIT kit we are testing and evaluating it in terms of usability at an early stage in the design process. Our approach is to enable students to build technological objects easily and thus to raise motivation through the feeling of accomplishment. Consequently, we hope to raise interest to pass on to more advanced topics such as programming or electronics which can be explored with the kit as well. In order to evaluate the approach workshops using EmbedIT in upper secondary schools are planned. A survey will be conducted to investigate the impact EmbedIT produces upon students, regarding change of attitude and self-confidence towards studying technology.

6. DISCUSSION

EmbedIT differs from other robotic or embedded systems kits with respect to usability and user-friendliness. Our goal is to enable users from different kinds of backgrounds and interests to explore the world of technology while accessing it in a creative, playful and non-engineering focused way. We believe that building own technological objects and gadgets results in a basic understanding about how they work. This can lower the barrier to move on to more advanced topics of engineering such as programming and electronics.

EmbedIT is contributing to education in such a way that it provides an alternative to common robot kits on the market. The goal is the same – teaching technology – but the approach is different. A technological gap between teachers and students exist since many teachers are not familiar with technological devices in the way their students are. The user-friendliness of EmbedIT can encourage teachers to incorporate it in class as a learning tool and consequently, may help to overcome that technological gap. We intentionally chose EmbedIT to be open source to encourage the growth of a contributing community. This could consequently lead to the availability of cheap hardware from which schools can profit as well.

7. CONCLUSION AND FUTURE WORK

In this paper we referred to the problem of decreasing enrollments in science and technology disciplines in Universities. Even though young people nowadays are exposed to technological gadgets and are using them with enthusiasm, little is understood by them about the technical details. We introduced our concept of promoting interest in science and technology through the understanding of these gadgets. To achieve this, we are developing EmbedIT, an open embedded systems kit that enables users to build technological objects in an easy and user-friendly way. This can be highly motivating and may lower the barrier to move on to more advanced topics in engineering.

Right now we are developing, testing and revising the hardware and software of EmbedIT. More modules have to be added to the repository. At the moment servos can be controlled through the EmbedITApp interface and sensor data can be read. To combine sensor data and servo commands a graphical user interface has to be implemented. In addition, interface solutions for attaching hardware modules to mechanical structures (for instance by using mechanical toolkits) have to be developed. Since we are interested in including the end-users feedback in our design process, it is important to communicate and publish the project idea as well as to test the kit with students at an early stage. Once the EmbedIT hardware and software achieves a first stable release state, it will be available under an open-source license on the project website [22].

REFERENCES

1. Miller, J., Ward, B., Sienkiewicz, F., Antonucci, P. "ITEAMS: an Out-Of-School Time Program to Promote Gain in Fundamental Science Content and Enhance Interest in STEM Careers for Middle School Students." **In: Proceedings of The 8th Intl. Conf. on Education and Information Systems, Technologies and Applications (EISTA 2010)**
2. Rocard, M. et al. **Science Education now: a renewed pedagogy for the future of Europe.** EUR22845, ISBN 978-92-79-05659-8, European Communities, Brussels (2007)
3. Piaget, J. **The origins of intelligence in children.** New York: International Universities Press. (1952)
4. Piaget, J. **The construction of reality in the child.** New York: Basic Books. (1954)
5. Sardone, N. "Developing Information Technology (IT) Fluency in College Students: An Investigation of Learning Environments and Learner Characteristics." **In: Proceedings of The 8th Intl. Conf. on Education and Information Systems, Technologies and Applications (EISTA 2010)**
6. Borka, M. "Technology and Teacher Education: Integration in Context." **In: Proceedings of The 8th Intl. Conf. on**

Education and Information Systems, Technologies and Applications (EISTA 2010)

7. Gaudiello, I., Zibetti, E., Carrignon, S. “Representations to go: learning robotics, learning by robotics.” **In Workshop Proceedings of Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2010)**, pp. 484-493, ISBN 978-3-00-032863-3
8. Bollaert, H. “Holistic Embedding of Interaction Generating Learning Objects.” **In: Proceedings of The 8th Intl. Conf. on Education and Information Systems, Technologies and Applications (EISTA 2010)**
9. Bredenfeld, A., Hofmann, A., Steinbauer, G. “Robotics in Education Initiatives in Europe – Status, Shortcomings and Open Questions.” **In Workshop Proceedings of Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2010)**, pp. 568-574, ISBN 978-3-00-032863-3
10. Bredenfeld, A., Leimbach, T. “The Roberta Initiative.” **In Workshop Proceedings of Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2010)**, pp. 558-567, ISBN 978-3-00-032863-3
11. Alimisis, D., Moro, M., Arlegui, J., Frangou, S., Papanikolaou, K. “Robotics & Constructivism in Education: the TERECoP project.” **In Proceedings of EuroLogo 2007**, Bratislava, (2007)
12. Kabátová, M., Pekárová, J. “Lessons learnt with LEGO Mindstorms: from beginner to teaching robotics.” **AT&P Journal Plus 2, Robotics in Education**, pp. 51-56, ISSN 1336-5010, (2010)
13. Menegatti, E., Moro, M. “Educational Robotics from high-school to Master of Science.” **In Workshop Proceedings of Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2010)**, pp. 639-648, ISBN 978-3-00-032863-3
14. Balogh, R. “Acrob – an Educational Robotic Platform.” **AT&P Journal Plus 2, Robotics in Education**, pp. 6-9, ISSN 1336-5010, (2010)
15. The Arduino project. http://www.arduino.cc/
16. Assaf, D., Aryananda, L., Pfeifer, R. “Robot Competition with Teachers.” **In Workshop Proceedings of Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2010)**, pp. 514-521, ISBN 978-3-00-032863-3
17. The Gadgeteer project. http://research.microsoft.com /en-us/projects/gadgeteer/
18. Phidgets. http://www.phidgets.com/
19. Borka, M. “Technology and Teacher Education: Integration in Context.” **In: Proceedings of The 8th Intl. Conf. on Education and Information Systems, Technologies and Applications (EISTA 2010)**
20. Sowden, D. “Delivering User-Centric Esystems.” **In: Proceedings of The 8th Intl. Conf. on Education and Information Systems, Technologies and Applications (EISTA 2010)**
21. Mendoza, A., Stern, L., Carroll, J. “Learnability as a Positive Influence on Technology Use.” **In: Proceedings of The 8th Intl. Conf. on Education and Information Systems, Technologies and Applications (EISTA 2010)**
22. The EmbedIT project. http://www.embed-it.ch
23. Lungarella, M., Hafner, V., Pfeifer, R., Yokoi, H. “Artificial Whisker Sensors in Robotics.” **In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**, (2002)