

Holistic Web-based Virtual Micro Controller Framework for Research and Education

<http://dx.doi.org/10.3991/ijoe.v8i4.2169>

Sven Seiler¹, Raivo Sell², Dennis Ptasiak¹, Martin Bölker¹

¹ Bochum University of Applied Sciences, Bochum, Germany

² Tallinn University of Technology, Tallinn, Estonia

Abstract—Education in the field of embedded system programming became an even more important aspect in the qualification of young engineers during the last decade. This development is accompanied by a rapidly increasing complexity of the software environments used with such devices. Therefore a qualified and solid teaching methodology is necessary, accompanied by industry driven technological innovation with an emphasis on programming. As part of three European projects regarding lifelong-learning a comprehensive blended learning concept for teaching embedded systems and robotics was developed by paper authors. It comprises basic exercises in micro controller programming up to high-level student robotic challenges. These implemented measures are supported by a distance learning environment. The programming of embedded systems and microcontroller technology has to be seen as the precursor for more complex robotic systems in this context, but with a high importance for later successfully working with the technology for further professional utilization with these technologies. Current paper introduces the most novel part; the online accessible Virtual Micro Controller Platform (VMCU) and its underlying simulation framework platform. This approach conquers the major existing problems in engineering education: outdated hardware and limited lab times. This paper answers the question about advantages of using virtual hardware in an educational environment.

Index Terms—Distance learning, Computer science education, Remote systems, Robotics, framework, virtual micro controller

I. INTRODUCTION

Embedded system and microcontroller field received a lot of attention over the last years and their importance is still increasing. This seems to be a logical process as embedded devices themselves have entered into everyday life, where smart products are already widely spread in many homes. Most contemporary electronic devices at home and in industry are embedded devices by their nature. This means, they consist of software in addition to mechanics and electronics, controlling the device behavior and offering functionality. The software aspect to this is continuously growing [1] in modern products. The programming of such devices, particularly in relation to conventional computer programming methods, is becoming more common. Due to the fact that almost every modern machine consists of embedded devices there is a high need to educate more people to design, develop and maintain these kinds of systems [2] on a professional level. One problem in microcontroller education is that (often) special hardware is required, which is (in total

costs) quite expensive for laboratories. Also the risk of them being destroyed by microcontroller beginners is quite high (over-voltage, wrong fuses amongst others). These issues led to the approach of developing an educational concept to counter the gap between rapid developments in technology and the state of knowledge of young engineers [3].

Since 2007 new conceptual blended learning solution is developed which have resulted a complete set of embedded hardware, named Robotic HomeLab kit, remote access to this hardware, named DistanceLab and virtualized microcontroller unit with virtualized peripheral electrical components, named Virtual Micro Controller (VMCU). All solutions are accompanied by several add-on modules for different engineering sectors (for instance Electrical Engineering, Automotive and Mechatronics). The solution is supported by Blended Robotic Teaching & Learning Concept [4, 5], completed by ready-to-use pedagogical material, distance access to hardware, and miscellaneous hands-on online learning material.

The integration of virtualized and remote laboratory solutions into the classical teaching processes within the fields of mechatronics, electronics and computer science became a major topic within the scientific community and among major stakeholders. The most important reasons for this development are the constant innovations in technology which make it quite hard for educational institutions to keep their lab equipment up to date. The application of distance lab technology raises also the mobility of students as well as the accessibility of lab equipment for students abroad, special needs learners or people in lifelong learning who have very limited access to classical lab.

To cover a full approach suitable for implementation into the educational routine, more than just pure technology is required. Our goal is to provide a concept to link the classical approach with distance education concepts and technology. To reach this goal, the VMCU and virtual hardware framework presented in this paper will be based on the existing real hardware components that are already established in education. Main strength of the concept is, that the learning material provided for the lab work will be exactly the same for real or distance labs. In the daily teaching process, virtualized and remote labs will become an auxiliary tool but will never fully replace hand on lab experience. The virtualized microcontroller must be seen as a supporting technology in this development. The didactical materials are set to cover a full approach utilizing all technologies (remote-, distance- and real labs).

Latest development in this modular and extendible concept and main focus of this paper, which technical part is illustrated in Figure 1, is a fully virtualized microcontroller system, named Virtual Micro Controller Unit (VMCU) and virtualized pin connections with external electronic components. The next section will pick up on the VMCU environment and its applications, concluding with further development prospects. In the third section, “Lab Description Language” (LDL), a dynamic approach for describing and mediating remote and virtual labs is introduced, followed by a section about the VirtualLab Environment. The last section shortly illustrates the virtual solution counterpart – a DistanceLab platform and describes the concept of concurrent use of both platforms.

II. VIRTUAL MICRO CONTROLLER UNIT

The newest part of authors blended learning concept [4, 5, 6] is the so named Virtual Micro Controller (VMCU) [7, 8], which is patterned on the Robotic HomeLab kit [9] hardware. It is a fully featured, but simulated web environment behaving like real hardware. VMCU is based on Avrora framework [10, 11] and Java programming language. It can be used for prototyping and research experiment as well as for educational purposes. The Robotic HomeLab kit, the robots as well as the VMCU are accessible over a web portal, called DistanceLab.

It interfaces remote laboratories and virtual experimentations that can be programmed and observed over the Internet. The system works with camera feedback. In addition to the hardware, the concept is offering additional material, with a lot of exercises for the real hardware and also for the VMCU.

The idea of the concept is to utilize the VMCU in the beginning of educating microcontroller technology or for self-studies at home, as only a modern web browser with Java enabled is needed to access the virtual hardware.

A. Design Conditions

For satisfying all needs to build products for an educational environment and reaching the desired high level of attractiveness to fascinate young people for this technology, it is important for the system design to rest on

web technology. Therefore the available tools appropriate for the development are limited.

First requirement formulated by the authors was to research a system based on modern Internet technology, running as a single web platform, accessible by any common web browser. Second condition was having the VMCU to work with binary files, so common C programming language development software can be interfaced to the VMCU environment. Author’s third framework condition concerns the cost of the system. The intention was to minimize costs as much as possible, without any annual/recurring fees, to encourage its acceptance. Forth requirement is about the demonstration purposes. For attractiveness reasons a graphical version of the controller is necessary; textual output only will not be sufficient. As fifth condition the general behavior of the VMCU environment should be comparable to real hardware belonging to the Robotic HomeLab kit. The last requirement is to enable the VMCU being included into the DistanceLab environment.

B. Interface / Inter-Connectivity concept

To create a microcontroller test stand, different environmental values and hardware components have to be taken into account. The main part is the microcontroller chip itself, as this chip deals with nearly all operations of a test stand. The deep state-of-the-art research prior to the VMCU development (described in detail in [10]) resulted, that Avrora framework is the best base micro controller simulation solution for given task, as it is Java based and freely available as Open Source, which fits the demands for developing it further in a modern web environment.

As the initial structure of the system and model is based on the Avrora framework the programming language Java was chosen, too. Because of this and the modular structure of Avrora, which is separated into Java-packages, the creation of a new Java-package known as *virtualmicroclab* was the base for further development. Due to the condition that the VMCU should be based on Web 2.0 technology, applets and an inter-applet communication schema for component interconnection was the appropriate way to design the overall system.

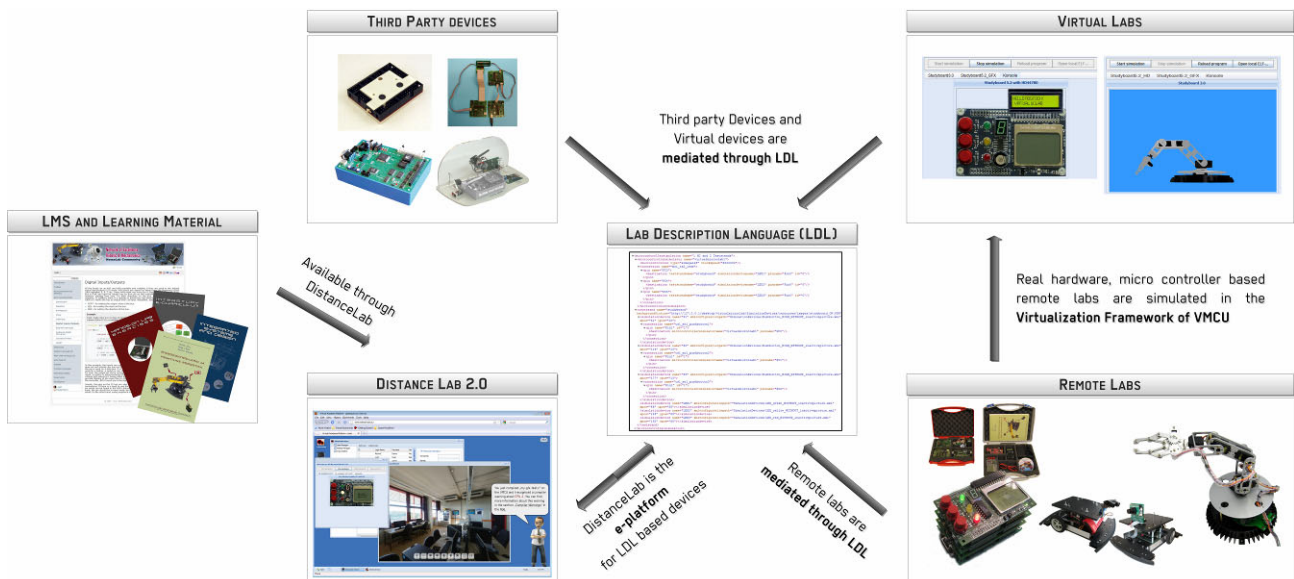


Figure 1. The VMCU in the full concept

The general system design is shown up in Figure 2. For high accuracy, each module connected to the controller handles an event in every cycle of the simulation. These events have to be implemented for each simulated hardware component. Therefore they are included as an abstract definition implementation in the interface of each device.

III. LAB DESCRIPTION LANGUAGE (LDL) FOR CONFIGURATION

All changes and extensions to Avrora, like running as an Applet are realized in a modular way to ensure a high level of integration of possible future Avrora versions into the VMCU Environment. The modular concept needs to be applied for all additional components (modules and devices) to enable their integration into more complex test stand scenarios.

Due to standardized interfaces, these modules can be combined by the user as necessary. The configuration is realized by utilizing a new description language for lab integration, called Lab Description Language (LDL) developed by the authors and presented in figure 3. LDL is based on the XML standard resting upon ontology for remote labs. So it can be easily understood by humans and machines. The configuration file is mainly used for the interconnection between virtual components like the controller or simulation modules. The arrangement of the user interfaces and graphical elements with the frontend solution are also defined in LDL. Lab Description Language allows transparent access to laboratories independent of the type of the system, because LDL allows labs/experiments to be defined in an abstract manner. LDL is similar to the RDF-based Lab2Go Description Model [12] or the LiLa Ontology [13]. While the Lab2Go approach and the Lila Ontology are focusing on easy integration of laboratories and their associated experiments, LDL is additionally able to describe components used in virtual experiments in detail. For instance, LDL in its current state holds an ontology for embedded system related electronic parts that can be virtually connected to the VMCU. The LDL approach is capable of mediating dynamically tailored experiments. For instance, a teacher may create a new VMCU module by just adding components like buttons and LEDs and virtually plug them together and share this new experiment with her/his class. Most important benefit of this approach is that a “lab creator” does not need neither advanced programming skills nor deep knowledge in electronics for creating new experiments.

LDL is meant as a potential prototype of a universal approach for specification and connection of remote and virtual labs and associated components. The consideration for adoption of such a draft idea by engineers, technologists and scientists will aid the long-term success of the networked labs and encourage participations. LDL provides the ontological basis to develop a web-based platform incorporating a comprehensive toolbox of components, component interfaces and access interfaces. The standardization of required interfaces and the development and free distribution of software components for lab integration and distribution solves the major problems of Internet accessible labs.

LDL will allow transparent access to laboratories independent of the type of the system. LDL allows labs/experiments to be defined in an abstract manner, all

objects needed to integrate existing Labs (button, editor, widget, interface to software tools, e.g., Matlab), which can represent different laboratory components, will be defined in LDL in terms of tags and attributes based on OWL2.

For instance a description of the green LED looks like the following LDL in fig. 4.

LDL in its current state can be utilized to describe the interconnection of micro controller based devices, their interfacing to remote lab platforms, such as DistanceLab. In addition LDL offers a comprehensive way to describe and mediate the interfaces of external labs to be included into a DistanceLab environment.

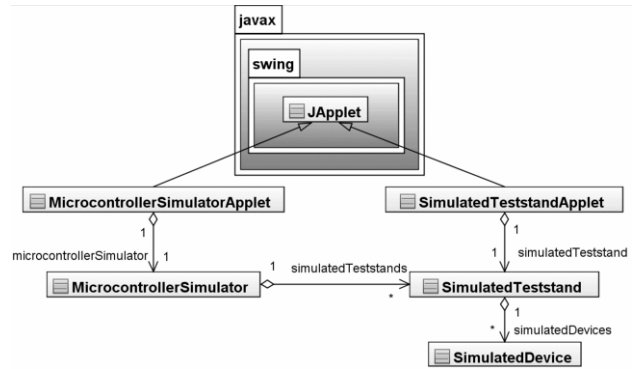


Figure 2. General VMCU System design

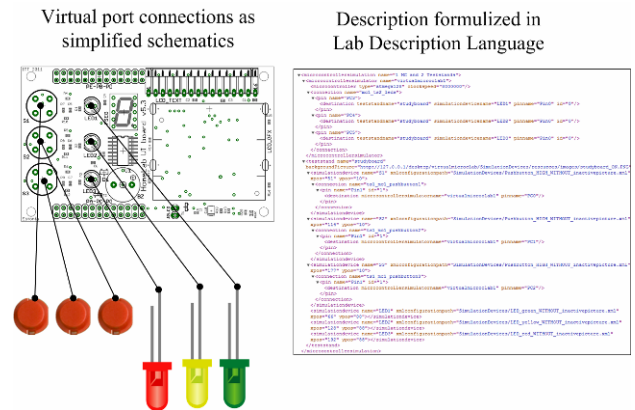


Figure 3. Description of virtual port connection in LDL

```
<simulationdevice type="LED"
  classfile="virtualmicroclab.simdev.led.LED">
<parameters>
  <parameter name="color" type="String" value="green" />
  <parameter name="connection information" type="String"
    value="add your device description here." />
</parameters>
<resources>
  <resource name="activePicture" type="image"
    path="components/electrical/LED/Green" />
  <resource name="inactivePicture" type="image"
    path="components/electrical/LED/Transparent" />
</resources>
<inputpins>
  <pin name="Pin0" id="0" />
</inputpins>
</simulationdevice >
```

Figure 4. LDL description example

A. Interface / Virtualized hardware inter-connections

For a microcontroller based simulation framework the abstraction and simulation of electrical signals and connection is a crucial part for the usability and extendibility of the whole approach.

As a microcontroller device uses input and output pins, they must be reflected in the software model. The Avrora system offers two interfaces to integrate external signals to the simulator. These interfaces define methods necessary for all external communication of the controller with the simulated environment. The output interfaces of the virtual controller require the implementation of certain events. These events are using for instance *read* and *write* methods and are very similar to common unbuffered I/O concepts. In Figure 5 the depth of the inheritance is presented. After the registration and connection of external device pins to the VMCU pins, they can interact without the need of explicit data exchange.

B. Communication in VirtualLab

The whole solution is a web browser based distributed system. Due to the utilized web technologies (HTML5) it is also run capable on modern java-enabled smart phones or tablet PCs. Various software components, developed by project partner institutions will operate in the same environment loosely connected to each other. The concept by itself sets very high requirements to the underlying communication engine. The challenge of real time embedded device simulation requires the system to grand very short response times. For some applications like sensor systems or communications it is important to have response times in the milliseconds scale. Within a single monolithic software solution this requirement would not be too hard to fulfill. Considering a distributed web based solution, where all components are programmed by different developers hosted on different servers, the real time and communication requirements turn into a crucial challenge for the system design.

The concept chosen to mediate this problem was the browser based inter-applet communication. In this concept the browser serves as host application and also as an arbiter between all components of the distributed system. The connection between all different simulation components like sensors, actors, visualization modules and the controller itself are defined by the LDL system model. As a part of the communication concept the conjunctions between the software components must be realized according the overall simulation model definition. In a final step the java representation of the virtual components are connected for inter-applet communication. To realize inter-applet communication, the applets have to be registered to each other. Based on this fact, the names of the desired parameters of the simulation device and the Test Stand applet are equal to the pin connection in the LDL configuration of each virtual test stand. The *TestStand* applet is able to search for an applet with the unique name of the microcontroller simulation device it is intended to connect with. Once found, the registration process is realized as shown in fig. 6. Another important component of the communication concept is the Communication Manager Applet. It serves as an arbiter to gain a lucid structure. Furthermore, this arbiter is able to handle the cyclic events to update the connected simulated components. As the Communication

Manager object is registered with a certain components applet, a mutual invoke of method calls is possible, as the applets are running in the same Java Virtual Machine. After successfully linking the applets and the parsing of the LDL definition, the user interface components and other parts, like simulated devices, the simulation can be started.

While starting, a cyclic event named “*processInformation*” is invoked by the Virtual Micro Controller.

C. Improvements of the Avrora system

As the original system was designed as a console based application, several improvements were necessary to use the system in a web environment. The improvements were realized following a modular concept, fitting into new Java packages and not directly in the Avrora source code to ensure lucidity and enabling easy integration of future software versions and further development into the framework. To ensure intuitive usage of the system, a Graphical User Interface (GUI) was developed, enabling control of the states of the simulation (start, stop, reload). Based on the platform concept, which offers a comfortable functionality to group microcontrollers and statically connected periphery devices to a platform, like the MicaZ mote[14] or instance, a loader has to be implemented, which is able to load virtual hardware modules dynamically into the system’s initialization procedure.

This new platform is called “*Dynamic Connection Platform*”. The necessary connection data is obtained from a corresponding LDL system model and used to generate the simulation environment. As the signal processing events have to be invoked in every cycle, their behavior has to be implemented outside the core system in each simulated device.

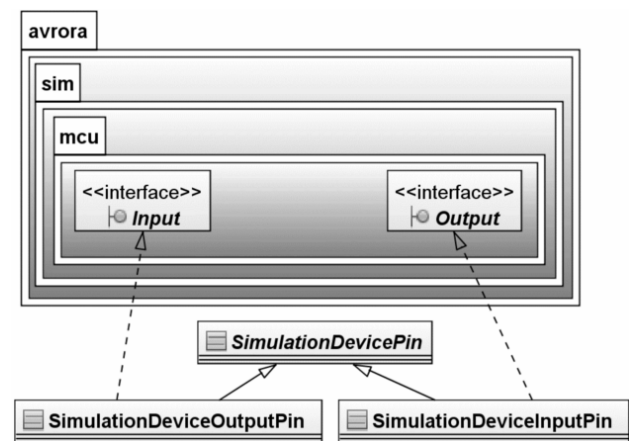


Figure 5. Virtualized I/O concept

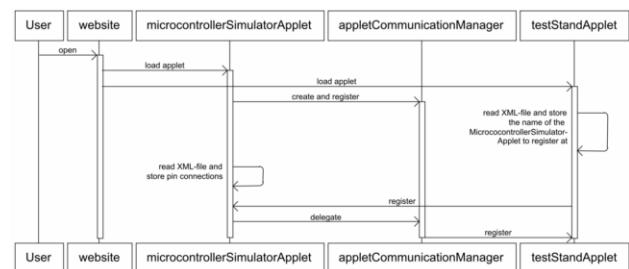


Figure 6. Sequence diagram of Communication manager

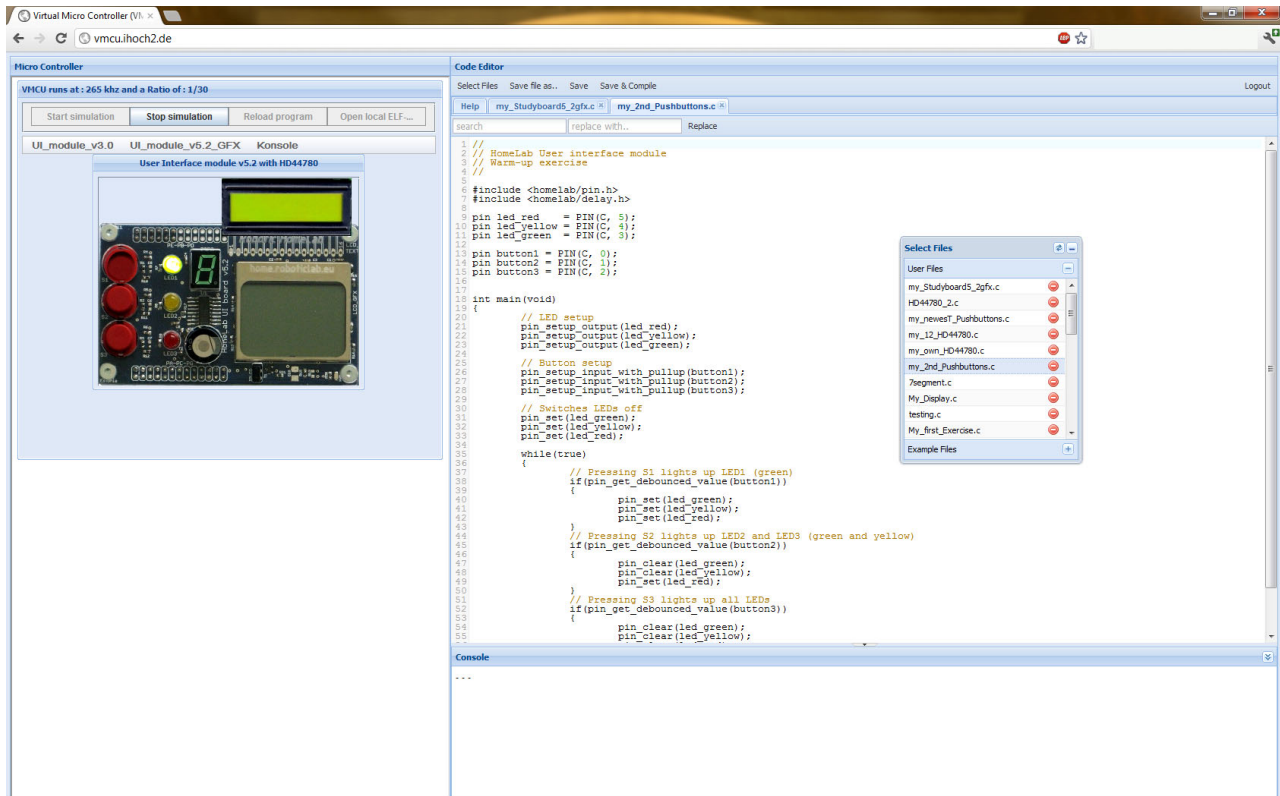


Figure 7. Virtual Lab Framework Platform, showing VMCU

IV. THE VMCU PLATFORM

The whole VMCU unit is integrated into a web environment (accessible free for use at [7]), using the ExtJS JavaScript framework and {}CodeMirror, as shown in Figure 7. The login is possible (after registration) for everyone interested, or by using Bochum University LDAP login. In the figure, the VMCU GUI is shown on the left and on the right the programming area is illustrated. The GUI offers a comprehensive development environment for the VMCU, including syntax highlighting, feedback about the compilation process and demonstration exercises, which are loaded into each new user profile. Strength of the system can be seen in the (almost) independence to operation systems, as only a few hard- and software conditions need to be fulfilled to work with the virtual system.

The controller unit itself provides buttons for controlling the simulation and an important feedback about the real speed of the current simulation compared to the internal clock speed. The behavior is like the real hardware from the Robotic HomeLab kit, but based on the LDL configuration files of the test stands; new modules may be added easily.

V. DISTANCE LAB WIRELESS PROGRAMMING

The counterpart of the presented VMCU platform is the real hardware based platform called Robotic HomeLab kit. Both technologies are accessible through a remote and virtual distance lab platform, named VAPVoS. This system is conceptually fully compatible with the distance lab approach presented in [3] [15] and [16]. However components in these advanced educational platforms are built up on virtual and real hardware which are combined with ICT systems to allow remote access to controller

programming feature and visual feedback. In that way both systems can be used simultaneously to overcome single system specific limitations.

DistanceLab platform has a two level architecture. First layer is based on Internet connected servers, located central web host and servers located in every lab. Main web server provides a web based user interface for several labs by allowing compiling the program or calculating the correct values depending on the specific lab device characteristics or interfaces. If user program has passed the validation e.g. microcontroller program is successfully compiled, program will be transferred to a program server located in lab, close to target devices, or send to a virtual hardware device through a software interface. In case of real hardware, the program server connects then with target device and identifies its state. If the device is available and active, the program server resets running program in the selected device, uploads a new program and restarts the device. When uploading is completed, system starts with new program and user can see visual feedback over the Internet connected cameras. In the case of the virtual hardware, the system is restarted on a software base and the visual feedback is directly rendered as an applet in a webpage to the browser instance.

Regarding the real hardware labs, there need to be different approaches for programming the devices. Therefore, a special remote programming solution, as presented in fig. 9 is required. Communication between the real hardware device and the program server is implemented as wireless 2.4 GHz connections. ZigBee wireless protocol is used, first on API mode to detect state of target device and then transfer mode to upload new program. The reason why wireless communication is implemented is to allow controlling not only static systems (wired systems) but also mobile systems. A solu-

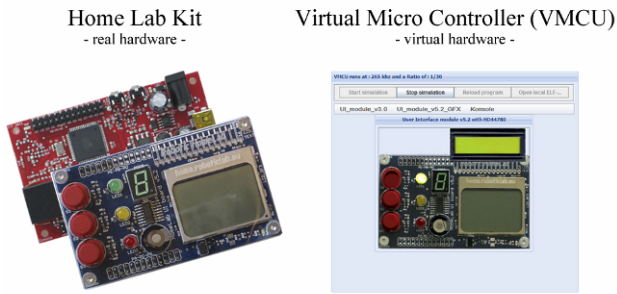


Figure 8. Robotic HomeLab kit and VMCU

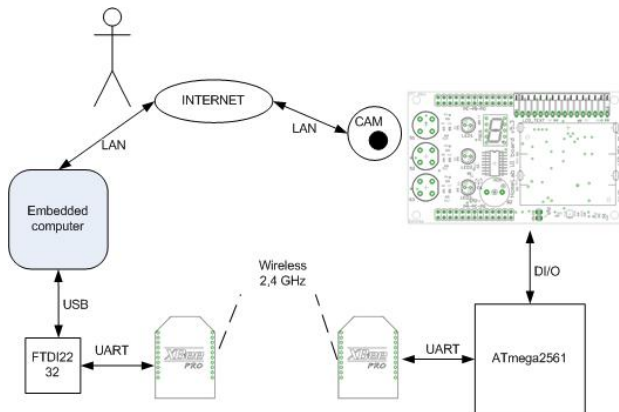


Figure 9. VMCU platform with user interface module

tion is developed to program wirelessly small mobile robots running for instance in an arena in university lab. The detail overview of mobile robotic remote lab is given in paper [17].

The benefit of combining the virtual and real remote labs is to give the extended possibility for students and researcher to study or make experiments of microcontroller based systems or experiment machinery over the distance. At first, the solution can be evaluated with the VMCU, where device instances can be multiplied as much as available computing power. After completing virtual tests, which are still in simulated world, user can move to real world platform and continue with same solution in real world environment. The limitations of using DistanceLab platform are in fact that real devices cannot be multiplied just copying them in computer. One device can be controlled only by one user at certain time. By combining presented solutions these limitations can be compensated by both platforms.

VI. DISCUSSION AND FURTHER STEPS

The structure of the VMCU, with respect to LDL as the descriptive language, offers an easy to use microcontroller simulation device that can be extended with additional microcontroller periphery devices. By utilizing the open source microcontroller simulation environment Avrora, different devices from the Atmel ATmega family can be simulated. As Avrora is written in Java, a high level of platform independence and comfortable integration into a web environment is achieved, so that larger audiences can be reached. Nevertheless the calling of information processing methods for each device in each simulation cycle leads to a performance loss. But this disadvantage also has a positive aspect, because as the simulation is working more slowly the underlying process can be better

understood, as details of the simulation may not be hidden anymore like in real hardware. By using the LDL configuration files the test stands can be created dynamically and different devices can be combined with the controller. The applets were also designed to work without an Internet connection, so the simulation device can be used without the need of permanently being online.

In the next development step the intention is to add new modules to the controller. It is also planned to create a “physics engine”, where real world physics can be simulated, like air pressure, light intensity, among others. Furthermore we intend to enable a math description of for instance temperature curves over time, so virtual sensors can be added. The first prototype for this physics engine is currently tested at Bochum University of Applied Sciences. While the Mixed Reality lab [18, 19] at the University of Bremen merges the real technical world with computerized simulation and visualization in two ways, the VMCU is only a virtualized version of the HomeLab kit, simulating its full behavior, but through LDL enhanced with the functionality of easily adding new virtual embedded system experiments without advanced technical skills.

A. Conclusion

In this paper the focus was set on a virtual, web based microcontroller learning environment.

The VMCU was realized to enable an easy implementation of new test stands and microcontroller periphery, like LCD screens, LEDs or even more complex motor and sensor modules.

Through the research, design and development and the following pilots runs in Estonia and Germany, authors asked about feedback about the products. As the pilot is still running and no final feedback is published, the interest expressed is really convincing to work further on innovative learning material for education.

From a group of currently 38 people answering the survey, to the question “Is the VMCU helping you learning micro controller technology”, currently 74% agree, while 8% disagree where 18% do not know yet. Asking about “Is the VMCU a good development you would like to tell your colleagues about?” currently 79% fully agree, 11% are not sure and 10% disagree.

Currently there are new modules for the controller, accompanied by a virtual “physics engine” in development. The idea about this physics engine is to simulate real world physics, like air pressure, light intensity, humidity among others. Furthermore it is intended to enable a math description of for instance temperature curves over time, so virtual sensors might be added to the VMCU platform.

REFERENCES

- [1] Federal Ministry of Education and Research, Germany (BMBF). ICT 2020, Research for Innovations. Bonn and Berlin, Germany, 2007.
- [2] Department for Employment and Learning, Research Study on High-Level Skill Needs in Northern Ireland ICT Sector, Oxford Economics, Lisburn, UK, 2009
- [3] S. Seiler, R. Sell, Comprehensive Blended Learning Concept for Teaching Micro Controller Technology, in eLearning Baltics 2011. Proceedings of the 4th International eLBA Science Conference, Rostock (Germany) May 26-27, 2011, ISBN 978-3-8396-0258-4, pp. 15-24.

- [4] R. Sell, S. Seiler, Integrated Concept for embedded system study, in MSM 2011. Proceedings of the 7th International Conference Mechatronic Systems and Materials (MSM 2011). July 7-9, 2011, Kaunas, Lithuania, ISSN 1822-8283
- [5] Sell, Raivo; Seiler, Sven (2012). Improvements of Multi-disciplinary Engineering Study by Exploiting Design-centric Approach, Supported by Remote and Virtual Labs. *International Journal of Engineering Education*, 28, 759 - 766.
- [6] Seiler, S.; Ptasiak, D.; Sell, R.: "Remote and virtual labs in a distance learning environment", p. 611-612, *Annals of DAAAM for 2011 & Proceedings of the 22th International DAAAM Symposium*, Editor B[ranko] Katalinic, Published by DAAAM International, Vienna, Austria 2011, ISBN: 978-3-901509-83-4, ISSN: 1726-9679
- [7] MoRobE and VAPVoS consortium, Virtual Micro Controller Webpage. Retrieved 2011-06-22, <http://vmcu.ihoch2.de/>.
- [8] M. Boelter, Virtual µCLab: Entwicklung einer Simulationssoftware für eine webbasierte Mikrocontroller Lernumgebung, Diploma Thesis, Bochum University of Applied Sciences, Bochum, Germany, 2010.
- [9] MoRobE Consortium (2011), Robotic HomeLab kit. Retrieved 2011-06-19, <http://home.roboticlab.eu/en/hardware/homelab>.
- [10] B. L. Titzer, Avroa: The AVR Simulation and Analysis Framework, Master's Thesis, University of California, Los Angeles, 2004
- [11] UCLA Compilers Group, Avroa - The AVR Simulation and Analysis Framework. Retrieved 2011-06-21, <http://compilers.cs.ucla.edu/avrora/index.html>.
- [12] Grube, P.P.; Boehringer, D.; Richter, T.; Spiecker, C.; Natho, N.; Maier, C.; Zutin, D.; "A metadata model for online laboratories," *Global Engineering Education Conference (EDUCON)*, 2011 IEEE , vol., no., pp.618-622, 4-6 April 2011 <http://dx.doi.org/10.1109/EDUCON.2011.5773202>
- [13] Thomas Richter, Yvonne Tetour, David Boehringer: Library of Labs. A European Project on the Dissemination of Remote Experiments and Virtual Laboratories. SEFI Annual Conference 2011, Lisbon, 28-30 September 2011.
- [14] MicaZ mode, Datasheet, Document Part Number: 6020-0060-04 Rev A, Crossbow Technology Inc., San Jose, California
- [15] Sell, R.; Seiler, S, Combined Robotic Platform for Research and Education. Proceedings of SIMPAR 2010 Workshops Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS. Darmstadt (Germany), 2010.
- [16] Sell, R; Otto, T. Remotely controlled multi robot environment. In: Proceedings of 19th EAEEIE Annual Conference: 19th EAEEIE Annual Conference, Tallinn, Estonia, June 29 - July 2, 2008. Tallinn., 2008.
- [17] S. Seiler, C. Köhn, R. Sell, T. Otto, INNOVATIVE STUDY KIT FOR ADVANCED MECHATRONIC EXPERIMENTS, 9th International Workshop on Research and Education in Mechatronics. September 18-19, 2008, Bergamo, Italy, ISBN: 88-88412-33-6
- [18] Müller, D.; Bruns, W.; Erbe, H.-H.; Robben, B. and Yoo1, Y.-H.: Mixed Reality Learning Spaces for Collaborative Experimentation: A Challenge for Engineering Education and Training, *International Journal of Online Engineering (iJOE)*.– Vol. 3, No. 4, 2007 , ISSN: 1861-2121
- [19] Müller, D.: Mixed Reality Systems, *International Journal of Online Engineering (iJOE)*.– Vol. 5, Special Issue 2 IRF'09, 2009 , ISSN: 1861-2121

AUTHORS

Sven Seiler, is with the Institute of Computer Science, Bochum University of Applied Sciences, Bochum, Germany (e-mail: sven.seiler@hs-bochum.de)

Raivo Sell, is with the Department of Mechatronics, Tallinn University of Technology, Estonia (e-mail: raivo.sell@ttu.ee).

Dennis Ptasiak, is with the Institute of Computer Science, Bochum University of Applied Sciences, Bochum, Germany (e-mail: dennis.ptasiak@hs-bochum.de)

Martin Boelter, is with the Institute of Computer Science, Bochum University of Applied Sciences, Bochum, Germany (e-mail: martin.boelter@hs-bochum.de)

This development was realized by funding of the European Commission Education and Training in the funding scheme "Leonardo da Vinci" by the projects "VAPVoS" and "NetLab" and the Estonian Science Foundation grant No. 8652. Manuscript received 28 June 2012. Published as resubmitted by the authors 23 October 2012.