Robotic Educational Tool to engage students on Engineering

Pedro Plaza Merino
HW R&D and WESTRACE SE
SIEMENS RAIL AUTOMATION, SAU
Tres Cantos, Spain
pedro.plaza@siemens.com

Elio Sancristobal Ruiz, German Carro Fernandez, Manuel Castro Gil
Electrical and Computer Engineering Department
Spanish University for Distance Education (UNED)
Madrid, Spain
elio@ieec.uned.es, germancf@ieee.org, mcastro@ieec.uned.es

***Abstract* — The aim of this paper is to summarize the Work in Progress related to the design of a Collaborative Robotic Educational Tool. This tool arises to improve STEM (Science, Technology, Engineering and Math) educational programs for school students. The design is intended to cover different specifications such as: scalability, modular capabilities, reconfiguration possibilities and compatibility with the aim of promoting the innovation and the motivation of the students during the learning process. Furthermore, the main objective of the mentioned platform is focused on getting a cost effective tool which can be included easily within educational institutions with a low budget restriction. This platform is intended for students which are able to use textual programming languages. Furthermore, the platform is expected to cost below 100 euros.**

Keywords — Arduino, Raspberry Pi, FPGA, Robotics, Education, STEM, IoT.

I. Introduction

There are a lot of needs to be improved in STEM (Science, Technology, Engineering and Math) education. The robotic represents a promising educational tool. Nowadays, it is easy to find a vast variety of robotic platforms which price is a great barrier for most of educational institutions when they want to include these robotic platforms as part as their educational programs. In the other hand, there are other cost-effective platforms such as remote laboratories [1] that eases the inclusion of robotic platforms for the development of educational activities where the proactive learning is empowered through experiments in the real world.

The Robotic Educational Tool which is being developed arises from an investigation carried out in the context of different development platforms. Additionally, the mentioned platforms are analyzed in order to know how developers are using the different platforms found with the aim of including these tools in educational applications. The results of this investigation can be found in [2].

The aim of this platform is to provide users with a learning experience that goes beyond the acquisition of knowledge or specific skills, since it offers real opportunities thanks to freedom of reconfiguration, flexibility, versatility and scalability. It also enables the development of active learning methodological practices mainly focused on the user and in enhancing relations group. Users can set it so that spaces own learning or group learning spaces are created. It also allows educators the possibility of producing creative, divergent and open activities.

This Work in Progress paper also describes the status of the Robotic Educational Tool development.

II. Work description

This paper covers the main characteristics of the development phase. The Robotic Educational Tool is based on FPGA (Field Programmable Gate Array) and Arduino [3]. The design is divided in three main phases: Hardware design, Firmware design and Software design.

The Hardware design covers the PCB (Printed Board Circuit) and connections side. Therefore, the PBA (Printed Board Assembly) is the platform which holds the Firmware and Software.

The Firmware design comprises the FPGA programming environment. Additionally, this stage includes interfaces and logic to accomplish functionalities specified for the platform.

And, finally, the Software design encompasses the Arduino code. During the development, the Arduino code is used for testing purposes. Once the development has been finished, all generated code will be provided as reference. The main objective is that students can reprogram the Arduinos in order to accomplish different educational sessions.

The specifications for Robotic Educational Tool are extracted from [2]. Table I summarizes the required functionalities and it also includes a brief description for all of them.

TABLE I. FUNTIONALITIES SUMMARY

Functionality	*Description*
Reconfiguration	Connections or configuration parameters can be modified before its use or while it is being used.
Scalability	The platform complexity can be modified according to the needs.
Compatibility	Connections with Arduino Shields and Smart Devices are possible and easy.

Functionality	Description
Concurrency	Processing and communications can be paralleled.
Protection	Integrity and minor protections are included. It is also allowed the inclusion of user protection.
Prototyping	User electronic prototypes can be included by teachers or students in order to conform the platform.
Flexibilty	The platform is susceptible of modification or adaptations.

Moreover, the platform is totally capable for IoT (Internet of Things) environments due to the possibility to be connected to Internet being easily integrated as a Remote Laboratory [1]. A clear example is the connection to Internet through a Raspberry Pi [4].

The proposal consists of a Main Module which holds the four sub-modules. In addition, Main Module is intended to communicate with the Secondary Modules. Furthermore, it has the ability to communicate via Bluetooth with Smart Devices physically separated from the platform. Moreover, Arduino Shields can be stacked. Arduino Shields are compatible with Arduino, whereby the latter can be easily assembled to the Secondary Modules too. External elements are boards that interfaces between the platform and different types of electronic circuits or robotic platforms so they can be integrated into the system. Finally, thanks to the Bluetooth communication, the platform can access or be accessed via the Internet, using smart devices for this purpose.

Due to the platform complexity, a working methodology is needed. Hence, the first step is to carry on an investigation about the methodologies which are being used these days.

A. Development methodology

The platform architecture is a part of three major areas: Hardware, Firmware and Software. Therefore, a sufficiently flexible and effective methodology must be defined.

[5] defines methodology as the science of the method and as a set of methods that are followed in scientific researches, or a defined doctrinal exposition. Therefore, a methodology is defined as that guide which refers to the path or set of rational procedures used to achieve the objective or the variety of objectives in order to manage a scientific research. It also means the approach which allows observing a problem from a complete, systematic and disciplined manner.

There are many methodologies for the analysis and design of systems, for this reason, has done research on the different existing methodologies and its advantages and disadvantages in order to choose the most appropriate methodology to be used in the presented Robotic Educational Tool development.

The analyzed methodologies are compiled in the following list:

- Rapid Application Development methodology: This Software development methodology is best known as Methodology RAD (Rapid Application Development), and was created by the computing guru James Martin in 1991. It is aimed at reducing drastically the time needed to design and implement information systems. The RAD has an intense participation, JAD sessions, prototyping, integrated CSE tools and code generators – see [6].

- Object Oriented methodology: according to [7], the object-oriented analysis and design is an approach to software engineering modeling a system as a group of objects that interact with each other. This approach represents a stranglehold in terms of composed verbs and nouns concepts, classified according to their functional dependence.

- Soft Systems methodology: This methodology was created by Peter and a group Checkland Systems Department of the British University of Lancaster. The methodology of SSM (Soft Systems Methodology), as defined in [8], soft systems has been developed from this continuous cycle of intervention in poor management structures of problems and learn from the results.

- V-Model methodology: the ISO / IEC 61508 functional safety for electrical, electronic and programmable electronic systems standard defined in [9] the V-Model as a variation of the waterfall model showing how the test activities are related to the analysis and design.

- Extreme Programming methodology: This is an agile methodology focused on strengthening interpersonal relationships as the key to success in software development, as defined in [10].

- SCRUM methodology: As defined in [11], SCRUM is an agile and flexible to manage software development methodology. The development is performed iteratively and incrementally. Each iteration, called Sprint, has a fixed duration of between 2 and 4 weeks, resulting in a new software version with performance ready for use.

Table II summarizes the methodologies analysis. It includes a brief description and a weight based on the ease of use of corresponding methodology for the development of mixed Hardware and Software systems. There are three weights, one for Hardware developments, other for Software developments, and the third one is an average of both. The weight is ranged from 0 to 5, where 0 means that a methodology is not intended to be used for with that kind of developments and a 5 means that a methodology improve the development when it is used for that purpose.

TABLE II. METHODOLOGIES ANALYSIS

Methodology	***Description***	***Ease of use for HW/SW systems***
Rapid Application Development	Software development methodology. Intended for design and implement information systems.	0 (HW) 4 (SW) 2 (SYS)
Object Oriented	Approach to software engineering modeling.	0 (HW) 4 (SW) 2 (SYS)
Soft Systems	Continuous cycle of intervention.	3 (HW) 4 (SW) 3.5 (SYS)
V-Model	Creates relationships between test activities and the analysis and design.	5 (HW) 5 (SW) 5 (SYS)
Extreme Programming	Agile methodology focused on strengthening interpersonal relationships for software development.	0 (HW) 5 (SW) 2.5 (SYS)
SCRUM	Agile and flexible to manage software development methodology	0 (HW) 5 (SW) 2.5 (SYS)

After the analysis shown above, V-Model methodology has been selected to be used as a basis because the V-Model methodology is intended for both, Hardware and Software development. Furthermore, the test-analysis-design cycle improve the development results. To do this the following steps are defined for the development of the whole platform architecture:

1. First stage: platform specifications.
2. Second stage: architecture design.
3. Third stage: components description.
4. Fourth stage: implementation.
5. Fifth stage: integration tests.
6. Sixth stage: system tests.
7. Seventh stage: testing with students.

Figure 1 depicts the different levels that compose the development methodology, the stages in which it has been divided, and the logical transitions that can occur in each of these stages.

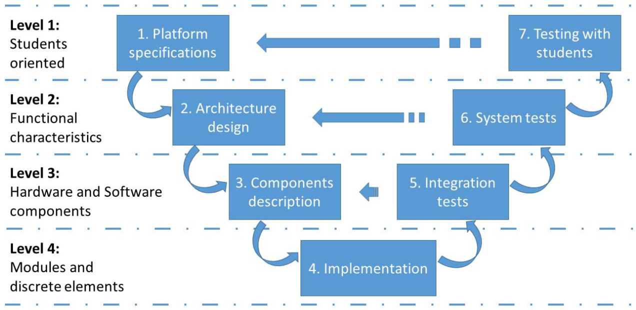

Fig 1. Architecture Methodology Diagram

Level 1 is oriented to students. It consists in specifications analysis and platform acceptance testing.

Level 2 is dedicated to the functional characteristics of the proposed system. It may be considered as a black box system, and characterize only those functions that are visible directly or indirectly by the end user.

Level 3 defines the hardware and software components of the final system. This level is called the system architecture.

Level 4 is the implementation phase, in which the unit elements or modules of the program are developed.

B. Hardware design

The Robotic Educational Tool Hardware design complexity degree is very high. Thus, a Hardware design methodology has to be elaborated specifically for the Hardware development. This methodology is based on the main V-Model methodology exposed above. It consists of several steps, steps followed for both the Main Module Hardware design and the Secondary Module Hardware design. First, the identified specifications that relate to the hardware design are common for both modules. Later the Hardware elements that will be part of the platform and that will be crucial throughout the design schematics are identified. Thirdly, the schematic elaboration is performed. After producing the schematic, physical design of the board is started, i.e., dimensions, connections and location of the different components. After completing the above steps, manufacturing files, GERBER files, are generated in order to be sent to the PCB manufacturer. Upon the reception of all PCB, the components are assembled, and finally verification tests are performed to verify that the hardware design meets the specifications.

When a design error or issue which does not fit with any specification is detected, an evaluation is performed in order to take the decision of continue with the following steps, or stop the process and make a redesign until a prototype which meets with the Hardware specification set is obtained.

The Main Module is based on FPGA. For the first prototype a basic system development based on Altera Cyclone II FPGA EP2C5T144 is used.

To be placed above the FPGA basic system, a PCB has been developed in order to connect the PBA with the

Secondary Modules, the Bluetooth communications module and Arduino Shields. In addition, the power supply will be distributed to secondary modules from the Main Module.

Secondary Modules are PCBs that connect the Main Module, the Arduino Nano, and Arduino Shield. The platform is composed of four sub-modules taking power from the main module. reset signals Main Module to the secondary modules are also passed to be able to reset the Arduino Nano from the FPGA.

Figure 2 shows a prototype setup of the Robotic Educational Tool. Figure 2 includes a Main Module and one Secondary Module. The Robotic Educational Tool can consist of up to four Secondary Modules. Each of these modules is formed by other components which are defined along this paper. Figure 2 illustrate the Main Module (green board) holding a Bluetooth communications module, an Arduino WiFi Shield and a Secondary Module (yellow board). Figure 2 also represents an Arduino Nano and an Arduino Bluetooth Shield, both connected directly to the Secondary Module.

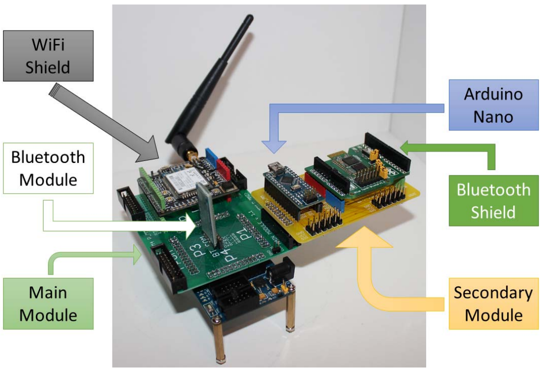

Fig 2. Robotic Educational Tool prototype example

III. Paper Contribution and Conclussions

The Work in Progress paper main contribution is to show the progresses related to the Robotic Educational Tool development status. The premises of this work are also shown in order to add the context of the investigation which was carried out. That results define clear specifications of a tool that meets the educational needs in the field of STEM (Science, Technology, Engineering and Mathematics) relying on robotics as a motivation enhancer for students in the classroom. The Robotic Educational Tool is intended to allow the use by teachers who can to include the proposed tool in its classrooms as a modular, reconfigurable, flexible, adaptable and cost effective Robotic Educational Tool.

The presented design is intended to produce a platform which can be used with students aged above 15 years. These students should be able to use textual programming languages.

Additionally, the platform is expected to be used in public and private schools Hence, the platform cost should be with an upper limit about 100 euros.

Furthermore, a methodology investigation results are included. This investigation was carried out in order to choose the better choice for the working methodology used for the Robotic Educational Tool development.

Finally, a description of the Hardware design phase work is described.

Acknowledgment

The authors acknowledge the support provided by the Engineering Industrial School of UNED, the Doctorate School of UNED, “Techno-Museum: Discovering the ICTs for Humanity” (IEEE Foundation Grant #2011-118LMF) and “Go-Lab: Global Online Science Labs for Inquiry Learning at School” (FP7-ICT-2011-8 – Project number 317601) [9].

Authors are especially grateful to the Electrical and Computer Engineering Department (DIEEC) of UNED for its support and advice in the preparation of this paper as well as for the support of his Special Project on Remote Laboratories equipment 2014-2015-2016 (2016-IEE13, 2016-IEE15 y 2016-IEE16), as well as to eMadrid excellence net, "Research and Development in educational technologies in the Community of Madrid" -S2013/ICE-2715.

References

[1] Carro Fernandez, G.; Carrasco Borrego, R.; Plaza Merino, P.; Canas Lopez, M.A.; Sancristobal Ruiz, E.; Castro Gil, M.; Mur Perez, F., "Mechatronics and robotics as motivational tools in remote laboratories," in Global Engineering Education Conference (EDUCON), 2015 IEEE , vol., no., pp.118-123, 18-20 March 2015

[2] P. P. Merino, E. S. Ruiz, G. C. Fernandez and M. C. Gil, "A Wireless robotic educational platform approach," 2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV), Madrid, Spain, 2016, pp. 145-152

[3] Arduino website. [Online]. Available on: http://www.arduino.cc, visited on 20 December 2015

[4] Raspberry Pi website. [Online]. Available on: https://www.raspberrypi.org, visited on 20 December 2015

[5] Methodology definition. [Online]. Available o: http://dle.rae.es/?id=P7eTCPD, visited on 9 April 2016

[6] E.J Davidson, Joint application design (JAD) in practice, Journal of Systems and Software, Volume 45, Issue 3, 15 March 1999, Pages 215-223, ISSN 0164-1212 J, Martin, “Rapid Application Development,” Macmillan, New York, 1991

[7] G. Booch, R. Maksimchuk, M. Engle, B. Young, J. Conallen, K., Object-Oriented Analysis and Design with Applications, 3rd Edition. Addison-Wesley Professional, 2007

[8] P.B. Checkland, J. Poulter, Learning for Action: A short definitive account of Soft Systems Methodology and its use for Practitioners, teachers and Students. Wiley, 2006. ISBN 0-470-02554-9

[9] Functional safety of electrical/electronic/programmable electronic safety-related systems: Part 3 Software requirements, IEC Standard ISO/IEC 61508, 2010

[10] Beck, K., Extreme Programming Explained: Embrace Change, Addison Wesley, 2000, ISBN: 0201616416

[11] Linda Rising and Norman S. Janoff, “The Scrum Software Development Process for Small Teams,” Focus process diversity, IEEE SOFTWARE July / August 2000

[12] Go-Lab project website. [Online]. Available on: http://www.go-lab-project.eu/, visited on 21 February 2016