The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
21-23 September, 2017, Bucharest, Romania

A Versatile Robotic Platform for Educational Interaction

M. Naya[1], G. Varela[2], L. Llamas[2], M. Bautista[1], J. C. Becerra[3], F. Bellas[1], A. Prieto[1], A. Deibe[1], R. J. Duro[1]
[1]Integrated Group for Engineering Research, Universidade da Coruña, Spain
{francisco.bellas, martin.naya, abprieto, moises.bautista, adeibe, richard}@udc.es
[2]Mytech, A Coruña, Spain
gervasio.varela@mytechia.com, luis.llamas@mythechia.com
[3]MINT, A Coruña, Spain
juancarlos@mintforpeople.com

***Abstract* - This paper describes a new educational robot called ROBOBO which was developed in order to provide an engaging and natural interactive experience with children. The objective is to motivate children to explore and cooperate with the robots while they learn concepts and abilities related to STEM education (Science, Technology, Engineering and Mathematics). The robot and the educational approach has been tested in different school environments corresponding to different educational levels with very successful results.**

Keywords—educational robotics; STEM education; interactive robotics

I. INTRODUCTION

STEM (Science, Technology, Engineering, Mathematics) [1-3] or STEAM (Science, Technology, Engineering, Arts, Mathematics) [4] education is becoming a very important approach to increase the proficiency of students in these technological fields and, at the same time, promote future careers in these areas. As indicated in [5], STEM is critical to many education reforms that are being undertaken in many countries. Especially in those that have fallen behind in these areas.

In the last few years robotics has slowly become one of the pillars of STEM education. The objective is to make the learning experience of the students more appealing by providing mobile platforms with capabilities that permit engaging the students' attention for long periods of time while they try out and experiment things. Through an adequate curriculum, based on enticing projects to be carried out with robotic platforms, the students are able to learn by playing, designing and constructing.

This concept has been successfully tested in the last few years in different school environments and with children at different levels [6]. However, the real application of robots to education have mostly been related with the processes of constructing and/or programming robots that, in general, were very simple and endowed with few sensors and actuators. Examples of these types of robots are presented in Fig. 1, and a review of their capabilities in table I.

While these platforms are quite adequate to teach students the basics of robot construction or, even programming, they lack a rich enough set of sensors and interactive capabilities to really become the desired learning or teaching assistant contemplated in STEM education. Most of the platforms only present rudimentary sensing abilities, such as infrared or ultrasonic collision detectors or maybe light sensors, and their only motor capabilities are turning two wheels, which basically allows them to move around the environment. This really limits what a student can do with them, and, especially their interaction capabilities. They have no way of expressing emotions or detecting the children or their expressions. It is becoming quite clear that to have a complete learning experience, teaching has to be considered as a social experience and, consequently, the more social the robots' capabilities the better [6].

This is where the robot we propose here, ROBOBO, shines. ROBOBO is based on the combination of a

Figure 1. Popular current educational robots. From top left: Dash & Dot, Lego EV3, Thymio II, Mbot Ranger.

smartphone as the brain of a robot with a robotic base that provides the basic sensing and motion capabilities, both those associated with moving around and those associated with expressiveness (shaking its head, for instance). This presents many advantages. On the one hand, the robot can make use of all of the state of the art sensors found in current and future smartphones, such as high definition cameras, accelerometers, inclinometers, microphones, touch sensors on the screen, etc. On the other, it can benefit from the extensive communication capabilities of a smartphone, including WIFI, Bluetooth, NFC, 3G, 4G, etc… It will also benefit from the extensive software available for smartphones as well as from the high computational capabilities these systems currently provide. Thus making it possible to use speech processing, face detection, or visual object identification capabilities within the activities or the interaction resources the robot provides. Pooling these computational capabilities with the aforementioned communication possibilities, allows the students to combine robot actions with information from the web (for instance, meteorological information, or traffic information) or to interact with other users by directly linking the robot through twitter, skype or any other social media site. This opens up a world of possibilities for collaborations in class projects among schools using ROBOBO, even when the schools are located in different countries (excellent chance to use a foreign language to collaborate). Finally, as smartphones are constantly being updated the robot will remain current with respect to sensing, computational or communication capabilities by just updating the smartphone, which, in general is becoming a small cost.

Table I: Popular educational robots

	LEGO EV3	**DASH & DOT**	**MBOT RANGER**	**THYMIO II**
CPU	Arm926ej-S Core@300 Mhz	Arm Cortex-M0	Arduino Mega 2560	PIC24 32 MHz
Distance sensor	Sonar / IR	IR	Sonar	IR
Sound	Speaker	Speaker /Mic	Buzzer /Mic	Speaker /Mic
Camera	No	No	No	No
Speech recognit.	No	No	No	No
Comms	Bluetooth /USB	Bluetooth	Bluetooth /USB	WIFI /USB

II. ROBOBO

ROBOBO [7] is a robotic platform that was designed as a reliable, high quality smartphone based educational tool for children that are starting in the robotics field and for the introduction of STEM type educational approaches (Science, Technology, Engineering and Mathematics) [1-3]. Notwithstanding this initial objective, due to its multipurpose design, it is also suitable for both makers and robotic enthusiasts. It is made up of a mobile sensing base platform (the ROB) that supports a smartphone (the OBO), which acts as its high level brain providing advanced sensing, such as vision, sound, GPS, actuation, such as generation of images on the screen or production of sound, and communication, Wifi, Bluetooth, 4G, capabilities.

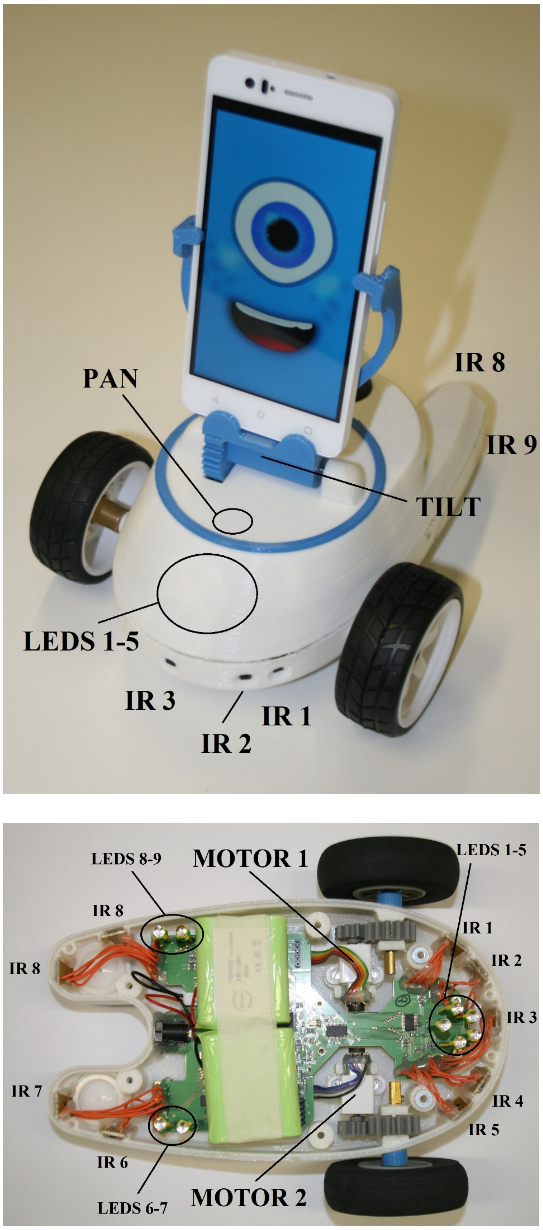

Figure 2. Robobo and its components

From a design point of view, the ROBOBO base platform, called ROB, was created seeking simplicity and naturalness with the aim of producing a system that once it is integrated with the Smartphone, the OBO, it is perceived as a unit, not as two different and separate parts. In the following sections we will provide a description of the main characteristics of ROBOBO in terms of its mechanical, electronic and control components as well as of the software structure that allows for the integration of advanced and intelligent sensing and control strategies with the aim of making the system as user friendly as possible while allowing it to tap the huge amount of resources that are available to designers and educators on the web. In fact, the platform is intended for use in the "adopt a robot" initiative within the DREAM H2020 European project in order to test the adaptation of new cognitive approaches within environments populated by children with no previous robotic knowledge.

III. ROBOBO HARDWARE DESIGN

The mechanical design of the ROB is divided into two parts (Fig. 2, top): the main platform and the pan-tilt unit that supports the smartphone. The basic components of the ROBOBO are in the main structure (Fig. 2, bottom). It contains the circuit boards, infrared sensors, batteries, docks for add-ons and the pan-tilt unit. It is also the support for all the mechanical components that allow the robot to move and interact with the world. Thus, the platform is driven by two wheels which are powered by two CC motors through gears with a pinion-crow ratio that allows the ROBOBO to achieve a velocity of 1m/s with 40Ncm of torque. This is quite enough for snappy performance and climbing small ramps. The two wheels can move backwards or forward with independent speeds, allowing the robot to turn. In addition to the wheels, the robot is supported by two caster balls located in the back of the robot on the two ends of the "U" to give stability to the robot. They allow an omnidirectional motion of the ROB.

The pan-tilt unit is one of the most important parts of the robot and a key characteristic in its Human Robot Interaction (HRI) capabilities [8-9]. It is made up of two geared motors that allow the pan and tilt motion of the OBO. This lets the mobile phone perform actions such as nodding or shaking its "head", which when combined with images displayed on the smartphone screen, the sounds it produces, the displacement of the main platform and the different colored LEDs, endows the robot with the capacity of expressing and transmitting emotions, feelings, etc. in a very natural and understandable manner [10], thus providing a huge amount of interaction possibilities. The objective is to allow the robot to actively interact with children, influencing their mood, predisposition to learn, or even making them more sociable [11]. Additionally, and in terms of vision through the smartphone camera, this pant-tilt support allows for almost omnidirectional semi-spherical vision all around the robot, providing a lot of information about the environment in which the robot operates.

IV. ROBOBO CONTROL STRUCTURE

The control structure of the ROBOBO is summarized in Fig. 3 where the main components of the hardware architecture can be observed:

This architecture is divided into two levels: the low level control structure located in the ROB platform and the mission control or cognitive high level structure which is located in the smartphone (in the OBO).

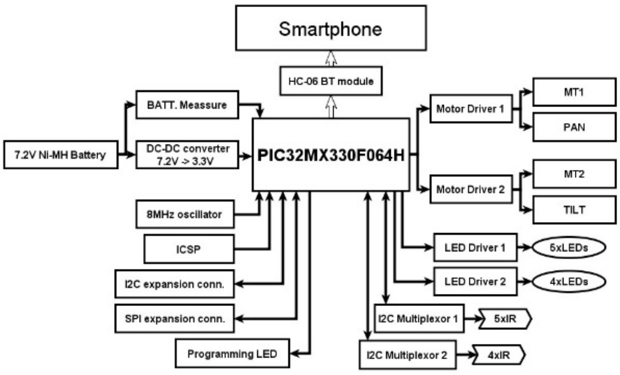

Figure 3. Diagram of the Robobo hardware architecture

The low level control mechanism is structured around a PIC32 microcontroller for real time operation and responsiveness. It is in charge of controlling the ROB's sensors, actuators and routine programs as well as the communications with the OBO. The system is powered by a 7.2V Ni-MH power supply. It provides the operating voltage for the motors and a lower voltage for the microcontroller and the remaining components of the circuit. This battery endows the system with an autonomy of 3-4h of normal operation.

In addition to the previously mentioned 4 motors (two for the wheels and two for the pan-tilt unit), there are 9 RGB LEDs, controlled by a LED driver with a resolution of 12bits per LED, for a total of 212 = 4096 different colors on each RGB LED. These LEDs can be used to provide any type of information to the user. For instance, by default they are associated to each one of the infrared sensors, allowing the user to assert what the sensor is sensing in a very simple and intuitive manner.

These infrared sensors (Vishay VCNL4040) are in charge of obstacle detection and fall avoidance as well as ambient light sensing. Regarding obstacle detection: 7 of the total of 9 infrared sensors are placed al around the ROB to sense the distance to the different objects and thus be able to carry out collision avoidance tasks when required. These sensors, unlike many other infrared sensors, are not binary sensors (detection-not detection) they can provide a measure of the distance to the obstacle, that is, they provide obstacle range detection up to 10-15 cms, depending on the color of the obstacle.

In the case of fall avoidance: 4 sensors are devoted to this task. Two frontal sensors are dedicated exclusively to the detection of "holes" (in the sense of lack of floor). These sensors are tilted towards the ground at a 70° angle to optimize hole detection and its range so that the robot has time to stop before falling. The other two sensors are placed on the back of the robot. These two sensors share the fall avoidance functionality with the obstacle detection functionality.

Finally, although the VCNL4040 sensors are configured by default to work as infrared sensors for obstacle detection and fall avoidance, there exists the possibility of configuring any of them as ambient light sensors if the user so desires. In this configuration, the sensor returns the light intensity value.

Apart for controlling low level sensing and actuation, the electronics in the ROB platform are also in charge of the communications to the OBO part of the system. This is achieved by means of a Bluetooth module together with a series of firmware programs whose function is to abstract the low level control into an API that the OBO can use. That is, the software developed on the smartphone just sees this abstraction of the ROBs functions. The choice of a wireless communications protocol over a wired one, such as USB, had to do with increasing the versatility and reducing the complexity of the platform due to the diversity of connectors and connector positions in different smartphones. Thus, the ROB platform can be used with any smartphone just as long as it has Bluetooth connectivity.

On the OBO side of ROBOBO, we have a smartphone that is in charge of high level control. It is the real brain of the Robot providing lots of very interesting sensing and interaction capabilities. In fact, we could consider the ROB part as the reflex system of the agent and the OBO as the thinking part. The key is the appropriate integration and coordination of these two parts in order to produce an agent that can interact with humans in a very natural manner. One great advantage of this approach is that the system can be upgraded by just introducing a more advanced smartphone.

V. ROBOBO SOFTWARE ARCHITECTURE

Current smartphones provide very good computing characteristics: fast and efficient processors, large amounts of memory, huge storage capabilities, and extensive communications abilities. Consequently, they can easily be used for introducing advanced and intelligent sensing, actuation, interaction and control strategies. Deep Neural Networks can be run on them for voice or image processing, complex cognitive architectures can be implemented. They can even run educational systems for programming, such as the well-known Scratch [13] or Blockley [14] block languages when used as teaching aids.

To benefit from all of these capabilities, an API and a software structure was developed. The main requirement of the Robobo software architecture was to support the programming of the robot using very different programming paradigms. The goal is to be able to use the robot for teaching STEM and robotics at very different levels of education, from primary school to universities, through secondary school or even use it for research purposes. This has led to a completely modular software architecture as shown in Fig. 4.

The architecture revolves around the concept of Robobo module and a very lightweight library, called the Robobo Framework. The Framework provides the mechanisms to manage and run those modules on an Android smartphone. Two different types of modules can be loaded on top of the library:

Functionality modules (the orange ones in Fig. 4), implemented in Java using the Android standard API, provide different functionalities to the robot, like speech recognition, face detection, environment sensing or physical movement. Furthermore, these modules complement the native API of the robot, which can be directly used for programming the robot using Java for Android, and create Robobo Apps that can be run in any Android smartphone.

A series of proxy modules (ros-proxy and remote-proxy modules in Fig. 4), also implemented in Java, provide particular interfaces to other programming environments, like ROS [12], Scratch [13] or the native Robobo Education IDE using Blockly [14]. These interfaces provide a translation between an external API or protocol, and the native Robobo API in Java.

A set of default modules is provided with the ROBOBO, but users can customize the modules, or even implement new modules to support new robot functionalities or new programming paradigms through proxy modules.

As indicated before, there exists a module for connecting the Robobo framework and its modules to the Robobo platform (ROB). The rob-interface module, shown in pink in Fig. 4, implements the Bluetooth communications protocol of the ROB and provides a control API for other modules to use (pink in the figure). This is an essential module to control the physical part of the robot. However, for homogeneity, it is implemented and loaded just as any other module of the framework.

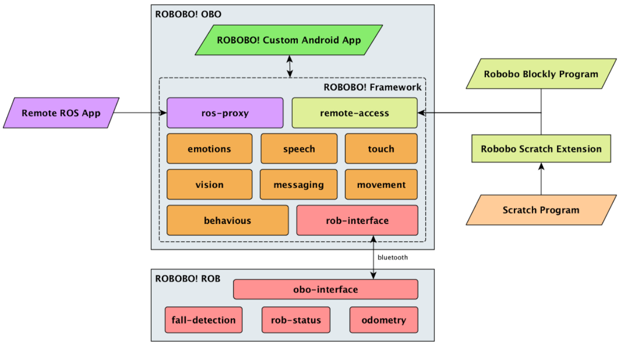

Figure 4. Block diagram showing the Robobo software architecture and the different programming paradigms supported by it.

VI. ROBOBO IN SCHOOL

As indicated above, ROBOBO was designed for its use at different educational levels, from primary school to Universities. To be able to fulfill these requirements, different implementations of its modular structure are necessary. More importantly, different educational programs and proposals are required.

In the case of primary education, ROBOBO has been mostly used as a tool for teachers to introduce concepts and knowledge to children in a fun and engaging way. To this end, functional modules related to color detection, or face detection are used within games to introduce children to concepts such as left or right, speeds, angles, music notes, etc… The usual procedure is to generate some type of game whereby the child interacts with the robot in order to become familiar with the corresponding concept.

As an example, we can consider a musical "Simon says" type of game. Two elements are required in this case, a ROBOBO and a piano (Fig 5 displays a configuration for this case). This game is usually played in low noise environment with the ROBOBO and the piano close to each other. Specifically, the game involves the following steps:

1. The game is started by clapping. This "wakes up" ROBOBO, which says "Ready".
2. A second clap corresponds to the beginning of the interaction. Here, the robot tries to challenge the user by saying "imitate the notes I produce if you can".
3. Then, the robot generates a single musical note in simple cases or a sequence of note when we want to make the game more complicated.
4. At this point, the facial expression presented on the ROBOBO screen is set to an attentive expression and ROBOBO starts recording the musical notes the user plays using the piano.
5. Once the user has played a note or sequence of notes, ROBOBO compares them to the ones it played.
   5.1. If they are the same, the robot expression is modified to a happy face and at the same time, the pan tilt unit moves pan-tilt unit moves forwards and backwards (a nodding gesture) while it produces a congratulating sound.
   5.2. If they are different, the expression is changed to a sad one, the pan-tilt unit moves from side to side (negating with the head) and a failure sound is emitted.
6. The program returns to step 3

This game is very adequate for primary school children, the robot is just a resource in their environment that captures their attention and makes learning notes fun. In this case the robot is a tool that engages children through interaction and expressiveness.

However, for older children (end of primary school or beginning of high school) this example can be made a bit more complex and used in order to introduce computer programming concepts and abilities. To do this, we must endow ROBOBO with appropriate programming tools. As indicated before, ROBOBO can be directly programmed in Java, but this is usually way out of the league of students that are just starting with programming and who need to learn the basic concepts of dividing a problem into steps, using recurrences, establishing decision points, etc… In other words, we do not want the students to get lost in the syntax of the programming language, but concentrate on the basic programming structures.

Figure 5. The Robobo and the piano

To this end, there are several educational programming tools that are very adequate, such as ScratchX or Blockley [13-14], which have been implemented as modules of ROBOBO. Thus, we can now ask the students to program the robot to play the game using, for instance, ScratchX.

Consequently, to complete the lesson, the students must apply the following ROBOBO specific ScratchX blocks: clap-Detector, speechProducer, noteProducer, noteDetector, soundProducer, emotion-Producer and motorCommand, as well as other ScratchX native blocks to create the algorithm. The configuration of the blocks to achieve a fluid and reliable interactive game is part of the students' work because, as previously highlighted, the goal is to solve problems that can be used in the real world.

As it can be observed, this lesson does not require any movement of the robotic platform, so one could think that a robot would not be necessary. However, the approach underpinning The Robobo Project is that human-robot interaction is more than just movement, and developing this type of static interaction is also very important. Additionally, the pan-tilt unit must be moved in this lesson. Therefore, from a didactical viewpoint, the students need to use ROBOBO motor commands.

Anyway, to address a case where robot motion is necessary, we will present another example where ROBOBO can be used to start students in learning the abilities that are necessary for them to be able to start programming as well as introducing some intuitions on mathematical problems. In this example the students must interact with the robot in order to make it follow and then draw different types of triangles on the floor.

The students will have to make use of very basic ROBOBO ScratchX commands such as move_forward and turn a given angle. The idea here is that by doing this, the students will be exposed to the programming facts that a process can be divided into steps, that each step must be expressed in a precise manner and that when one links the appropriate number of correct steps in the right order a

Figure 6. Top: Teacher presenting a class on angles and triangles. Middle: Students learning to control ROBOBO. Bottom: Result in the case of a particular triangle drawn with ROBOBO.

whole is achieved. Simultaneously, the students will be grappling with the concepts of line lengths (as a function of speed and time) and turning angles, which are basic mathematical and physics related concepts. By testing different combinations, they will obtain different triangles (once they are able to obtain triangles). In this case, and to make the exercise more fun and engaging the teacher had introduced funny sounds and different face expressions for the robot when it was turning or going at different speeds.

Fig. 6 displays the result obtained by one group of students in a real class. What is relevant here, is that the students had a great time and, more importantly, they actually assimilated the concepts involved in a very intuitive manner through exploration and interaction.

VII. CONCLUSIONS

Robobo is a compact robotic platform that has been designed as the ideal complement for STEM or STEAM ducation. It leverages the power of current smartphone technologies in terms of computing capabilities and advanced sensing with the mobility of a dedicated platform, the ROB. This platform has been endowed with expressiveness in addition to its displacement capabilities. This expressiveness is achieved by the capability of moving the head (the smartphone) sideways as well as forward and back, thus allowing for nodding and negating and by a set of led lights all around it that can be used to represent states or emotions. These expressive qualities together with those that can be achieved using the smartphone through images on its screen and sounds provide for a very interactive system that gets children involved with it.

The system has been tested on several school classes and the results are very promising. We are now extending the tests to other age groups and different types of schools, including schools for handicapped children.

ACKNOWLEDGMENT

This work has been partially funded by the EU's H2020 research and innovation programme under grant agreement No 640891 (DREAM project) and by the Xunta de Galicia and redTEIC network (ED341D R2016/012).

REFERENCES

[1] Alimisis, D., Moro, M. (2016). Special issue on educational robotics. Robotics and Autonomous Systems, Vol 77, pp 74-75
[2] Barker, B. S., Nugent, G., Grandgenett, N., & Adamchuk, V. I. (2012). Robots in K-12 Education: A New Technology for Learning, IGI Global, pp 1-402
[3] Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. Computers & Education, 58(3), 978-988.
[4] Kim, Y., and Namje P.. (2012). The effect of STEAM education on elementary school student's creativity improvement. Computer Applications for Security, Control and System Engineering. Springer Berlin Heidelberg, 115-121.
[5] Thomasian, J.. Building a Science, Technology, Engineering and Math Education Agenda. National Governors Association Center for Best Practices, Dec. 2011.
[6] Mubin, O., Stevens, C. J., Shahid, S., Al Mahmud, A., & Dong, J. J.. (2013). A review of the applicability of robots in education. Journal of Technology in Education and Learning, 1, 209-0015.
[7] The Robobo Project web page: http://theroboboproject.com
[8] M. A. Goodrich and A. C. Schultz, (2007) Human-robot interaction: a survey. Foundations and trends in human-computer interaction, vol. 1, no. 3, pp. 203–275,
[9] Sheridan, T. B. (2016). Human–robot interaction status and challenges. Human Factors: The Journal of the Human Factors and Ergonomics Society, 58(4), 525-532.
[10] Belpaeme, T., Baxter, P. E., Read, R., Wood, R., Cuayáhuitl, H., Kiefer, B., & Looije, R. (2012). Multimodal child-robot interaction: Building social bonds. Journal of Human-Robot Int., 1(2), 33-53.
[11] Kandlhofer, M., Steinbauer, G. (2016) Evaluating the impact of educational robotics on pupils' technical-and social-skills and science related attitudes. Robotics and Autonomous Systems. Vol 75, Part B, pp 679-685
[12] "Robot operating system (ROS)," http://www.ros.org.
[13] ScratchX web page: http://scratchx.org
[14] Blocky web page: https://developers.google.com/blockly