Studies in Computational Intelligence 982

Monica Malvezzi
Dimitris Alimisis
Michele Moro *Editors*

Education in & with Robotics to Foster 21st-Century Skills

Proceedings of EDUROBOTICS 2020

Springer



Mendieta, One Robot Per School: Multi-user Robot for Technology Education

Gonzalo Zabala[1](✉), Ricardo Morán[1,2], and Matías Teragni[1]

[1] Universidad Abierta Interamericana, Centro de Altos Estudios en Tecnología Informática, CABA, Buenos Aires, Argentina
{gonzalo.zabala,ricardo.moran,matias.teragni}@uai.edu.ar
[2] Comisión de Investigaciones Científicas, La Plata, Buenos Aires, Argentina

Abstract. In recent years, the use of physical computing devices in Argentinian schools has been growing steadily. The insertion of robotics in the classroom allows addressing a set of technology-related subjects, such as mechanics, electronics, and programming. It is no coincidence that the Argentine educational system has incorporated these subjects into its curricular design, as well as the introduction of new technical schools aimed at programming and robotics. However, the cost of the equipment, the difficulty of maintenance, and the technical knowledge required to be able to properly use this resource in an educational context are limiting factors that hinder the potential to reach all schools. For these reasons, a project has been designed to overcome these difficulties: a low-cost autonomous robot controlled by an Arduino Nano, with a web server built into an Orange Pi computer, which allows programming by multiple users at the same time, queuing the execution orders in such a way that with a single device an entire class can work seamlessly. This allows the introduction of robotics in a school with a cost of less than USD 180. Additionally, all the development of both hardware and software is open source, offering the community a framework for building other robots with the same architecture.

Keywords: Technology education · Robotics · Web server · Arduino · Orange Pi

1 Introduction

Since 2010, the introduction of robots in Argentine classrooms as a didactic resource has encouraged an increasing interest in incorporating the discipline at the curricular level along with programming (Ripani 2017). We can mention, among others, the project "Todos a la robótica" in the province of San Luis (Munizaga 2013), which from 2011 to 2014 incorporated material, training and a set of activities in all its primary schools. Another project worthy of mention is "Aprender conectados" (primarily called Escuelas del Futuro 2017) that reached about 3,000 schools throughout the country, both at primary and secondary level, including mechanics, electronics and robotics materials, as well as training and activities for use in the classroom (Ministerio de Educación – Presidencia de la Nación 2017). In the Province of Buenos Aires, we find the Plan Provincial de Robótica

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021
M. Malvezzi et al. (Eds.): EDUROBOTICS 2021, SCI 982, pp. 52–63, 2021.
https://doi.org/10.1007/978-3-030-77022-8_5

Educativa (2018) that also reaches primary and secondary schools with different types of kits (Banchoff Tzancoff et al. 2019). Finally, in 2018 the Ministry of National Education defined for the first time learning objectives for compulsory education in programming and robotics, to start the path of their formal inclusion in the teaching and learning proposals in the schools of Argentina (Ripani 2017).

Even though there is a growing interest in the discipline, the sustainability of the activity over time is difficult to achieve. Physical materials suffer natural wear and tear in their use (which does not happen, for example, with computers or software), in addition to having an accelerated level of obsolescence. On the other hand, public policies related to the insertion of technology in schools are modified in each administration whenever there is a change of party in government. These experiences lead us to conclude that simply purchasing kits and training teachers is not enough: long-term support is necessary in order to ensure financial resources to periodically renew the material and activities.

In this paper we present "Mendieta, one robot per school". The objective of this project is to facilitate the insertion of robotics in secondary schools and its sustainability over time.

Given the aforementioned problems, this project should meet the following requirements:

- The development of the robot must be very low cost and highly reusable and upgradeable.
- The construction of the robot must be simple enough that can be implemented directly at the school.
- The hardware and software used must be fully open[1] to build a community that would be able to maintain and create new models.
- Building a community of teachers and educators is essential to provide activities that other teachers can use and extend.

Regarding low-cost solutions for educational robotics, various proposals can be found in the literature (Aroca et al. 2012; Darrah et al. 2018; Junior et al. 2013; Lopes Filho et al. 2011; López-Rodríguez and Cuesta 2016; Rubenstein et al. 2012; Saleiro et al. 2013). However, all these proposals imply the need for a robot per group of students and, in general, the usage of external devices such as smartphones. Additionally, the hardware they propose is too complex to be built by teachers without sufficient knowledge of electronics. To further reduce the cost, and meet the aforementioned objectives, Mendieta has been designed as a robot with a web server hosted on it, using a software system that allows for concurrent access to the programming tools using any modern web browser. In this way, with a single robot (or with a small number of them) the entire class can work simultaneously.

For the robot electronics, we have selected low-cost, standardized, easy to assemble, and easily available components. Regarding the robot structure, models are provided in STL format that can be printed on any of the cheapest 3D printers. Both electronics and 3D design have broad Creative Commons licenses.

[1] All the source code and design files will be published on https://github.com/MendietaProject.

In addition to the robotics kit, a website will be developed where teachers can consult activities classified by level of difficulty and associated subject. They will also be able to propose their own activities from a template form, which will then be moderated by our group before its final publication.

In the following sections we will describe the technical challenges and design decisions behind Mendieta. In the first section we will make an analysis of various educational robotics hardware architectures, ending with a description of the one chosen for Mendieta. And in the second part we will focus on the software architecture, both on the robot's firmware and on the programming platform mounted on the web server.

2 Hardware Architecture

In order to reduce the cost of the materials there are several things to consider. On the one hand, robotics equipment is not in permanent use in classrooms. Therefore, the robotics kit must be designed in a way that allows it to be shared between classes. This leaves out kits that involve construction processes that exceed the duration of a single lesson, otherwise the equipment can only be used by one class instead of being shared with the whole school.

At the same time, the kit must have the versatility to provide the students challenges of diverse levels of complexity. To that end, several low-cost robotic kits have already been proposed that meet these requirements. One of them is Cellbots (Aroca et al. 2012), which uses a smartphone as its processing unit. It proposes the use of the microphone input to control the motors and to read external sensors, as well as the use of other phone sensors to enrich the information about the robot surroundings. Another similar proposal is the Andruino - A1 (López-Rodríguez and Cuesta 2016), which includes an Arduino Uno as the device responsible for connecting sensors to an Android phone. In both cases, the model is still one robot per team, although the form of communication proposed in this last project would allow a similar use to the one offered by Mendieta. However, both projects assume the students would use their own smartphones, so they do not include it when estimating the final cost for each kit. And forcing students to use their own smartphones can be problematic, given the well-known risk of accidents involving mobile robots.

Other options such as those found in (Junior et al. 2013; Lopes Filho et al. 2011; Rubenstein et al. 2012) are harder to build by inexperienced users, and maintain the model of one robot per group of students.

For these reasons, we understand that the least possible amount of materials in a course is that of a single robot. With this objective, the proposed robot not only has the responsibility of coordinating its sensors and actuators based on the programs developed by the students, but it must also be able to receive, interpret and coordinate the simultaneous work of said students. To allow concurrent access to write, compile and execute programs, students use their own computers or cell phones to access a web page hosted on the robot, acting as local server, and through which they can program, debug, request the execution, and monitor the device status.

Since the established requirements exceed the limited capabilities of the most common microcontrollers used in educational robotics, there are two alternatives: either use

specialized and expensive hardware, or complement the microcontroller with a microprocessor that has the necessary computing power and capabilities to function as a web server**. It is important to note that when we refer to both microcontrollers and microprocessors, we refer to integrated boards that complement the corresponding chips with the electronics and firmware necessary for an application without the need for complex technical knowledge**.

Popular and relatively low-cost microprocessors that have access to sensors and actuators already exist, but most do not include an analog-digital converter out of the box. Therefore, if you want to read analog sensors, such as light, humidity, distance, etc., you must add additional hardware and lose a significant amount of the ports that are available for sensors. Furthermore, this solution involves more complex electronic development.

Based on the above, the proposed architecture has two processing units. On the one hand, a microprocessor responsible for hosting the web server that lets teachers to monitor and configure the robot, as well as allowing students to program, debug and execute code. And on the other hand, a microcontroller in charge of executing the students' program, accessing the connected actuators and sensors.

There are multiple hardware options that can meet the established requirements, but given the nature of this project, which seeks to be open-software and open-hardware, the ease of access to the components required for assembly is an important factor to consider. The use of standard components in the educational and maker community worldwide was prioritized. Therefore, an Arduino was chosen as the microcontroller, being an open platform, low cost and widely accepted by the community, and an Orange Pi as the microprocessor, which has wide availability and documentation.

In particular, regarding the microcontroller, the Arduino Nano (Arduino Nano | Arduino Official Store, s/f) was chosen because it is one of the versions of the board most used in Argentina today, and because it can be replaced by other versions with a minimal work if necessary. It is less expensive and consumes less power than Arduino Uno, offering similar features, which are sufficient for this project.

Regarding the microprocessor, the Orange Pi Zero (orange pi zero - Orangepi, s/f) was selected mainly due to its low power consumption. Despite being less popular than Raspberry Pi, it has more processing power for a similar cost, allowing for better performance. However, a Raspberry Pi Zero W could be used instead, albeit at a lower performance.

So far, the minimum base of the hardware architecture has been defined. The objective is that different robot proposals could be developed based on this framework. Next, we will describe a first prototype which we will use to make the first experiences in the classroom.

Regarding the sensors, with just two types of sensors we can cover multiple different use cases: the ultrasonic sensor HC-SR04 (Freaks 2016) and the TCRT5000 module (Semiconductors 2009). Both sensors are low cost and have a strong market presence, making them an ideal choice for a robotics kit. We included one HC-SR04 to allow measuring distances with a range of 2 to 400 cm and an accuracy in the order of 3 mm. And we added two TCRT5000 modules, which are reflective optical sensors widely used

for line and obstacle detection. The TCRT5000 modules have another advantage in that they already include the electronics required to connect with the Arduino.

Regarding the movement, two direct current motors with a built-in reduction box were selected (Motor Yellow 3-12VDC 2 Flats Shaft, s/f). These are also popular in the maker community. In order to synchronize the speed of each motor, an optical encoder is used. We decided to use an opto-switch fc-03 (MOCH22A Interrupter Datasheet pdf - OPTO Interrupter. Equivalent, Catalog, s/f) based on an LM393 comparator. The motors are controlled by the module based on the L298N driver (Handson Technology, s/f).

For power supply we decided to use two rechargeable lithium ion 14500 batteries of 3.7 V each, which have the same size as traditional AA batteries, thus being able to use a conventional battery holder. We also incorporated the TP4056 module (NanJing Top Power ASIC Corp., s/f) that allows to recharge them with an input voltage of 4.5 to 5.V and a miniUSB plug like that of traditional cell phone chargers.

Regarding the visualization of the robot data, in addition to the software tools that will be shown later, we added a 16 × 2 liquid crystal display compatible with the Hitachi HD44780 driver (Hitachi, s/f).

Finally, the mechanical structure of this first prototype is still in a design stage. The objective is to make a minimum amount of parts that can be obtained from the cheapest 3D printers on the market. It is for this reason that none of them will exceed 14 cm in any of its dimensions, since the printing beds have a limit of 15 cm in the economic models. To simplify the printing process and make the components easy to locate, the design was done in two layers. In the image below, the items in gray will be printed while the other components are placed to show the final structure of the robot (Fig. 1).

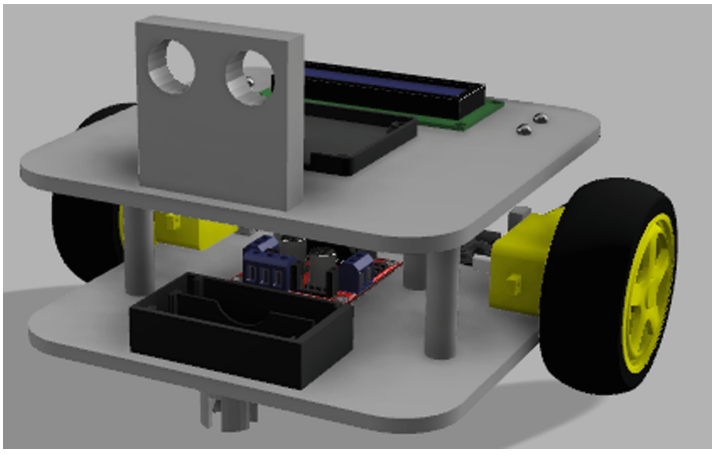


Fig. 1. 3D design of the first prototype based on the proposed architecture

We believe this design would benefit the sustainability of the project due to the minimum coupling between the hardware components. However, changes in the selected components have the potential to disrupt the hardware structure in ways that might require some changes in the design. We believe, though, that the open-source nature of the project

and the flexibility of the design should allow for those changes to be performed without too much trouble.

3 Software Architecture

The inclusion of a web server in the robot itself is a proposal that was already present in the world of educational robotics. In (Saleiro et al. 2013) the objective is to enable the programming of the robot from any Wi-Fi device through a browser with javascript, and without installing any additional software. The use of the Blockly framework as a programming tool is also proposed. In other cases (Erdemir et al. 2015), the server has the function of allowing the use of the robot remotely. But in none of the examples found, the objective is to make a concurrent use of the robot by the students.

The following architecture diagram shows the different components of the system and how they interact to allow all the students in the classroom to simultaneously program the robot (Fig. 2).

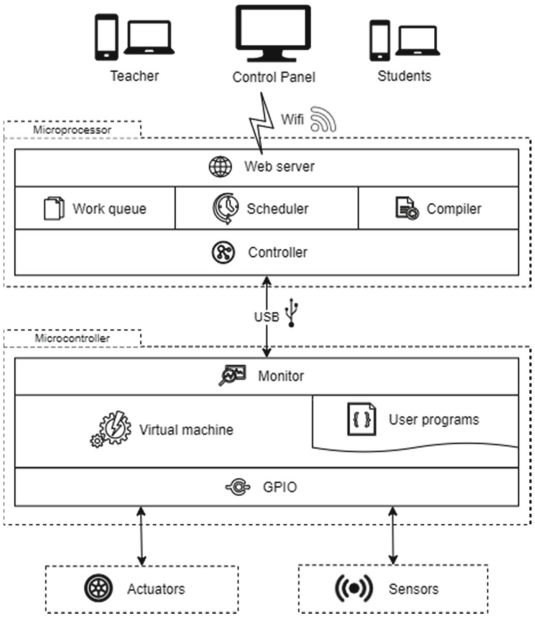


Fig. 2. Mendieta's software architecture diagram

Both students and teachers will be able to access the device through the web browser (already present on users' devices). In order to do this, the robot's microprocessor will expose a web server that will grant access to a graphical user interface including both the configuration and administration of the programming queue, as well as the integrated development environment from where students can program the robot.

This design will allow the robot to be programmed wirelessly (through the Wi-Fi network) and using any type of device (notebook, desktop computer, cell phone, tablet,

etc.) without requiring users to install anything. Optionally, a control panel can also be connected to allow the entire classroom to view the queue of pending programs.

The scheduler module is responsible for deciding when each program will be executed. To do this, it uses the work queue, which stores each of the programs sent by the students. By default, the order of arrival will be respected, but if the teacher so wishes, the order of execution of the assignments can be manually modified, as well as eliminating programs from the queue. Whenever a program is ready to be executed a message will appear in the students' device notifying them that it is their turn to use the robot. At that time, students will have a fixed amount of time to test their program and evaluate how well it performs the assigned task.

The compiler module is responsible for generating the instructions that the microcontroller will execute as well as ensuring that the programs added to the queue are syntactically valid. The compiler can be invoked by any user connected to the robot at any time, this way the students can verify their programs before submitting them to the work queue.

The controller module will be responsible for the communication with the microcontroller. This includes managing the program execution and allowing both control of the robot as well as receiving status information that can then be made available to the web server.

The microcontroller will be preprogrammed with a firmware that allows interactive programming of the robot from the microprocessor using the Serial port. This firmware consists of a monitor program that will be responsible for the communication protocol with the controller module, and a virtual machine that will execute the programs written by the user.

The monitor program will periodically send the status of the robot to the controller module. This status data includes the current values of all sensors in the robot, as well as all the variables in the program. The controller module can also send commands to the monitor, which will be executed as soon as they arrive. These commands allow the controller to fully control the virtual machine, including directly manipulating all variables and pins, debugging the current program, or executing a new one.

The virtual machine implements a stack-based bytecode interpreter to execute the user programs. This implementation was chosen mainly because of its simplicity. Since the purpose of this project is educational, performance is not currently considered a high priority.

The GPIO module allows the user program and the virtual machine to access the pins, sensors, and actuators with a generic interface that facilitates porting the firmware to other microcontrollers, if necessary.

Regarding the programming tools, Mendieta uses a fork of Physical Bits,[2] a web-based programming environment for educational robotics that supports live coding and autonomy using a hybrid blocks/text programming language (Fig. 3).

Several changes to Physical Bits were needed for this project, the biggest being porting the middleware code from Squeak Smalltalk to Clojure. Our early tests showed that performance on the selected microprocessors (at the time, Raspberry Pi Zero W and Orange Pi Zero) was less than ideal and well below our requirements. We experimented

[2] Physical Bits website: https://gira.github.io/PhysicalBits/.

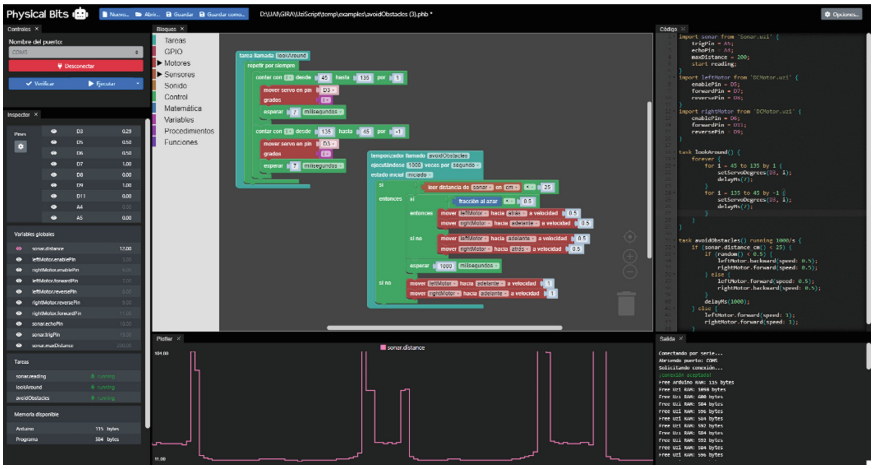


Fig. 3. User interface for robot programming

with different languages and eventually settled on Clojure due to its performance and language characteristics. In particular, the possibility of deploying the code on top of either Java or Javascript (thanks to ClojureScript), and the REPL and its live coding facilities (which reminded us of Smalltalk and its live image).

Additionally, to make a more efficient use of the microprocessor limited resources we are working on moving most of the compiler from the middleware to the clients. This way the cost of compiling the students' programs will be shared by all the computers in the classroom instead of falling into the microprocessor. For this change we are taking advantage of the Clojure and ClojureScript compatibility mentioned above.

We plan on making some other smaller changes that mostly involve changing the firmware to reduce the reporting frequency of pins and globals, as well as adding support for optical encoders and car movement primitives.

Considering that users are expected to spend most of their time working on their programs without immediate access to an actual robot (until their turn arrives), we are developing a simple simulator that would allow students to perform some basic validations for their programs while they wait for their turn to test in the actual robot. Although limited, we believe this to be an effective development technique and we have used it extensively for the development of the Physical Bits firmware.

In order to allow students to visualize both the sensors available in the robot and the variables defined in the program, a monitoring tool will allow the students to choose what variable they want to observe and display its value over time using different visualizations. This monitoring tool would be available to all students regardless of their respective turns, allowing students to monitor the robot values even while it is executing another student's program. An early prototype of the monitoring interface can be seen below:

In the case of complex programs, some kind of debugging tool is essential to allow stopping the execution of the program, executing each instruction step by step, and observing the status of the program in detail in order to understand and resolve errors in its logic (Fig. 4).

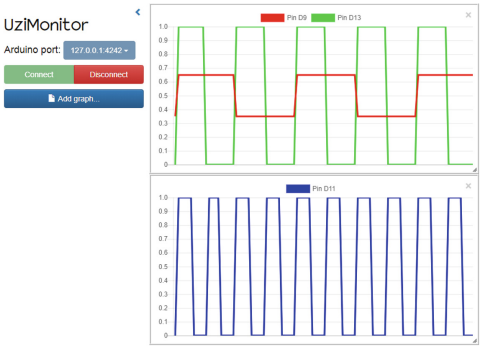


Fig. 4. User interface for monitoring variables and pins

This debugger prototype was fully functional in the Smalltalk version of the Physical Bits IDE but is not yet ported to Clojure (Fig. 5).

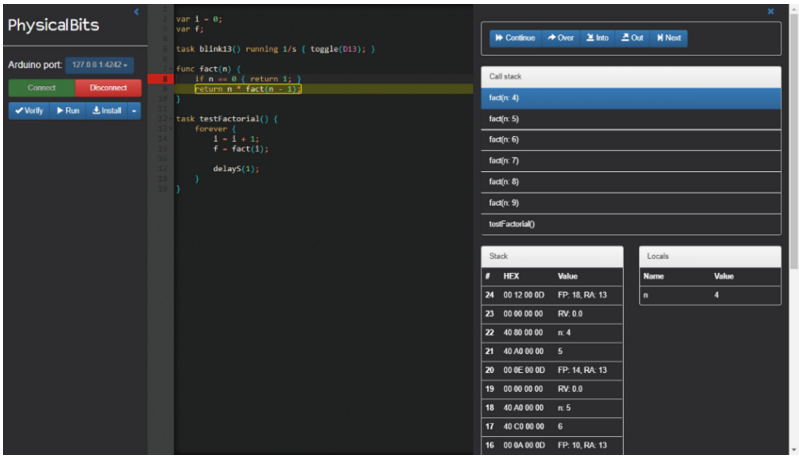


Fig. 5. User interface for debugging (early prototype)

Finally, the teacher will be able to access an administration interface that will allow him to view and control the queue of programs to be executed in the robot. This interface will allow to open and inspect the programs that were submitted to the robot, change their ordering, or even remove them from the queue. A prototype of the administration interface is shown below (Fig. 6).

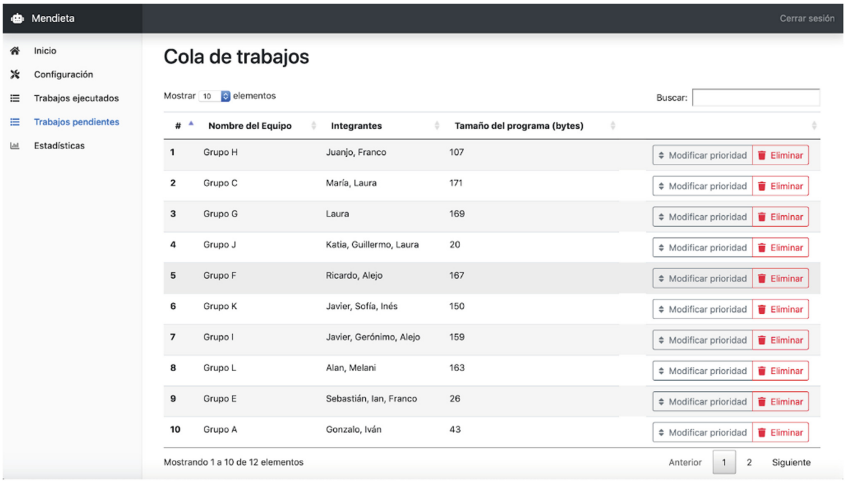


Fig. 6. User interface for managing the work queue

4 Schools' Involvement in the Pedagogical Decisions

Based on the principles of design thinking, for requirements analysis and testing cycles, we are working with five schools in our country with different characteristics. Some of them have experience using robots in the classroom, while the others are doing their first experience. In this way, in the first rounds of discussion, the view provided by the most experienced was essential. But at the same time, when we begin the testing cycles, inexperienced teachers will give us a perspective that will not be influenced by all previous experience and knowledge. In this way, by having the two groups of experimentation, we will be able to determine more clearly if the success or failure is given by our proposal or by the teaching experience.

Before designing the robot, we had four meetings with the schools to define the requirements for both hardware and software, which are what defined the characteristics previously presented. Subsequently, the emergence of the pandemic and the quarantine imposed in our country prevented us from carrying out the construction and testing stages, which we hope to be able to carry out this year.

5 Conclusions and Future Work

Mendieta is still in active development, we are currently working on a first prototype that will allow us to evaluate its effectiveness in the classroom. It is very likely that there will be changes to both its hardware and software architectures. Even though in this first iteration both the physical structure as well as the electronics components are well defined, the prototype we have developed so far is only a test model. More important than just this prototype is the framework we have defined that will allow building different robots under the same scheme of use (one robot per class).

It should be noted that it is possible to extend the use of Mendieta to support virtual education, by developing a site where students can watch the robot through one or

more cameras and remotely program it using the same web tools that we have already developed. In that case, both the robot and its environment should be able to return to its initial state automatically.

Although we have decided to use a blocks-based environment as the default programming model we hope that other groups can take this as a starting point to create other types of interfaces.

It should be noted that this proposal was presented in 2019 for the Clarín Zurich Award for Education, obtaining a mention that has allowed to finance a large part of the project. We have also been working with five schools around the country in the design and testing of this prototype. These schools have different orientations as well as various levels of experience with robotics, so their input on the project has been invaluable.

References

Arduino Nano | Arduino Official Store. (s/f). Recuperado el 15 de junio de 2020, de. https://store.arduino.cc/usa/arduino-nano

Aroca, R.V., Gomes, R.B., Tavares, D.M., Souza, A.A.S., Burlamaqui, A.M., Caurin, G.A., Goncalves, L.M.: Increasing students' interest with low-cost Cell Bots. IEEE Trans. Educ. **56**(1), 3–8 (2012)

Banchoff Tzancoff, C.M., Martin, S., Gómez, S., López, F.E.M.: Experiencias en robótica educativa: Diez años trabajando con escuelas argentinas. XIV Congreso Nacional de Tecnología en Educación y Educación en Tecnología (TE&ET 2019), (Universidad Nacional de San Luis, 1 y 2 de julio de 2019) (2019)

Darrah, T., Hutchins, N., Biswas, G.: Design and development of a low-cost open-source robotics education platform. In: ISR 2018; 50th International Symposium on Robotics, pp. 1–4 (2018)

Erdemir, G., Kuzucuoglu, A.E., Kaplanoglu, E., El-Kahlout, Y.: Design and Implementation of Web Based Mobile Robot Control Platform for Robotics Education. Applied Mechanics and Materials; Trans Tech Publications Ltd. (2015). https://doi.org/10.4028/www.scientific.net/AMM.704.283

Finocchiaro, A.: Núcleos de Aprendizaje Prioritarios para Educación Digital, Programación y Robótica-Resolución CFE 343/2018

Freaks, E.: Ultrasonic ranging module hc-sr04. HC-SR04 datasheet (2016)

Handson Technology. (s/f). L298N Motor Driver. Recuperado el 15 de junio de 2020, de. http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf

Hitachi. (s/f). HD44780U (LCD-II), (Dot Matrix Liquid Crystal Display Controller/Driver). 60

Hostler, A., Benson, B., Warner, J.: BudgetROV: an ultra low cost robotics platform for education and research. In: OCEANS 2017-Aberdeen, pp. 1–4 (2017)

Junior, L.A., Neto, O.T., Hernandez, M.F., Martins, P.S., Roger, L.L., Guerra, F.A.: A low-cost and simple arduino-based educational robotics kit. Cyber J. Multidisciplinary J. Sci. Technol. J. Sel. Areas Robot. Control (JSRC), December edition, **3**(12), 1–7 (2013)

Lopes Filho, J.A.B., Almeida, W.R.M., Martins, S.G.: Development of a multitasking mobile robot for the construction of educational robotics kits. In: 2011 International Conference on Electronic Devices, Systems and Applications (ICEDSA), pp. 213–216 (2011)

López-Rodríguez, F.M., Cuesta, F.: Andruino-A1: low-cost educational mobile robot based on android and arduino. J. Intell. Robot. Syst. **81**(1), 63–76 (2016). https://doi.org/10.1007/s10846-015-0227-x

MOCH22A Interrupter Datasheet pdf—OPTO Interrupter. Equivalent, Catalog. (s/f). Recuperado el 15 de junio de 2020, de. https://datasheetspdf.com/pdf/685501/Junye/MOCH22A/4

Motor Yellow 3-12VDC 2 Flats Shaft. (s/f). Wiltronics. Recuperado el 15 de junio de 2020, de. https://www.wiltronics.com.au/product/10137/yellow-motor-3-12vdc-2-flats-shaft/

Ministerio de Educación - Presidencia de la Nación. (2017). Escuelas del Futuro. Recuperado el 12 de junio de 2020, de. http://www.bnm.me.gov.ar/giga1/documentos/EL005852.pdf

Munizaga, R.A., Moleker, C., Yonzo, G., Zabala, G.: Everyone to robotics: an educational robotics project in Argentina. In: Robocup 2013 (2013)

NanJing Top Power ASIC Corp. (s/f). TP4056 1A Standalone Linear Li-lon Battery Charger with Thermal Regulation in SOP-8. Recuperado el 15 de junio de 2020, de https://www.semtronics.net/shop/resources/pdf_document/18/c1/9.pdf

orange pi zero—Orangepi. (s/f). Recuperado el 15 de junio de 2020, de http://www.orangepi.org/orangepizero/

Ripani, María Florencia. Programación y robótica: Objetivos de aprendizaje para la educación obligatoria. Ministerio de Educación - Presidencia de la Nación (2017). https://www.argentina.gob.ar/sites/default/files/programacion_y_robotica_0.pdf

Rubenstein, M., Ahler, C., Nagpal, R.: Kilobot: a low cost scalable robot system for collective behaviors. In: 2012 IEEE International Conference on Robotics and Automation, pp. 3293–3298 (2012)

Saleiro, M., Carmo, B., Rodrigues, J.M., du Buf, J.H.: A low-cost classroom-oriented educational robotics system. In: International Conference on Social Robotics, pp. 74–83 (2013)

Semiconductors, V.: TCRT5000, TCRT5000L, August 2009