Facultad de Ingeniería
ISSN: 0121-1129

Universidad Pedagógica y Tecnológica de Colombia

Bacca-Cortés, Bladimir; Florián-Gaviria, Beatriz; García, Sergio; Rueda, Sebastian

Development of a platform for teaching basic programming using mobile robots

Facultad de Ingeniería, vol. 26, no. 45, 2017, pp. 61-70
Universidad Pedagógica y Tecnológica de Colombia

DOI: https://doi.org/10.19053/01211129.v26.n45.2017.6054

Development of a platform for teaching basic programming using mobile robots

Desarrollo de una plataforma para enseñar programación básica usando robots móviles

Desenvolvimento de uma plataforma para ensinar programação básica usando robôs móveis

Bladimir Bacca-Cortés[*]
Beatriz Florián-Gaviria[**]
Sergio García[***]
Sebastián Rueda[****]

Fecha de recepción: 24 de octubre de 2016
Fecha de aprobación: 2 de abril de 2017

Abstract

Mobile robotics is being used in different education contexts, such as basic, middle, and high-level education. A literature review showed that 197 papers have been published in this area of knowledge over the past 10 years. Nowadays, Latin America faces a serious problem due to the low student enrollment in engineering programs, where, depending on the country, the ratio of graduate engineers can be 1 per 4500 to 1 per 10 000 people. In Colombia, the SPADIES program of the Ministry of Education affirms that the lack of motivation and interaction with real artifacts relating theory and practice is an important aspect for dropout. In this paper, a platform composed by a set of programmable mobile robots, and a WEB-responsive software tool for programming at different levels of knowledge was implemented. The set of mobile robots included sensors such as proximity, trajectory, light, inertial, and vision; also, communication and user interaction tools, such as Bluetooth and colored LEDs-ring, and a mechanical support for an erasable marker were included. The WEB-responsive tool supports graphical programming for novice; Python programming, for middle; and ANSI-C, for advanced learners. This platform consolidates a hands-on tool to introduce students to STEM concepts. Results are reported in the context of platform functionality, using all three programming environments, and beta tests with real users.

Keywords: Engineer education; Mobile robotics; WEB-responsive.

Resumen

La robótica móvil es utilizada en diferentes niveles educativos, tales como básica, intermedia y universitaria. Una revisión del estado del arte desarrollado por Benitti mostró que 197 artículos científicos han sido publicados en esta área del conocimiento en los 10 últimos años. Actualmente, América Latina enfrenta serios problemas relacionados con el número de estudiantes de Ingeniería matriculados; dependiendo del país, existe una relación del ingeniero

[*] Ph. D. Universidad del Valle (Cali-Valle del Cauca, Colombia). ORCID: 0000-0003-0113-4134.
[**] Ph. D. Universidad del Valle (Cali-Valle del Cauca, Colombia).
[***] Universidad del Valle (Cali-Valle del Cauca, Colombia).
[****] Universidad del Valle (Cali-Valle del Cauca, Colombia).

por 4500 a 10 000 personas. En Colombia, el programa SPADIES, del Ministerio de Educación, afirma que la falta de motivación e interacción con artefactos reales que relacionen teoría y práctica es un importante aspecto de deserción. En este trabajo fue implementada una plataforma compuesta por un conjunto de robots móviles programables y una herramienta software WEB responsiva para programar a diferentes niveles de conocimiento. El conjunto de robots móviles incluyó sensores de proximidad, trayectoria, luz, inerciales y visión; también, herramientas de comunicación y de interacción con el usuario, tales como Bluetooth y un anillo coloreado de LEDs, y un soporte mecánico para un marcador borrable. La herramienta WEB responsiva soporta programación gráfica, para novatos; programación en Python, para usuarios intermedios, y programación en ANSI-C, para usuarios avanzados. Esta plataforma consolida una herramienta práctica para introducir a los estudiantes en los conceptos STEM. Los resultados son reportados en el contexto de la funcionalidad de la plataforma, usando todos los tres ambientes de programación, y en pruebas beta con usuarios reales.

Palabras clave: Educación en ingeniería; Robótica móvil; WEB responsiva.

Resumo

A robótica móvel é utilizada em diferentes níveis educativos, tais como básico, intermediário e universitário. Uma revisão do estado da arte desenvolvida por Benitti mostrou que 197 artigos científicos têm sido publicados nesta área do conhecimento nos 10 últimos anos. Atualmente, a América Latina enfrenta sérios problemas relacionados com o número de estudantes de Engenharia matriculados; dependendo do país, existe uma relação de 1 engenheiro por 4500 a 10000 pessoas. Na Colômbia, o programa SPADIES, do Ministério de Educação, afirma que a falta de motivação e interação com artefatos reais que relacionem teoria e prática é um importante aspecto de deserção. Neste trabalho foi implementada uma plataforma composta por um conjunto de robôs móveis programáveis e uma ferramenta software WEB responsiva para programar em diferentes níveis de conhecimento. O conjunto de robôs móveis incluiu sensores de proximidade, trajetória, luz, inerciais e visão; também, ferramentas de comunicação e de interação com o usuário, tais como Bluetooth e um anel colorido de LEDs, e um suporte mecânico para um marcador borrável. A ferramenta WEB responsiva suporta programação gráfica, para principiantes; programação em Python, para usuários intermediários, e programação em ANSI-C, para usuários avançados. Esta plataforma consolida uma ferramenta prática para introduzir aos estudantes nos conceitos STEM. Os resultados são reportados no contexto da funcionalidade da plataforma, usando todos os três ambientes de programação, e em provas beta com usuários reais.

Palavras chave: Educação em engenharia; Robótica móvel; WEB responsiva.

Cómo citar este artículo:
B. Bacca-Cortés, B. Florián-Gaviria, S. García, and S. Rueda, "Development of a platform for teaching basic programming using mobile robots," *Rev. Fac. Ing.*, vol. 26 (45), pp. 61-70, May. 2017.

I. Introduction

Robotics is a valuable learning resource that allows students building their own concepts on science, technology, engineering, and math (STEM). The main goal of using robotics on education is learning STEM concepts through hands-on learning activities [4], and in this way encouraging team work within a multi-disciplinary environment. In the past few years, robotics on education has become an interesting research topic; for instance, [1] and [5] reported 197 different scientific works on this area over the past 10 years. These studies showed that robotics on education is an important tool to improve the learning process, not only at secondary level, but in Universities as well. Nowadays, Latin America faces a serious problem of student enrollment in engineering programs [2]. In Colombia, according to the Ministry of Education [3], there is 1 engineering student per 1864 people, and only 5 % of the students enrolled in engineering programs finish their studies. Furthermore, according to the Desertion Prevention System for High Education (SPADIES) program, one of main causes of dropout of engineering students in first semesters is the lack of interaction with real learning artifacts.

Currently, many mobile robotic platforms are available for engineering education. A summary of those more representative platforms is shown in Table 1. In this table, we compare the used development environment, the supported programming languages, the operative system, the level of knowledge required to deal with the mobile robot, and the type of licensing that is essential to guarantee long-term use of the learning platform. Observing Table 1, most platforms strictly constraint the type of users able to deal with the mobile robot, *i.e.* basic, advanced or intermediate. In addition, most of them have commercial licenses, which could limit their use. Advanced robotic platforms normally have a slow learning process; however, basic robotic platforms have a limited life cycle, which is an important observation made in the Mobile Robotics Seed Bed developed by the Perception and Intelligent Systems research group [6].

Table 1

State of the art summary of mobile robotics platforms for education and research

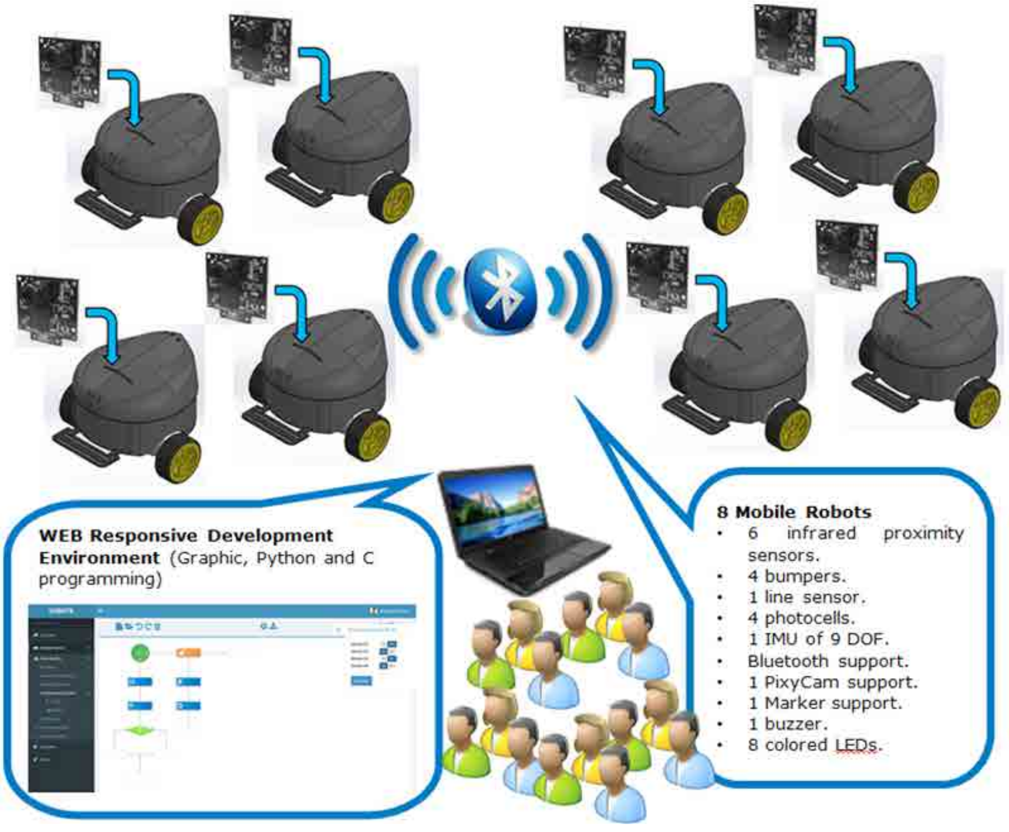
Mobile Robot	Dev. Env.	Prog. Lang.	Operat. Syst.	Learner Level	License
Hemisson [7]	BotStudio	Graphic	Multi-platform	Basic	Commercial
Handy Crickets [8]	Cricket Logo	Scripts	Windows, MAC	Basic	Commercial
Khepera [7]	WebBots	C, C++ Matlab, Labview	Multi-platform	Advanced	Commercial
Pioneer 3DX [9]	ARIA	C++, Java Phyton	Multi-platform	Advanced	Commercial
Pekee [10]	PekeeAPI	C, C++, Java Matlab, C#	Multi-platform	Advanced	Commercial
iCreate [11]	ROS	C++, Java, Python	Multi-platform	Advanced	Commercial
Pob-bot [12]	RisBee	Graphic, Basic and C	Multi-platform	Basic, Middle, Advanced	Commercial
ArduinoBot [13]	Arduino IDE	Arduino	Multi-platform	Middle	GNU
Thymio II[14]	Aseba Studio	Graphic and Aseba	Multi-platform	Basic and Middle	Commercial
The Finch [15]	Finch Dreams	Multi-language	Multi-platform	Basic, Middle and advanced	GNU
Pololu M3Pi [16]	AVR IDE	C, C++, Arduino	Multi-platform	Middle and advanced	GNU
Edison [17]	EdWare	Graphic	Multi-platform	Basic	GNU
Scribbler 2 [18]	S2 GUI	Graphic, Basic, C	Multi-platform	Basic, Middle	GNU

Recently, platforms such as Pob-Bot [12], Thymio 2 [14], The Finch [15], and UVBots1 [19, 20] bring to the users important properties such as multiplatform support, platform programming using various languages, and GNU licensing. Nowadays, STEM concepts, and specifically basic programming concepts, are also learnt using WEB applications, and on-line mini-courses. For instance, CODE [21] is a mini-course of 20 hours where learners acquire the fundamental concepts of computer science using animations, and popular movies or game characters for kids; depending of the learner knowledge level there are many available mini-courses to choose. App Inventor [22] is an innovative programming environment for beginners app developers; it uses drag-and-drop blocks to create simple apps for Android SO. It is worth noting that all these software applications are oriented only two one user profile, and they do not support an increasing level of knowledge complexity. This paper proposes a hands-on platform composed by a set of programmable mobile robots, and a WEB-responsive development environment; both of them support increasing level of complexity, which allow to include learners with any knowledge on mobile robotics or programming, multiplatform support, and GNU licensing. The main goal of this platform is to provide a mobile robot as a real artifact to learn STEM concepts at early semesters of engineering programs.

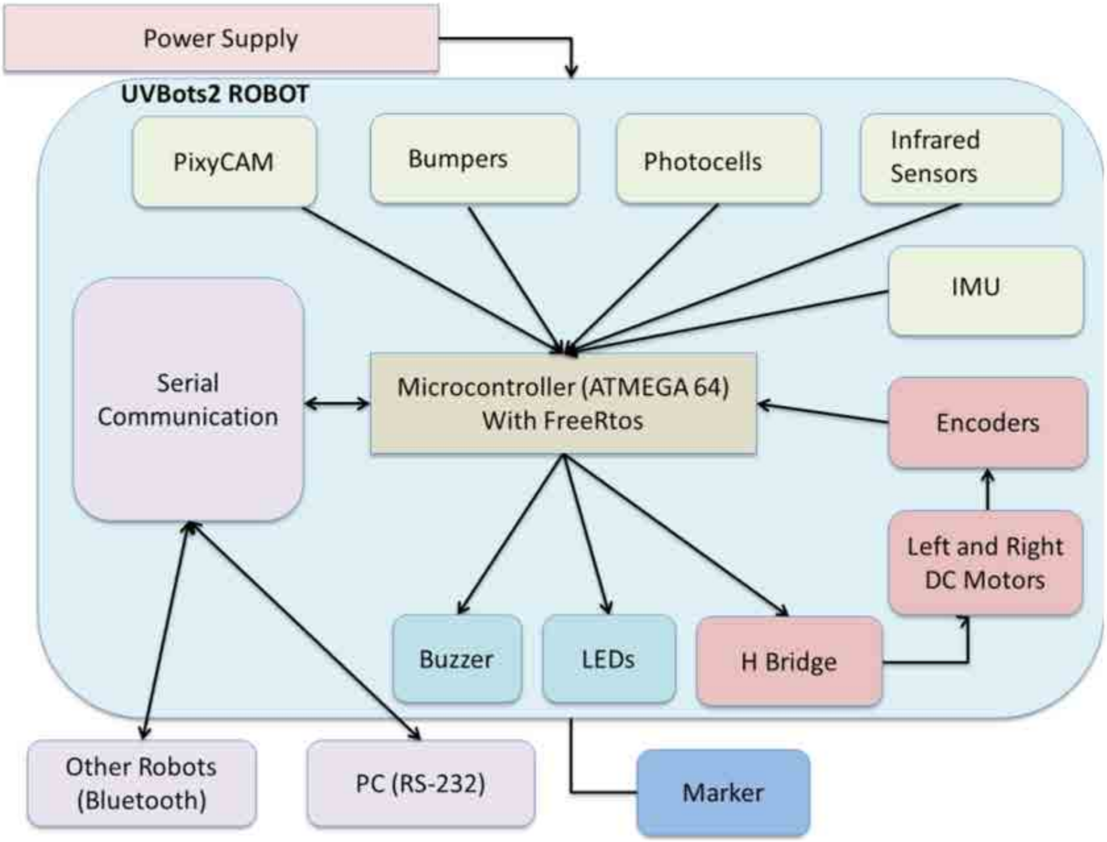
II. Platform Description

The proposed platform (Fig. 1a) is composed by eight mobile robots and a WEB-responsive application that can be programmed considering three levels of complexity, depending on the learner expertise. The mobile robots support Bluetooth communication, but they are programmed using a PC through a serial link. The WEB-responsive application used to program the mobile robots runs into a PC or Laptop. The three levels of complexity offer to learner three programming environment interfaces: graphical (basic level), Python (intermediate level), and C (advanced level). Within all levels of complexity, learners are able to use all the perception, motion, communication, and interaction tools from the mobile robots.

(a)



(b)



(c)

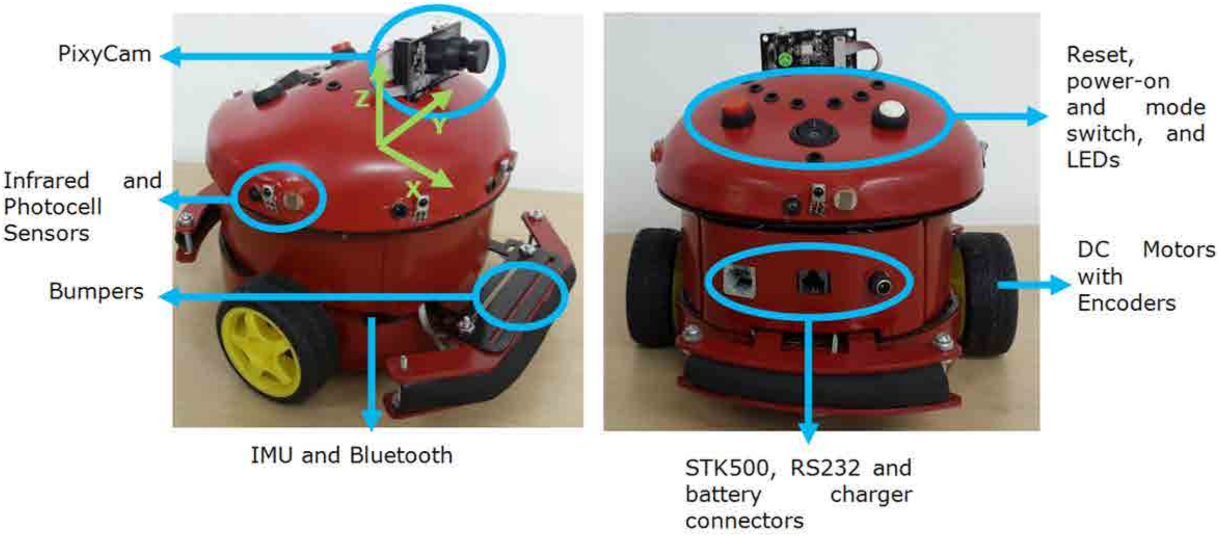

PixyCam
Z
Y
X
Infrared and Photocell Sensors
Bumpers
IMU and Bluetooth
Reset, power-on and mode switch, and LEDs
DC Motors with Encoders
STK500, RS232 and battery charger connectors

Fig. 1. a) Hands-on platform conceptual diagram. b) Hardware block diagram. c) UVBots2 mobile robot and Placement of the perception system and coordinate frame.

A. Mobile robots hardware description

Each one of the eight UVBots2 was equally designed from a hardware and firmware point of view. This section briefly describes hardware modules, such as motion, perception, and communication. The UVBots2 hardware block diagram is shown in Figure 1b. The UVBots2 mobile robot has the following hardware requirements: a power system with a 12V at 1800mAh battery pack, a differential motion system with position sensing, a perception system including proximity, light, inertial, and vision sensors, Bluetooth communication, user interaction system composed by eight colored LEDs, and a CPU running FreeRTOS 7.4.2 [23].

The differential kinematics model assumed for each robot is shown in (1). The DC motor model assumed was lineal with K = 5.81 and t = 221.2ms, then a PI controller was designed considering the following requirements: a steady-state time of 750ms, no overshooting, and a natural system frequency of 5.33 rad/s. The PI controller designed is shown in (2).

$$\begin{bmatrix} V_R \\ W_R \end{bmatrix} = \begin{bmatrix} c/2 & c/2 \\ c/b & \text{-}c/b \end{bmatrix} \begin{bmatrix} w_r \\ w_l \end{bmatrix} \qquad (1)$$

$$C(s) = \frac{1.3595*(1+0.2161s)}{0.2161s} \qquad (2)$$

Where, V_R and W_R are the lineal and rotational robot velocities, c is the wheel radius, b is the wheel distance, w_r and w_l are the rotational wheel speeds.

The UVBots2 perception system is composed by the following modules: proximity sensing, light intensity measurements, inertial measurement, and vision (Fig. 1c); these systems are briefly described as follows: *Proximity perception* is implemented using four bumpers, and six GP1UD28YK infrared receivers; *Light intensity perception* is implemented with four photocells; *Inertial measurements* are performed by a MiniIMU9v3 IMU with 9 DOF; and *Vision perception* is implemented using the PixyCam CMU5.

Human-robot interaction is an important aspect to know what the robot is doing at any moment. UVBots2 has two different types of feedback for learners: first, basic audio systems implemented using a buzzer, and second a set of eight colored LEDs (Fig. 1c). Also, UVBotsV2.0 robots are able to communicate to each other using Bluetooth by means of a HC-05 module, which can be configured as master or slave. However, to perform the PC–Robot communication, a RS232 link is needed.

III. Programming Environment

The programming environment was designed using WEB-responsive technology from the DJango framework [24]. This environment supports three different programming interfaces: basic, intermediate, and advanced. The basic level uses a graphical interface, which is designed for learners with basic knowledge on programming and robotics; the intermediate level uses Python programming language because it is easy to use and understand; and the advanced level uses ANSI-C programming language.

The software development process for this project was carried out using eXtreme Programming methodology [25]. The software functional requirements for the developed modules are described as follows (Fig. 2): *User module*: it performs user registration, sessions, user challenges, and program management; *Graphical programming module*: its aim is to show all graphical programming interface and tools; *Python programming module*: it implements the Python based programming interface and tools; *C Programming module*: it shows the C programming interface and tools; *Data model*: it defines a general schema for the UVBots2 functionalities and user data persistence; *Program conversion, compilation, and download*: its aim is to perform translation from graphical to Python programs, from graphical to C programs, and from Python to C programs; and *Challenges and samples module*: it shows a list of user challenges including the problem to solve, and the expected results.

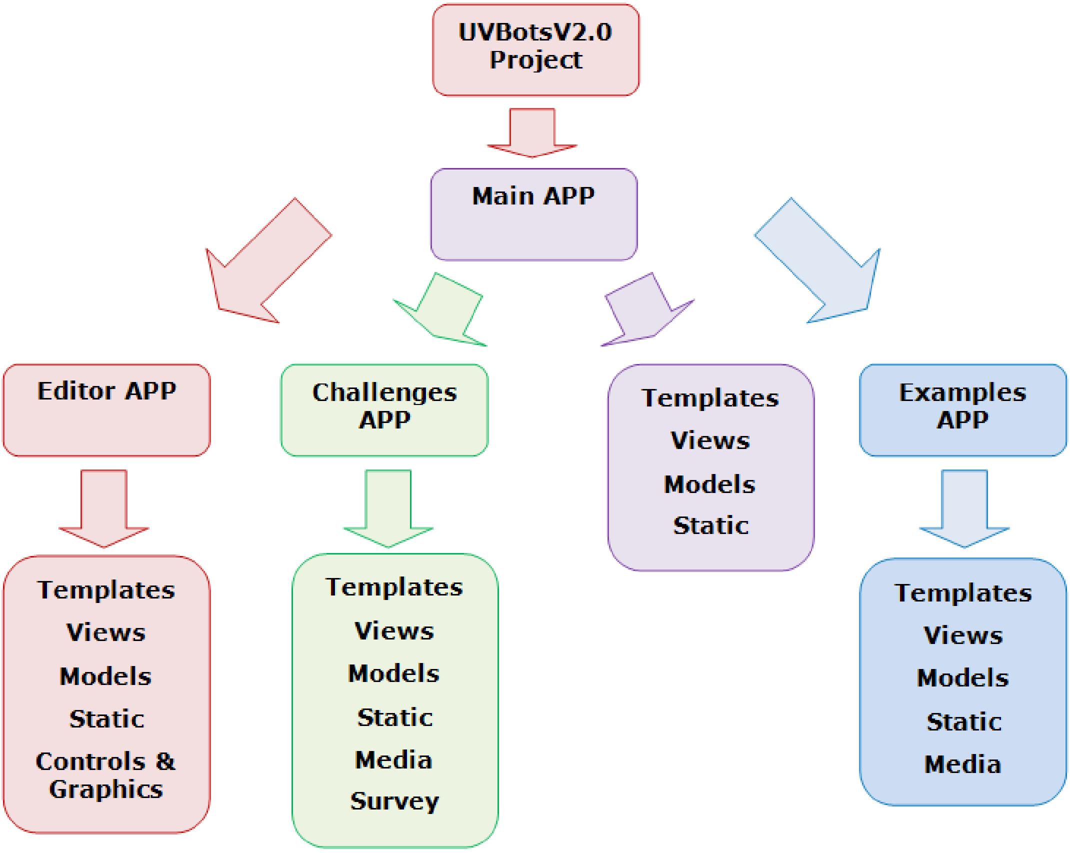


Fig. 2. Components of the UVBots2 software architecture.

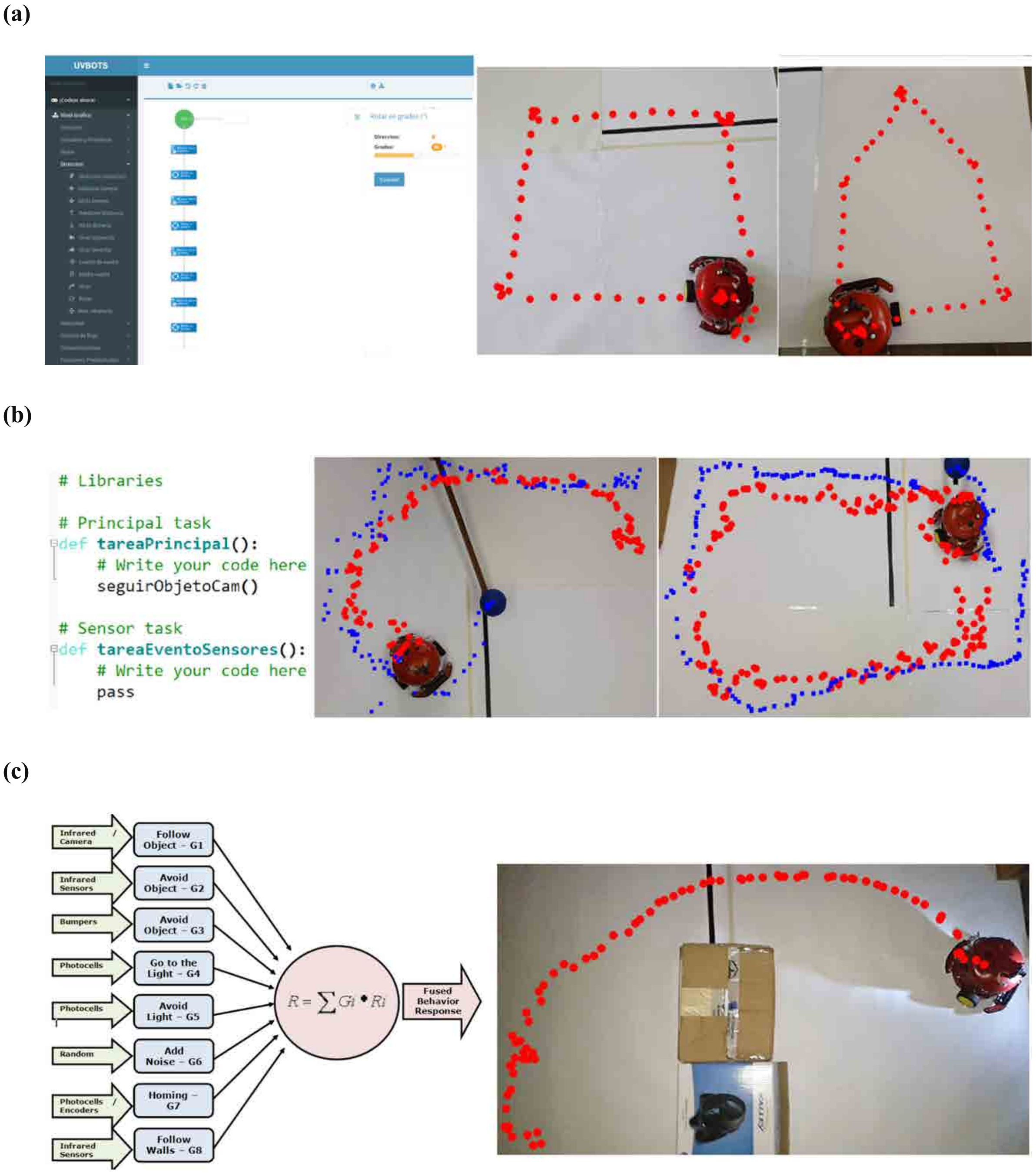
Graphical Programming Module: Learners can start programming using drag-and-drop operations from the programming blocks available on the left side. These blocks are added downwards into the main routine; however, robot sensor blocks are added horizontally, and they create an independent execution thread. All learner graphical programs are stored in XML files. Among these files, the most important stored data are user and program data, main and sensor routine blocks, block attributes, and C code to implement the functionality of each block; the latter is implemented using Python dictionaries. In this work, three different dictionaries were created: first, to translate block programming to C; second, block programming to Python; and third, Python programming to C.

Python Programming Module: In this module, learners can start programming using all built-in functions available for the UVBots2. Here, learners must consider two main routines: "*tareaPrincipal*" (*mainTask*), and "*tareaEventosSensores*" (*taskSensorEvents*). The former is used to introduce all Python sentences corresponding to the main execution thread, while the latter is used to introduce all Python sentences of the UVBots2 sensors.

Programming Module: The interface of this module, which is aimed at advanced learners, shows two C program templates: the "*tareaPrincipal.c*" or *mainTask*, and the "*tareaEventos.c*" or *eventTask*. In the same way as the previous module, they are used to store C code related with the main program, and C code related with the robot sensors. The FreeRTOS RT was used to program all UVBots2 functionalities.

III. Test and Results

To validate the proposed platform, two kinds of tests were performed. First, three individual integration tests, ranging from basic to advance programming, were conducted in order to show the main UVBots2 functionalities at different levels of learner knowledge; and second, beta tests were performed on real users.



```
# Libraries

# Principal task
def tareaPrincipal():
    # Write your code here
    seguirObjetoCam()

# Sensor task
def tareaEventoSensores():
    # Write your code here
    pass
```

Fig. 3. a) Geometric figures (square and house) drew using basic programming interface; b) Using a Python program, the mobile robot is able to follow a blue object using the PixyCam; c) Homing and obstacle avoidance behaviors programmed using the advanced programming interface.

The first integration test aimed at programming the UVBotV2.0 using the graphical environment to draw geometric figures (Fig. 3a). For this test, a Matlab program was developed to process the video captured while the test was performed. The red dots in Figure 3a show the UVBotsV2.0 trajectory. In the second integration test, a Python program was written to use the PixyCam; this program was in charge of following the blue ball in front of the UVBotV2.0 robot. Figure 3b shows the Python program and two different trajectories performed while the UVBot2 was following the ball. The third integration test used the

ANSI C programming environment, and implemented a behavior based programming application [26]. The robot homing behavior is very important in robotics, it allows mobile robots to return to the recharging stations, and to go to specific goals in the environment. In this work, the robot goal is the more illuminated region in Figure 3c, which is placed on the right, while avoiding obstacles. The set of behaviors supported by the UVBots2 are shown on the left side of Figure 3c. All the behaviors motion commands are coordinated using a weighted sum, each weight represents the behavior importance within the behavior stack. We employed the *Avoid Object*, and the *Go to the Light* behaviors in the test.

The beta test was performed by 10 students of Electronic Engineering at the Universidad del Valle. The students had neither information about the programming environment nor the mobile robot. All the information to perform this test was consolidated into a PDF document, which contained two challenges: one, using the graphical programming environment, and another, the C programming environment. At the end of these tests, a survey took place, and the questions and resulting statistics are shown as follows:

1. How do you evaluate your experience with the GUI? 50 % answered "Fun" and "Innovative", and 50 % answered "Intuitive"

2. In general, is the application slow to load its content? 90 % answered "No", and 10 % "Yes".

3. How easy is to understand the notification messages? 80 % answered "Easy", 20 % answered "Very easy".

4. How was your experience with the graphical programming interface? 50 % answered "Fun" and "Innovative", and 50 % answered "Intuitive".

5. Were the blocks easy to add to your graphical program? 70 % answered "Easy", 20 % answered "Very easy", and 10 % answered "Difficult".

6. Were the blocks easy to delete from your graphical program? 60 % answered "Easy", and 40 % answered "Difficult".

7. Were the block parameters easy to edit? 90 % answered "Yes", and 10 % answered "No".

8. Was the block category easy to differentiate by its color? 80 % answered "Yes", and 20 % answered "No".

9. Was the block based program easy to understand? 90 % answered "Yes, and 10 % answered "No".

10. What do you think about the C translation of the block based program? 70 % answered "Good", and 30 % answered "Very good".

Considering the beta test results described above, the programming environment is easy to understand, and use; and specially, this programming environment is suitable for novice, intermediate, and expert users.

IV. Conclusions

In this work, we presented the UVBots2 platform that includes a programming environment, and a set of eight mobile robots having three levels of knowledge. The development environment supports graphical, Python, and ANSI C programming languages, each one corresponding to basic, intermediate, and advanced levels of knowledge on programming or robotics. The UVBots2 platform offers an important experimentation tool, mainly used to execute all user programs; however, basic programming, robotics, or STEM concepts can be learnt. The development environment was implemented considering the following properties: use of WEB-responsive technologies, platform independence, ability to translate block based programs to Python, or ANSI C code, and from Python programs to ANSI C using dictionaries, and the possibility of saving, editing, downloading, and executing programs using the mobile robot. Finally, this development environment considers the user previous knowledge on programming and robotics, since it offers an increasing complexity learning methodology.

Recent works such as the Thymio II [14], the Finch [15], and the Edison [17] mobile robot platforms are examples of how different levels of knowledge, and programming languages provide high flexibility in the learning process. In the same context, UVBotsV2.0 platform provides longer-term learning experiences than other commercial platforms as Mindstorms EV3. In addition, UVBotsV2.0 platform can be used by users with advanced knowledge of programming and robotics, as well as by novice users. This capability allows users to use the UVBotsV2.0 platform in an incremental way, and they can notice their progress on programming skills depending on the type of programming language used.

Finally, considering the current governmental efforts in developing novel educational strategies [27] and the experiences in the Robotics Seedbed [6] the hands-on platform presented in this work represent an interesting option to bring children and young people to science, technology, engineering, and math.

References

[1] F. B. V. Benitti, "Exploring the educational potential of robotics in schools: A systematic review," *Comput. Educ.*, vol. 58 (3), pp. 978-988, Apr. 2012. DOI: http://doi.org/10.1016/j.compedu.2011.10.006.

[2] M. E. Argentina, "Plan Estratégico de Ingeniería 2012-2016," *Ministerio de Educación - Argentina*, 2014. Available: http://pefi.siu.edu.ar/. [Accessed: 13-Feb-2014].

[3] M. de Educación, "SPADIES - Ministerio de Educación Nacional de Colombia," 2014. Available: http://www.mineducacion.gov.co/1621/w3-article-156292.html. [Accessed: 13-Feb-2014].

[4] P. De Cristóforis, A. Member, S. Pedre, S. Member, M. Nitsche, T. Fischer, F. Pessacg, and C. Di Pietro, "A Behavior-Based Approach for Educational Robotics Activities," vol. 56 (1), pp. 61–66, 2013.

[5] L. Major, T. Kyriacou, and O. P. Brereton, "Systematic literature review: teaching novices programming using robots," in *15th Annu. Conf. Eval. Assess. Softw. Eng. (EASE 2011)*, pp. 21-30, 2011. DOI: http://doi.org/10.1049/ic.2011.0003.

[6] E. Milena, J. Jojoa, E. C. Bravo, E. Bladimir, and B. Cortés, "Tool for Experimenting With Concepts of Mobile Robotics as Applied to Children ' s Education," vol. 53 (1), pp. 88-95, 2010.

[7] C. K-Team, "K-Team Corporation, Mobile Robotics," 2016. Available: http://www.k-team.com/. [Accessed: 01-Jan-2016].

[8] HandyBoard, "Cricket Logo," 2009. Available: http://handyboard.com/cricket/. [Accessed: 01-Jan-2016].

[9] A. MobileRobots, "Pioneer Robots," 2013. Available: http://www.mobilerobots.com/ResearchRobots/. [Accessed: 01-Jan-2014].

[10] W. Robotics, "Pekee Robot," 2009. Available: http://www.wanyrobotics.com/robots.html. [Accessed: 01-Jan-2014].

[11] Ir. Corporation, "iCreate Robot," 2013. Available: http://www.irobot.com/us/learn/Educators/Create.aspx. [Accessed: 01-Jan-2014].

[12] A. Corporation, "Pob-Bot Robot," 2013. Available: http://education.awabot.com/. [Accessed: 01-Jan-2014].

[13] Arduino, "Arduino Robot," *Arduino*, 2013. Available: http://arduino.cc/en/Main/Robot. [Accessed: 01-Jan-2014].

[14] Aseba, "Thymio II Robot," *Aseba*, 2013. Available: http://aseba.wikidot.com/en:thymio. [Accessed: 01-Jan-2014].

[15] BirdBrain-Technologies, "The Finch Robot," 2016. Available: http://www.finchrobot.com/. [Accessed: 01-Jan-2016].

[16] Pololu-Corporation, "Pololu M3Pi Robot," 2016. Available: http://www.pololu.com/product/2151. [Accessed: 01-Jan-2016].

[17] Meet-Edison, "Meet Edison - Cheap Programable Lego," 2016. Available: http://meetedison.com/. [Accessed: 01-Jan-2016].

[18] Parallax, "Scribbler Robot," 2014. Available: http://www.parallax.com/product/28136. [Accessed: 01-Jan-2014].

[19] F. Gómez, F. Muñoz, B. E. Florián, C. A. Giraldo, and E. B. Bacca-Cortés, "Diseño y prueba de un robot móvil con tres niveles de complejidad para la experimentación en robótica. Design and testing of a mobile robot with three levels of complexity for robotics experimentation," *Ing. y Compet.*, vol. 74 (2), pp. 53-74, 2008.

[20] C. Giraldo, B. Florián, B. Bacca, F. Gómez, and F. Muñoz, "A programming environment having three levels of complexity for mobile robotics," *Ing. e Investig.*, vol. 32 (3), pp. 76-82, 2012.

[21] Code.org, "CODE - Anybody can learn," 2016. Available: http://code.org/. [Accessed: 01-Jan-2016].

[22] MIT, "MIT App Inventor," 2016. Available: http://appinventor.mit.edu/explore/about-us.html. [Accessed: 01-Jan-2016].

[23] R. Barry, "FreeRTOS - Market leading RTOS," 2016. Available: http://www.freertos.org/. [Accessed: 01-Jan-2016].

[24] DJango-Software-Foundation, "The Web Framework for Perfectionists with Deadlines - DJango," 2016. Available: http://www.djangoproject.com/. [Accessed: 01-Jan-2016].

[25] J. Shore, and S. Warden, *The Art of Agile Development*. Sebastopol, CA: O'Reilly Media, 2008.

[26] R. C. Arkin, *Behavior-Based Robotics (Intelligent Robotics and Autonomous Agents)*, Third. A Bradford Book, 1998.

[27] E. C. for the D. of V. T. (CEDEFOP), "Skills for Green Jobs (European Synthesis Report)," 2010.