

Development of a set of mobile robots for basic programming experimentation

Desarrollo de un conjunto de robots móviles para la experimentación en programación básica

Sebastián Rueda¹, Beatriz Florián - Gaviria², Bladimir Bacca - Cortes³

¹ Percepción y Sistemas Inteligentes, Escuela de Ingeniería Eléctrica y Electrónica, Universidad del Valle, Colombia.

Email: sebastian.rueda@correounivalle.edu.co

² Estudios Doctorales en Informática, Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle, Colombia.

Email: beatriz.florian@correounivalle.edu.co

³ Percepción y Sistemas Inteligentes, Escuela de Ingeniería Eléctrica y Electrónica, Universidad del Valle, Colombia.

Email: bladimir.bacca@correounivalle.edu.co

RECIBIDO: Mayo 19, 2016. ACEPTADO: Marzo 1, 2017. VERSIÓN FINAL: Mayo 15, 2017.

ABSTRACT

Mobile robots platforms are being used in different Educational contexts such as primary, middle and high school levels of education. A review of the state of the art shows that 197 papers have been published in this field of knowledge over the last 10 years [1]. Nowadays, Latin America faces a serious problem regarding student enrollment in engineering programs with a ratio of one graduated Engineer per every 4500 to 10000 people [2] depending on country. In Colombia, the SPADIES program [3] states that an important aspect of the dropout was due to the lack of motivation and interaction of students with real artifacts that linked the theory and the practice. In this work, a set of programmable mobile robots were developed considering different levels of knowledge, as a tool for basic programming experimentation. The set of mobile robots embodies sensors for proximity, line, light, inertial, and vision. Moreover, this set also incorporates tools such as Bluetooth, LEDs-ring and, a mechanical support for an erasable marker. This set of mobile robots consolidates a hands-on tool to introduce students to Science, Technology, Engineering and Math. The results reported in this paper show the mobile robots functionality when they are programmed using different levels of complexity.

KEYWORDS: Mobile robots, Engineering education, Basic programming.

RESUMEN

Las plataformas de robots móviles se usan contextos educativos a nivel de básica primaria, secundaria y a nivel universitario. Una revisión del estado del arte muestra que 197 artículos han sido publicados en esta área de conocimiento en los últimos 10 años [1]. Actualmente, América Latina enfrenta serios problemas en la inscripción de estudiantes en los programas de Ingeniería, registrando un graduado en Ingeniería por cada 4500 a 10000 personas [2] dependiendo de cada país. En Colombia, el programa SPADIES [3] afirma que la falta de motivación e interacción con artefactos reales relacionando la teoría y la práctica es un aspecto importante en la deserción. En este trabajo, se desarrolló un conjunto de robots móviles programables para personas con diferentes niveles de conocimiento, como una herramienta para la experimentación en programación básica. El conjunto de robots móviles incorporan sensores de proximidad, línea, luz, inerciales y visión; también los robots integran herramientas tales como Bluetooth y un anillo de LEDs son incluidos, y un soporte mecánico para un marcador borrable. Estos robots móviles consolidan una herramienta práctica para introducir a los estudiantes a la Ciencia, Tecnología, Ingeniería y Matemáticas. Los

resultados reportados en este artículo muestran la funcionalidad de los robots móviles al ser programados usando los diferentes niveles de complejidad.

PALABRAS CLAVE: Robots móviles, Educación en Ingeniería, Programación básica.

1. INTRODUCTION

Robotics is a valuable learning resource which allows students to acquire concepts of science, technology, engineering and math (STEM). The main idea of using robotics on education is learning STEM concepts through hands-on learning activities [4], and in this way team work is encouraged in a multi-disciplinary environment. Robotics on education has become an interesting research topic over last years. In [1] and [5] authors report 197 different scientific studies over the last 10 years. These studies indicate that robotics on education is an important tool to improve the learning process, not only at the secondary level but at University level as well.

Nowadays, Latin America faces a serious problem of student enrollment into Engineering programs [2]. In Colombia, according to statistics from the Ministry of Education in [3] there is one Engineering student in every 1864 people. In addition, these statistics show that only 5% of enrolled students in engineering programs finishes their studies. These statistics are an important concern for the Ministry of Education, which has implemented the Desertion Prevention System for Higher Education (SPADIES). One of the most important observations made by this program is the lack of student interaction with real learning artifacts during first semesters of engineering programs.

Therefore, this work is focused on developing a set of mobile robots with the aim to allow students with real learning artifacts interaction using a basic programming language as C with increasing complexity functionalities.

2. RELATED WORKS

Currently, many mobile robotic platforms are available and ready to use in engineering education. A summary of representative platforms is shown in Table 1. This table displays a comparison considering the programming languages supported, the learner's level of required knowledge, and License. The latter is essential to guarantee a long-term usage 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, intermediate or advanced users. Furthermore, most of these platforms have commercial licenses, which could limit their use. Table 1 also shows the existence of two

marginal positions: the robot platform requires either an advanced learner, or an amateur one. In addition, advanced robotic platforms normally have a slow learning process. However, basic robotic platforms become expensive toys. This important observation was also made in the Mobile Robotics Seed Bed developed by the Perception and Intelligent Systems research group [6].

Recently, platforms such as Pob-Bot [10], Thymio 2 [12], The Finch [14], and UVBots1 [21] [22] offer users properties such as multiplatform support, multi-languages programming, and GNU licensing. In addition, in Colombia other developments in this context are [23], [24] and [25].

Currently, STEM and more specifically basic programming concepts are also learnt by using WEB applications and on-line courses. Most of these applications or on-line courses do not need a real world platform to execute the programs implemented by learners. Then, descriptions of software applications using robotics platforms as well as teaching programming applications are described here. First, RoboLab [26] is a free graphical programming environment which supports RCX 2.0 and NXT Lego bricks. Bot-Studio [7] is a desktop application which uses GRAFCET language to program Hemisson and K-Junior robots; Bot-studio is a licensed application, and it is only distributed when buying these robots. Scribbler GUI [18] and Risbee [27] are free graphical programming environments for the Scribbler2 and POB robots respectively; the former are desktop and platform-dependent applications. Scratch [28] is a free graphical programming tool to introduce learners to STEM concepts using images, sounds, and multimedia resources. Later on, Scratch was used to create animated stories. CODE [29] is a 20-hours course where learners acquire the fundamental concepts of computer science using animations, and popular movies or game characters for children. Depending of the learner knowledge level there are many mini-course to choose. App Inventor [30] is an innovative programming environment for app developers in their beginner level; 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 to a unique user profile, they are not intended for users with different levels of knowledge.

Table 1. Summary of mobile robotics platforms for education and research.

| Robot | Prog. Lang. | Learner Level | Licens e |
|------------------|---|--|----------|
| Hemisson [7] | Graphic | Basic | Comm. |
| Khepera [7] | C/C++, Matlab, Labview | Advanced | Comm. |
| E-Puck [7] | C, Matlab | Advanced | Comm. |
| AmigoBot [8] | C++, Java Phyton | Advanced | Comm. |
| Pioneer 3DX [8] | C++, Java Phyton | Advanced | Comm. |
| iCreate [9] | C++, Java, Python | Advanced | Comm. |
| Pob-bot [10] | Graphic, Basic and C | Basic, Intermediate, Advanced | Comm. |
| ArduinoBot [11] | Arduino | Intermediate | GNU |
| Thymio II[12] | Graphic and Aseba | Basic and Intermediate | Comm. |
| Robotino [13] | C/C++, Java, Labview | Advanced | Comm. |
| The Finch [14] | Graphic, Java, Python, Scratch, C, C++, Matlab, VBasic, Labview... | Basic, Intermediate and advanced | GNU |
| Pololu M3Pi [15] | C, C++, Arduino | Intermediate and advanced | GNU |
| mBot [16] | Graphic, Scratch, Arduino | Basic | Comm. |
| Edison [17] | Graphic | Basic | GNU |
| Scribbler 2 [18] | Graphic, Basic, C | Basic, Intermediate | GNU |
| TurtleBot2 [19] | C, C++, Python | Advanced | GNU |
| MarxBot [20] | Graphic, Scratch, Aseba | Advanced | Comm. |
| K-Junior [7] | C, C++ | Advanced | Comm. |

This work proposes a hands-on platform built around a set of programmable mobile robots, since it is important to have a mobile robot as real artifact to learn STEM concepts in the first years of engineering programs. To

program de mobile robots previously discussed, it is necessary to use an adaptive level-complexity set of functionalities. In this way, learners with none or previous knowledge can be included in mobile robotics or programming concepts. Finally, the robot firmware supports GNU licensing.

3. HARDWARE DESCRIPTION OF MOBILE ROBOTS

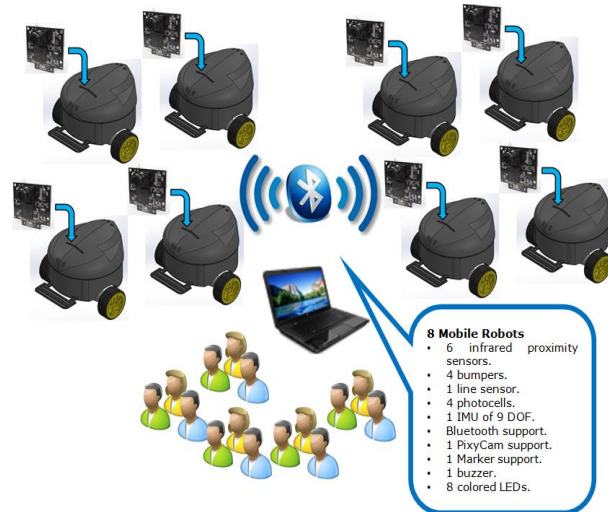
**Figure 1.** Conceptual diagram of the platform. **Source.** The authors.

Figure 1 shows the proposed platform. Here, 8 mobile robots can be programmed considering functionalities at three levels of complexity depending on the learner expertise. The mobile robots support Bluetooth communication among each other, and they are programmed using a PC or Laptop through a wired serial link. The PC or Laptop runs a standard programming environment such as AVRstudio. Within all levels of complexity, learners are able to use all functionalities related with perception, motion, communication, and interaction tools from the mobile robots. These include: six infrared proximity sensors, four bumpers, four photocells, one line sensor, one Inertial Measurement Unit (IMU) of nine degrees of freedom (DOF), one PixyCam, two incremental encoders of 562 ticks per revolution, two DC motors in differential drive configuration, one Bluetooth communication device, one buzzer, and eight colored LEDs.

Each one of the eight UVBots2 was equally designed from their hardware and firmware point of view. The next section briefly describes hardware design (Section 3.1) and modules relating to perception (Section 3.2), motion (Section 3.3) and communication (Section 3.4).

3.1 Design Requirements

The UVBots2 hardware platform was conceived considering the block diagram shown in Figure 2a. Technical specifications of UVBots2 are described next:

- The Power System: It consists of one rechargeable battery package of 12V at 1800mAh, and two power regulation sub-systems which are the robot electronics and the motion system.
- The Motion System: It is a differential drive robot with two drive motors with 50oz/in (Pololu 6VDC), and two incremental encoders with 562 ticks by revolution.
- The Perception System: It includes four bumpers, six infrared proximity sensors (GPIUM28YK), four photocells, one IMU (MiniIMU9v3), one line sensor, and one camera (CMUcam5).
- The Communication and User Feedback System: It consists of a PC-Robot communication using RS-232, Robot-Robot communication uses Bluetooth (HC-05), one buzzer, and eight colored LEDs. And an ATMEGA644-20PU as the main CPU running FreeRTOS 7.4.2 [31].

Figure 2b shows the UVBots2 robot mechanical structure with the following requirements: first, an aesthetically appealing housing to encapsulate a metallic chassis, the battery pack, motors and hardware electronics. Second, three upper buttons to power on/off, reset and change the execution mode of the mobile robot. Third, a marker-handle support at the bottom part of the robot. Last, three connectors for downloading the boot-loader, the PC-Robot communication and recharging the battery pack.

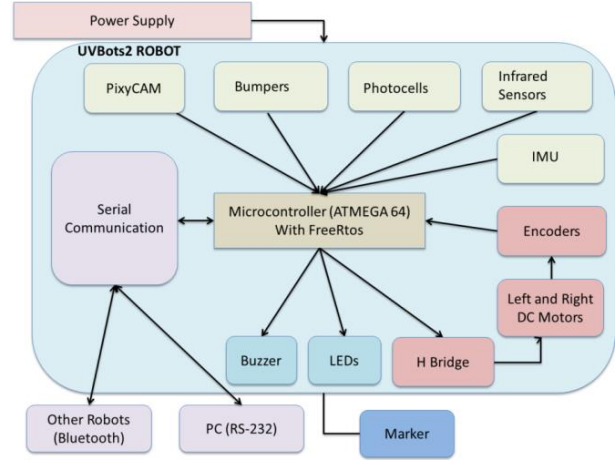
3.2 Motion System

The UVBots2 mobile robots have a differential kinematic model, as shown in Equation 1 considering the robot's space speed.

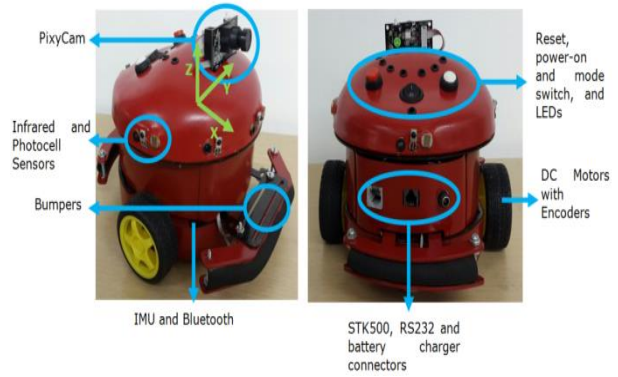
$$\begin{bmatrix} V_R \\ W_R \end{bmatrix} = \begin{bmatrix} c/2 & c/2 \\ c/b & -c/b \end{bmatrix} \begin{bmatrix} W_r \\ W_l \end{bmatrix} \quad (1)$$

Where, V_R and W_R are the lineal and rotational robot velocities measured in cm/s and rads/s respectively, c is the wheel radius in cm, b is the wheel distance in cm, and w_r and w_l are the rotational wheel speeds measured in rads/s. It is well known that the transfer function between the rotational speed of a DC motor and the voltage applied to it is linear [32]. Then, assuming that both DC motor models are equal, Equation 2 shows the DC motor model for the linear velocity of each robot wheel.

$$v_r(s) = \frac{K}{\tau s + 1} u(s) \quad (2)$$



a)



b)

Figure 2. a) Hardware block diagram. b) UVBots2 mobile robot and Placement of the perception system and coordinate frame. **Source.** The authors.

Where, $v_{r/l}$ is the lineal velocity of the right or left wheel, K is the model gain, and τ is the model constant time. Using the standard 2% criteria to identify these parameters [32], the resulting values were $K = 5.81$ and $\tau = 221.2\text{ms}$, which means, a steady-state time of 884.8ms. Hence, to improve the steady-state time, avoid overshooting, and reducing the steady-state error to zero, a PI (Proportional – Integral) controller was designed for each robot wheel. The PI design requirements include: a steady-state time of 750ms, no overshooting, and a natural system frequency of 5.33rad/s. Considering a PI controller design based on zero-pole analysis, the resulting proportional and integral constants are: $K_p = 1.3595$, and $T_i = 0.216$. Consequently, the PI controller transfer function is shown in Equation 3.

$$C(s) = \frac{1.3595 \cdot (1 + 0.2161s)}{0.2161s} \quad (3)$$

3.3 Perception System

As introduced previously, the UVBots2 perception system has several modules: proximity sensing, light intensity measurements, inertial measurement, and vision. Proximity sensing is commonly used to detect obstacles; therefore, UVBots2 uses contact and infrared sensors. The four bumpers are placed as shown in Figure 2b, three at the front, and one at the rear. The infrared proximity sensing is computed by measuring the infrared beam reflected from obstacles. This beam is modulated at 40kHz, since the receiver is a GPIUD28YK Sharp module. A total of six emitter/receiver pairs are placed around the robot as shown in Figure 2b. The raw data provided by UVBots2's proximity sensing is discrete, it measures obstacles presence only.

Sensing environmental variables is important for mobile robots, thus UVBots2 robots are able to measure light intensity around the robot. To do so, a four photocells set is placed as described in Figure 2b. The light intensity around the UVBot2 robot is provided in continuous percentage values. Nowadays, inertial data has become popular in mobile robots and devices. UVBots2 has an IMU of 9DOF; this IMU is a Pololu MiniIMU9v3 containing three different sensors: 3-axis accelerometer, magnetometer, and gyroscope. Figure 2b shows the coordinate frame assumed for this sensor.

Although, UVBots2 robots have a vision sensor, their ATMEGA64 CPUs are not able to process images; consequently, the Pixy Cams CMU5 was used. One of the advantages of using this vision sensor is the compatibility to process an entire image on board while issuing the processing result over several communication channels such as RS232, I2C and SPI. For the terms of this work SPI channel was used to detect up 7 color signatures within the field of view (FOV) of the camera.

3.4 Learner Feedback and Communication System

Human – robot interaction is an important aspect to know what the robot is doing at a given moment. UVBots2 robots have two different types of feedback for learners which are the basic audio systems implemented using a buzzer, and second a set of eight colored LEDs. The buzzer is placed inside the mobile robot, but the set of LEDs are placed on top of the mobile robot as shown in Figure 2b.

The UVBots2 robots are also able to communicate among each other via Bluetooth. To do so, the HC-05

module was configured as master or slave. Bluetooth communication is used to exchange data between robots. However, to perform PC – Robot communication a RS232 link is used to download the program from the PC.

4. FIRMWARE DESCRIPTION

Figure 3a shows the interaction between users, robot firmware, and robot hardware. Users can employ functionalities at three different levels of complexity depending on their knowledge on robotics and programming, namely: basic, intermediate, and advanced. These functionalities interact with the concurrent tasks implemented using FreeRTOS through a data structure. This data structure stores all configuration values corresponding to the entire robot hardware.

Figure 3b shows all the 10 concurrent tasks in charge of operating the mobile robot at three different levels. These concurrent tasks are: the proximity task to deal with infrared and bumper sensors, the mobility task to drive the robot wheels, the perception task to capture measurements from the photocells, the odometry task to estimate the robot relative pose, the IMU task to handle the IMU sensor, the pixyCAM task to deal with the robot camera, the communication task to handle the Bluetooth module, and the main and event tasks which are available for user programming.

The design requirements for the basic, intermediate, and advanced level functionalities were conceived to easily program the mobile robot using a structured language as C. A summary of these functionalities are described as follows:

- **Basic Level.**
 - *Motion:* It includes turn on/off motors, setting speed and direction, displacements by time and velocities, and 90° turns.
 - *Perception:* It comprises basic odometry, activating individual sensors, getting proximity data, getting illumination data and IMU data such as roll-pitch-yaw angles.
 - *Communication:* It includes receiving and transmitting data through Bluetooth, playing sounds, notes and beeps, turning on LEDs or sequence of them.
 - *Pre-designed:* It comprises going to lighter and darker areas, following/avoiding objects using proximity sensors, following lines and walls.

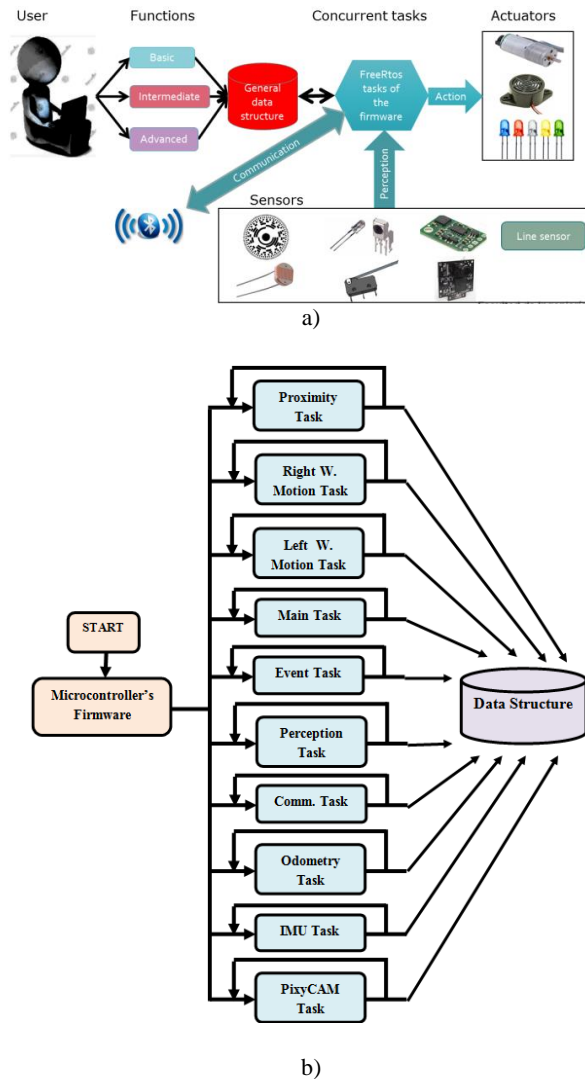


Figure 3. a) User-firmware-robot interaction. b) FreeRTOS firmware tasks. **Source.** The authors.

- **Intermediate Level.**

- *Motion:* It includes displacements by distance, or until some sensor activates, turns by IMU, kicking objects.
- *Perception:* It comprises getting vectors of motion, getting pose filtered by IMU, getting lineal and rotational velocities.
- *Communication:* It includes sending strings, integers, and floats through Bluetooth, pairing with other robots, changing the name of the robot, connecting to a specific robot, changing between master and slave.
- *Pre-designed:* It comprises follow colored objects using the PixyCAM, processing color codes from camera, motion on predefined trajectories, displacements to specific coordinates, following the leader

- **Advanced Level.**

- *Motion:* It includes Setting lineal and angular velocities, getting pose measurements, modifying PI controller parameters, modifying robot heading or displacement.
- *Perception:* It comprises getting vector representation of sensors (proximity, contact and light), sensing obstacles, sensing more/less illumination vectors, getting data from the PixyCAM, heading the robot towards camera objects
- *Pre-designed:* It comprises random heading vectors, behavior programming (follow, avoid, go to, noise, homing, follow walls), coordinating behaviors, setting behaviors thresholds and priorities.

5. TEST AND RESULTS

In order to test the proposed mobile robot set, four experiments were performed (as shown in Figure 4). Test No. 1 shows results obtained when the UVBots2 robot was programmed to draw different geometric figures; the programs for these figures were implemented using functionalities for basic learners. Test No. 2 presents the obtained results when the UVBots2 robot tries to follow an object placed in front of it using the PixyCam; the latter was programmed using functionalities for intermediate learners. As for test No. 3, it describes the obtained outcome from the mobile robot when using the behavior-based programming [33] to reach the most illuminated zone of the environment, while avoiding obstacles. This test was implemented using advanced learners functionalities. Finally, experiment No. 4 shows a beta test performed with real users. The aim of these tests is to show the main UVBots2 functionalities at different levels of learner's knowledge.

Test No. 1 – Geometric Figures: the aim of this test is to program the UVBot2 using basic functionalities to perform sequential steps in order to draw geometric figures. Figure 4a shows the resulting square and a house construction as an additional result to be seen in Figure 4b. Due to image resolution, the marker trace cannot be observed. For this reason, Matlab-based software was developed to process the captured video while the test was taking place. The circles in Figure 5a show the UVBot2 robot's trajectory.

Test No. 2 – Camera based Object Following: in this test, the intermediate level functionalities were used to show how an object can be followed by the UVBot2 robot using the PixyCam. Here, the program that performs this task has a more complicated syntax in comparison to the basic level functionalities. Figure 4c and 4d show two different circle-like trajectories performed by the

UVBot2 robot while following the ball. Here, circles show the robot trajectory, and the squares show the ball trajectory. Basically, the UVBot2 robot firmware

captures the ball coordinates with respect to the image principal point. The former is used by the *seguirObjCam()* procedure to drive the mobile robot.

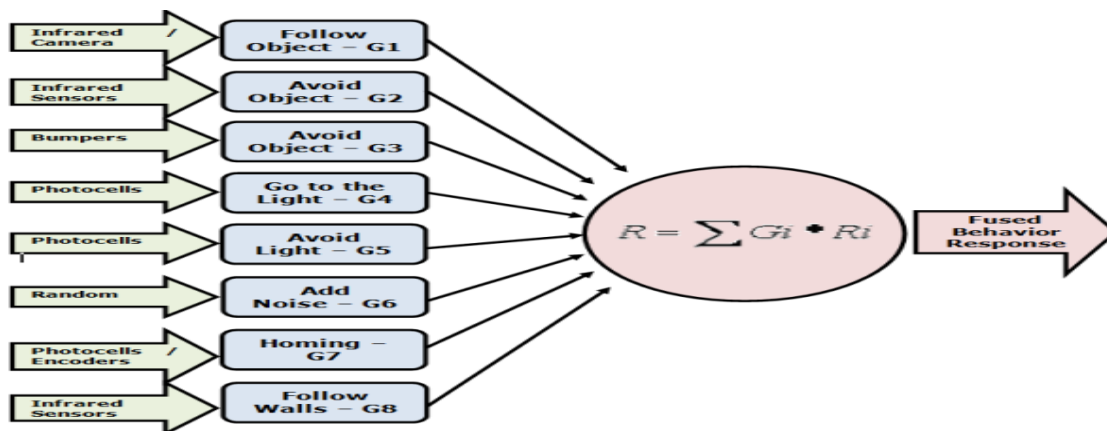
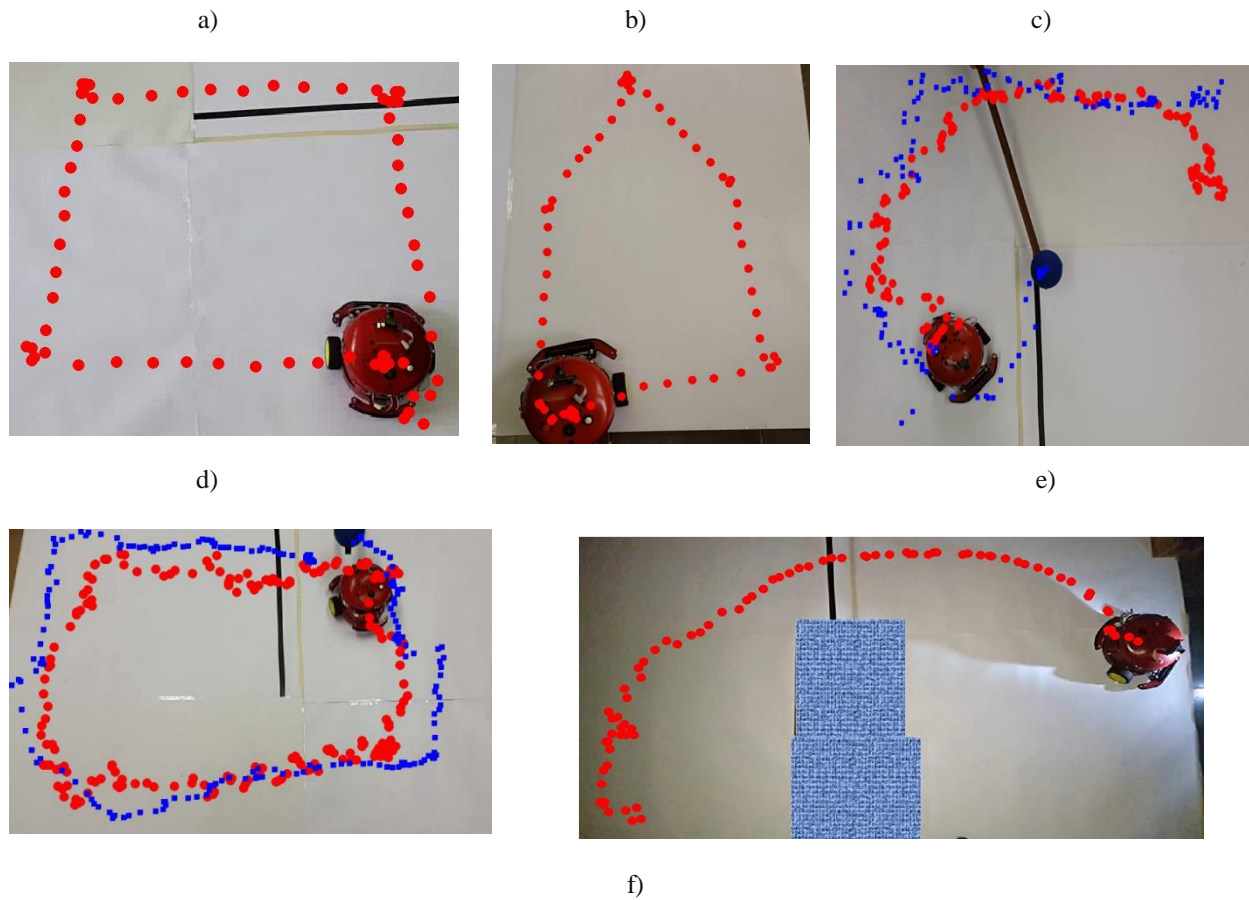


Figure 4. Drawing geometric figures: a) square and b) house. The mobile robot is able to follow a blue object using the PixyCam c) and d). e) Behaviors stack available into the robot firmware. f) Homing and obstacle avoidance behaviors. **Source.** The authors.

Test No. 3 – Light based Homing with Obstacle Avoidance: in order to test the advanced programming functionalities, a behavior-based application was implemented. Behavior based programming [33] is a control architecture used to handle this problem. The robot homing behavior is very important in robotics, it allows mobile robots returning to recharging stations, and going to specific targets in the environment. Here, the robot target is the more illuminated region in Figure 4f, which is placed at the right. However, avoiding obstacles is important while going to goal; in this test, two boxes were placed between the starting robot position and its goal. Then, two simultaneous tasks must be coordinated in order to perform this task: going to the goal, and avoid obstacles. To do so, in Figure 4e is shown the set of behaviors supported by the UVBots2 robots. All the motion command behaviors are coordinated using a weighted sum. Each weight represents the behavior importance into the behavior stack. In this test, the *Avoid Object* (using the infrared sensors), and the *Go to the Light* (using the photocells) behaviors were employed using weights of 0.7 and 0.3 respectively. Figure 4f also shows the resulting trajectory of the UVBot2. Initially, the mobile robot was moving to the light source. Although the boxes were in the way, the robot avoided them to reach the most illuminated region of the environment.

Test No. 4 – Experiment with Real Users: it was performed by 10 students of Electronic Engineering of the Universidad del Valle. They did not know any information about the robot and programming environment, but they have experience with microcontrollers and C programming. Two challenges were performed: to move the robot in a square Archimedes spiral, and to track a red ball using the PixyCam. At the end of the test, a survey took place, the questions and resulting statistics are shown as follows:

1. How easy is to understand the source code comments? 80% answered “Easy”, 20% answered “Very easy”.
2. In general, is the loading process on the robot slow? 90% answered “No”, and 10% “Yes”.
3. Was it easy to add source code to your application? 70% answered “Easy”, 20% answered “Very easy”, and 10% answered “Difficult”.
4. Was it easy to edit the user tasks in the source code? 90% answered “Yes”, and 10% answered “No”
5. Was it easy to differentiate the main and sensors tasks? 80% answered “Yes”, and 20% answered “No”

6. Was it easy to understand the structure of the source code? 90% answered “Yes”, and 10% answered “No”.

7. What do you think about using FreeRTOS in the C program? 70% answered “Good”, and 30% answered “Very good”.

6. CONCLUSIONS

In this work, the UVBots2 platform was presented; it includes a set of eight mobile robots supporting three levels of knowledge on programming concepts and robotics. At each level of knowledge (basic, intermediate, and advanced), a set of functionalities were developed to handle all robot sensors and actuators. As for the mobile robots equipment each one has been set up with six proximity sensors, four bumpers, four photocells, differential drive with encoders, one 9DOF IMU, Bluetooth communication, one buzzer, eight colored LEDs and one PixyCam camera. The UVBots2 platform is an important experimentation artifact, since it allows executing all learner programs ranging from basic programming to advanced robotics projects. These projects are hands-on tools for learning STEM concepts. The firmware for the mobile robots was implemented using the FreeRTOS real-time kernel in order to concurrently execute all the perception, communication, motion, feedback, and user tasks. Nowadays, there are European [34] and North American [35] governmental efforts to improve, and develop novel educational strategies at different levels of the education process. Considering the previous experiences on the Robotics seedbed [6], and the hands-on platform presented in this work, both initiatives are an interesting option to bring children and young people to science, technology, engineering, and math.

7. ACKNOWLEDGEMENTS

This work was founded by the research project “Desarrollo de una Plataforma para la Enseñanza Interactiva de Cursos Básicos de Programación Usando Robots Móviles Programables” granted by Universidad del Valle with identification number (ID) CI 2808.

8. REFERENCES

- [1] F. B. V. Benitti, “Exploring the educational potential of robotics in schools: A systematic review,” *Comput. Educ.*, vol. 58, no. 3, pp. 978–988, Apr. 2012.
- [2] M. E. Argentina, “Plan Estrategico de Ingenieria 2012-2016,” *Ministerio de Educación - Argentina*, 2014.

- [Online]. Available: <http://pefi.siu.edu.ar/>. [Accessed: 13-Feb-2014].
- [3] M. de Educación, “SPADIES - ...:Ministerio de Educación Nacional de Colombia:..,” 2014. [Online]. 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, no. 1, pp. 61–66, 2013.
- [5] L. Major, T. Kyriacou, and O. P. Brereton, “Systematic literature review: teaching novices programming using robots,” *15th Annu. Conf. Eval. Assess. Softw. Eng. (EASE 2011)*, pp. 21–30, 2011.
- [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, no. 1, pp. 88–95, 2010.
- [7] C. K-Team, “K-Team Corporation, Mobile Robotics,” 2016. [Online]. Available: <http://www.k-team.com/>. [Accessed: 01-Jan-2016].
- [8] A. MobileRobots, “Pioneer Robots,” 2013. [Online]. Available: <http://www.mobilerobots.com/ResearchRobots/>. [Accessed: 01-Jan-2014].
- [9] Ir. Corporation, “iCreate Robot,” 2013. [Online]. Available: <http://www.irobot.com/us/learn/Educators/Create.aspx>. [Accessed: 01-Jan-2014].
- [10] A. Corporation, “Pob-Bot Robot,” 2013. [Online]. Available: <http://education.awabot.com/>. [Accessed: 01-Jan-2014].
- [11] Arduino, “Arduino Robot,” *Arduino*, 2013. [Online]. Available: <http://arduino.cc/en/Main/Robot>. [Accessed: 01-Jan-2014].
- [12] Aseba, “Thymio II Robot,” *Aseba*, 2013. [Online]. Available: <https://aseba.wikidot.com/en:thymio>. [Accessed: 01-Jan-2014].
- [13] Festo, “Robotino,” 2013. [Online]. Available: http://www.festo.com/cms/en_corp/11367.htm. [Accessed: 01-Jan-2014].
- [14] BirdBrain-Technologies, “The Finch Robot,” 2016. [Online]. Available: <http://www.finchrobot.com/>. [Accessed: 01-Jan-2016].
- [15] Pololu-Corporation, “Pololu M3pi Robot,” 2016. .
- [16] MakeBlock, “mBot Educational STEM Robot,” 2016. [Online]. Available: <http://www.makeblock.cc/mbot/>. [Accessed: 01-Jan-2016].
- [17] Meet-Edison, “Meet Edison - Cheap Programmable Lego,” 2016. [Online]. Available: <http://meetedison.com/>. [Accessed: 01-Jan-2016].
- [18] Parallax, “Scribbler Robot,” 2014. [Online]. Available: <http://www.parallax.com/product/28136>. [Accessed: 01-Jan-2014].
- [19] Open-Source-Robotics-Foundation, “TurtleBot,” 2016. [Online]. Available: <http://www.turtlebot.com/>. [Accessed: 01-Jan-2016].
- [20] M. Boanani, T. Baabura, P. Retornaz, F. Vaussard, S. Magnenat, D. Burnier, V. Longchamp, and F. Mondada, “Miniature Mobile Robots Group - Mobots,” 2016. [Online]. Available: <http://mobots.epfl.ch/marxbot.html>. [Accessed: 01-Jan-2016].
- [21] F. Gómez, F. Muñoz, B. E. Florián, C. A. Giraldo, and E. B. Bacca-cortes, “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, no. 2, pp. 53–74, 2008.
- [22] C. Giraldo, B. Florian, B. Bacca, F. Gómez, and F. Muñoz, “A programming environment having three levels of complexity for mobile robotics,” *Ing. e Investig.*, vol. 32, no. 3, pp. 76–82, 2012.
- [23] D. Salas, I. Guarín, and R. Llamasa, “ISO SPICE EN SISTEMAS HIPERMEDIA EDUCATIVA,” *Rev. UIS Ing.*, vol. 2, no. 1, pp. 63–72, 2003.
- [24] H. Gonzáles and C. Mejía, “ESTUDIO COMPARATIVO DE TRES TÉCNICAS DE NAVEGACIÓN PARA ROBOTS MÓVILES,” *Rev. UIS Ing.*, vol. 6, no. 1, pp. 77–84, 2007.

[25] M. Suell, J. Archila, and O. Lengerke, "DISEÑO MECATRÓNICO DE UN ROBOT TIPO AGV 'AUTOMATED GUIDED VEHICLE,'" *Rev. UIS Ing.*, vol. 7, no. 1, pp. 65–76, 2008.

[26] LEGO-Company, "Robolab on-line WEB site," 2016. [Online]. Available: <http://www.robolabonline.com/home>. [Accessed: 01-Jan-2016].

[27] Awabot, "Awabot Education," 2016. [Online]. Available: <http://www.pob-tech.com/>. [Accessed: 01-Jan-2016].

[28] MIT-Media-Lab, "Scratch - Imagine, Programming, Share," 2016. .

[29] Code.org, "CODE - Anybody can learn," 2016. [Online]. Available: <https://code.org/>. [Accessed: 01-Jan-2016].

[30] MIT, MIT-Media-Lab, and MIT-CSAIL, "MIT App Inventor," 2016. [Online]. Available: <http://appinventor.mit.edu/explore/about-us.html>. [Accessed: 01-Jan-2016].

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

[32] K. Ogata, *Sistemas de control en tiempo discreto*. Prentice Hall, 2009.

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

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

[35] USA-White-House, "Computer Science for All - Whitehouse," 2016. [Online]. Available: <https://www.whitehouse.gov/blog/2016/01/30/computer-science-all>. [Accessed: 01-Jan-2016].