

Robobo: The Next Generation of Educational Robot

Francisco Bellas¹(✉), Martin Naya¹, Gervasio Varela², Luis Llamas², Moises Bautista¹, Abraham Prieto¹, and Richard J. Duro¹

¹ Integrated Group for Engineering Research, Universidade da Coruña, A Coruña, Spain
{francisco.bellas, martin.naya, m.bautista, abprieto, richard}@udc.es

² Mytech, A Coruña, Spain
{gervasio.varela, luis.llamas}@mytechia.com

Abstract. This paper presents Robobo in the context of higher education. Robobo is a low-cost educational mobile robot that combines a simple wheeled base with a smartphone, which provides the latest technology to the robot. With Robobo, students can develop their own projects through ROS using cameras, microphones or high-resolution screens, bringing teaching closer to the real requirements of the market they will find when they finish their studies. In this work, the hardware and software development that has been carried out is described in detail. Furthermore, it is presented an exemplifying case of student project that shows the potentiality of Robobo in this context.

Keywords: Educational robots · ROS · Human-robot interaction

1 Introduction

University degrees in computer science and engineering have been using mobile robots in different subjects from a long time [1–3]. They are a basic tool when teaching autonomous robotics but, in addition, they have shown to be very useful in other subjects like computer vision [4] or control systems. Up to now, mainly due to economical limitations, these educational robots have been quite simple in technological terms. Thus, the most popular educational robot in higher education is the Lego Mindstorms [5], a modular robot equipped with an Arm926ej-S Core@300 MHz processor in its last version (EV3). This robot is equipped with a sonar or infrared sensor, a gyroscope and a color sensor, and it has an average price of 350–400€. A very popular and small mobile robot in European Universities is the Thymio II [6], suitable for multi-robot teaching. It only costs around 120€ and it is equipped with infrared sensors, a temperature sensor and an accelerometer. It has a low performance PIC24@32 MHz microcontroller. As a consequence of this simplicity, the practical use of real robots in higher education has been limited to simple control mechanisms and laboratory applications.

This issue has not been a real problem because robotics was not a real market in industry, so robots were used in classes as simple prototypes, without a rigorous translation to reality. In fact, the use of robotic simulators has been very popular. But, as it is known, the current and near future situation is that robotics will be a key market for

engineers and computer scientists [7]. As a consequence, the robots used in University classrooms must be updated to include all the features of the real robots that will be present in our reality, both in industrial and domestic domains. This implies that robotic classes must change simple platforms or simulated robots for realistic ones. But if one moves to robots with more powerful sensors, the price increases. For example, the e-puck robot [8] is a mobile robot that was very popular in the last decade in several European Universities. It has infrared sensors too and a dsPIC30F6014A@64 MHz controller, and it includes a low-resolution camera of 640×480 pixels. The average price of one unit is 600€. A more powerful robot used in higher education is the Kephra IV [9], which uses a 752×480 resolution camera, an ARM Cortex 800 MHz processor, infrared and sonar sensors, but with a price of around 2500€. Considering high-end mobile robots like the NAO [10], with an Atom Z530 1.6 Ghz CPU, two 1280×960 cameras, sonar and tactile sensors, the price becomes even higher (around 6000€), which is unreachable for most higher education institutions when considering the large number of robots that intensive teaching of these topics would imply (no more than two or three students per robot, and preferably only one). The most remarkable platform in terms of cost-features balance is the TurtleBot [11]. It is a low-cost (520€) and open-source mobile robot, which is very popular for ROS (Robot Operating System) users as a testing platform, mainly for SLAM (Simultaneous Localization and Mapping) research, because it contains a 360° LIDAR system in the latest version. Its application to a wider range of robotic domains, as human-robot interaction, using speech recognition, cameras or tactile interfaces, is limited.

Consequently, a few years ago, the authors of this paper decided to start the development of low-cost educational mobile robot that can incorporate state-of-the-art technology to be used in engineering and computer science degrees. This robot has been called Robobo [12], and it combines a simple wheeled base with a smartphone. With Robobo, students can develop engineering projects using cameras, microphones or high-resolution screens, bringing teaching closer to the real market they will find when they finish their studies. This paper is focused on the presentation of the current version of the Robobo hardware and software, which is explained in Sect. 2. Furthermore, its potentiality in higher education will be illustrated through an exemplifying case of student project, in Sect. 3. This project exploits the Human-robot Interaction (HRI) features that are provided by the smartphone based on its visual, auditory and tactile capabilities.

2 The Robobo Robot

The Robobo hardware is made up of two elements: a mobile wheeled base and a smartphone that is attached to the base, as shown in Fig. 1. It must be pointed out that the Robobo robot is not only the mobile base, but the combination of base and smartphone. This configuration provides the following features:

- *Low-cost*: The educational institution must acquire only the mobile base, because the smartphone can be provided by the student. Our experience in this aspect is that students perceive this extra application of their own smartphone as positive. Robobo

will be commercialized in classroom packs with a reference cost per unit that will be lower than the most popular educational robots, TurtleBot 3 or LEGO EV3.

- *Continuous update:* The smartphone technology, both in hardware and software, is in continuous update, so Robobo can be easily improved by simply upgrading the smartphone. It is well-known that students usually have the latest models, so new hardware features can be easily incorporated in classes. The Robobo base will support new smartphone features by simply updating its firmware and software.
- *Long-term duration:* Related to the previous two features, it can be concluded that the Robobo validity is longer than other educational robots, and the investment in the mobile base lasts longer.
- *State-of-the-art technology:* Apart from the sensors and actuators that are in the mobile base and that will be explained later in detail, students can use all the hardware elements that are included in typical current smartphones:
 - 2 high-resolution cameras
 - Proximity, light and temperature sensors
 - Gyroscope, accelerometer and magnetometer
 - GPS
 - Microphone
 - High-resolution LCD screen
 - Connectivity: 4G, WI-FI, USB, ...
 - Speaker
 -



Fig. 1. The Robobo robot with the smartphone attached (left). The mobile base with a 3D printed pusher (right)

In addition to the Robobo hardware, a software architecture that allows to program the Robobo through ROS (Robot Operating System) [13] has been developed. In the

following sections, the Robobo hardware and software design and implementation will be described.

2.1 Robobo Hardware

The Robobo hardware is composed by two elements: the smartphone and the mobile base. Any commercial smartphone with the following requirements is supported:

- Operating system: Android 5.1 or higher (iOS version is under development)
- Display size: minimum 4.7 inches and maximum 5.5 inches
- Thickness: 95 mm maximum

The main elements of the base hardware are displayed in Fig. 2. The mobile base has two *driving motors* (MT1, MT2) which are in charge of performing the displacement of the platform. They are 150:1 gear motors with encoder plus a pinion-crown design. This extra ratio increases the maximum available torque and the robot's acceleration. At the back of the base, there are two caster balls which allow an omnidirectional movement. Furthermore, the base is equipped with two additional motors that allow an independent movement of the base and the smartphone:

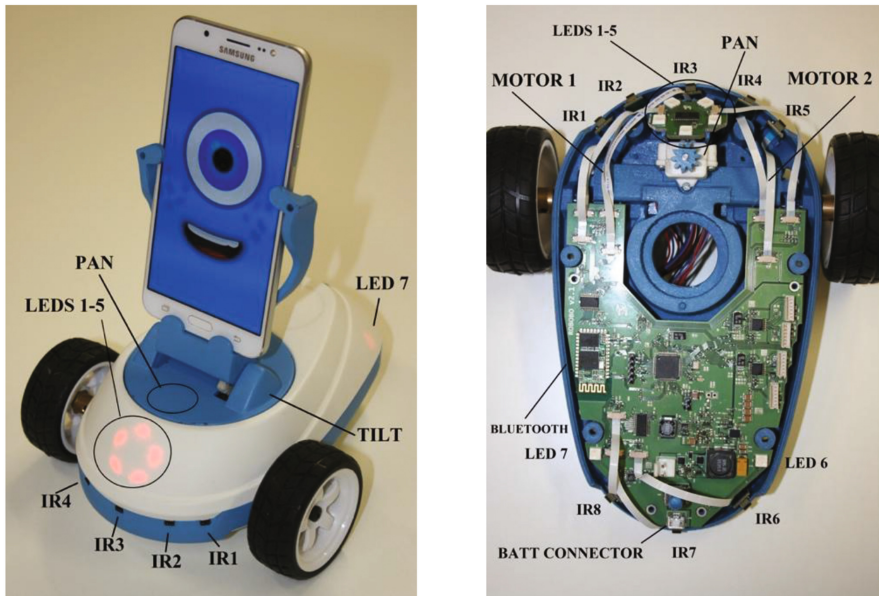


Fig. 2. Main hardware elements of the Robobo base

Pan motor (PAN): It is a 150:1 gear motor with encoder with pinion-crown reduction. This reduction provides the possibility of performing soft and precise movements, with a 4.86° minimum angular displacement.

Tilt motor (TILT): It is a 1000:1 gear motor with encoder. This is a high ratio for two reasons:

- 1000:1 reduction implies that the smartphone is self-supported. No current is needed to maintain the smartphone in the desired position.
- This ratio gives us the required torque to move the smartphone in the worst-case conditions. The gear motor gives a maximum of 44 Ncm, which is higher than the 13 Ncm in the worst considered case obtained in our tests.

These two motors provide Robobo with a many different movements, and with a richer type of HRI possibilities.

A specific Printed Circuit Board (PCB) has been designed for Robobo in order to maximize the performance and capabilities of the base, and to adapt the electronics to the external design. These are the main electronic components:

Microcontroller: The electronic structure of the Robobo is built around a PIC32 microcontroller. The CPU runs at 80 MHz, being enough to control the sensors, actuators and carry out miscellaneous tasks that the PIC must perform. It is in charge of carrying out simple calculations and operations, such as motor-encoder adjustments or analog battery readings, leaving the complex and high cost computing tasks to the smartphone.

Sensors: Robobo uses 8 VCNL4040 Vishay I2C infrared sensors (IR1 to IR8) for three purposes, which depend on their orientation:

- *Object detection (IR1, IR3, IR5, IR7):* They are placed in the front and rear part of the base oriented parallel to the ground. They are used for obstacle detection and avoidance in cluttered environments. Furthermore, they can perform a soft object approximation thanks to their gradual detection range.
- *Fall avoidance (IR2, IR4, IR6, IR8):* Placed in the front and rear part of base, they are placed at a 45° degree orientation towards the ground, with the aim of detecting holes and avoiding falls when on tables or stairs.
- *Ambient light sensing:* The VCNL4040 can also be used as an ambient light sensor.

Actuators: As commented above, 4 DC-DC motors are used to move and control the base. The 4 motors have a 6-pole magnetic encoder shaft to precisely measure the motor revolutions. In addition, they are controlled by two motor drivers that are limited to a maximum of 1.3A each. The motor drivers are controlled by a PWM signal from the PIC microcontroller.

RGB LEDs: There are 7 RGB LEDs (*L1 to L7* in Fig. 2). Five of them are placed in the front part of the base and two at the rear. The LED driver has a 4096 bit resolution per color, which provides a high color resolution. The LEDs have a user interface functionality, showing information to the user, such as the battery status or infrared obstacle detection.

Bluetooth module: A Microchip BM78 Bluetooth module is used to establish communication between the base and the smartphone. This module was selected for being a simple, low cost, CE and FCC certified Bluetooth module. It uses the 4.2 version, which supports the SPP (Serial Port Profile) for serial communication with the microcontroller.

Power system: A 2500 mAh and 3.7 V LiPo battery is mounted in the base. This battery was selected for its high charge density, in order to provide a high level of autonomy in the minimum possible space. The voltage range of these batteries is 4.2 V–3.0 V. Two integrated circuits (IC) complete the rest of the power system:

- A *Buck-Boost* converter to obtain a fixed 3.3 V used by the microcontroller, Bluetooth, LEDs, etc. This IC is needed due to the variable voltage range of the battery.
- A *Step-up* converter, to boost the battery voltage to the motor’s 7 V.

Furthermore, there is a battery manager in the circuit board for recharging the battery through a BATT USB connector (see Fig. 2).

2.2 Robobo Software

The Robobo software development can be organized into low-level (firmware) and high-level (software architecture) blocks:

Low-level: Figure 3 shows the general hardware architecture of the mobile base, where only the main connection paths are represented to clarify the scheme. The core of the system is the PIC32 microcontroller, which is in charge of controlling the different integrated circuits, sensors and actuators on the base. Furthermore, the PIC32 manages the communications between the microcontroller and the Bluetooth module, which sends information to the smartphone and receives motor commands from it.

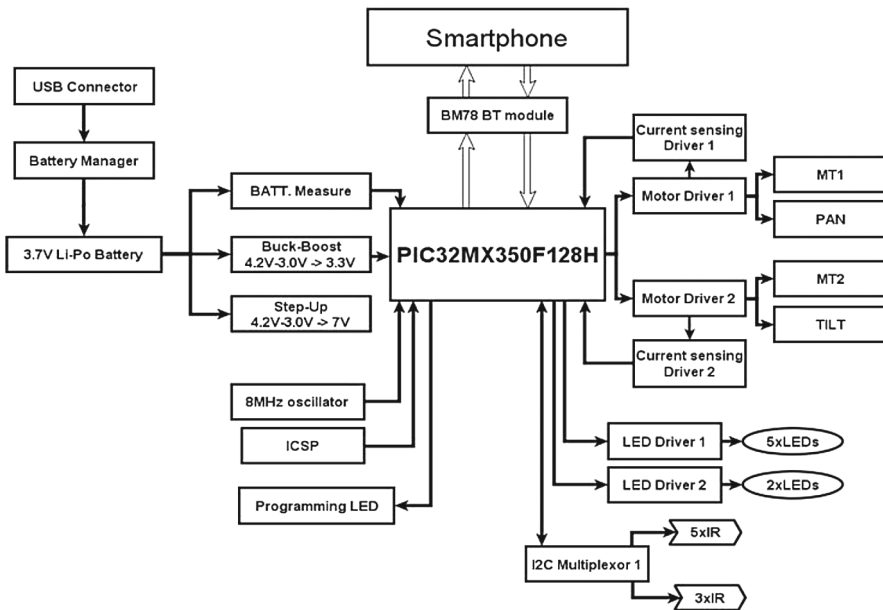


Fig. 3. Robobo electronic architecture

High-level: The main requirement of the Robobo software architecture was to support the programming of the robot using different programming paradigms like ROS, Java or Scratch. A completely modular software architecture, shown in Fig. 4, has been designed. It is based around the concept of Robobo module and a very lightweight library, called the Robobo Framework, which provides the essential mechanisms to manage and run those modules on an Android smartphone. On top of this library, two different kinds of modules can be loaded:

- Functionality modules (the orange ones in Fig. 4): Implemented in Java using the Android standard API, they provide different functionalities to the robot, like speech recognition, face detection, environment sensing or physical movement. Furthermore, these modules complement the native API of the robot, which can be directly used for programming it using Java for Android, thus creating Robobo Apps that can be run in any Android smartphone.
- A series of proxy modules (ros-proxy and remote-proxy modules in Fig. 4), also implemented in Java, which provide particular interfaces to other programming environments. These interfaces provide a translation between an external API or protocol, and the native Robobo API in Java.

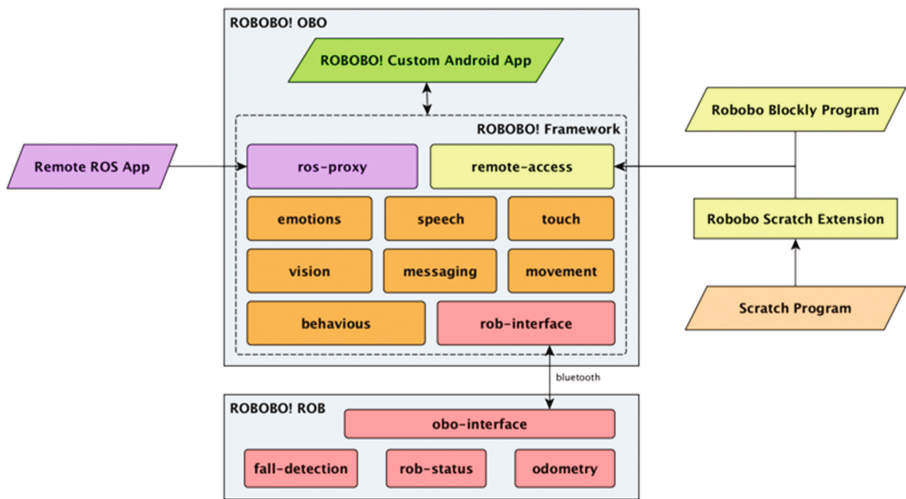


Fig. 4. Block diagram of the Robobo software architecture

Robobo comes by default with a particular set of these modules but, as they are completely decoupled between them and with respect to the framework, advanced users can customize the set of modules, and even implement new modules to support new robot functionalities, or even new programming paradigms through proxy modules.

Finally, it is important to note that there exists a module for connecting the Robobo framework and its modules to the Robobo robotic base. The rob-interface module, shown in pink in Fig. 4, implements the Bluetooth communications protocol of the base and provides a control API for other modules to use.

Java programming is directly supported by the native API provided by the different modules. Using the Robobo framework users can create custom Android Apps that control the behavior of the robot. These apps use the programming interfaces of the Robobo modules to access the different functionalities of the robot and build new robot behaviors or solve new tasks.

For block programming, two different approaches are currently supported: Scratch, and our own block-based IDE that uses Blockly. As can be seen in Fig. 4, both approaches are connected to the framework using the same proxy, the Robobo Remote Access Protocol (RRAP). This is a simple JSON based protocol that allows remote access to the Robobo modules' APIs.

Finally, Robobo can also be programmed using the Robot Operating System (ROS), which is the main programming language for teaching robotics at higher education levels. ROS is supported by a ros-proxy module that translates between the native APIs of the modules and ROS messages and topics.

3 Example of Student Project

To illustrate the type of project that can be carried out using Robobo in computer science or engineering degrees, this section describes a specific example that was proposed to students in the "Robotics" subject of the Computer Science degree, at the University of Coruna, during the 2016/17 academic year. The project was focused on HRI, specifically, on programming the Robobo to act as a pet. That is, the robot must ask the user for food, attention and "affection". The way it is performed must be selected by the students, giving them the opportunity of creating highly realistic solutions.

The functionality modules required to solve this project use all the interactive modalities of Robobo:

Speech: The speech capability of the robot was deployed using the Android libraries and a *speechProducer* module was implemented.

Sound recognition: *SpeechRecognition* and *soundRecognition* were implemented using the Pocketsphinx [14] and TarsosDSP [15] libraries.

Vision: A *faceRecognition* module was implemented using the Android libraries, and a *colorDetection* module using the OpenCV library [16].

Tactile sense: The *touchDetection* module performs the recognition of tactile gestures over the smartphone screen. It allows a *tap* (quick and simple touch on the screen), a *touch* (sustained touch over the screen), a *caress* (slide over the screen) and *flings* (end a slide in a fast way). The Android libraries were used to deploy this module.

Visual emotion: The *emotionProducer* module allows to display different images or animations in the smartphone screen, giving a powerful interaction capability to users to show the robot emotions.

Movements: The *motorCommand* module allows the user to move the four motors of the Robobo base.

With these functionality modules and the programming language selected by students (and supported by ROS), the specifications of the pet behavior were the following:

- Basic instincts: Robobo must store hunger and thirst levels and, when they are below a threshold, it must say “I am hungry” or “I am thirsty” until they are refilled.
- Movement: Robobo must move in order to capture user attention if no event has been executed for a predefined time-lapse. For instance, it can spin or emit a sound.
- Touch: Two different behaviors must be implemented depending on the type of screen touch modality:
 - Tap: If the user touches the screen with a single tap, Robobo must react differently depending on the part of the “face” that is touched. If it is in the “eye” or in the “mouth”, it will move the tilt motor backwards and show an angry expression (see Fig. 5a and b for examples). If it is in any other point, it will show a laughing face and emit a sound saying “tickles” (see Fig. 5c and d for examples)

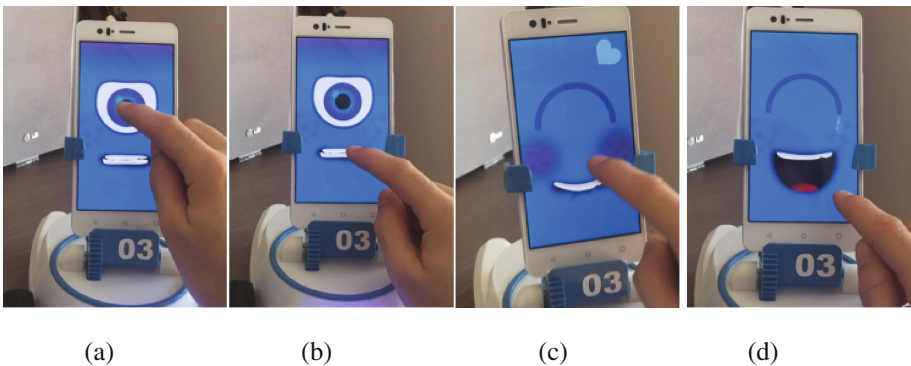


Fig. 5. Expressions of the touching behavior of the Robobo pet

- Flip: If the user slides a finger over the screen, the pan-tilt unit moves accordingly. The possible slide directions must be discretized into four pan-tilt movements: tilt backwards or forwards and tilt rightwards or leftwards.
- Voice: Robobo must react to the following predefined phrases:
 - “Here comes the food”: The robot prepares to receive food
 - “Here comes the drink”: The robot prepares to receive drink
 - “Hello”: The robot answers by saying “Hello”
 - “How are you?”: The robot will respond by saying its thirst and hunger levels
- Vision:
 - Color: Robobo must detect 2 different colors, green for food and blue for drink, but only after the corresponding voice command has been detected. In both cases, it must say whether the color is correct or not. For instance, if the user says “Here comes the drink”, the robot must look for a blue color area of a predefined size in its field of view (the left image of Fig. 6 shows an example), and after a time-lapse, it must say “Thank you for the drink” or “I don’t see the drink”.

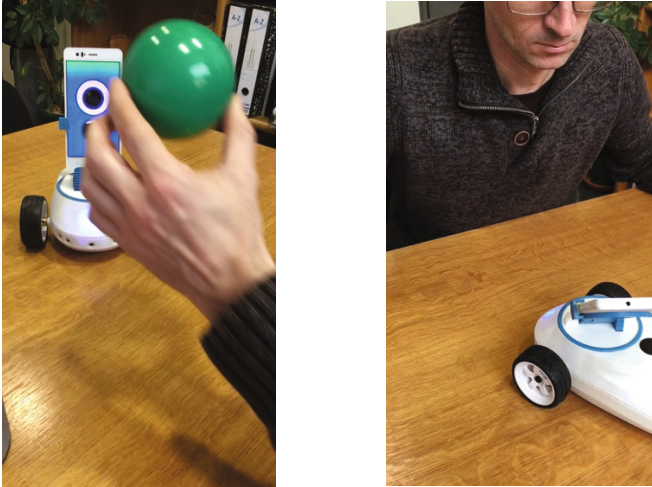


Fig. 6. Green ball that represents “food” (left) and user face recognition (right)

- Face: Robobo must detect the user face. If it is below a threshold, the robot will move backwards (Fig. 6 right)

This example provided an engaging experience for the students. As shown, it addresses many topics within robotics and programming while, at the same time, provides a fun way for the students to get acquainted with them. In fact, many of the functionalities can be initially programmed without using the base, and the students would often do this at home playing around with their mobile phone.

Summarizing, even though this example is simple and it is an initial test, it is easy to see that there is a lot of potential in using this type of simple robots with high end functionalities (sensing and actuation structures provided by the combination of the base and the smartphone). It is our aim to progressively introduce this approach in more courses during the next year.

4 Conclusions

Traditional robots used in higher education must be updated in order to prepare engineering and computer science students to the real market they will face in the near future. In this line, a low-cost educational robot called Robobo has been designed, developed and tested, which is made up of a simple mobile base that has a smartphone attached to it. With this configuration, state-of-the-art technology can be used by students in classes now and in the future, because Robobo can be easily updated only by upgrading the smartphone model. The robot will be available in November 2017 through the Robobo Project web page: <http://en.theroboboproject.com>.

References

1. Schilling, K., Rösch, O.: Mobile mini-robots for engineering education. *Global J. Eng. Educ.* **6**(1), 79–84 (2002). IGI
2. Fagin, B., Merkle, L.: Quantitative analysis of the effects of robots on introductory Computer Science education. *J. Educ. Resour. Comput.* **4**, 1 (2002). Article 2
3. Williams, A.: The qualitative impact of using LEGO MINDSTORMS robots to teach computer engineering. *IEEE Trans. Educ.* **46**(1), 206 (2003)
4. Sankowski, D., Nowakowski, J.: *Computer Vision in Robotics and Industrial Applications*. World Scientific, Singapore (2014)
5. Danahy, E., Wang, E., Brockman, J., Carberry, A., Shapiro, B., Rogers, C.B.: Lego-based robotics in higher education: 15 years of student creativity. *Int. J. Adv. Rob. Syst.* **11**(2), 27 (2014)
6. Mondada, F., et al.: Bringing robotics to formal education: the Thymio open-source hardware robot. *IEEE Robot. Autom. Mag.* **24**(1), 77–85 (2017)
7. Frey, C., Osborne, M.: The future of employment: how susceptible are jobs to computerisation? *Technol. Forecast. Soc. Chang.* **114**, 254–280 (2017)
8. Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotocz, A., Magnenat, S., Zufferey, J., Floreano, D., Martinoli, A.: The e-puck, a robot designed for education in engineering. In: *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions* (2009)
9. Soares, J., Navarro, I., Martinoli, A.: The Khepera IV mobile robot: performance evaluation, sensory data and software toolbox. In: *Proceedings of Robot 2015: Second Iberian Robotics Conference*, pp. 767–781 (2016)
10. Shamsuddin, S. et al.: Initial response of autistic children in human-robot interaction therapy with humanoid robot NAO. In: *Proceedings of the 2012 IEEE 8th International Colloquium on Signal Processing and its Applications*, pp. 188–193 (2012)
11. TurtleBot web page. <http://www.turtlebot.com>
12. The Robobo Project web page. <http://en.theroboboproject.com>
13. ROS web page. <http://www.ros.org>
14. Pocketsphinx web page. <https://github.com/cmusphinx/pocketsphinx>
15. TarsosDSP web page. <https://github.com/JorenSix/TarsosDSP>
16. OpenCV web page. <http://opencv.org>