Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Low Cost Platform for Automatic Control Education Based on Open Hardware. ⋆

A. Soriano, L. Marín, M. Vallés, A. Valera, P. Albertos

Department of Systems Engineering and Control, Institute of Control Systems and Industrial Computing, Universidad Politécnica de Valencia, Camino de Vera, E-46022 Valencia, Spain
e-mail: {ansovi, leomarpa, mvalles, giuprog, pedro}@ai2.upv.es.

Abstract: Automatic control, as well as the introduction to programming, are subjects increasingly being taught to young engineering students. For this reason the availability of a suitable platform for the laboratory work becomes a critical decision, in order to promote students' motivation to experience the theoretical concepts studied in the classroom. Until recently, the only option to perform laboratory work in this kind of subjects was to use closed platforms commercially available but today, thanks to advances in technology, many open hardware options supporting standard and intuitive programming languages are available. In order to provide a multidisciplinary low cost open platform for the automatic control introductory courses, an example of open-source educational platform easily programmable composed by a differential mobile robot with a robotic arm manipulator is presented in this paper.

1. INTRODUCTION

Teaching in control and automation and particularly the related to mobile robotics, is becoming an essential subject (Weinberg and Yu, 2003) in different courses taught both in middle and high school as well as in vocational education and university. In these courses, the use of an experimental platform is fundamental in the learning process (Greenwald and Kopena, 2003) as it allows a visual demonstration of the basic concepts maintaining the students' interest and motivation.

Until recently, the existing educational material was usually closed hardware and remarkably expensive. Each commercial platform was often compatible exclusively with its own hardware and software, typically away from standardizations preventing the use of any other protocol, connector to link devices or programming language different from their own products. But nowadays the open-source hardware philosophy is booming, offering many options in open platforms and new low cost products. Working with open hardware offers users the ability to add or modify hardware devices at will, making the system more flexible, scalable and totally customizable. In the case of mobile robots, there is a tending towards open hardware products (Fisher and Gould, 2012) of low cost, easy implementation and standardization. All of these aspects provide great benefits in the research and education communities.

Another critical aspect in choosing an experimental platform is the programming environment that should be simple, with a fast learning curve and powerful enough to allow the use of all the sensors, actuators and devices (communication, identification, etc.) of the platform. For this reason, the experimental platform can also be used in programming education if the programming language is intuitive enough. For example, visual languages based on a flowchart or block diagrams such as *Scratch* (developed by MIT, Resnick et al. (2009)), *Blockly* (by Google, Fraser (2013)), *Alice* (by Carnegie Mellon University, Cooper et al. (2000)), *Greenfoot* (by the University of Kent, Henriksen and Kolling (2004)), *Simulink* (by MathWorks, (2013)) and *Labview* (by National Instruments (2013), Johnson and Jennings (2001)) can offer a more natural approach to programming as well as faster results, encouraging the students' motivation.

Another aspect recently appearing in automatics and programming education is the widespread use and knowledge of wireless technology. Devices such as tablets and mobile phones are starting to be used to control, program and supervise the mobile robot systems. Due to the popular use of these devices (the students normally are very familiar with this technology) the learning process is easier and more attractive.

Considering these factors, this paper introduces a way to build a multidisciplinary educational platform developed using low-cost open-hardware components, with intuitive and easy programming by using block diagrams. In addition, some possible activities applied to the platform and some examples to interact with tablets or mobile phones, are proposed.

After this review, the paper is organized as follows. In section 2, the different open hardware devices to build a prototype of educational platform are introduced. In section 3, a graphical programming language to develop applications for the proposed platform is described along with the necessary software. In section 4, the building of

⋆ This work has been partially funded by FEDER-CICYT projects with references DPI2011-28507-468 C02-01 and DPI2010-20814-C02-02, financed by Ministerio de Ciencia e Innovacin (Spain). Also, the financial support from the University of Costa Rica is greatly appreciated.

Table 1. Microcontroller boards comparison

						
Board:	MintDuino	Arduino Uno	Arduino Due	Netduino 2	Raspberry Pi	BeagleBone Black
Price:	$24.99	$29.99	$49.99	$34.99	$39.99	$45.00
Processor:	ATmega328	ATmega328	ARM Cortex-M3	STMicro Cortex-M3	ARM1176JZF-S	ARM Cortex-A8
Processor Speed:	16 MHz	16 MHz	84 MHz	120 MHz	700 MHz	1 GHz
Analog Pins:	6	6	12	6 (12-Bit)	-	7
Digital Pins:	14 (6 PWM)	14 (6 PWM)	54 (12 PWM)	22 GPIO (6 PWM)	8 Digital GPIO	65 GPIO (8 PWM)
Memory:	SRAM 2KB EEPROM 1KB	SRAM 2KB EEPROM 1KB	SRAM 96 KB	Code 192KB RAM 60KB	RAM 512MB	DRAM 512MB DDR3L, eMMC 2GB
Language:	Arduino C Variant	Arduino C Variant	Arduino C Variant	Microsoft C# or Visual Basic	Any	Any

a prototype platform is presented and, in section 5, some concepts of automatic control applied to the platform are introduced. Finally, some conclusions are drafted in the last section.

2. HARDWARE APPROACH

The platform should offer versatility without increasing its cost. By analyzing the different open-source hardware platforms available in the market, a prototype platform has been developed based on low-cost hardware components and/or recycled devices. The main components of the platform are: the microcontroller, the mechanical structure with motors, a manipulator arm and a remote control interface. The following sections detail the different devices that could be part of the platform for learning concepts applied in Automatics.

2.1 Microcontroller Card

Nowadays, there are many types of low-cost open-source microcontroller cards based on ARM processors. A comparison among the most popular boards on the market ordered from lowest to highest processor speed is shown in Table 1.

As the main purpose is to build a low cost open-source platform to introduce control concepts, any card can be suitable since high computational resources are not required. However, its programming should be simple because maybe the students do not know much about programming or maybe they are learning it. Recently, the well known company MathWorks has developed a Matlab library to develop applications with Arduino boards using Simulink blocks. For these reasons, Arduino board has been selected for assembling the platform prototype developed in this work.

Arduino (Thompson (2010)) arose in 2005 and due to its low cost and open hardware policy, in a few years it has achieved widely spread within the scientific and research community as the brain of various projects such as 3D printers (Evans (2012)), biped robots (Al-Busaidi (2012)), two-wheel self-balancing robots (Juang and Lum, 2013), *ROS* enabled mobile robots (Araujo et al., 2013), remote DC motor experiments, (Neto et al., 2012), data acquisition (Shajahan and Anand, 2013), perception and modern display fabrication lab kits (Sarik and Kymissis, 2010), network management (Faludi (2010)), manipulator arms controllers (Kadir et al. (2012)), research on underwater vehicles (Busquets et al. (2012)), and even in their lighter versions, at unmanned aerials vehicles (Lim et al. (2012)).

Among the current models of Arduino boards, Arduino MEGA (Fig. 1) has been chosen for this work. This card is very similar to the Arduino UNO but with a larger number of inputs and outputs. It provides serial communication libraries and an Analog to Digital converter. It has 54-pin digital I/O, 16 analog inputs, 4 UARTs (serial ports), a flash memory of 256 KB, a clock frequency of 16 MHz and an USB port to connect to any computer. The four UARTs provide the system more flexibility since each one can run at a different baud rate.

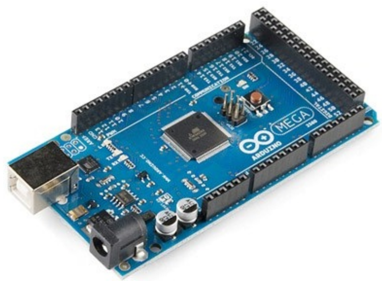

Fig. 1. Arduino MEGA board.

As the idea is to build a mobile platform prototype, it is necessary to control at least two motors. For the motor control, the same company offers an expansion board called Arduino Motor Shield (Fig. 2) which is mounted on the same MEGA board and it allows to control two DC motors with the possibility of reversing the direction (it incorporates an H bridge).

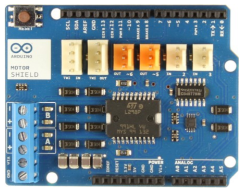

Fig. 2. Arduino Motor Shield.

To program the Arduino board there is a free multi-platform software called Arduino IDE which is available on its own website. Also, the drivers for the FTDI port incorporated in the board, must be installed when the board is connected to the computer via USB. The Arduino IDE environment is a simple text editor that allows to cut or paste and find or replace text to write the code. It has a message area, a text console useful for debugging and verifying the code, a code compiler for C/C++ as well as a toolbar with buttons for common functions such as compiling the code, downloading the program to the board, creating, opening and saving programs, monitoring them, etc. However, due to the serial data transmission between the computer and Arduino, the programming can be done from any language supporting this, as Java or Simulink.

2.2 Structure and Motors

There are many different ways to build a platform, but in this work, the structure of the platform prototype is formed by recycled LEGO pieces that can be easily assembled, which offering the possibility of providing several models of vehicles to the teacher and therefore the student can choose which of them wants to build. For the same reason, LEGO motors have been selected for the vehicle.

The LEGO Mindstorms platform arose in 1998 and its NXT version was released in 2006 (Kelly (2007)). It has been a commonly tool used in introductory courses of automatic control and robotics. It was one of the first educational tools appeared on the market with a wide range of possibilities to take advantage of the product. However, while technology evolves offering better performance and new protocols, these closed devices get obsolete in terms of computation and communication capabilities. Despite the fact that the LEGO brick is a very closed system, the sensors and actuators are connected via I2C, so they can be used for any other system which supports I2C protocol. In this project, LEGO Mindstorms DC motors have been recycled for the prototype open-source platform. These motors incorporate encoders (Fig. 3), providing a feedback of the motor position to calculate the control action.

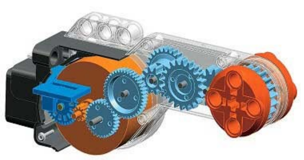

Fig. 3. LEGO Mindstorms NXT motor.

These motors work at 9 V through an I2C port that contains 6 cables. Two of them are supplying the encoders with 4.3 V and ground. Two others are responsible for sending the PWM power signal at a frequency which determines the speed and the direction of the motor. The last two are used to read the encoders. As shown in Figure 4, the encoder pulses are offset so it is possible to determine the direction of the motor according to the detected sequence. Thus, despite the motors offer a real resolution of 720 degrees per turn, in practice it is halved because it is necessary to analyze two cycle times to detect the motor direction.

The way to control these motors from the Arduino board is exposed later on, in section 5.

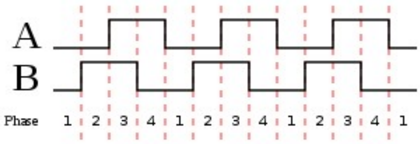

Fig. 4. Relation between encoders sequences.

2.3 Manipulator Arm

Manipulator arms are typical elements of study in automatic control. In order to expand the versatility of the platform prototype, a low cost robotic arm manipulator with 4 degrees of freedom has been incorporated therein. The arm is shown in Fig. 5. It consists of 5 servomotors: one at the base, one shoulder function, one elbow, another as wrist and the fifth one to open and close the clamp. The motor control is performed by a servo controller SSC-32 that allows a serial connection to send and receive commands. The commands keep in the following format: “#X **P**Y **T**Z **<cr>**”, where X is the ID of the servo that has to be moved, Y corresponds to the position that the servo has to achieve and Z is the time that the movement must last. It is always required to include the carriage return character in order to finish any command.

Fig. 5. Robotic arm and SSC-32 controller.

2.4 Tablet or Mobile Device Based on Android

An Android application for tablets or mobile devices has been developed in order to interact with the prototype platform through wireless communications and test its functionality. The integration of daily used devices for teaching involves savings in material costs and the advantage that the students already know the device handling. There are many wireless devices based on the Android operating system. An Android device has the advantage that, due to it is free software, the brand offers everything you need for the applications development. In addition, its publication and distribution within the Android market is more economical than other systems. For this work a Tablet Samsung Galaxy Note 10.1 has been used (Fig. 6), with a 10.1-inch screen with a resolution of 1280x800, a 1.4 GHz quad-core processor, A-GPS, 32 GB of memory, 2 GB of RAM, WI-FI, Bluetooth and 3G communication options. Nevertheless, any Android device with Bluetooth connectivity is enough for the development of the application proposed in this paper.

Fig. 6. Samsung Galaxy Note 10.1.

3. SOFTWARE APPROACH

Two programming languages have been used, one for the prototype platform and another one for the programming of the tablet.

The Arduino board manages the motor control and the arm control algorithms. Arduino IDE can be used to perform the programming directly on the board but in this work Matlab-Simulink has been chosen for this, since one of the goals is that the programming should be simple and intuitive for any student with any level of programming knowledge.

On the other hand, in order to interact with the platform prototype remotely, an android application for the tablet previously mentioned has been implemented using the Eclipse development environment with the Android SDK library.

3.1 Arduino Support for Simulink

As it is well known, Matlab is a mathematical software that provides a development environment with its own high-level language programming, widely used in the fields of control and automation. Also, on the Matlab programming environment, Simulink is an extremely useful tool for systems analysis, model and simulation. Although Matlab already uses a high-level language, Simulink allows programming directly and graphically, performing transparently the code generation. Due to the large number of available blocks in Simulink, the complexity and functionality of implemented models are determined by the designer's skill level. This point makes it a really interesting working tool in any academic field.

In order to use Simulink blocks for Arduino, the free library "*Arduino support form Simulink*" has been recently distributed. It offers a series of blocks specifically designed for Arduino boards. In this way, the scheme in Matlab-Simulink can be implemented and transferred to the Arduino in a transparent manner for the user, without worries in the conversion between Simulink blocks and the Arduino code.

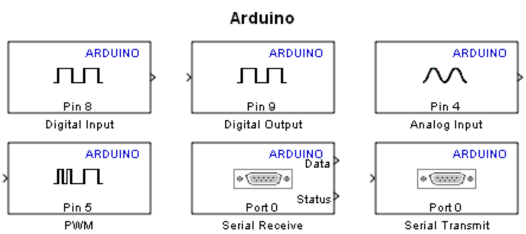

Fig. 7. Simulink Support Package for Arduino Hardware.

Among the blocks provided for Arduino, the most relevant for this work are (Fig. 7):

- Arduino Analog Input: by this module, the voltage that is being applied by a particular pin can be read. The output accuracy provided in this block is 10 bits.
- Arduino PWM: through this block a PWM signal is sent to the selected pin. The frequency of the signal pulse is set at 490 Hz, the duty cycle may be modified with a precision of 8 bits (values between 0 and 255).
- Arduino Digital Input/Output: from these two blocks digital signals can be read and written on the selected ports.
- Arduino Serial Receive/Transmit: send and receive data bytes via the serial port is possible using these blocks.

Thus, the Arduino board, along with Simulink integration, allows students to understand the process of software development for embedded systems, without the need for programming via code but in a way so simple and interactive as adding, connecting and modifying blocks with full functionality.

3.2 Android Programming

Recently the mobile devices are very popular and there are many students interested in developing applications of such devices. However, their programming is not a simple task as there is no programming methodology oriented to flowcharts or block diagrams. Therefore, a basic application for test and control remotely the platform has been developed, and it can be offered it to the students compiled and ready to use.

4. SOFTWARE-HARDWARE INTEGRATION

This section details the assembly performed in order to build the prototype platform.

A solid base resting on two motors in differential configuration has been built. Each motor has a non orientable caterpillar-track as shown in Figure 8:

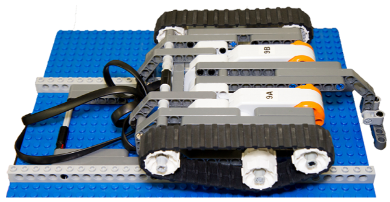

Fig. 8. Bottom view of the base of the platform prototype built by LEGO pieces.

In the base center, the robotic arm with the SSC-32 controller has been placed such that the platform is balanced with the weight of the arm. Thus, the wheels do not rise off the ground despite any position that the manipulator may adopt. In addition, two Arduino cards have been situated on each side of the robotic arm, to provide the possibility of controlling the movements of the arm and the motor speed at the same time, independently. This is necessary because the multithreaded operations in the microcontroller are not possible due to the type of high-level programming used. Thus, one of the Arduino boards is responsible for managing the positioning of the manipulator and the other one, by installing the Arduino motor shield, manages the speed control and the direction of the motors. To give autonomy to the platform, a two-cell LiPo battery of 5 A and a voltage of 7.4 V powers all the hardware. Also, for wireless communication between the platform and the tablet, a Bluetooth device has been installed and connected to the two micro-controllers. The different elements of the prototype can be observed in Fig. 9.

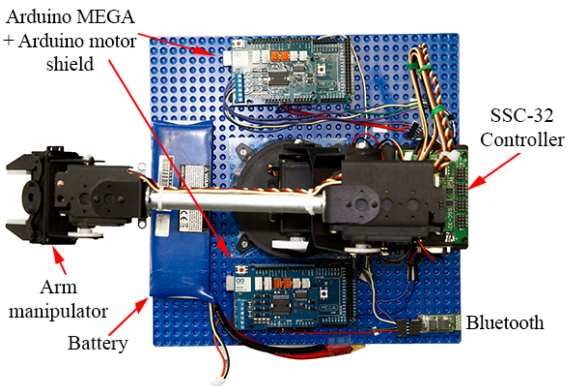

Fig. 9. Top view of the elements that are part of the educational platform prototype.

As previously mentioned, the programming of the Arduino boards has been done exclusively with the new Simulink library for Arduino. The following section shows different concepts related to the field of automatic control which have been implemented to reinforce the learning methods in an attractive way.

5. PROPOSED ROBOTIC EDUCATIONAL PLATFORM

Due to the many features of the introduced platform in this paper, it can be a very useful and versatile tool to develop a variety of practical activities related to the automatic control learning. In this section, a set of activities are proposed to the students. Of course, depending on the built platform and the concepts that the teacher desires to address, they could be changed or expanded.

5.1 Electronics

To know the hardware and the interconnection options between the elements is necessary to build the platform. For example, it is needed to know: the voltage required for each element, the maximum power consumption of the system in the worst case, the identification of the input and output ports, the low-level working of DC motors and actuators, the correct use of measuring instruments such as oscilloscopes and multimeters to verify connections and parameters, etc. By this, the platform is very useful for students since they can work in a practical way, with different issues related to theoretical concepts of electronics.

5.2 Inverse Kinematics Calculation

The inverse kinematics calculation is an essential issue in robotics related to manipulator arms. The robot arm incorporated into the platform allows the students to learn, in a practical way, how to calculate the necessary movements of the arm joints to lead its terminal tool to a position $X-Y-Z$. For this, students can use geometrical methods or the well known homogeneous transformation matrices. The idea is to implement the solutions of the issues raised in a theoretical way and visually check them on a real robot. In the prototype platform, the inverse kinematics has been developed with an *Embedded Function* block of Simulink. In this block, a simple check of the dimensions of the various elements of the arm (base, shoulder, elbow and tool) is performed.

5.3 Motor Control

The motors are well known elements within the field of automatic control and they are studied in depth in subjects related to process control, mechatronics and robotics. Using the proposed platform a variety of task can be performed such as motor identification, applying filters or velocity and position control based on the feedback value of the encoders.

5.4 Kinematic Control

The differential configuration defined in the prototype is considered one of the simplest, it consists of two caterpillar wheels diametrically opposed on a perpendicular axis to the direction of movement of the robot. This configuration allows turns on the central axis of the robot without advancing, which is known as holonomic configuration.

In a first step, students can do a simulation of the kinematic configuration of the robot, analyzing the motor speeds, the movement control, etc. And once the kinematic model is understood, students can perform tasks related to motion trajectory generation and tracking control of these paths, such as decentralized point or path control by pure pursuit.

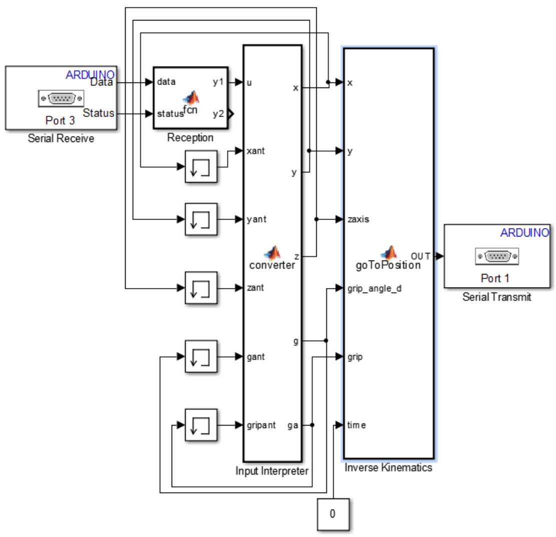

Fig. 10. Simulink scheme for robotic arm control via Bluetooth messages to Arduino.

5.5 *Blocks Programming*

As already mentioned, a programming by blocks using Matlab-Simulink has been carried out which allows easy understanding and development of programs for the provided platform. Of course, the level of complexity depends only on the programmer. Figure 10 shows for example a diagram developed in Matlab-Simulink for programming the Arduino board that manages the robotic arm by receiving messages via Bluetooth.

In this case, the Bluetooth device is connected to the serial port number 3 of the Arduino board. The *reception* block performs an active standby of the data frame and the *input interpreter* block is responsible for breaking down and interpreting the received frame as positions X, Y, Z, where the clamp of the robotic arm should be placed. Finally, the inverse kinematics block calculates the necessary commands to place the robot in the required position and send them via the serial port number 1, which is where the SSC-32 controller is connected.

For motor control through commands via Bluetooth, the prototype platform uses a second Arduino. The scheme for such control is shown in the following Figure 11:

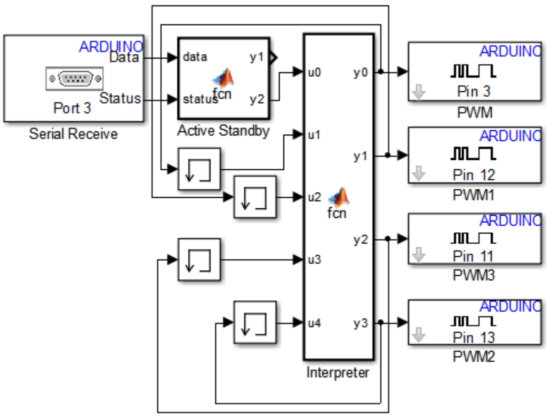

Fig. 11. Simulink scheme for motor control via Bluetooth messages.

As with the first board, the Bluetooth device is connected to the serial number 3 of Arduino. In this case the *interpreter* module translates the received data frame to the output modules of Arduino PWM Motor Shield. The right motor is connected to the PWM ports 3 and 12. The port number 3 defines the power that has to be applied to the motor and the 12 defines the direction. Similarly, the left motor is connected to pins 11 and 13. Thus, depending on the students level and the desired contents of the course, different activities can be proposed. The simplest would be that the students had to verify the various modules and generate code to program the Arduino control boards. Of course, more complex tasks can be proposed, for example they can program some embedded function in Matlab for PWM signal generation for the motors, for serial communications, etc.

5.6 *Tester and Remote Controller Android Application*

Once programmed the two Arduino boards, an application for Bluetooth interaction between the robotic platform and any mobile device can be offered to the students to test it. In this work, for the remote control and handling of the prototype platform, an Android application has been developed. It is based on a sample provided by Google for communication with Bluetooth devices, in which a server handles all incoming connections using Bluetooth sockets proper of the Android operating system. As with any Bluetooth device, the first step for the application user is to go to the options menu of the application, search and connect to the receiving Bluetooth device of the platform. In this way, the bidirectional communication necessary for sending commands from the Android application control panel to the proposed platform is established.

Fig. 12. Android application interface developed to control the platform.

For controlling the motors and the robotic arm of the prototype mobile platform, the application has two joysticks, left and right, as shown in Figure 12. The right joystick operates the motor control platform and it is composed of four intuitive directional arrows to move forward, backward or to rotate the robot. On the other hand, the left joystick, is composed of the necessary controls to move the tool of the robot arm in its 3 axes of movement (x, y, z). This last joystick also has two arrows more (up and down) to open and close the clamp of the arm, and in this way pick up and manipulate objects of the environment.

Once again, depending on the level of the students, they can only download the compiled application directly to their phone, or different activities can be proposed providing the source code of the application. The proposed platform is shown in Figure 13.

6. CONCLUSIONS

A low cost open robotic platform oriented to the education in the field of automatic control has been introduced, taking advantage of the most current open hardware and software technology. The system gathers typical elements studied in automation control problems such as motor control, electronic wiring, serial port management, the inverse kinematics of an arm manipulator, etc., and the difficulty is adjustable according to the students level. Also, the interaction between the platform and wireless devices such as tablets or smart phones based on the Android operating system has been incorporated.

Fig. 13. The low cost prototype platform manipulating pieces which can be assembled.

The assembly and the didactic concepts applicable to the platform have been detailed, as well as how to deal with its programming through flowcharts and block diagrams in an attractive and intuitive way for students.

The following link hosts a video explaining the assembly and the development of the prototype platform: `https://idecona.ai2.upv.es/videos/robotsmoviles/video12`

REFERENCES

A.M. Al-Busaidi. Development of an educational environment for online control of a biped robot using matlab and arduino. In *Mechatronics, 2012 9th France-Japan 7th Europe-Asia Congress on and Research and Education in Mechatronics (REM), 2012 13th Int'l Workshop on*, pages 337–344, 2012.

Andre Araujo, David Portugal, Micael S. Couceiro, and Rui P. Rocha. Integrating arduino-based educational mobile robots in ros. In *Autonomous Robot Systems (Robotica), 2013 13th International Conference on*, pages 1–6, 2013.

J. Busquets, J.V. Busquets, D. Tudela, F. Perez, J. Busquets-Carbonell, A. Barbera, C. Rodriguez, A.J. Garcia, and J. Gilabert. Low-cost auv based on arduino open source microcontroller board for oceanographic research applications in a collaborative long term deployment missions and suitable for combining with an usv as autonomous automatic recharging platform. In *Autonomous Underwater Vehicles (AUV), 2012 IEEE/OES*, pages 1–10, 2012.

Stephen Cooper, Wanda Dann, and Randy Pausch. Alice: a 3-d tool for introductory programming concepts. In *Proceedings of the fifth annual CCSC northeastern conference on The journal of computing in small colleges*, CCSC '00, pages 107–116, USA, 2000. Consortium for Computing Sciences in Colleges. URL `http://dl.acm.org/citation.cfm?id=364132.364161`.

Brian Evans. *Practical 3D Printers: The Science and Art of 3D Printing*. Apress, Berkely, CA, USA, 1st edition, 2012. ISBN 1430243929, 9781430243922.

R. Faludi. *Building Wireless Sensor Networks: with ZigBee, XBee, Arduino, and Processing*. O'Reilly Media, 2010. ISBN 9781449302740.

D. Fisher and P. Gould. *Open-Source Hardware Is a Low-Cost Alternative for Scientific Instrumentation and Research*. 2012.

N. Fraser. *Blockly - a visual programming editor*. Google, 2013. URL `http://code.google.com/p/blockly/`.

L. Greenwald and J. Kopena. Mobile robot labs. *Robotics Automation Magazine, IEEE*, 10(2):25–32, 2003. ISSN 1070-9932.

Poul Henriksen and Michael Kolling. greenfoot: combining object visualisation with interaction. In *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, OOPSLA '04, pages 73–82, New York, NY, USA, 2004. ACM. ISBN 1-58113-833-4.

National Instruments. Labview, 2013. URL `http://www.ni.com/labview/`.

G.W. Johnson and R. Jennings. *LabVIEW graphical programming: practical applications in instrumentation and control*. Number v. 1. McGraw-Hill, 2001. ISBN 9780071370011.

Hau-Shiue Juang and Kai-Yew Lum. Design and control of a two-wheel self-balancing robot using the arduino microcontroller board. In *Control and Automation (ICCA), 2013 10th IEEE International Conference on*, pages 634–639, 2013.

W. Kadir, R. Samin, and B. Ibrahim. Internet controlled robotic arm. 2012.

James Floyd Kelly. *LEGO MINDSTORMS NXT-G Programming Guide*. Technology in action series. Apress, 2007. ISBN 9781590598719.

Hyon Lim, Jaemann Park, Daewon Lee, and H. J. Kim. Build your own quadrotor: Open-source projects on unmanned aerial vehicles. *Robotics Automation Magazine, IEEE*, 19(3):33–45, 2012. ISSN 1070-9932.

MathWorks. Simulink, 2013. URL `http://www.mathworks.es/products/simulink/`.

J. M. Neto, S. Paladini, C.E. Pereira, and R. Marcelino. Remote educational experiment applied to electrical engineering. In *Remote Engineering and Virtual Instrumentation (REV), 2012 9th International Conference on*, pages 1–5, 2012.

Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. Scratch: programming for all. *Commun. ACM*, 52(11):60–67, November 2009. ISSN 0001-0782.

J. Sarik and I. Kymissis. Lab kits using the arduino prototyping platform. In *Frontiers in Education Conference (FIE), 2010 IEEE*, pages T3C–1–T3C–5, 2010.

A.H. Shajahan and A. Anand. Data acquisition and control using arduino-android platform: Smart plug. In *Energy Efficient Technologies for Sustainability (ICEETS), 2013 International Conference on*, pages 241–244, 2013.

C. Thompson. Build it. share it. profit. can open source hardware work? *Wired Magazine*, 2010.

J.B. Weinberg and Xudong Yu. Robotics in education: Low-cost platforms for teaching integrated systems. *Robotics Automation Magazine, IEEE*, 10(2):4–6, 2003. ISSN 1070-9932.