RoboBlock: A Remote Lab for Robotics and Visual Programming

Ignacio Angulo, Javier García-Zubía, Unai Hernández-Jayo, Iñigo Uriarte
Faculty of Engineering
University of Deusto
Bilbao, Spain
zubia@deusto.es

Luis Rodríguez-Gil, Pablo Orduña, Gabriel Martínez Pieper
Deusto Institute of Technology, DeustoTech
Deusto Foundation
Bilbao, Spain

***Abstract*— Robotics is part of K12 curricula in different subjects and countries because it is exciting and formative. To teach, the teacher and the school need a laboratory with robots, and this is a challenge because they are not cheap and they need to be maintained. In this scenario the use of a remote lab for robotics is a good solution. There are several remote labs for robotics, the main advantage of RoboBlock is that it offers in the same interface the robot and a visual tool based on Blockly to program the robot.**

Keywords—remote labs, robotics, visual programming

I. INTRODUCTION

The promotion of STEM among the young people is one of the objectives of different countries and institutions like EU. To increase the interest of the young people to science, engineering and technology the schools are including subjects that combine programming and robotics [1, 2]. The main effect of these initiatives is in K12.

Currently, programming is part of the curricula in different subjects and countries. To face this challenge, schools are using visual programming (e.g., Scratch, Blockly) and/or educational robots. Some combinations of these two tools are very popular, such as the Lego Mindstorm robots. However, they still have some disadvantages, such as a high purchasing and maintenance cost and limited availability.

When a teacher tries to introduce robotics in the curriculum he needs a laboratory with the following problems:

- The laboratory needs financial support to buy the robots that are not always cheap (LegoMindstorm costs around 350 euros).
- The teacher must maintain the robots in a good condition. He has to set up the robots every time they a lab session.
- During the lab session the students first need to review the robot before using it. In general they have around 60 minutes, and doing this first task they are losing time. Before leaving the laboratory they have to collect the robots. So, finally the session does not have enough time to work properly.
- Not all the robots are in the same condition, so the lab session will have many “branches”, one for each group of students. In this situation the teacher will be stressed.
- The students only can practice with the robots during the lab sessions and also if the teacher is with them. So, the teacher has to assume this task.

In this scenario, the use of a remote lab is a good option. The remote lab should not substitute the real robot (in the hands of the students), but to complement it and help the teacher during the teaching process.

There are several remote laboratories offering robots around the world but not with the same characteristics that RoboBlock [3, 4].

II. ROBOBLOCK’S DESCRIPTION

The RoboBlock robotics remote lab has three main parts:

- Hardware: description of the robot.
- Software: description of the programming environment based on Blockly.
- Remote lab: platform to perform the remote experiment.

A. *Hardware description of RoboBlock*

The robotic platform consists on the Zumo 32u4 Pololu robot and a low-cost embedded system (Raspberry Pi 3) that allows remote programming the controller (Arduino) and interact with the inputs and outputs of the Pololu platform. This robot is based on the microcontroller compatible with Arduino Atmega32u4. It includes DC motors, decoders, line sensors, accelerometer, magnetometer, gyroscope and buzzer.

B. *Software description of RoboBlock*

The RoboBlock platform offers the user an interface to write the code in a visual way. The programming language is

978-1-5386-0810-4/17/$31.00 ©2017 IEEE

based in Blockly (https://developers.google.com/blockly/) and it is offered through a web based interface.

New specific blockly modules to interact with actuators and sensors of the robot have been built. Combining specific modules with the blockly generic ones, provides a very simple tool to model the algorithms, easily affordable by secundary students. Fig. 1 shows some functions of RoboBlock language.

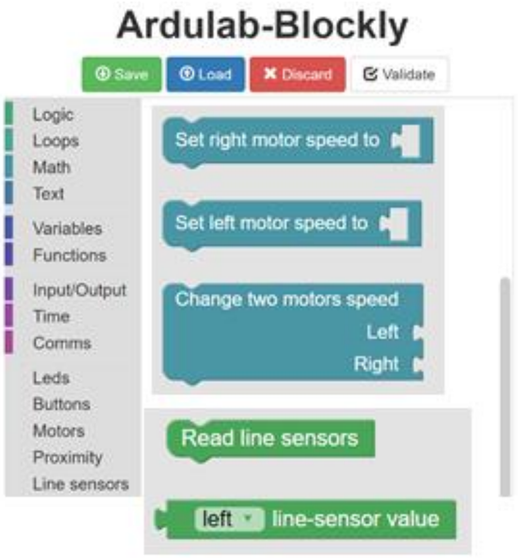
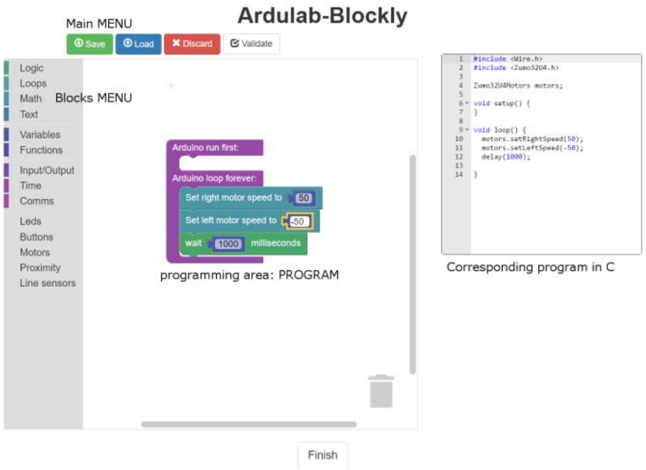


Fig. 1. RoboBlock language based on Blockly

C. *RoboBlock Remote Lab*

The hardware and the software are offered as a remote lab. They are integrated in the WebLab-Deusto RLMS (Remote Laboratory Management System).

At this moment the RoboBlock remote lab has two parts. In ArduLab the user draws the code, compile it and test it. In Fig. 2 the visual area can be seen, and in the right side the user can see the C code that is associated to the Blockly program. For advanced students an Arduino based interface is also provided.



```
#include <Wire.h>
#include <Zumo32U4.h>

Zumo32U4Motors motors;

void setup() {
}

void loop() {
  motors.setRightSpeed(50);
  motors.setLeftSpeed(-50);
  delay(1000);

}
```

Fig. 2. ArduLab programming environment

In the second part (Zumoline) the user can download the code generated in Ardulab into the RoboBlock hardware. After doing this the user can see, through a webcam, the behavior of the robot. He can decide if Roboblock is running properly or not. Formula One challenge is shown in Fig. 3.

In this case the user has to program the RoboBlock (using the sensors, etc.) trying to run the circuit as fast as possible.

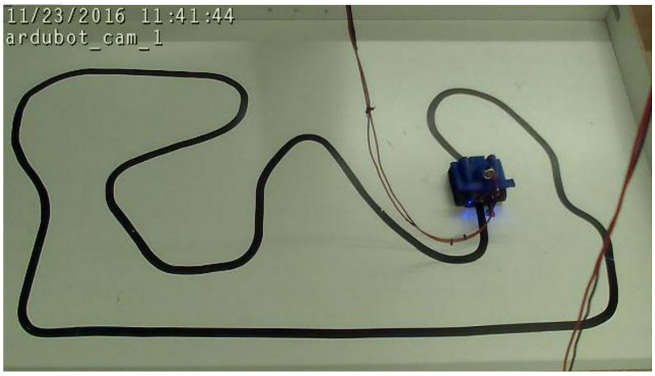


Fig. 3. Formula One Challenge in RoboBlock

As RoboBlock is connected to WebLab-Deusto a user can access the robot using any device (lap-top, tablet, smart phone..) with any OS and without security problems (open ports, firewalls). RoboBlock can be included in any LMS (moodle, etc.).

Another special feature of WebLab-Deusto is that the teacher can see who accessed the RoboBlock, when, how many time, what code was written and downloaded, etc. This helps the teacher to control and evaluate the students' activity.

III. Conclusion and Future Work

RoboBlock allows the teachers and schools to deploy programming robotics with the students without assuming some risks. All the activities are made in the web, and this is an important novelty in this area.

RoboBlock is offered by WebLab-Deusto RLMS.

Future work will be focused in improving the user experience: integrating the two interfaces (Ardulab and Zumoline) in one interface and adding new challenges. Also RoboBlock will be tested in schools.

References

[1] Roscoe, J. F., Fearn, S., y Posey, E. (2014) Teaching Computational Thinking by Playing Games and Building Robots, *Int. Conf. Interactive Technologies and Games (iTAG),* pp. 9-12.

[2] Merrill, M. D. (2002) First principles of instruction. *Educational Technology Research and Development*, Vol. 50 (3), pp. 43–59.

[3] Islamgozhayev, T. U. et al. (2015) IICT-bot: Educational robotic platform using omni-directional wheels with open source code and architecture, *Int. Siberian Conf. Control and Communications (SIBCON)*, pp. 1-3.

[4] Antonio, C. P. et al. (2015) Remote experiments and 3D virtual world in education. *3rd Experiment@, Int. Conf., exp'at 2015,* pp. 65-70.