

# The use of programmable robots in the education of programming

Zoltán Istenes<sup>a</sup>, Attila Pásztor<sup>b</sup>

<sup>a</sup>Department of Software Technology and Methodology,  
Faculty of Informatics  
University of Eötvös Loránd  
e-mail: istenes@inf.elte.hu

<sup>b</sup>Department of Information Technology  
Faculty of GAMF  
College of Kecskemét  
e-mail: pasztor.attila@gamf.kefo.hu

## Abstract

In higher education great emphasis has been laid on teaching IT as well as teaching programming. Students entering into higher education have various background knowledge of computer programming. Some of them studied some basic programming languages at secondary school; some have never ever studied anything about programming at all. This is the reason why it is so important to introduce programming problem-situations besides the basic programming knowledge, syntaxes, basic algorithms that can be solved with the creative usage of the already known algorithms amongst the favoured environment of a programming language. If these ready-made programs have “visible” results (not only the mass of characters on the display), they can motivate those who already have some background knowledge in programming and can help beginners to become experts. In order to achieve our goal programmable robots developed by LEGO for educational purposes, seem to be one of the most suitable devices. Some program developing softwares running under graphical operating systems have been made to the device that can make programs and can load them into the memory of the robot where these programs can run independently under the control of the robot’s own operating system. We have tried the use of programmable robots in both university and college education. In college education our goal is to make students like the subject with the help of this device at the beginning of their studies, while at university, getting to the end of the courses, the goal is to put the already studied programming knowledge well into practice (including Artificial Intelligence, communication, image recognition, parallel processing). To demonstrate our results, we also show some simple tasks (like following routes or looking for a spotlight) as well as more difficult ones (like “stone - paper - scissors” game, scanner or the use of the web cam). Applying this

device in teaching can help the motivation of the students to get stronger and both “beginners” and “experts” can acquire a deeper knowledge in programming through solving problems with various difficulties. To present the usage of one or two fields, or to make some creative algorithms can also be a proper theme of dissertation and scientific lectures (TDK). According to our experiences, programmable robots proved to be very useful devices in teaching programming and at present we are looking for new fields where these devices can be involved in teaching Information Technology.

*Keywords:* Robotics, Automated systems

*MSC:* 68T40, 93C85

## 1. Introduction

This paper, beyond some problems in teaching programming, deals with how new toy-like devices can be used in the education of programming both at beginner and intermediate levels. By using model robots we can make teaching programming more interesting and we can also give the students the impression of just playing. Building and programming robots motivate the beginners just as well as the intermediate students’ creativeness. The robots make students able to meet several simple but real problems and give them the possibility to try in practice the already studied theoretical knowledge from other fields of informatics. Students at intermediate level can employ their knowledge that they have already studied in Algorithms, Artificial Intelligence, Shape-recognition, Communication and Networks subject.

## 2. Some problems in teaching programming

At the Kecskemét College Faculty of GAMF IT students study programming in C/C++ language, four lessons per week, during three semesters [9]. Together with my colleagues we have seen that students’ results are getting worse and worse over the years and more and more students have to study programming again, because of their unsuccessful exams. Furthermore they liked this subject less and less. In order to explore the reason of these bad results we have made a survey among students who have already finished the first semester. The survey was made with one hundred participants and it turned out that students entering into higher education have various knowledge in programming. Forty-nine per cent of them have never studied programming at all and the other half was varied. Some studied it for only one hour per week while others had six or eight lessons per week for years.

Of course the exams’ marks show that students who already had background knowledge, reached better results, the average marks of students, who have not studied informatics before was 2.86, while the average marks was 3.31 for those who have studied informatics before. Much more beginners failed their first, second and

third exams. The survey results clearly show that only twenty-two out of forty-nine of those students who have not studied informatics at all in high school passed the exams for the first time, compared to thirty-nine out of fifty-one of those who have already studied informatics.

Answer	percent
hate	8
dislike	13
indifferent	36
like	38
love very much	5

Table 1: How do you like programming?

These bad results (see Table 1) were depressing, sixty per cent were indifferent or did not like programming and fifty per cent did not think, that programming could be important in their future carrier.

### 3. A device, which can help in teaching programming

Because of these bad results and the students indifferences, we started to search for such new methods and devices that can help us to teach in a more effective, spectacular and easier way, as well as to make them enjoy the subject. One of the devices we found was introduced in the VI. ICAI conference in 2004 (see [12]). It was a programmable robot set by LEGO. The robot can be seen as a toy at first sight, but with the help of it the teaching can be more interesting, effective and the subject can be more spectacular.

#### 3.1. The hardware parts of robot

The main part of the robot is the Mindstorms Brick, which has three input ports and three output ports and a small display. Motors, touch sensors, light sensors, temperature sensors and rotation sensors can be connected to the ports. In the brick there are RAM and ROM memories, a processor and the communication between the robots and the computer is done trough the infrared communication port.

#### 3.2. The software

The robot controller programs can be written in a C-like programming language called NQC, C/C++ language, amongst many others. In NQC language there are

various loops, if-then and switch-case sequences, functions with value and reference passes, parallel tasks and include files like in the most C-type programming languages. Its greatest advantage is that, after loading the ready-made programs into the robot, the results can be seen immediately and visually.

## **4. Simple programs written by beginners**

### **4.1. Route - following**

The first example is that the robot follows a route with two light sensors. One of the sensors is next to the route and the other one is on it. Similar example is that the robot goes around in one determined area. Both of these very simple programs were made by students after six–seven hours of practice.

### **4.2. Maximum selection**

In this example students put the algorithmic theories into practice. In their lesson they choose the biggest members of a sequence or a set and put them into practice. The robot with the help of its light sensor starts from any parts of the room to look for the lightest place of the room. A similar problem is when the robot, with its temperature sensor, seeks for the hottest part of the room. This practical application were done by the students after ten–twelve hours of practice.

### **4.3. Scanner**

In this practical application a simple scanner reads a character drawn on the paper. The robot controlling program moves the reading light sensor head above the paper at fifty by twenty pixel resolution and records the data in its datalog. This program written in NQC language is a very simple one; it has only about twenty lines and it was also written by students after fourteen fifteen hours of practice. At the end of the scanning, the robot sends the collected datalog through the infra red port to the computer, into a text file. This file is then processed by a C++ program, running on the computer and written in Visual C environment. The result of the scanning is then displayed at the computer display.

## **5. The robotics course at the ELTE**

At the Eötvös Loránd University, Faculty of Informatics, the robotics course is placed at the MSc level. At this level, the students can already program in several programming languages and have also acquired a lot of knowledge from the different areas of informatics. Therefore the aim of the robotics course is not to help the teaching of programming, but reuse, to try in practice and to reinforce the students previously acquired knowledge from different fields of informatics (see

[13]). The arrival (2006 summer) of the new Mindstorms NXT kit (see [1]) has extended the possibilities.

## 5.1. Different ways of the robot controlling

As described in the Section 3, an RCX robot can be programmed in the simple C like NQC language, but there are many other open source and free languages and programming environment for PC compatible and Macintosh computers, for Linux, Mac OS and Windows operating systems.

One of the simplest way, called “direct control”, is to send the controlling code directly from the computer to the robot, one-by-one, like: “start motor A forward” or: “stop motor B”. The sensor input is also received by the computer. There is no computation in the robot, the robot firmware interprets all the commands and the control program is running only on the computer. The robot must stay within the transmission range, which is rather limited when using the infrared communication of the RCX (1-2 meters and preferably in line of sight), but is more flexible when using the Bluetooth communication of the NXT (some dozens of meters). The controlling code is just a sequence of bytes (see [2, 3]), thus the control of the robot is possible with any programming languages, which can send bytes to devices.

The most common way of robot controlling is to write the robot controlling program on the computer itself, compile, then download the program to the robot and finally, run the program on the robot. After the downloading, the robot executes the program autonomously and there is no more communication between the robot and the computer. The simplest environment to construct programs for RCX is the original LEGO Robolab (see [4]) and for the NXT is the NXT-G Labview (see [5]). Both are graphical block-building drag-and-drop style environments.

Probably the best known programming language is the Not Quite C (NQC) for the RXC and the Not eXactly C for the NXT (see [6]). This API and compiler toolkit is based on a subset of the C programming language but has serious program control structures and data type limitations. There exists an integrated development environment called “Command Center” (see [7]).

All the previous environments and languages work with the original LEGO firmware, but there exists a possibility to change this firmware. The LeJOS (see [8]) and brickOS (see [9]) are two firmware replacements and programming languages, the former includes a Java virtual machine and the latter provides C and C++ programming environments. An eclipse plug-in allows to develop Java code for the leJOS JVM.

In a so called “combined control” mode, one part of the robot control computation is done on the robot, while the other part of the computation is done on the computer. The most common example is the use of a web camera. The robot is moving in an environment using its onboard program. The web camera image is processed on the computer, since to connect a web camera to an RCX or to a NXT is not easy, and because to process the camera’s image in real-time requires more computation power then what an RCX or a NXT have. Afterwards, the camera image processed by the computer, it can “help” or “assist” the robot onboard

program.

Last but not least, two or more robots (and computers) can communicate via their infrared or Bluetooth connection, forming an agent network and realising like this very interesting cooperative tasks.

## **6. Programming the robots using more IT knowledge**

Students at the MSc level can use their previously acquired knowledge from different fields of informatics to program robots. This section presents some interesting projects made by students.

### **6.1. Computer architecture**

A simplistic model of a CD player (or rather hard disk) demonstrated the lecture of data blocks from a rotating surface. The model used 3 light sensors and 2 motors, one light sensor for data read, one light sensor and motor for positioning the reader arm to the desired cylinder, one light sensor to synchronise the lecture to the sectors and one motor to turn the disk.

An other machine could “read” LEGO blocks having 7 positions with hole or without hole corresponding to 7 bits. The first 3 bits were interpreted as instruction code and the last 4 bits were interpreted as a number. Introducing into the machine different blocks, the machine could simulate a simple accumulator machine and realise simple arithmetic.

### **6.2. Game playing using artificial intelligence algorithms**

Several game playing projects were realised, such as number guessing, battleship or gomoku (called also five-in-a-row). In the battleship game two RCX were the two players and the third RCX was the game master and plotted the game state on a paper. The three RCXs communicated via infrared communication.

In the gomoku game a human could play against the robot. After the robot has marked his move on the paper, the human could mark his own move on the paper then the robot scanned the paper to find the human move and according to that, it marked his new move on the paper and so one.

### **6.3. Using a web camera and image processing to navigate**

A number of other projects used a web camera mounted above an environment. The robot moved around in this environment and was assisted by the computer processed image. The robot could avoid obstacles and could find its way out of a labyrinth or a maze.

In some projects it could even find and kick a ball into a goal.

## 6.4. Other projects

Recently, in a diploma work we used the NXT robot with its ultrasound distance detector to detect all the objects around the robot and to build a map of these objects through the robots moves, like in a simultaneous localisation and mapping problem.

In an other project the robot was controlled from a computer using the certified proved-property-carrying code architecture, where the robot executed the downloaded program only if it satisfied certain properties.

## 7. Summary

Robots are often just seen as children's toys or as mysteriously complicated things. Today, programmable robot building kits can motivate the beginner programmers to write different programs for robots since it allows to see immediately the results of the build robot in a real world environment. The more experienced programmers can implement more complicated programs, using, combining and trying their previously acquired knowledge from different fields of informatics. This kind of active learning and discovery learning (see [11]) is more than useful, since students affirm that these kind of courses are more motivating, interesting and entertaining.

## References

- [1] NXT <http://mindstorms.lego.com>
- [2] RCX INTERNALS <http://graphics.stanford.edu/~kekoa/rcx/>
- [3] NTXREME <http://mindstorms.lego.com/Overview/NXTreme.aspx>
- [4] LEGO ROBOLAB <http://www.lego.com/dacta/robolab>
- [5] NATIONAL INSTRUMENTS LABVIEW <http://www.ni.com/academic/mindstorms/>
- [6] NQC, NXC <http://bricxcc.sourceforge.net/nqc/>
- [7] COMMAND CENTER <http://bricxcc.sourceforge.net/>
- [8] LEJOS <http://lejos.sourceforge.net/>
- [9] BRICKOS <http://brickos.sourceforge.net/>
- [10] KLASSNER, F., ANDERSON, S., Lego MindStorms: Not Just for K-12 Anymore, *IEEE Robotics and Automation Magazine*, in Press (January 2003). <http://www.csc.vill.edu/~klassner/pubs/klassner-mindstorms-IEEE-RA02.pdf>
- [11] BONWELL, C., EISON, J., Active Learning: Creating Excitement in the Classroom, *AEHE-ERIC Higher Education Report* No. 1. Washington, D. C.: Jossey-Bass., (1991), ISBN 1-87838-00-87.

- [12] ISTENES, Z., Learning serious knowledge while “playing” with robots *ICAI’04 6th International Conference on Applied Informatics*, Eger, (2004).
- [13] ISTENES, Z., Classroom Knowledge and Laboratory Experience in Robotics *IATED, INTED 2007 Proceedings* Valencia, (2007), ISBN 978-84-611-4517-1.

**Zoltán Istenes**

University of Eötvös Loránd  
Faculty of Informatics  
Department of Software Technology and Methodology,  
1117 Budapest, Pázmány Péter sétány 1/c.  
Hungary

**Attila Pásztor**

College of Kecskemét  
Faculty of GAMF  
Department of Information Technology  
6000 Kecskemét, Izsáki út 10., Pf. 91.  
Hungary