2009

# Micro Information Systems Programming With Lego Mindstorms

Rasmus Ulslev Pedersen
*Copenhagen Business School*, rup.inf@cbs.dk

Matthias Paul Scholz
*Interactive Objects Software*, matthias.paul.scholz@gmail.com

Ralph Hempel
*Hempel Design*, rhempel@bmts.com

# MICRO INFORMATION SYSTEMS PROGRAMMING WITH LEGO MINDSTORMS

Rasmus Ulslev Pedersen
Department of Informatics
Copenhagen Business School
rup.inf@cbs.dk

Matthias Paul Scholz
Interactive Objects Software
Freiburg, Germany
matthias.paul.scholz@gmail.com

Ralph Hempel
Hempel Design
Ontario, Canada
rhempel@bmts.com

## Abstract:

In this paper we discuss and compare how LEGO MINDSTORMS NXT can be used to teach information systems (IS) students selected computer science subjects. We present the wide-spread robotics platform named LEGO MINDSTORMS NXT from LEGO Systems. It is an educational platform with an open source operating system. This open source property of LEGO MINDSTORMS has resulted in several alternatives to LEGO's own operating system. Therefore this platform can be tailored used to teach students a programming language that fits their needs. Teenagers and non-programmers usually settle for LEGO's graphical programming environment, while other groups choose a different language. We demonstrate some of the most obvious choices of programming language for IS students.

**Keywords:** IS curricula, Introductionary programming, Java, pbLua, C, LEGO MINDSTORMS NXT

## I. INTRODUCTION

IS student curricula can benefit from the inclusion of a programming class. Programming a system is traditionally something that is associated with computer science majors, but we argue that programming with the LEGO MINDSTORMS NXT is so rewarding that it can enrich IS education. The vast majority of systems that surround us are not personal computers or mainframes even though we sometimes focus almost all of our IS curricula in that direction. Everywhere in our daily environment there are little computer systems that take care of some specialized function. That can be a GPS device, an EKG monitor, a room thermostat, a mobile phone, an intrusion detection system or much more – in fact, there's a universe of embedded systems around us.

LEGO MINDSTORMS NXT is an educational system for such embedded systems. When an embedded system is used in an IS context it is called a micro information system (micro-IS)(Pedersen 2008; Pedersen 2006b).To prepare the IS student body for a future as both professionals and potential researchers, we need to direct some of our focus to the programming of such small systems. It may seem overwhelming at first to claim that we can teach non-technical individuals in regard to how such systems work and what the main design constraints are.

However, that is precisely what we will do in this paper: we will present some programming languages such as pbLua and Java (Scholz 2009). Both of these languages are good starting points for the novice programmer. In addition, we also point to earlier studies (Pedersen 2007), where a so-called "difficult" programming language was used with success for freshmen at an interdisciplinary IS program.

The paper is structured such that we start out with an introduction of LEGO MINDSTORMS NXT, followed by a section on a very popular programming environment in Java for LEGO MINDSTORMS. After that, we show how pbLua works as a programming language for LEGO MINDSTORMS NXT. To add depth to the paper, we include a section on a programming environment that could be useful in an electrical engineering setting - we include it here to show the difference to pbLua and Java. Finally, we briefly touch the programming environment that originally comes with the NXT and compare and discuss the different options (Java, pbLua, and C) for programming LEGO MINDSTORMS NXT in terms of usability for IS students.

# II. JAVA PROGRAMMING WITH LEJOS FOR MICRO INFORMATION SYSTEMS

leJOS NXJ[1] is a programming and runtime environment for the LEGO MINDSTORMS NXT that allows for programming Lego NXT robots in Java and executing them directly on the NXT hardware (also called "the Brick").

It includes a dedicated firmware for the NXT with an implementation of a Java Virtual Machine, a library of Java classes that implement the leJOS NXJ Application Programming Interface (API) for execution on the brick, a library of Java classes for computer programs that communicate with the brick via USB or Bluetooth as well as various tools for debugging and flashing the firmware and for compiling, linking and uploading programs to the NXT brick.

leJOS NXJ comes with its own firmware that replaces the original one provided by The LEGO Group™ for the NXT hardware, which means that this firmware must be flashed onto the NXT before any leJOS NXJ program can be uploaded to and executed. However, the original LEGO® NXT firmware can be restored at need, using the software supplied by The LEGO Group™.

While NXT-G, the official programming environment for the NXT by The LEGO Group™ (see below), is targeted at programming beginners and suited mainly for comparatively basic programs, leJOS NXJ addresses a more advanced user group. For them, there are many advantages to NXT-G that leJOS NXJ offers: next to being based on a standardized and popular programming language and the modern object-oriented programming paradigm which is supported by many state-of-the-art IDEs like Eclipse[2] or Netbeans[3], it also offers cross-platform support, additional in-built robotic-oriented programming frameworks, multi-threading, support for third-party sensors, and is much more efficient and compact when it comes to complex programming tasks[4].

leJOS is an Open Source Project hosted on the popular Sourceforge OS repository[5] and originally was a spin-off from another project that implemented a Java VM for the LEGO® MINDSTORMS RCX system. Not only the NXT version, but also the predecessing RCX version of leJOS gained a lot of popularity in the LEGO® MINDSTORMS Robotics community, with close to 200,000 downloads of all versions of leJOS until today.

[1] http://www.lejos.org

[2] http://www.eclipse.org

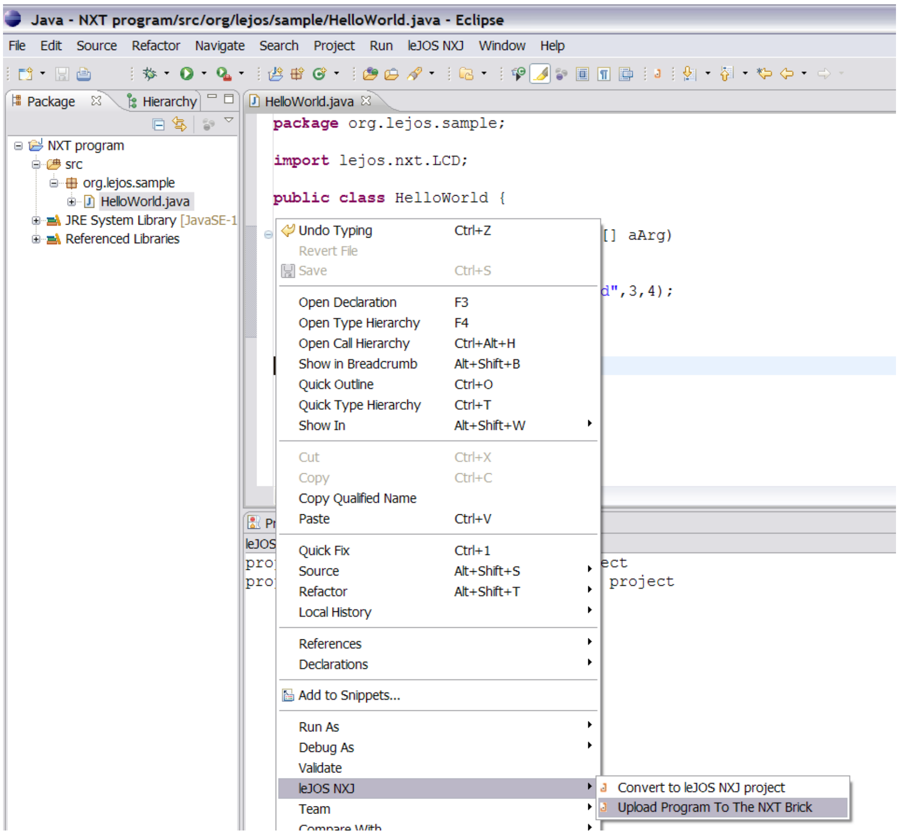
[3] http://www.netbeans.org

[4] For a further discussion on this topic, see http://lejos.sourceforge.net/nxt/nxj/tutorial/Preliminaries/Intro.htm

[5] http://sourceforge.net

# III. LEJOS ECLIPSE PLUGIN

The Eclipse plug-in for leJOS NXJ is an addition to the popular Eclipse platform and is meant to facilitate developing Java programs that are able to run directly on the NXT brick.



```
package org.lejos.sample;

import lejos.nxt.LCD;

public class HelloWorld {
```

Figure 1. Eclipse leJOS plug-in

Using Eclipse's powerful plug-in extension mechanism, it enriches the platform with various features that are seamlessly integrated into Eclipse's user interface (see Figure 1. Eclipse leJOS plug-in):

- uploading the leJOS NXJ firmware replacement for the NXT to the hardware brick
- creating, editing, compiling and linking leJOS NXJ Java programs for the NXT, integrated in Eclipse's editor, project and builder frameworks
- uploading leJOS NXJ programs to the NXT brick
- running leJOS NXJ programs on the NXT brick
- displaying additional information for debugging
- connection to the NXT brick is possible via USB or Bluetooth.

By using a state-of-the-art integrated development environment and enhancing it for the NXT developer's needs, the plug-in not only frees the developer from having to switch between different environments like text editors and command shells when developing and testing NXT robot programs. It also provides access to the most powerful features of one the best Java development and runtime platforms presently around.

The plug-in is based on the most recent leJOS NXJ release and supports the current Eclipse 3.x versions and hence all the operating systems they are available for. Presently, this includes Windows, Linux and Mac OSX. It can easily be downloaded from the web[6] and installed using Eclipse's online update mechanism.

## IV. PBLUA FOR PROGRAMMING MICRO INFORMATION SYSTEMS

The pbLua project is developed by Ralph Hempel. It is an implementation of the Lua[7](Ashwin 2007) interpreter that runs entirely on the NXT brick. The main differentiator between pbLua and other NXT programming languages is the very low software requirement on the host computer. There is no IDE needed, and no special compiler or other development tools. In fact, the NXT can be programmed with any host machine that supports a text editor and a terminal program.

## LUA BACKGROUND INFORMATION

The core Lua language was created in 1993 by Roberto Ierusalimschy, Luiz Henrique de Figueiredo, and Waldemar Celes, members of the Computer Graphics Technology Group (Tecgraf) at PUC-Rio, the Pontifical Catholic University of Rio de Janeiro, in Brazil. Its history from then until now (Roberto et al. 2007) is marked by methodical addition and pruning of features to the point where it is now considered very stable.

Lua is remarkable in that it offers many features from the functional programming domain, and at the same time has a very low learning curve for beginning programmers. The primary data structure in Lua is the table - a combination of conventional indexed access and with the added bonus of associative array keys.

[6] http://www.lejos.org

[7] Lua Website - http://www.lua.org

The Lua language is the premier embedded scripting language in online game development with World of Warcraft as the most often cited example. The Adobe Photoshop Lightroom product is about 40% Lua. More recently, Google has announced support for Lua scripting on the Android platform[8].

There is a wealth of documentation for Lua, most notably the excellent Programming in Lua (Ierusalimschy 2006) by the language's chief architect. It is a slim book that is both readable and a great reference for Lua programming style and idioms.

## LUA IS EMBEDDABLE AND EXTENSIBLE

Lua is small enough to be embedded as the scripting language within larger applications. This also makes it ideal for embedding into small devices as an operating system. Lua also has a well documented API for binding with standard C code, which allows the language to be extended with additional functions.

The pbLua firmware is made up of three main components. The first is the Lua source code, which is pure ANSI C and was not modified at all to run on the NXT. The second part is a selection of drivers that was chosen from the open sourced LEGO firmware for the NXT. The third component was the binding between the Lua language and the LEGO drivers.

The process of porting Lua to the extremely constrained microcontroller in the NXT is documented elsewhere(Luiz Henrique de et al. 2008).

All of the hardware subsystems on the NXT brick are available for use in student applications. There are functions to interface with the display, motors, analog and digital sensors, speaker, and the front panel pushbuttons. These basic APIs are enough to teach even young students the basics of robotics.

Once the students have mastered the basics, they can move on to Bluetooth and RS485 networking, as well as floating point math and file handling. They can also explore more advanced Lua scripting with numerically indexed or associative keyed tables.

Perhaps more than any other third party language option, pbLua is ideal for experimenting with the many sensor hardware development kits, as it offers discrete control of the digital I/O pins on the 4 sensor ports.

## PBLUA IN EDUCATION

The minimal host side requirements for pbLua make it an ideal teaching language for students from high-school to post-secondary levels. When the pbLua firmware is flashed onto the NXT, the NXT presents itself to the host computer as COM or serial device on the USB bus. It is also possible to pair the NXT with the Bluetooth radio on the host computer and use that as an untethered console.

As long as there is a terminal program that can connect to a serial device, it is possible for students to produce simple programs that interact with the console within minutes:

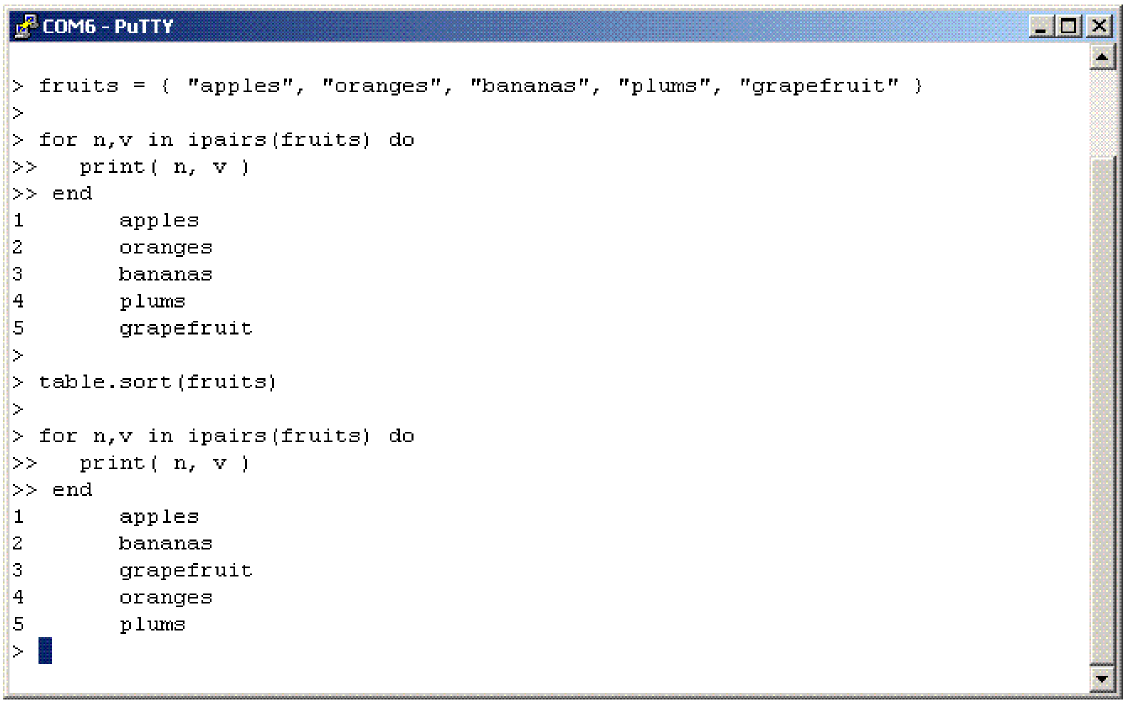
```
print("Hello World")
```

[8] http://google-opensource.blogspot.com/2009/06/introducing-android-scripting.html

From there, they can be exposed to simple timing loops that make use of the NXT API for Lua:

```
t=nxt.TimerRead()
for i=1,10000 do
end
print(nxt.TimerRead()-t)
```

And further to manipulating associative arrays:



```
> fruits = { "apples", "oranges", "bananas", "plums", "grapefruit" }
>
> for n,v in ipairs(fruits) do
>>    print( n, v )
>> end
1       apples
2       oranges
3       bananas
4       plums
5       grapefruit
>
> table.sort(fruits)
>
> for n,v in ipairs(fruits) do
>>    print( n, v )
>> end
1       apples
2       bananas
3       grapefruit
4       oranges
5       plums
>
```

Figure 2. The pbLua Console (using PuTTY)

The Lua programming language is a class of interpreted languages that has a Read-Eval-Print-Loop (REPL) front end. This means that it is possible to enter and test new functions right at the console.

The author of pbLua writes and tests most of the sample programs by simply cutting the functions out of the text editor and pasting them to the terminal for immediate evaluation. This rapid prototyping method encourages experimentation and makes it easy to modify code as conditions change.

While it's certainly not an appropriate development method for large scale software, it's entirely suitable for the smaller robotic applications that the NXT will control. The simplicity of the development environment removes a significant barrier to getting fast results with students. When educators first encounter the development environment, they often expect a more complex interface between their ideas and the NXT brick. Once they become used to simply cutting and pasting their code to the NXT, their acceptance increases dramatically.

## IV. NXTGCC

The primary purpose of **THE LEGO GROUP™** having published the NXT source code was to open up the possibility for modifying it by the NXT community that encompasses a lot of very skilled software experts also.

There are two routes one can take: One is to slightly modify the existing operating system such that it can still run original NXT-G compiled programs. The second and more sophisticated option is to program a new operating system from scratch. This is time consuming, but since there is a complete tool chain (NCTGCC) (Pedersen 2006a) with all the source code available, this task is manageable.

## V. NXT-G

NXT-G is the block programming language originally provided by **THE LEGO GROUP™** for NXT programming; as said, it is targeted at programming beginners and suited mainly for comparatively basic programs. NXT-G comes with a graphical syntax; programs can be created using an IDE provided by **THE LEGO GROUP™** that interfaces with NXT either via USB or Bluetooth. Among other things, the operating system of the NXT brick can be upgraded by features located in the IDE. A NXT-G program created by the user gets compiled by the IDE and downloaded to the NXT brick.

During that process, the NXT-G program and its sibling Labview components from National Instruments Labview are compiled into a file in an executable bytecode format. This file is downloaded then into the NXT file system onto the brick and run there by the virtual machine.

## VI. COMPARISON AND DISCUSSION

The programming environments presented in the previous sections are quite different. They range from the more technical C programming environment to the programming choices of pbLua and Java. Both pbLua and Java are languages that are easy to learn. We consider pbLua having a flat learning curve because it is a textual language and it is not compiled before the actions are carried on the NXT. On the other hand Java is a language used on a wide scale of software systems, ranging from mobile phones to large business enterprise systems. It is a very popular programming language because of its wide applicability and well-documented application programming interfaces (APIs). Being object oriented, its learning curve is more steep, though, and it takes some time before the student can make full advantage of its power.

We might add that it is also possible to program the LEGO MINDSTORMS NXT system in C, but that approach is quite technical and hence will be better suited for technical oriented students.

It is important to note, though, that both the Java programming environment (leJOS) and pbLUA use a C-based hardware interface layer.

## VII. CONCLUSION

In this paper we have investigated how the robotics platform LEGO MINDSTORMS NXT can be leveraged for IS teaching. The motivation is that most systems are actually embedded systems despite most current IS teaching focus solely on PC and server programming. We argue that the LEGO MINDSTORMS platform is a well-suited base for teaching IS students how such embedded systems works and what design choices are dominant in that very domain.

Java and pbLUA are presented as two programming choices for IS students. Java has the advantage that is it well-established, well-documented and popular, so there might be a good chance are that students already have tried it. pbLua is a good option because the learning curve is very flat and there exist online tutorials which enable the student to start experimenting right away.

The actual advantage of the topic associated with this paper is that we teach prospective researches and future professionals how micro information systems (micro-IS) do work in theory and practice.

# AUTHORS

**Rasmus Ulslev Pedersen** is maintaining a project for LEGO MINDSTORMS NXT called nxtgcc (see http://nxtgcc.sourceforge.net/). He is member of the LEGO MINDSTORMS user panel. At Copenhagen Business School, he is associate professor and manages the embedded software laboratory at the Department of Informatics.

**Matthias Paul Scholz** is the creator of the popular leJOS Eclipse plugin for the leJOS project and is one of the developers of leJOS, the Open Source Java platform for the NXT. He also is a member of the MINDSTORMS Communiyt Partner Program of The LEGO Group™.

**Ralph Hempel** was one of the 4 users selected for the Mindstorms User Panel and is profiled as a firmware expert. He is the creator of pbLua which is an ideal programming language for the new LEGO MINDSTORMS NXT.

## LIST OF REFERENCES

Ashwin, H. "Traveling Light, the Lua Way," *IEEE Softw.* (24:5) 2007, pp 31-38.

Ierusalimschy, R. *Programming in Lua, Second Edition* Lua.Org, 2006.

Luiz Henrique de, F., Waldemar, C., and Roberto, I. *Lua Programming Gems* Lua.Org, 2008, p. 368.

Pedersen, R. "NXTGCC: Open Source Firmware Development Kit for Lego Mindstorms NXT," 2006a.

Pedersen, R. "Micro Information Systems: Toward an experimental platform," in: *Ubiquitous Knowledge Discovery Workshop, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2008.

Pedersen, R.U. "Proceedings of the 1st Workshop on Knowledge Discovery from Data Streams (IWKDDS, ECML PKDD 2006)," Hard Real-time Analysis of Two Java-based Kernels for Stream Mining, Berlin, Germany, 2006b.

Pedersen, R.U. "TinyOS Education with LEGO MINDSTORMS NXT," in: *Learning from Data Streams,* J. Gama and M.M. Garber (eds.), Springer, 2007, pp. 231-241.

Roberto, I., Luiz Henrique de, F., and Waldemar, C. "The evolution of Lua," in: *Proceedings of the third ACM SIGPLAN conference on History of programming languages*, ACM, San Diego, California, 2007.