Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 625 – 628

ISBN 978-83-60810-14-9
ISSN 1896-7094

Real-time Control Teaching Using LEGO® MINDSTORMS® NXT Robot

Wojciech Grega,
AGH University of Science and Technology,
Department of Automatics, 30-059 Krakow, Poland,
Email: wgr@ia.agh.edu.pl

Adam Piłat
AGH University of Science and Technology,
Department of Automatics, 30-059 Krakow, Poland,
Email: ap@ia.agh.edu.pl

***Abstract*—A current trend in learning programs, even at high-degree studies, is applying the concepts of "learning by projects". In this context, the LEGO® MINDSTORMS® NXT modular robot appears as a simple, flexible and attractive educational platform to achieve the referred challenge in many domains of information technologies, especially in real-time control. The project team can operate with the real system (e.g. constructed robot) to realize a number of particular tasks. The ready-to-use embedded platform and dual microcontrollers architecture represents actual trends in the modern control systems designs. The open hardware and software architecture gives unlimited possibility to handle all devices connected to the main control brick and write effective control algorithms.**

The proposed rescue robot project is a good example of the simple system where, a number of real-time control problems can be analyzed. The project on this scale requires detailed planning, cooperation in teams, extensive literature survey, and comprehensive software design. That is exactly, what we need for "learning by projects" concept.

I. Introduction

A current trend in learning curricula, especially at engineering studies, is applying the concepts of "learning by experiments" or "learning by projects" [1]. The method gives the students the opportunity to get familiar with some practical problems that faces project development teams and organizations in real situations. It was found, that learning achievements resulting from the application of a collaborative work methodology based on the "learning by projects" are much higher [2].

This concept was also proposed for ILERT (International Learning Environment for Real-Time Software Intensive Control System) EU/US project [3]. This study leads to establishing a methodology for a multinational, engineering program, producing graduates capable of working efficiently in multidisciplinary teams engaged in international collaboration on industrial projects. One important output the proposed study is creation of an interdisciplinary specialization in Real-Time Software-Intensive Control (RSIC). As a part of the pilot implementation phase of ILERT project selected courses were introduced as a RSIC curriculum units, acceptable for engineering programs in four partner organizations.

This work was supported by the AGH research grant

During the research phase of the ILERT we did an intensive literature study demonstrating, that several technical universities have practically implemented robotic design and control experiments with the LEGO MINDSTORMS kits [4], [5]. The LEGO kits give the students a rich and flexible material, which they can use in their design of their robotics projects. The LEGO ability to link directly with Simulink® and access the MATLAB® toolboxes has classified this robot kits as an "open" laboratory. The LEGO MINDSTORMS NXT design is based on the advanced 32-bit ARM7 microcontroller, which can be programmed with the LabView based block-oriented language. However, many other programming tools and languages applicable for real-time LEGO robot control experiments are available in the Internet. The published hardware [9] and software [10] documentation makes the LEGO® MINDSTORMS® NXT system open for any kind of modifications and new applications. Several authors are replacing the operating system of the MINDSTORMS microcontroller with a real-time operating system suitable for C/C++ or similar applications. Most popular is NXC language [17], which supports all of the commands provided by the ARM7 microcontroller. This upgrade might be advantageous for some real-time control applications. The "upgraded" LEGO MINDSTORMS experiment can teach students the importance of real-time computing, periodic tasks development, timers and tradeoffs involving embedded processor size and cost versus performance relations within the context of control systems.

II. Teaching requirements

From educational point of view, if control aspects are the key element of the curricula, student are expected to design control algorithms based on the mathematical model, simulate how the controlled system works and then implement the controller for the robot during the limited time of a regular academic course. In this case integration of real-time environment with rapid prototyping platforms such as MATLAB/Simulink or LabView is essential.

If other topics of real-time control system must be covered, for example, task scheduling, resource management, real-time communication or fault-tolerance then text based or object oriented languages are preferable.

978-83-60810-14-9/08/$25.00 © 2008 IEEE

It is also important to demonstrate to the students how to fulfill the needs for embedded systems, e.g. how to minimize the application memory requirements. It will be also valuable, if the real-time system monitoring tools provide on-line facilities for measuring execution times.

Working with communication algorithms the LEGO MINDSTORMS built-in Bluetooth media or IEEE 802.11 extension can be explored. The communication tasks can be focused on information interchange between two or more robots and host PC. The PC can support a higher level control algorithms or just to be used for monitoring and data acquisition purposes.

III. Real-time operating systems for LEGO® MINDSTORMS® NXT

The NXT-G standard graphical programming environment for LMNXT is useful only for very beginning experiments with robots. Its limited performance does not allow to create and diagnose in real-time the specific algorithms. To improve the performance of robot control a number of software tools were developed and are available [8]. The programming environments available for the LEGO® MINDSTORMS NXT robot and representing real-time features are listed in Table 1. Two most popular solutions for academia and teaching purposes are: LEGO MINDSTORMS Toolkit for LabView and Embedded Coder Robot NXT - based on the graphical environment. Both of them need the licensed software and a number of toolboxes installed. The LabView can operate with a standard LEGO firmware while MATLAB requires the replacement by the leJOS system.

Text programming solutions are represented by NXC and leJOS – both of them are free and open source solutions. The NXC (Not Exactly C) is a text language for MINDSTORMS® NXT robot microcontroller programming with multitasking features, based on standard LEGO firmware [6].

With free programming environment IDE [17] it is possible to create complex and advanced data acquisition and control algorithms operating in real-time, including file management and Bluetooth communication features implementation.

Using text language the programming skills of students together with their understanding of data precision, time dependencies and I/O devices access methods to can be improved.

The leJOS is a Java-based replacement firmware for the LEGO MINDSTORMS RCX microcontroller [14]. The leJOS environment is based on object oriented language (Java) and has the following features: preemptive threads (tasks), arrays including multi-dimensional, recursion, synchronization, exceptions. Java variable types includes Float, Long, and String.

The nxtOSEK [13] (previous name – up to June 2008 was leJOS OSEK) is a hybrid of existing two open source projects (leJOS NXJ and TOPPERS OSEK):

- leJOS NXJ is a API device for NXT sensors, motors, and other external devices [14],
- TOPPERS OSEK provides real-time multitasking features proven in automotive industry [16]. OSEK was originally designed for embedded real-time control systems which are used in real cars [15].

The nxtOSEK is focused on real-time control applications for LEGO MINDSTORMS RCX, thus user-friendly GUI/file system was out of target. Additionally, one can use a graphical modeling, simulation, and code generation environment which is called Embedded Coder Robot NXT - specific MATLAB/Simulink Toolbox.

A unique feature of nxtOSEK application is that users do not need to apply time wait API, which is frequently used by other NXT programming languages to execute control algorithm at desired timing. The nxtOSEK provides accurate preemptive and periodical/event driven task scheduling.

An interesting programming interface, based on client/server architecture , is URBI (Universal Real-time Behavior Interface) [7]. This scripting language can be interfaced with several popular programming languages (C++, Java, MATLAB,...) and OS (Windows, Mac OSX, Linux).

Table I.
Selected programming tools for LEGO MINDSTORMS NXT

	LabVIEW Toolkit	**Matlab/ Simulink**	**NXC**	**RobotC**	**leJOS NXJ**	**nxtOSEK**	**URBI**
Programming	Graph	Graph	Text	Text	Text	Text	Text
Syntax	NI blocks	Simulink blocks, C	Like C	C	Java	C/C++	Like C/C++
Firmware	Standard	Repl.	Standard	Repl.	Repl.	Repl.	Standard
License	LabView	Matlab/Simulink	Freeware	Yes	Open source	Open source	Open source
Events	No	Yes	No	Yes	Java events	Yes (OSEK RTOS)	Yes
Multithreading	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Bluetooth communication: Brick to PC	Yes	Yes	Yes	Yes	Yes	Yes	n.a.
BluetoothBrick to Brick	Yes	Yes	Yes	Yes	Yes	Not yet	n.a.
Floating point	No	No	No	Yes	Yes	Yes	Yes

The URBI language uses an C++ like syntax. It allows to obtain parallel execution of commands, event programming, command tagging, dynamic variables. With the event feature a user can react on sensor inputs or create individual events and emit them ones or periodically.

IV. Example: NXC – based system for robot DC motor control

The demo LEGO MINDSTORMS NXT robot has been developed for RSIC teaching purposes. The robot uses two drives for movement and one for area scanning with distance and light sensor.

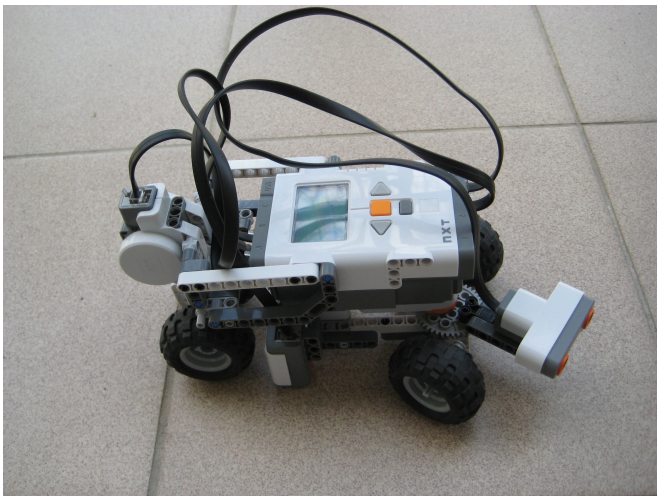

Fig 1. Sinusoid

The problem of target search in the maze was defined as a practical example ("rescue problem"). One can imagine a few algorithms solving this problem. The basic problem of the proposed robot is accurate and precise motor control. The error in the motor control can affect the correct navigation and robot movements. From the practical point of view, it is not so important for rescue operation in the maze that the robot moves, but how precise it operates and what is the accuracy of the controlled devices. Thus, the tasks execution details and precision of the robot controller are important.

The aim of this part of the project was to analyze the motor control using two tasks dedicated to motor PID controller and motor encoder measurements. Before the any multitasking operation is started it is required to initialize semaphores, files, read timer values and start simultaneous task execution. The aims of these two tasks were as follows:

- TASK 1 – "Motor control": to read Tick Counter and Log data to file, start Motor PID control, set semaphore when PID control loop terminates, read Tick Counter and Log data to file,
- TASK 2 - "Encoder Means": to read Tick Counter and Log data to file, read motor rotations and speed and log them to file, till semaphore is inactive, read Tick Counter and Log data to file (Fig.2).

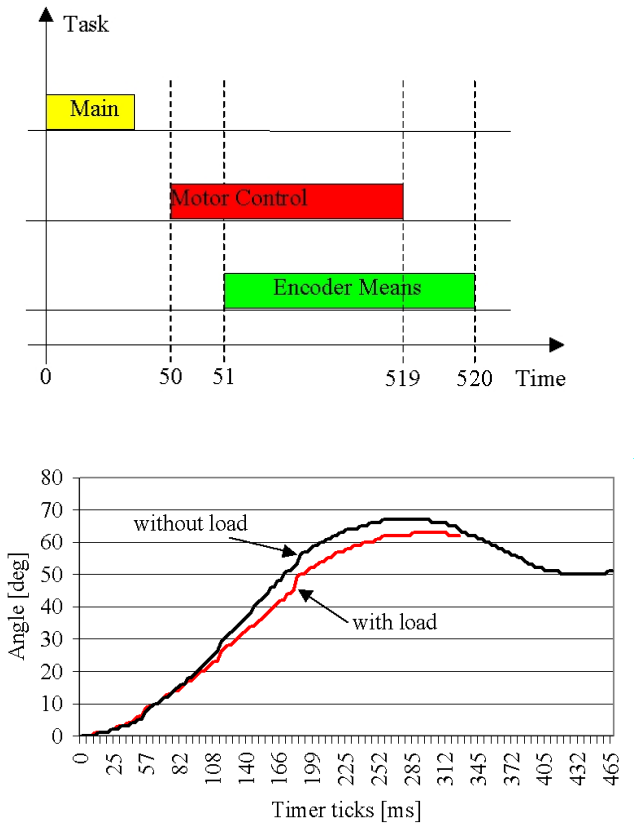

Fig. 2. Motor control in real-time with dual tasks: a) task schedule, b) measured rotor position

The control task was to set the desired angle of the motor axis (45 degrees) using 75% of the total motor power and setting the PID controller parameters. The realized experiments and logged data with a time resolution of 1ms show how the particular tasks are executed. The tasks time diagrams (Fig. 2a) show the execution sequence and confirm the data log time slots. The motor position time diagrams demonstrate the difference in the PID control quality due to the motor load. The steady-state error shows the regulation mismatch and the length of the logged data sequence points the time when the control task has been finished.

The realized testing program written in NXC shows its possibilities for real-time control education. Working with motor control problem students can analyze performance of the built-in, firmware digital PID controller or prepare a custom version of control algorithm. Running two (or more) tasks in the concurrent mode and logging time slots, the scheduling and tasks execution subroutines can be traced. The running tasks can be analyzed in the case of start-up, progress and terminate action.

V. Conclusions

The idea of open hardware and software solutions for LEGO® MINDSTORMS® NXT offers an unlimited number of application in the field of control education. If focusing of the real-time aspects, the text-based languages are required to demonstrate a full control over the executed code.

Analyzing several solutions it seems that the NXC and nxtOSEK are the best candidates to support real-time control experiments. The simplicity of NXC allows to start immediately with programming, while to start nxtOSEK a number of software replacements is required.

At this moment there are some limitations of Bluetooth communication using nxtOSEK, but hopefully this task will be extended soon. The presented simple example shows that it not so easy to obtain a precise and perfect results of the low level task. One can enjoy with working robot, but the important question facing the students is: does we have a full control providing a repetitive behavior of the robot?

Acknowledgment

The authors gratefully acknowledge the initial work done by their students K. Balwierz, D. Bieniek J. Blicharski and Ł. Łucek during the first stage of the project.

References

[1] Bassam A. Hussein, Kjetil Nyseth , "A method for learning in project management, 'Learning by projects' ", *9th International Workshop on Experimental Interactive Learning in Industrial Management,* "New Approaches on Learning, Studying and Teaching", Espoo, Finland, June 5-7, 2005

[2] Grega W., Piłat A., "Platforms for Laboratory Experiments: Low-cost Kits vs. Dedicated Trainer"s, *Proceedings of 18th EAEEIE Conference,* Praha July 2-4, ISBN 978-80-01-03745-4

[3] Grega W., Kornecki A., Sveda M., and Thiriet J.M., "Developing Interdisciplinary and Multinational Software Engineering Curriculum", *Proceedings of the ICEE'07,* Coimbra, Portugal, Sep. 3-7, 2007

[4] Gawthrop P. J., McGookin E., "A LEGO®-Based Control Experiment", *IEEE Control Systems Magazine,* v.24, October 2004, pp. 43-56

[5] Wang E. L., LaCombe J., and Rogers C., "Using LEGO® Bricks to Conduct Engineering Experiments," *Proceedings of the ASEE Annual conference and exhibition,* session 2756, 2004

[6] LEGO firmware, http://mindstorms.lego.com

[7] URBI, www.urbiforge.com, 2008

[8] http://www.teamhassenplug.org/NXT/NXTSoftware.html, 2008

[9] LEGO® MINDSTORMS® Hardware Developer Kit (HDK), http://mindstorms.lego.com, LEGO Group, 2006

[10] LEGO® MINDSTORMS® Software Developer Kit (HDK), http://mindstorms.lego.com, LEGO Group 2006

[11] Embedded Coder Robot NXT Demo, http://www.mathworks.com, 2006

[12] NXTway-GS (Self-Balancing Two-Wheeled Robot) Controller Design, http://www.mathworks.com, 2008

[13] nxtOSEK, http://lejos-osek.sourceforge.net, 2008

[14] leJOS, http://lejos.sourceforge.net, 2008

[15] OSEK, http://portal.osek-vdx.org, 1993

[16] TOPPERS Project, http://www.toppers.jp/en/index.html, 2003

[17] Hansen J., "Not eXactly C (NXC) Programmer's Guide", 2007