Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-6 (2016) 010–015

Teaching real-time programming using mobile robots ⋆

Carlos Rodríguez * **José L. Guzmán** ** **Manuel Berenguel** ** **Sebastián Dormido** *

* *Department of Computer Science and Automatic Control, UNED, Madrid, 28040 Spain (e-mail: carlos@bec.uned.es).*
** *Department of Informatics, University of Almería, CIESOL-ceiA3, Almería, 04120 Spain (e-mail: joseluis.guzman@ual.es, beren@ual.es)*

Abstract: Teaching theoretical concepts of real-time systems may be a challenging task without real hardware support. This paper discusses the use of the LEGO MINDSTORMS NXT mobile robots for teaching real-time programming to bachelor's students. The stable real-time control of a segway-like robot with a PID controller is used as case study to demonstrate the teaching methodology. Ada was used as real-time programming language. In general, very positive reviews were collected from the students after the course.

Keywords: Teaching, Real time, Mobile robots, PID control, Ada tasking programs

1. INTRODUCTION

Real-time programming encapsulates many abstract concepts that may be difficult to properly transmit to students. The physical feedback obtained with the use of real hardware may help to strengthen the most remarkable concepts and to motivate the students (Shakouri et al., 2013; Sobota et al., 2013). Over the last two decades, open-source hardware and software tools are emerging to provide an inexpensive solution to enhance control education.

The LEGO MINDSTORMS NXT (NXT) is an introductory robotics kit that allows students to rapidly develop real control applications making it an adequate complement even for courses which last few months. These applications usually require a specified time rate to perform each control loop step:

- Reading the controlled variables from the sensors.
- Computing the control actions.
- Sending the control signals to the actuators.

Thus, since real-time constraints must be generally met, software tools with integrated real-time features are desirable (Bradley et al., 2011). LEGO MINDSTORMS NXT has been already used within the framework of real-time programming and control mainly for robot competitions (Gómez-de Gabriel et al., 2011; Chaos et al., 2013; Grandi et al., 2014). They are also used for teaching concepts related to data acquisition, control systems engineering and real-time systems (Cruz-Martín et al., 2012), where NXC programing language is used for programming purposes. In (Iversen et al., 2000), a method for automatic verification of real-time control programs running on LEGO RCX bricks is performed, using the verification tool UPPAAL. In (Grega and Pilat, 2008) LEGO MINDSTORMS NXT has been used within a learning by doing project in a real-time subject also using NXC programming language and the Java-based firmware LeJOS as a tool to teach how to develop structured software. A rescue robot project is proposed where a number of real-time control systems can be analyzed, requiring planning, cooperation in teams, extensive literature survey and software design. References therein are also interesting and a table including programming tools is included. A similar approach using the same tools can be found in (Grandi et al., 2014), which also includes a list of interesting references. The approach followed in this paper is based on Ada programming language.

Ada is a high-level computer programming language targeted at embedded and real-time systems. It integrates concurrency and real-time features, e.g. tasks, synchronous message passing and protected objects, making it an ideal language for real-time programming courses. Together with the Ravenscar profile — a subset of the Ada tasking features designed for safety-critical hard real-time computing — makes Ada the ideal language for the NXT (Burns et al., 2004).

In this paper we present the results obtained by a group of students of real-time systems using mobile robots. The stability control of a segway-like NXT robot in Ada is presented as the case study. A PI controller with anti-windup mechanism and dead-zone avoiding strategy is used as the feedback controller. Furthermore, the robot features control parameters retuning at runtime.

The remainder of the paper is organized as follows. Section 2 describes the hardware characteristics and the development kit. The stability control of a segway-like robot is presented in Section 3. Finally, Section 4 conducts the conclusions of the work.

⋆ This work has been partially funded by the following projects: DPI2014-55932-C2-1-R and DPI2014-56364-C2-1-R (financed by the Spanish Ministry of Science and Innovation and EU-ERDF funds)

10.1016/j.ifacol.2016.07.145

2. MATERIALS AND METHODS

2.1 Teaching framework

Real-time systems lectures are taken by the students in the third course of the degree of computer science engineering at the University of Almería (Spain). This degree is a multidiscipline career which offers its graduates the adequate training to approach computer problems from different spheres of knowledge, including automatic control. The main objectives of the real-time systems lectures are:

- To expose the concepts of real-time systems in the industrial scope.
- To study time-based resource scheduling within a multitask framework.
- To develop software applications including analysis, design and coding in a real-time environment.
- To apply the acquired knowledge to physical real-time systems.

To fulfill all of the aforementioned objectives, the following teaching methodology is followed. During the first month the students take theoretical lessons that introduce the fundamental real-time aspects. Furthermore, they are also asked to solve some exercises individually. Next months, these lessons are complemented with Ada programming practical classes. Finally, the students must use their theoretical, practical, and teamwork skills in a real case study during the last month.

2.2 Hardware specifications

The NXT starter kit contains a programmable controller (also known as Intelligent Brick), several sensors, some actuators, and other mechanical pieces such as wheels. This Brick, that is depicted in Fig. 1, features a 32-bit ARMv3 main processor with 64 KB of RAM and 256 KB of Flash memory. Furthermore, it also has three bidirectional output ports to connect actuators, e.g. electrical motors, four input ports that support both analog and digital sensors, e.g. bump and sound sensors, and an 8-bit AVR co-processor that handles low-level tasks, e.g. the generation of PWM signals and the A/D conversion. Interacting with the Brick at runtime is also possible using the four rubber buttons and a 100 x 64 pixel LCD display. For schematics and further information refer to the LEGO MINDSTORMS website (Lego, 2016).

All the steps that are required to execute a working real-time Ada application in the NXT are described next.

2.3 Software development

AdaCore developed NXT drivers in Ada and provided periodic major updates in the last years. However, API documentation is not supplied with the drivers and it is convenient to revise the code during the development. AdaCore has also ported the GNAT compiler toolchain — compiler, assembler and linker — to the unsupported ARM architecture by using part of the LEON-based Open Ravenscar Real-Time Kernel (ORK+) developed by a team of the Department of Telematics Engineering of

Fig. 1. LEGO MINDSTORMS NXT programmable controller

the Technical University of Madrid (DIT/UPM) (Bradley et al., 2011).

With the aforementioned tools it is possible to execute Ada code in the NXT by following the next steps:

(1) Adding the Ada driver libraries within the software project and importing (at least) the **NXT.AVR** package which is always required even if its functions are not used.
(2) Generating a GNU *make* script (**Makefile.inc**) to compile the run-time libraries and builds the binary image.
(3) Completely removing the NXT's original firmware for the main processor and replacing it by a binary image of the Ada application that is executed from RAM. This is achieved using the default Boot Program (SAM-BA Boot Assistant) located in the first two sectors of the Flash memory while the NXT is connected through the USB port.

In what follows, we present the results obtained by a group of students that developed a real-time application to control a segway-like NXT robot.

3. RESULTS

A segway is a two-wheeled, self-balancing vehicle produced by Segway Inc. The device is kept upright when powered on using the electric motors in its base. In Fig. 2, the proposed segway-like NXT robot is shown. It consists of a main controller, two sensors and two actuators:

- The *ultrasonic sensor* is used to warn about possible head-on collisions.
- A *light sensor* gives an estimation of the angle that forms the robot and the surface normal. This measure is only slightly reliable under the specified conditions: An appropriate relative position of the robot with respect to the ambiental light and a clean homogeneous surface are required.
- The *wheel motors* are used for control purposes to maintain stability. Both motors are physically connected by a mechanical piece, thus both wheels move at the same speed [1].

[1] In a real Segway vehicle, the rider can go forward or backward by shifting its weight forward or backward on the platform.