Royal Institute of Technology

PROGRAMMING A ROBOT

INTRODUCTION TO COMPUTER PROGRAMMING WITH LEGO MINDSTORM - LAB REPORT

Simone Stefani
sstefani@kth.se

Introduction to Computer Studies (II0310)
25 August 2015

ABSTRACT

This report is written with the purpose of documenting a lab activity conducted as part of the Introduction to Computer Studies course. Such activity has been organised in order to help new students understanding programming methodology.

The students are asked to modify a file with NXC code in order to program a LEGO robot to follow a black line on the floor, stop when it hits an obstacle and display some text on the screen. The lab is carried out in pairs.

INDEX

1. INTRODUCTION

This report has been written with the purpose of documenting the work done during the lab activity concerning the LEGO MINDSTORM robots.

1.1 Background

The students attending the Bachelor in ICT are often required to read write and understand programming code. To help them to make the first steps in the world of programming, KTH offers an introductory course in computer science. This includes a lab activity to help them to familiarise with the programming process.

1.2 Purpose and objectives

The main purpose of the lab activity was to help students to gain a basic knowledge of the programming process, from installing the correct software to debugging the code. To successfully finish the lab, the student had to program a LEGO MINDSTORM robot to follow a black curved line on the ground and to correctly display a piece of text. The starting point was a program containing mistakes and useless code and the goal consisted in fixing it in order to make the robot accomplish his task.

2. EXECUTION

The lab activity was carried out in pairs. I worked with Erkki Kanninen. One student sat at the keyboard and operated on the code, while the second student, the "observer", read and checked the code. Often the roles were switched.

The robot used had two wheels connected to servo-motors independently controlled, a light sensor pointed to the ground and two touch sensors pointed frontward.

The first step was downloading and installing the required software, namely the programming IDE for LEGO robots "Bricx Command Center" and the drivers to connect the LEGO NXT Brick to the computer. The corrupt file of the program linefollower.nxc was also downloaded and opened in the IDE. After the connection between the and the IDE was established Erkki and I started reading the code contained in the file linefollower.nxc.

The first important step was to understand the purpose of the program: follow a black line using the light sensor and when the touch sensors detect an obstacle, stop the motors, play a tune and display the names of the students working on it on the screen. Then we started modifying the code. After few correction the robot was tested to evaluate the improvements and spot the problems.

This process was repeated until the robot succeeded in the task.

3. Results

After several attempts and code correction the robot succeeded in following the black line on the floor in both direction, stopping at the end and correctly displaying the lines of text on the screen.

Here are listed the correction to the program:

line 2: **`SpeedSlow`** value changed from 80 to 40

line 3: **`SpeedFast`** value changed from 100 to 60

lines 34-36: values 1 and 2 in the array **`groupMembers[]`** replaced with strings **“Erkki”** and **“Simone”**, array type changed from int to string

line 46: code inside loop changed from **`TextOut(0,(LCD_LINE2-(8*i-16)),names[i]);`** to **`TextOut(0,(LCD_LINE2-(8*i)),names[i]);`**

lines 50-55: **`dance()`** task commented out because useless

line 76: light sensor port changed from **`IN_1`** to **`IN_3`**

line 92: code changed from **`OnFwd(OUT_A, SpeedSlow);`** to **`OnFwd(OUT_A, SpeedFast);`**

line 100: code changed from **`OnFwd(OUT_B, SpeedFast); to OnFwd(OUT_B, SpeedSlow);`**

lines 108-116: order of tasks calls rearranged. First **`SetSensor`**, then **`OnFwd`** and finally **`Precedes`**; **`dance`** task call commented out because useless.

4. Analysis

The lab has been generally successful. The goal was completely achieved in a short amount of time. While the problems with the display and the touch sensors were solved in few minutes it took a longer period of time to properly program the robot to follow the line.

The difficulties were both logical and practical. On one hand was difficult to figure out what kind of actions and responses would have made the robot follow the line; on the other we had to learn how to implement these actions in the code and then into the hardware.

The program is organised in five tasks:
`main`, ***`printNameToScreen`***, ***`readTouchSensors`***, ***`readLightSensor`*** and ***`followLine`***. The task ***`main`*** calls the tasks ***`readTouchSensors`*** and ***`followLine`***, which includes ***`readLightSensor`***. The two tasks run in a loop (while) that is interrupted (break) only if a touch sensor is pressed. In this case the robot stops moving and the task ***`printNameToScreen`*** is called; the names contained in the array ***`groupMembers[]`*** are displayed on the screen.

5. Discussion

The main purpose of this lab was to help students to familiarise with the programming process and tools. This goal was completely fulfilled. We had also the opportunity to try the process of debugging which will be extremely important for the future courses in programming and electronics.

The language NXC resulted very easy even if I had a very little prior knowledge of programming languages. The activity has been generally informative and entertaining.

References

The code used as starting point for the lab is in the file linefollower.nxc. The instructions and reference manual necessary for the lab have been provided to us through the KTH Bilda webpage.