KTH ROYAL INSTITUTE OF TECHNOLOGY

Lego Lab

Programming a lego robot to follow a line and to display names of group members when reaching a wall

Katariina Martikainen

3.9.2015
martikai@kth.se
II0310   Introduction to Computer Studies

Abstract

The NXC programming language is C-based programming language used for programming LEGO Mindstorm robots. In order to understand C language it is good to start with easy practises with NXC languages. The lego robot lab during the course “Introduction to Computer Studies” provided easy practical tasks which helped the students to understand the logic behind the NXC language.This report has been written based on that lego robot lab and it describes the different procedures during the lab. Our tasks in the lab were to change the code of the lego robot’s programme so that the robot would follow a black half circle on the floor and display group member names on its screen when hitting the wall. We got the wanted results by altering certain parts of the code. Code changes are shown below on a page 5. Understanding the basic idea of programming with NXC is a good introduction for future programming studies and will make it easier to start improving one’s skills since the subject is not completely new anymore after the Lego Robot lab.

Table of contents

1. Introduction

This report is part of a course "Introduction to Computer studies". Its main purposes are practising writing scientific reports and describing what happened during the lego lab. If one wants to go back to what happened in the lego lab, the easiest way to do that is to study the report.

1.1 Background

We carried out the lego lab in order to understand the logic in programming and observe how the smallest changes in the code change the behaviour of the robot. It is very important to understand the idea of programming in order to continue one's studies in the field of ICT. The lab was carried out in order to introduce students to NXC, one of the programming languages and to create a good foundation which will help students to deepen their knowledge in programming.

1.2 Purpose and objectives

The purpose of the lego lab was to fix a code given to us. We had to change the code so that the lego robot would follow a black half circular line on the floor. When the robot reached the classroom's wall and the touch sensors got pushed in by the wall, the robot was supposed to display the names of the group members on its screen.

In order to finish the Lego robot lab we had to finish the following tasks:
-download the programme BricxCC and a document called "linefollower",which contained the code we were supposed to fix
-make the robot follow a black half circular line on the floor
-display names of group members on its screen when reaching the wall
-write down the code changes and the corresponding row numbers

2. Execution

We worked in pairs of two when executing the lab. First we downloaded the programme "BricxCC" and the document "linefollower" which contained the faulty code. Then we went through the code and tried to understand how each part of the code would affect the robot. We also tested how the robot would move with the original code. With the original code the robot turned a bit to one side and went straight for 3 seconds. After that it started to spin around itself.

After seeing how the robot acted with the original faulty code we tried to see which parts of the code needed changing. One of us changed the code and the other one marked the changes down to a piece of paper. When we were done editing we compiled the altered program by pressing a blue down pointing arrow. After this we could execute the program in the lego robot. To do this we went to the front of the classroom and played the program in the robot. By observing the behaviour of the robot we tried to figure out what kind of changes we would have to do next. We kept doing this until we had successfully fixed the code so that the behaviour of the robot would fulfil the requirements of completing the lab.

3. Results

We could make the robot do what we wanted by changing nine rows of code.Here are the changes in code.

Row number	Original code	Changed code	comments
2	#define SpeedSlow 80	#define SpeedSlow 70	We increased the difference of speed of the wheels so that the robot would turn more sharply. This was not a necessary change when it came to fulfilling the requirements for passing the lab.
34	int groupMembers [] = {	string groupMembers[] = {	Different variables can store different kind of data. We changed “int” to “string” because “string” stores phrases and “int” numeric values. Our names can be stored in a “string” but not in an “int”.
35	1	“Jiho Moon”	Here we wrote the names of our group members.
36	2	“Katariina M”	Here we wrote the names of our group members.
46	TextOut(0, (LCD_LINE2 - (8*i-16)), names[i]);	TextOut(0, (LCD_LINE4-(8*i-16)),name[i]);	We changed the LINE2 to LINE4 because that dropped our names 2 rows down on the lego robot’s display, which made them appear on the screen.

Row number	Original code	Changed code	comments
76	lightIntensity = SensorRaw(IN_1);	lightIntensity = SensorRaw(IN_3);	The cord from the light sensors of the lego robot went to the interface number 3, not to number 1 in the robot so we had to change it in the code.
94	OnFwd(OUT_A,Speed Slow);	OnFwd(OUT_A,Speed Fast);	If the robot moves from the black tape to the light floor we want that motor A goes fast and B motor goes slow. This causes the robot to turn to motor B's direction and back to the black tape.
98	OnFwd(OUT_B, SpeedFast);	OnFwd(OUT_B,Speed Slow);	The commands on this row happen if the robot is on the light floor. In this case we want the motor B go slow and motor A go fast so that the robot turns to motor B's direction and back to the black tape. When the robot went to the black tape we wanted in to do the opposite than on top. In that case we wanted the motor B go fast and motor A go slow. Then the robot would turn to motor A's direction and off the black tape.
115	dance();		We had to remove the command "dance" since it caused the robot to spin around itself and that was useless for our task.

4. Analysis

It was quite a hard task to understand the code without any earlier experience in programming. Despite of the preparation done before the lab (reading "Programming LEGO NXT Robots using NXC.pdf")

By going through the code little by little with our pairs and testing the code we eventually succeeded in our task.

The code in “linefollower” contains a task noted by “main”. This task tells the robot to move forward and study the surrounding with its light and touch sensors. If the sensors detect a condition defined in the code, the program executes certain parts of the code. For example if the light sensor detects certain light intensity the speed of the motors which move the wheels will change. This is the main structure of the code and working principle of the program.

In my opinion the most important things which needed realisation were adding the names of the team members to the string “groupMembers” and removing the command “dance”.

It was also important to understand how the light intensity detected by the light sensors should affect on the speed of the wheels so that the robot would follow a line. For example if the robot wandered off the black tape the light sensors detected the highest threshold for the amount of light reflected from the floor and that caused the speed in the motors change so that other wheel would go faster than the other and the robot would turn back to the black tape. In this way the robot was constantly changing its movement according to the changes in light intensity. This caused the robot to waggle from the black tape to the light area and back. That way it kept itself on the line and moved forward in a half circle.

It was also important to understand which kind of data different kind of variables could store and how those variables function. This knowledge was needed in completing the task of getting the robot display the names of the group members.

5. Discussion

The biggest problems were understanding how different parts of the code affect the behaviour of the robot and understanding the logic behind the commands. It took some time to understand what the commands of the code corresponded in real life. For example that the condition “if(lightIntensity> BotThreshold)…” happens when the robot drives off the black tape.

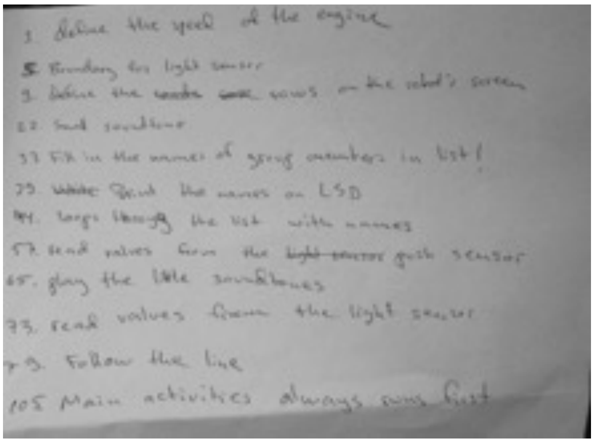
Eventually the lab taught us how to interpret simple NXC language and how the smallest changed in the code can lead to errors or unwanted results. I learned that the code can be altered in many different ways in order to get the desired results. Some ways worked better than others. I feel like the lego lab was not enough to make me skilled enough to deeply understand the NXC language or to use it in programming myself but it worked as a good introduction to the basic idea of programming. I had no earlier knowledge in programming so the lego lab showed me what it is all about. It convinced me that I want to learn more about programming.
I think that the software "BricxCC" was simple and easy to use. The language NXC was manageable yet challenging for beginners. Nevertheless I feel like I could have used some more advice on how to get started, how the software worked etc.

References

I have used the original code from "linefollower", slides form the lecture 3 and the lab instructions when writing this lab report.

Appendices

Here are the translations of some parts of the code since they were written in Swedish in the code. We started by translating the code into english.

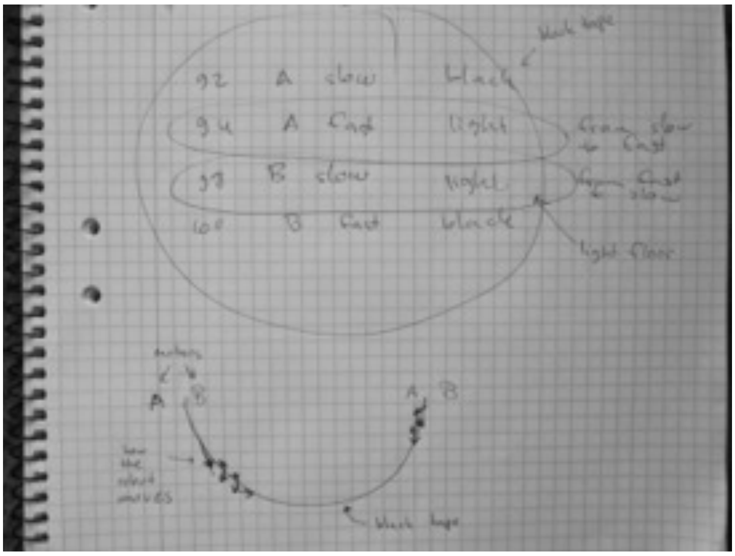


1. define the speed of the engine
5. Boundary for light sensor
9. Define the ~~[illegible]~~ ~~[illegible]~~ rows on the robot's screen
22. Send soundtone
37. Fill in the names of group members in list!
45. ~~White~~ Print the names on LSD
44. loops through the list with names
57. read values from the ~~light sensor~~ push sensor
65. play the little soundtones
73. read values from the light sensor
89. Follow the line
105 Main activities always runs first

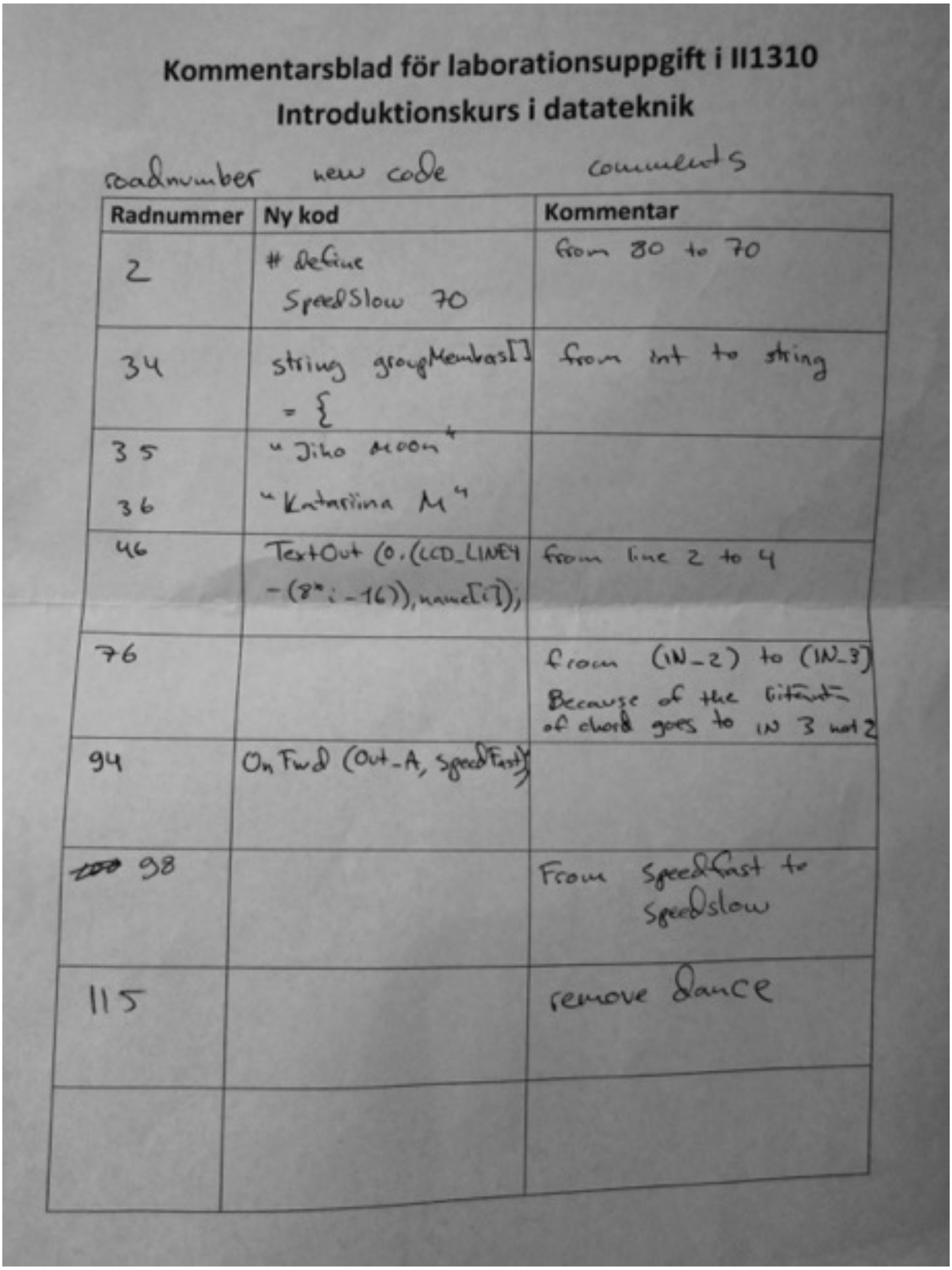
Here are some sketches I did during the lab. I tried to understand what speed which motor should have depending on whether the robot was on the black line or on the light floor.

I also tried to sketch the movement of the robot according to these commands given in the code.

92,94,98 and 100 stand for the row numbers in the code.



Here are the notes from the lego lab.



Kommentarsblad för laborationsuppgift i II1310
Introduktionskurs i datateknik

roadnumber new code comments

Radnummer	Ny kod	Kommentar
2	# Define SpeedSlow 70	from 80 to 70
34	string groupMembers[] = {	from int to string
35 36	"Jiho Moon" "Katariina M"	
46	TextOut (0,(LCD_LINE4 -(8*i-16)),name[i]);	from line 2 to 4
76		from (IN_2) to (IN_3) Because of the [illegible] of chord goes to IN 3 not 2
94	OnFwd (Out_A, speedFast);	
98		From speedfast to speedslow
115		remove dance