Assembly Programming using Simple Lego Mindstorms RCX Robots

Wen-Jung Hsin, Chris Baker, Joshua Branscum, Eric Bunce,
Aris Czamanske, Nick Kreeger, Rodrigo Neri, Philip Nganga, Amanda Warren
Information and Computer Science
Park University

***Abstract*— During 2007 Spring semester, the students in Computer Architecture course were introduced to Lego Mindstorms RCX assembly programming. During the semester, they produced a few simple playable games using Lego Mindstorms RCX robots. This paper describes the project motivation, student background, student projects, and experience gained by doing the projects.**

I. Introduction

In the current Computer Science curriculum at Park University, the Computer Architecture course is the only course with the chance of teaching assembly programming, without adding an extra course load to the curriculum. In Spring 2007 the Computer Architecture course students were introduced to Lego Mindstorms RCX assembly programming. During the semester, they produced a few simple Lego Mindstorms RCX playable games, designed and developed using the RCX assembly program language. All these games used only a RCX brick, 3 touch sensors, 3 LED lights, and 3 wires. Figure 1 shows the robot design. This simple design is used in the *Simon Says* project provided by Lego Mindstorms in Computer Science Education (LMICSE) [2], an NSF funded project. This paper describes the project motivation in section II, student background in section III, Simon Says project in section IV, student projects in section V, and experiences gained in section VI.

II. Project Motivation

Lego Mindstorms RCX robots were chosen in the Computer Architecture course for several reasons.

- The Information and Computer Science department at Park University has several Lego Mindstorms RCX robotic kits readily available as a result of conducting many girls scout robotic camps in the past several years.
- The instructor of this course attended Lego Mindstorms in Computer Science Education (LMICSE) [2], an NSF-funded summer robotic workshop for faculty. Through the workshop, she learned the simple Lego Mindstorms RCX robot design, the RCX assembly programming language, and assembly programming environment. By showing the students RCX robots and RCX assembly, she wishes to utilize the Lego Mindstorms RCX robots in ways other than that in the girls scout robotic camps.
- By using RCX robots, the students can obtain touchable results, as opposed to the result shown on the computer screen using other assembly programming environment such as SPIM Simulator for MIPS assembly programming language [3].
- Lastly and most importantly, by writing a few programs in RCX assembly, the students gain concrete experiene in using Complex Instruction Set Computer (CISC) assembly language. When Reduced Instruction Set Computer (RISC) was introduced later in the course, the projects in CISC using RCX helped students understand the difference between the CISC assembly and RISC assembly (such as MIPS) clearly and easily.

III. Student Background

At Park University, the students taking Computer Architecture course have no prior assembly programming experience. They have taken higher programming language courses using Java and C++ before coming to this course.

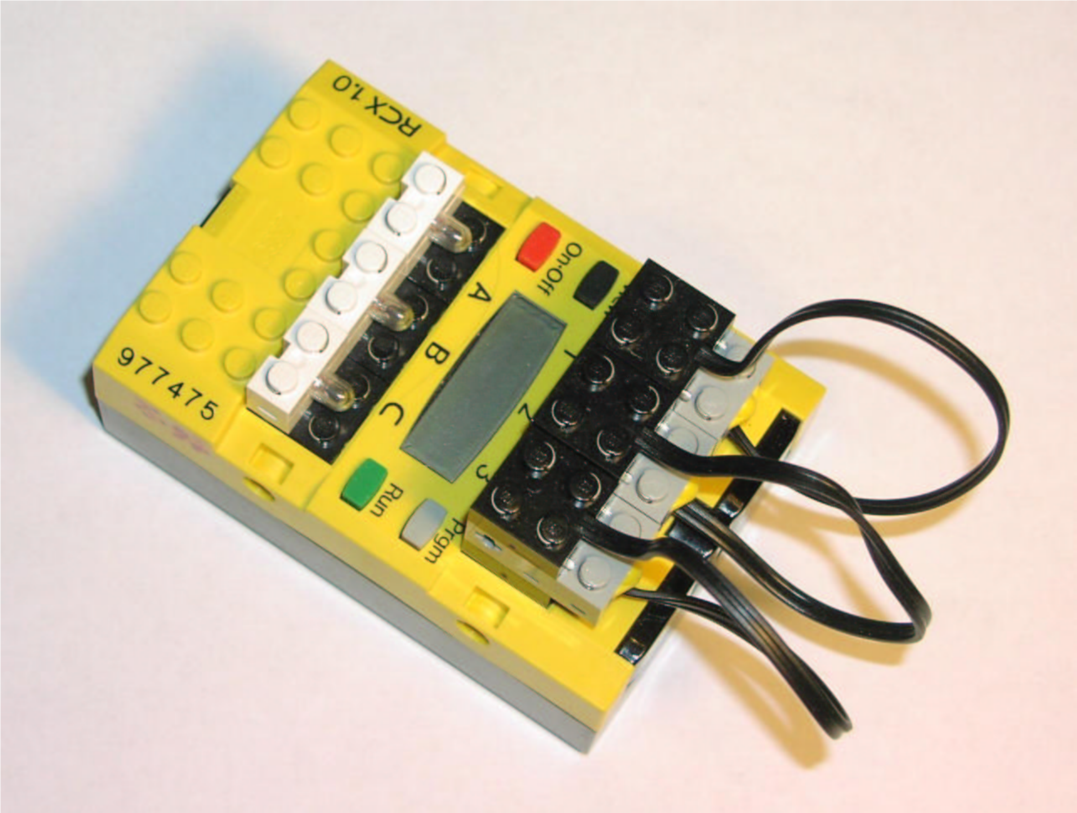

Fig. 1. A Simple Lego Mindstorms RCX robot

Lego Mindstorms RCX has its own assembly programming syntax and structure, and its own programming environment (i.e., Script Editor). Lego Mindstorms provides a specification on the Lego Mindstorms RCX 2.0 Firmware Command Overview [1]. The students were introduced to this new programming structure and environment in class. They then spent time on their own in understanding the command syntax and structure in more details.

IV. Simon Say Project

The *Simon Says* project is one of the two computer architecture laboratory materials provided by LMICSE [2]. Simon Says is a traditional follow-the-leader game. In this project, a RCX brick is the leader (i.e., Simon). A player of the RCX robot is the follower.

As depicted in Figure 1, a RCX brick has one LCD display, three input interfaces (labeled as 1, 2, and 3), and three output interfaces (labeled as A, B, and C). The Simon Says robot uses one RCX brick, 3 input touch sensors, 3 output LED lights, and 3 wires to connect inputs to the brick.

In this project, to output a value, the RCX brick displays a value n (where $1 \leq n \leq 3$) on the LCD display, and turns on the corresponding output LED light. Specifically, value 1 corresponds to output interface labeled A, 2 to B, and 3 to C.

When the RCX brick outputs a value, the user is expected to press the correspondent touch sensor. That is, output value 1 corresponds to input touch sensor at label 1, output value 2 to touch sensor at label 2, and output value 3 to touch sensor at label 3.

The user wins a game if he/she is able to follow m RCX brick's values consecutively, where m is a positive integer such as 12. As soon as the user makes a mistake by not touching an appropriate sensor, the game is over and the user loses the game.

LMICSE [2] provides two designs for the Simon Says project. In the first design, to simplify the coding complexity, the pattern outputted by a RCX brick is fixed. So anytime when a game starts, it outputs the same number pattern in which a pause is inserted after each number, awaiting for the user's input. As a more realistic variation to the first design, the second design has the RCX robot output a random pattern each time a game starts.

V. Student Projects

After the introduction to the Lego Mindstorms RCX assembly programming and programming environment, and the Simon Say projects [2], students were able to design their own simple games in which the only requirement is to write the programs using RCX assembly language.

After some studies, all students chose to stay with the same hardware pieces as the Simon Say project, i.e., a RCX brick, 3 touch sensors, 3 LED lights, and 3 wires as in Figure 1.

Students produced simple games, including memory and slot-machine (lottery) games. The following subsections introduce these games.

A. Memory Games

As a variation of the Simon Say project, a robot flashes a series of lights in random order and displays corresponding values on the LED display. A user is to memorize the random pattern of m numbers (where m is a positive integer value) and repeat the same sequence by touching the corresponding sensors to win the game. Thus, there is no pause waiting for the user's inputs inbetween numbers. This is different from the Simon Say project described in section IV in which an output and input pair operates one value at a time. That is, in Simon Say project, a RCX brick displays one value and then waits for the user's input, and when the user enters the same value correctly, the process continues.

Another memory game is similar to a game with lots of random colors in which a user is expected to remember the colors to win the game. Here, for our simple robot, a color corresponds to a number. The robot displays a sequence of random numbers with increasing length, i.e., length 1, length 2, length 3, etc. Essentially, each successful input from the user increases the difficulty level by adding one more random number. Specifically, the game starts by displaying one random number (between 1 and 3, inclusively). If a user repeats the number correctly, the robot displays two random numbers. The game continues until a user inputs an incorrect sequence in which case the user loses the game.

B. Slot Machine (Lottery) Games

In a traditional slot machine game (i.e., a typical game played in casinos), a player starts a game by pulling a lever or pushing a button. After some operation, the machine displays a set of numbers on the output display. If all numbers match a certain pattern (e.g., for example, all numbers are the same), then the player wins a prize.

In our game, a player pushes a button to signal the start of a game. If all three output lights are flashed, then the player wins the game. Otherwise, the player loses the game.

As an idea and a variation to the traditional slot machine, one can have a user select a set of input numbers. The robot then generates a set of output numbers. If the player's input numbers match the robot's output numbers, then the player wins the game. This is more akin to the lottery game such as Powerball in the market currently.

VI. Experience Gained

The purpose of the Simon Says project and the student projects is for the students to learn the RCX programming language, RCX programming environment, and environmental controls of the RCX brick. Overall, the students in the course learned the concepts and achieved this purpose.

As a side benefit in doing the RCX projects, since RCX is a CISC, the students have concrete experience in using CISC assembly, which helps them understand the differences between CISC and RISC assemblies.

VII. Summary and Conclusion

This paper describes student projects in using simple Lego Mindstorms RCX robots. Additionally, it describes how this project comes about. The students learned the RCX programming language, RCX programming and control environment. Furthermore, learning the RCX assembly which is a CISC assembly helped the students understand the difference between between a CISC assembly and a RISC assembly.

References

[1] Specification on RCX 2.0 Firmware Command Overview. LEGO Mindstorms Software Developers' Kit (SDK).

[2] Scott Anderson, Frank Klassner, Pam Lawhead, Myles McNally Lego Mindstorms in Computer Science Education (LMICSE): A project funded by NSF grants 0088884 and 0306096. http://www.mcs.alma.edu/LMICSE/

[3] David Patterson and John Hennessy Computer Organization and Design. Morgan Kaufmann. 2004