

Learning C Language Using Robots

Seung Han, Kim*, and Jae Wook, Jeon**

School of Information and Computer Engineering, Sungkyunkwan University,
300 Chunchun-dong, Jangan-gu, Suwon, Gyeonggi 440-746, Korea

* (Tel : +82-31-290-7233; E-mail: bbobbo1623@hotmail.com)

** (Tel : +82-31-290-7129; E-mail: jwjeon@yurim.skku.ac.kr)

Abstract - Lego company created a set called Robotic Invention System as a kind of Mindstorm set. This system helps to understand the technology of both robot and programming language. It also improves creativeness by building and controlling the robot we make. This paper will introduce basic idea of controlling the RCX(Programmable Lego Block) in C language. Also, this paper will show different idea of teaching C language by using other types of robots.

Keywords: LEGO Mindstorm, RCX, C Language, Robot

1. INTRODUCTION

LEGO Mindstorms Robotics Invention System (RIS) is a programmable toy which contains assortment of blocks, pieces, sensors, motors, and most important a Robotic Command Explorer(RCX). RCX is a programmable block using infrared(IR) transmitter to communicate with personal computers. Lego provides a graphic user interface(GUI) application to control the RCX. Although a given application is very simple and powerful, it will be more useful if it could be controlled by popular languages like C. This means we can learn language C in an alternative way. Most of education on language C is held through personal computer. We write the source code with a text editor and use a compiler to run the program. Outputs are always through the monitor only. It is most traditional way to learn language but it is also very limited and fixed. With RCX, we can look at the program running in 3 dimension, which is more interesting compare to monitor based programs.

The basic idea to control RCX by language C is that by using a cross-compiler, users can write codes in American National Standard Institute C (ANSI C) and compile and debug it in the way we knew. Unfortunately, not all codes are compatible with RCX due to its characteristic of the structure. Yet, it flows the main structure of C language. Therefore, it is alternative way to teach C language efficiently.

In section II, more specific structure of LEGO Mindstorm RIS will be explained. The method of controlling the RCX using C language will be explained in section IV. Section V will mainly explained with the potential of learning C language with robots. The last section describes the conclusion and future plans toward this paper.

II. LEGO MINDSTORM ROBOTIC INVENTION SYSTEM

The standard LEGO Mindstorm comes with a Microsoft Windows software CD which provides GUI for programming. It is in structure of "jigsaw puzzle" pieces representing commands to control over the available actions by using respective sensors and hardwares. Supplied software is simple enough for children to understand and get attracted to it.

Most important part of the kit is called the RCX, the programmable LEGO brick. It is shown in Fig. 1. We have to notice that it is a brick. We can build robots based on the RCX or we can design a creative and unique robot depending on our

imagination. The RCX brain contains a microprocessor (Hitachi H8300) inside of a plastic case. There are three input ports and three output ports. Three output ports can attach motors and lamps, and three input ports allow light, touch, temperature and angle sensors. It has external liquid crystal display (LCD) can show five characters at one time. Four buttons are on the top for users to control the RCX. IR communication device is supported for users to communicate with personal computers using a universal serial bus (USB) port.

There are other important parts in the set. We have gears, pipes, blocks, and others to use them to make a robot. Most people should have seen and have been playing at least one LEGO model when they were young. It is not a big problem even though he or she did not have any chance to play with it before. LEGO is easy to learn. We would like to explain some special parts that are exclusively included in Mindstorm.

First of all, motors are included. The main purpose of motor is to make a robot to move. The basic set comes with two motors. We can create robots using the motor to move their hands, legs, wheels or even to shake hands. There is no limit in designing a robot. Two directly cabled motors will move exactly the same way. We can attach any motors to any ports as our wish.



Fig.1 LEGO Mindstorm RCX

Secondly, there is one light sensor in the set. Light sensor has two abilities for users to control, first is lamp and the other is intensity sensor. Lamp is often used to help the intensity sensor to have closer value. However it also disturbs the reading of intensity sensor. The light sensor is not a good device to measure the light. The light sensor gives value in numerical form from range around 20,000 to 65,000.

Third, there are two touch sensors included as basic set. Not like the light sensor, touch sensor returns value binary value, on/off status. It works like a door bell. By pressing the sensor the circuit is completed and the electricity flows through it. RCX is able to detect this flow to give appropriate value for programs to read. Motor and sensors are shown in Fig. 2.

III. LEGO MINDSTORM ROBOTIC INVENTION SYSTEM FOR TEACHING C LANGUAGE

Language is the mechanism through which communication is realized. Without language we would be isolated with our thoughts. Computer programming languages exist because humans have a need to communicate instructions to a computer. The computer however, understands a cryptic series of symbols representing switches in the on or off state. This is language called machine language and is too much detailed for users to understand it. It is almost impossible for users to fully understand the machine codes and write codes in machine language. As time flows, programmers made programming language to communicate with computer using appropriate compiler. Programming language for users are usually called high-level language. C language is one of the high-level languages that can communicate with computers[1].

The RIS has been used to teach programming in several countries such as the United States, Hungary, and Norway. Penn State Abington College uses the RCX in their introductory programming course for engineers. According to Patterson-McNeill, the greatest use of LEGO robots has been in engineering courses and AI courses. Iowa State uses RIS and Not Quite C (NQC) to teach K-12 students and their teachers. They also have courses for women in science and engineering that utilizes the RIS. North Carolina State has a science program for children 4 – 12 that included a LEGO amusement park. Tufts University has several courses in mechanical engineering that uses the RIS. Wellesley uses the RIS in a physics/computer science course. The RIS was used at Texas A&M University-Corpus Christi to teach an introductory course to programming. Although most of the students were teachers without programming experience and afraid of computers, the use of Robots was a success because the students could implement a programming concept like a decision or a loop and test it in a real application[2].

Two factors combined to seriously limit the use of robots in teaching programming languages. First was the high cost of robotic kits, and second was the lack of a framework for comprehensively integrating robots. These two factors lead college or educational departments to avoid the use of robots in teaching programming. Since the mid 1990s, several manufacturers have released standardized, low-cost robot platforms and LEGO RIS 2.0 is one of the kits developed. There are several reason to choose LEGO among the robots.

COST: A single LEGO Mindstorm kit with 750 construction pieces, sensors, and programmable hardware, costs around US \$200.

FLEXIBILITY: The LEGO Mindstorm kit supports sensors(touch, rotation & temperature), effectors(motors, lights, IR emitters), building blocks, and a programmable control unit that can serve as the basis for user to program.

STUDENT INTEREST: Many students have played with LEGO building blocks as children, therefore, they are intrigued with it. Not like PC based lecture, student will fall into it even though it is classroom work[3].

IV. CONTROLLING RCX WITH C LANGUAGE

The C language is fast becoming the language of choice for teaching engineering technology students. C is a useful language. C is general-purpose language. It can be a real tool for writing programs in a concise and efficient manner. More important to the beginning engineering technology student, C is a language that relates directly to the operating system of many UNIX based hardware components[4].

C language is a “low level” language among “high-level languages”. This means C language is close to machine or assembly language. C language has replaced the study of assembler language in many programs. It closely represents the same fine control associated with assembly language.

The software that replaces the RIS firmware to program in C is called “legOS[3].” It is a multitasking operating system running on the RCX using a C-cross-compiler running on a host computer to program. LegOS uses library system calls for use in a C program for the RCX. LegOS also comes with memory management, call stacks, threads and other high-level language features[5].

In order to use the C compiler we need several free software like CYGWIN, Bricx Command Center and BrickOS. The cross-compiler works in the Cygwin shell. Users can write code in Bricx Command Center and then compile it. BrickOS include the library files needed for compiling and coding[5-7].

Among many program languages, C language was chosen for this paper. The main reason is because it has general characteristic as a language. C language has many of the characteristics of a so-called lower-level language. Users are allowed to use features of the hardware as they are needed. RCX has restriction for large programs or programs which need numerous resource. In order to maximize the use of the RCX with programming, C language is most suitable.

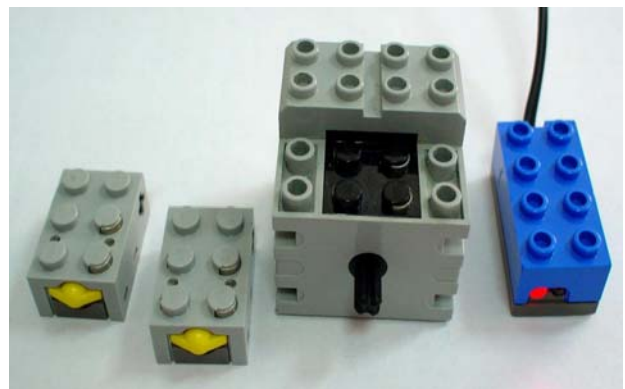


Fig. 2 Devices included in the set. (Motor & Sensors)

Although BrickOS is built to use C for coding, it does not perfectly match with ANSI-C codes. However keywords and structure are identical, so all users have to do is to learn exclusive code to control the RCX. We can control the motors, touch sensor, and light sensor. The RCX can make sounds as well and show text through its LCD panel. Here are some example codes for RCX.

There are three output ports on the RCX. A motor is connected to one of the three ports with cable to control. Code (1) means, motor connected to port A is set to speed of 255. Users can select the speed between the range of 0 to 255.

```
motor_a_speed(255); (1)
```

Code (2) is the code to control the direction of the motor. There are four choices forward(fwd), reverse(rev), brake(brake), and off(off). The difference between brake and off is quite obvious. Brake command will immediately stop the motor and off command will stop supplying the current to the motor so that the motor will stop gradually.

```
motor_a_dir(fwd); (2)
```

Code (3) is essential for the motor to run. If this command is not included, the motor will not move at all. The number written is milli-seconds users have to assign. Code (3) gives delay so that above codes are continued longer.

```
msleep(200); (3)
```

Above codes will run the motor attached to port A in the direction of forward at speed 255 for 200 milli-second. To control the other ports, we just have to change the letter "a" within the code to corresponding port like B or C. Table 1 shows a function includes above codes.

Table 1 Function to control motor connectd to Port A.

```
void setmotor_a()
{
    motor_a_speed(255);
    motor_a_dir(fwd);
    msleep(200);
}
```

Table 2 is the normal structure of coding. If we compare with standard ANSI-C codes, there are some differences. If we do not follow the structure, the code will not be compiled and gives error messages. This code is written to do the task with a Roverbot(Fig. 3).

Table 2 A complete code for Roverbot to have soft right turn.

```
#include <config.h>
#if defined (CONF_DMOTOR)
#include <conio.h>
#include <unistd.h>
#include <dmotor.h>

int main(int argc, char *argv[]) {

    motor_a_speed(255);
    motor_c_speed(150);
```

```
motor_a_dir(fwd);
motor_c_dir(fwd);
msleep(1000);
motor_a_dir(off);
motor_c_dir(off);

return 0;
}

#else
int main(int argc, char *argv[]) {
    return 0;
}
#endif
```

Most of the RCX related codes are in function type. For controlling the motor, code should be written in the main function. For RCX to make sounds, however, a song to play is made up of an array of these structures which is then handed to *dsound_play*. Table 3 is the one example of RCX to make sounds. Code (4) must be declared within the main function to make sounds. *PITCH* can be declared in range from A0 to A8. The alphabet shows the musical note and the according number is the octave. Note H is equivalent to American note B.

```
dsound_play(beep); (4)
```

Table 3 Structure for RCX to make sound

```
static const note_t beep[] = {
    {PITCH_H4, 1},
    {PITCH_H3, 1},
    {PITCH_END, 0}
};
```

The API is provided by the BrickOS. All the needed codes to control RCX in C codes are listed here. The header files needed for compilation is all listed here. Specific codes for each motors and sensors are introduced detail. List of all functions, variables, defines, enums, and typedefs are well organized. For more information on codes, refer to the API[8]. Table 4 shows some list of codes to control RCX.

Table 4 List of codes for RCX

Motor	
<i>motor_x_speed(num);</i>	<i>x={a,b,c} & num={0-255}</i>
<i>motor_x_dir(dir);</i>	<i>x={a,b,c} & dir={fwd,rev,brake,off}</i>
Touch Sensor	
<i>ds_active(TOUCH_x);</i>	<i>x={1, 2, 3}</i>
<i>ds_active(TOUCH_1);</i>	<i>ds_active(TOUCH_2);</i>
<i>(TOUCH_x!=y);</i>	<i>x={1, 2, 3}, y={0.1}</i>
Light Sensor	
<i>ds_active(&SENSOR_x);</i>	<i>x={1, 2, 3}</i>
<i>ds_active(&SENSOR_1);</i>	<i>ds_active(&SENSOR_2);</i>
<i>y = (SENSOR_x);</i>	<i>y = {float type variable} x={1, 2, 3}</i>
Beep	
<i>{PITCH_XI, T},</i>	<i>X={a~h}, I={1~7}, & T={1,2,4,8,16}</i>



Fig. 3 Roverbot

V. POTENTIAL OF LEARNING C LANGUAGE WITH ROBOTS

LEGO Mindstorm and related software provide alternative method to learn C language. This alternative method will be more effective to teach young students who did not learn any languages before. Even for college students, learning C language with LEGO will be marvelous. It will overcome the 2 dimensions environment and go to 3 dimensions.

Many beginners have hard time of learning language, since it is difficult to understand about the error message. The compiler inspects the code and gives suitable error messages. However, users are not able to understand about the meaning of it. Users have to know many of the library procedure for programming. This might be the obstacle for users to get close to coding. Even naming rules can make users get confused. Some keywords assigned for variables and functions should be learned. Users can easily abandon their will to learn programming.

Until today, most of C language courses were based on the text book and compilers. We wrote the code to a text editor and saved to a C file. The compiler will do the next action to show the output through the monitor. With the idea using Mindstorm , we can extend the limit of having the output to the monitor to moving robot. Mindstorm is a wonderful kit that provides necessary idea of robots. With this idea, we can build a special robot for education of C language only.

From very early age, children find movement captivating. Recent robot research is also examining robots aimed at teaching children about the world. This research includes classes and activities focused on teaching about robots, which was not easily taught with traditional computers.

VI. CONCLUSION

While the LEGO Mindstorm kit is still somewhat limited in coding C language perfectly, it has advantage to teach basic skills on coding to young programmers. Another merit of this kit is that it is inexpensive and it is durable. Mindstorm and C language, joined together, is ideal way to teach coding. Many computer science classes around the world actually use Mindstorm and even some artificial intelligence college courses do. Lego Mindstorm is not just a toy, we can create robots and also program it to work.

We are planning to open a class. The class is on teaching C language using LEGO Mindstorm. The expecting candidates are aged 12 to 15. The class will be prepared with personal computers and appropriate software to program the RCX. Currently, we expect the class to be held for total 40 hours in four weeks. We will check the ability, potential and interest of candidates toward programming. After the class is over we hope we can conclude with a blueprint of a robot for exclusively teaching C language.

REFERENCES

[1] Dr. Tri Nguyhen and Antoine F. Sajous, "Understanding C Through English Grammer," *Northcon/98 Conference Proceedings*, pp. 219-226, 21-23 October 1998.

[2] Mario A. Garcia and Holly Patterson-McNeill, "Learn How To Develop Software Using The Toy Lego Mindstorms," *32nd ASEE/IEEE Frontiers In Education Conference, Boston, MA, Session S4D*, pp. 7-10, 6-9 November 2002.

[3] Frank Klassner and Scott D. Anderson, "LEGO Mindstorms: Not Just For K-12 Anymore," *IEEE Robotics & Automation Magazine*, pp. 12-18, June 2003.

[4] John Z. McRee and Maurice Walworth, "Hardware Platforms For Teaching C to Engineering Technology Students", *Frontiers in Education Conference, 1991. Twenty-First Annual Conference. 'Engineering Education in a New World Order'*, pp. 753-757, 21-24 September 1991.

[5] <http://brickos.sourceforge.net>

[6] <http://www.cygwin.com>

[7] <http://briccc.sourceforge.net>

[8] <http://brickos.sourceforge.net/docs/APIs/html-c/>