Learning Computer Programming with Autonomous Robots

Shuji Kurebayashi[1], Toshiyuki Kamada[2], and Susumu Kanemune[3]

[1] Shizuoka University eskureb@ipc.shizuoka.ac.jp
[2] Aichi University of Education tkamada@auecc.aichi-edu.ac.jp
[3] Hitotsubashi University kanemune@acm.org

Abstract. This paper reports on a programming lesson using autonomous robots in junior high school. First, the design of the low cost circuit board for the lesson is described. The structure of a general programming language "Dolittle" which controls a robot is also explained. Then, we introduce lessons of manufacturing and controlling robots in "Information and Computer" area of "Technology and Home Economics" subject for students (from 14 to 15 years old). From the result of the lessons we found that (1) learning programming is "hard fun" for students and (2) robot programming is effective for students those who have difficulties in learning. We propose introduction of learning programming with autonomous robots to IT education of junior high school.

1 Introduction

In Japan, IT education at the junior high school level is performed in "Information and Computer" area of "Technology and Home Economics" subject. The primary goal of the area is that all of the students have the ability to operate computers. However, because there sometimes occur severe problems in our daily life caused by flaws in computer software, teaching only how to use computer is not sufficient. It is impossible to foster citizens with IT literacy who can think of the potential risk of the highly information-oriented society. The reason comes from the fact that learning only how to operate computers cannot give students the substantial knowledge of fundamental structures and mechanism of computers. Thus, we think it is crucial to teach computer programming in IT education and the most important point is to show that every computer program is created by humans [1, 2]. Normally, general programming languages are designed for IT specialists. Such language is too difficult and not suitable for children. Therefore, we have chosen the programming language "Dolittle"[3] for our educational practice because "Dolittle" is simple but has enough capacity to provide computer programs for children. The characteristic features of "Dolittle" are that (1) it has simple language syntax and that (2) it can use local language such as Japanese, Korean and English for instructions and identifiers including variable names. We also developed a low cost autonomous robot as teaching material and the course of IT education using this robot. Then, we implemented the course. The reason of choosing a robot is that we found learning through controlling tangible objects is more effective than learning with operating virtual objects on the screen. Consequently, we have realized our teaching material can enhance the motivation of students to learn programming and assist them in learning that a lot of electrical products in our society are controlled by software.

In this paper, the reasons of introducing robot manufacture to our lessons are considered. Then the autonomous robots that we developed are presented. Finally, the performed lessons of "Technology and Home Economics" subject in junior high school are reported.

2 Meaning of Learning Programming in Elementary and Secondary Education

2.1 Learning Programming is Learning "Manufacturing"

Hardware and software are both necessary for running computers. Software is constructed by humans as well as hardware.

2.2 The Effect of Robot Programming

Learning of computer science is essential for educating computer specialists. However, it's also important to teach both hardware and software to common students.

Instructions on computer hardware to common students are not easy. Therefore we thought that robot programming was effective for students. By experiencing robot control, students can learn programming from the viewpoints of both sides of hardware and software. Controlling real robots, students can see and understand each steps of the program. This provides a realistic feeling and motivation to students.

Thanks to robot programming experience, students start to figure out the motion of robots, think of algorithms, complete their programs and evaluate them. The skill of actual program development cycle is thus acquired. Students can learn hardware and software are both important.

2.3 Autonomous Robots are Desirable for Lessons

In order to raise educational effects, autonomous robots are desired. It is because many electric appliances whose structure we want students to understand are nowadays equipped with microcomputers to enable autonomous operation.

Most of current computer software is downloaded from a computer network before being installed. Because autonomous robots need to download their software from the host computer before operation, they are quite suitable for providing concepts of computer communication with network and embedded system. Furthermore, from the property that software is transferable, students can easily conceive that software is an entity independent of computer hardware. Given this perspective, we think autonomous robots are a very good teaching material to explain "programming is also a kind of making things" to students with persuasiveness.

2.4 The Educational Effect of Robot Manufacturing Experience

Students can see and look into the circuit board while building robots. They can observe the connection between sensors, motor drivers and a microcomputer. Students may not be able to understand the exact mechanism of the microcomputer or meaning of the circuit only by observing. Through handling electrical parts during wiring work, they come to understand the function of each part explained by a teacher; e.g. signal voltage for driving motors is output to signal pin of motor drivers; electric signals are sent from sensors and they are input to microcomputers etc. Then, they would understand the computer system deeper than before. They become aware that the most important part is CPU and that the computer system is not composed of only visible devices such as keyboards, mice or displays. Once this knowledge is obtained, students feel a sense of accomplishment that they completed the robot work all by themselves. We think this will encourage the interest and willingness to learn programming and increase teaching effectiveness.

3 Our Autonomous Robotic Car

We developed a new robotic car which can move autonomously. Students built their robots and wrote their programs to control them. The circuit and the controller board were designed by an expert. We adopted the PIC microcontroller which consists of Central Processing Unit (CPU), Random Access Memory (RAM), Read Only Memory (ROM) and Input and Output (I/O). ROM includes a monitor program (firmware) to interpret commands transferred from computers. RAM can store a program of up to 39 steps and two subroutines of up to 7 steps. I/O can control 2 motors which enable moving the robot forward, backward and turning it. This board can connect a switch as a sensor. The Program is transmitted from computers by means of an infrared ray interface. Fig. 1 is the controller board and an infrared ray interface.

4 Dolittle

4.1 Programming Language “Dolittle”

In Japan, teachers and students speak Japanese at school. Kanji and Kana characters which are unique to Japanese are used to express the language. Because students learn English in junior high schools, children are not familiar with the english/roman alphabet in elementary schools. That is why we adopted the Dolittle programming language. Dolittle is an object-oriented language designed for school education. Using Dolittle, students can write programs in Japanese, Korean and English. Fig. 2 is a sample of turning pentagon animation.

```
kameta = turtle ! create.
```

In this statement, the “turtle” object receives a “create” message, then its “create” method creates a new object named “kameta”. The basic syntax of

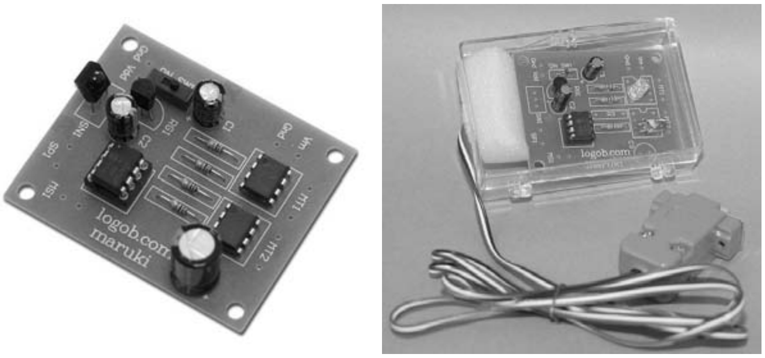


Fig. 1. The controller board and the infrared ray interface

Dolittle is like “`obj ! arg...msg.`”. object “obj” receives message “msg” with some arguments.

```
pentagon = [kameta ! 100 forward 72 leftturn] ! 5 repeat makefigure.
```

“`[...]`” is a block. It makes an object that includes pieces of program code. By sending a “repeat” message to a block, we can repeat the code within the block. As a result, a turtle moves on the display, then a pentagon is drawn.

A figure drawn by the turtle graphics is only lines. But we can make a figure object from lines by executing the “makefigure” message.

```
clock = timer ! create.
```

A timer object runs program code periodically. In the above sentence, a timer object named clock is created.

```
clock ! [pentagon ! 10 0 moveby] execute.
```

“clock” moves the “pentagon” periodically on the display. Students can make animation programs using timer.

Students can make more diverse programs using buttons. They are very interested in programming because they can use buttons they made. Fig. 3 shows an example of a program made by students.

```
button1 = button ! "turn" create.
```

This makes a button object named “button1”.

```
button1:click = [kameta ! 90 rightturn].
```

This defines a method which will be executed when “button1” is pressed. When the “button1” is pushed, the statement “kameta ! 90 rightturn” is executed, “kameta” turns 90 degrees.

4.2 Control Program

Dolittle can control external devices by accessing to external ports. As students can write programs in Dolittle syntax, they don’t need to write bytecode data which is transferred to robots. Fig. 4 is a sample program of controlling robotic cars using Dolittle. This program makes a communication object named

```
kameta = turtle ! create.
pentagon = [kameta ! 100 forward 72 leftturn]
! 5 repeat makefigure.
clock = timer ! create.
clock ! [pentagon ! 10 0 moveby] execute.
```

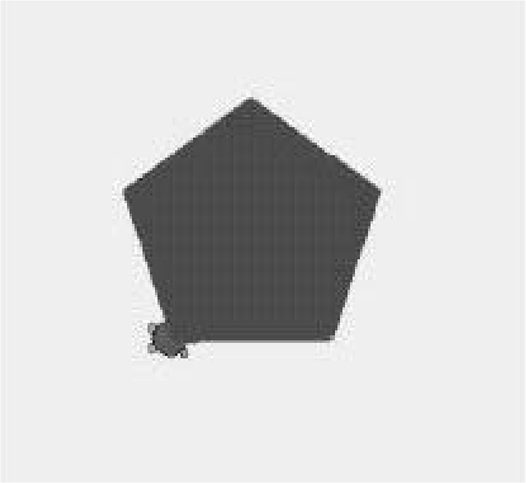


Fig. 2. Sample program of turning figure

"aRobot", then defines the method to send commands to the robotic cars. This method will be executed after opening the communication port. By this method, robotic cars move according to the following steps.

- Start when the switch is pushed.
- Move forward.
- When it collides with something, go back then turn left.
- Move forward again.
- When it collides with something, go back then turn right.

5 Lessons at a Junior High School

5.1 Curriculum

We conducted lessons at Nishimashizu Junior High School in Shizuoka prefecture. All of 135 students in the 2nd grade (14 years old) attended the lessons. Kurebayashi took charge of these lessons.

Table 1 is the curriculum of these lessons. First of all, student prepared programs in Dolittle language. They made animation programs using timer objects. Then they manufactured robots. They also wrote their own programs to control robot movement. Finally they produced game programs of Dolittle.

```
kameta = turtle ! create.
button1 = button ! "turn" create.
button1:click = [kameta ! 90 rightturn].
button2 = button ! "walk" create.
button2:click = [kameta ! 50 forward].
```

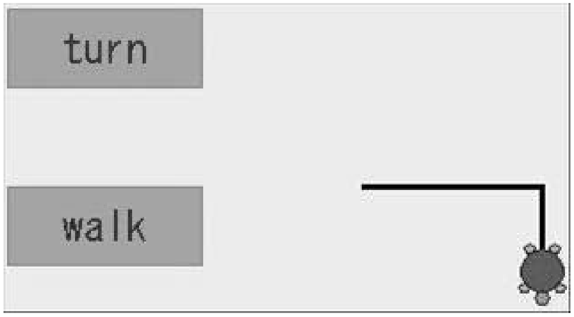
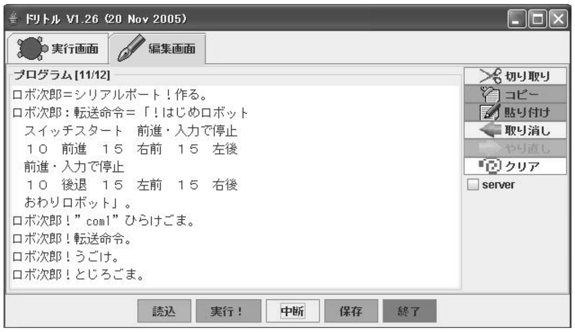


Fig. 3. Sample program using buttons



```
ロボ次郎＝シリアルポート！作る。
ロボ次郎：転送命令＝「！はじめロボット
  スイッチスタート  前進・入力で停止
  １０  前進  １５  右前  １５  左後
  前進・入力で停止
  １０  後退  １５  左前  １５  右後
  おわりロボット」。
ロボ次郎！”com1” ひらけごま。
ロボ次郎！転送命令。
ロボ次郎！うごけ。
ロボ次郎！とじろごま。
```

Fig. 4. Sample of controlling robotic cars (written on the edit screen of Dolittle)

5.2 Results

(1) Robot Construction Students developed their robotic cars that have own shapes in 10 lessons. The body of the robots was constructed using the parts of a vehicle chassis kit for kids. Half of the students were beginners who have never used solder and screwdrivers. But all of them strived to assemble their own robotic cars seriously. Thus, all students were able to complete robot building. Fig. 5 is a robot made by one of the students.

(2) Controlling Robots Students wrote control programs to run in maze courses in 8 lessons. Fig. 6 is a photo of the maze course and students.

Table 2 shows the results of a questionnaire for evaluating this class. More than 90% of students answered that it was fun to learn about robots. Most stu-

Table 1. Curriculum of lessons

Contents	Lesson hours
Let's make programs	8
Let's make robots	10
Let's control robots	8
Let's make programs	8

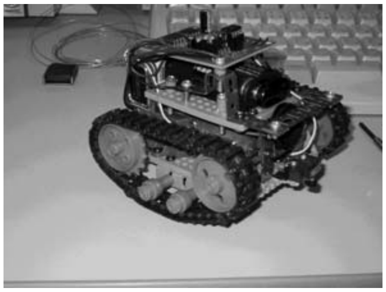


Fig. 5. Student's robot

dents were satisfied with learning of controlling robots. However, more than 70% students answered that it was difficult to control the robots by programs. Many students found it "hard fun". Fig. 7 presents the comments of some students.

(3) Programming of Dolittle In the first and the last lessons, students enjoyed Dolittle programming. They tried to develop game programs using button objects. They utilized their original idea. Fig. 8 is a screen-shot of the game and the student who programmed it.

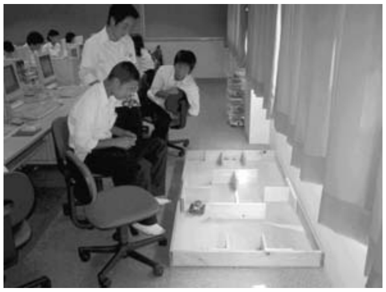


Fig. 6. Maze and students

Table 2. Results of questionnaire (%)

question	Yes	⟷		No
Programming of robots is fun	44	51	5	0
Programming of robots is easy	4	23	47	26

(1) I really enjoyed learning the latest robot technology. Programming was not easy, but I was very happy when my program was completed. Through these lessons, I felt that the age of AI (Artificial Intelligence) will come soon. Accuracy of robot was not satisfactory, but I could adjust it by the program. So, I was surprised to find that. I think this technology is amazing because some animal robots like AIBO (SONY's dog robot) use the same technology. I thought programming that we experienced is basis of many things in the world.

(2) I enjoyed programming. I like using computers. I felt a sense of achievement when I finished my robot construction and when it could reach the goal of the maze. Because I could make a robot and a program by myself. I could learn latest and high level technologies during these lessons. I was impressed with the subroutine of programs.

Fig. 7. Comments of students

6 Discussion

The student in front of Fig. 7 said, "Accuracy of robot was not satisfactory but I could adjust it by the program. So, I was surprised to find that." This comment shows he solved the mechanical problems by a program. His robot couldn't move exactly straight when "forward" movement was commanded. By the experiences of controlling robotic cars, students could learn about inside and outside of the computer and linking the program implementation with the actual movement.

The second student shown in Fig. 7 said, "I felt a sense of achievement when I finished my robot construction and it could get to the goal of the maze. Because I made the robot and the program all by myself." This comment indicates that students felt a sense of fulfillment through making robots and writing pro-

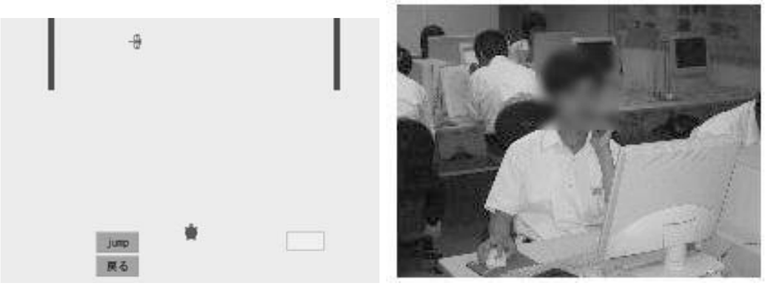


Fig. 8. Student and the programmed game

grams to control them. Students can feel deep achievement and motivation by developing their own robots.

In this class, students experienced two types of Dolittle programming: Dolittle programming and robot control programming. Fig. 3 shows the results of the respective questionnaire.

Table 3. Results of questionnaire (%)

question	Dolittle	Robot
Which is more fun?	57	43
Which is more difficult?	56	44
What make you interested in computers?	58	42

We correlated the above result with the students' scholastic achievements in 5 major subjects (Japanese, Social Studies, Mathematics, Science, and English). The result indicated that low-achieving students tend to answer "Robot is more fun", "Dolittle is more difficult" and "Robots made me interested in computers". Because the style of robot control programming is a "trial and error" approach, students who have difficulties in learning can also understand easily. On the other hand, high-achieving students found robot programming boring. As this programming is not so complicated, the students can estimate the result of their programs before execution. Thus they tend to find intellectual interest in Dolittle programming. Therefore, we concluded that robot control programming is suitable and effective material for the introductory stage of programming.

7 Future Work

7.1 From Biaxial to Triaxial Robot as Teaching Material

Use of robots as a teaching material enabled us to conduct lessons which combine programming and robotic control. In these lessons, robots are biaxial and they only move around the field. Although the lessons were very successful, we consider that tasks are too simple to apply in the field of education. We need more extensible teaching material with more teachers can conduct lessons according to their original ideas. Thus we are planning to implement a triaxial robot control lesson. In this lesson, students establish a goal to accomplish certain work such as carrying some objects from one point to another by using a robot arm with the third motor. We also think robots need to equip more sensors such as light sensor etc. If more sensors are provided with the robot, more flexible and complicated work can be attained. To realize this, we requested another expert to design and develop another controller board which can control three DC motors and install up to four sensors. Fig. 9 shows this controller board and a trial model of triaxial robotic car.

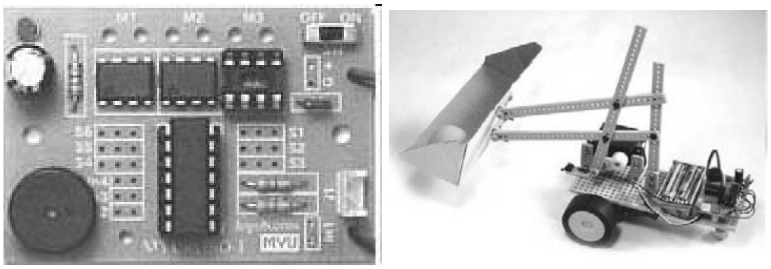


Fig. 9. Triaxial controller board and an example robot

By using this controller board, students can work on more practical problems than conventional biaxial robotic car control. Furthermore, triaxial robots have room to exercise ingenious mechanism of the arm. This enables school teachers to run robot contests to compete for superiority of both programming skill and sophistication of the mechanism. In most of the popular robot contests for junior high school students in Japan, robots are controlled by hand with a wired remote controller (switch board). In this type of contest, the effect of control skill is sometimes superior to that of the design of the robot. However, in creating an autonomous robot, student must consider the motion of the robot beforehand. This makes students think how their robots capture the information of the surrounding environment and process it. We think students will be able to enjoy solving problems throughout this process. Currently the experimental lessons of triaxial robot control are in progress. We are going to evaluate and analyze these lessons after finishing them.

7.2 Coherency Between Screen Objects and Real-World Robots

Two robots mentioned in this paper i.e. biaxial and triaxial robots run with programs downloaded from a PC to the on-board CPUs. These CPUs are not designed to execute complicated and long programs. The programs must be written as procedural, not object-oriented; this means the language syntax of robot control code is different from that of the Dolittle programming language. Consequently, there is a semantic gap between Dolittle and robot control code.

We are planning to improve both Dolittle and the firmware of the controller board so that consistency in language syntax can be achieved between screen objects like turtles and robots in the real-world. With this, we hope students will be able to learn the concept of object-oriented programming not only from Dolittle but also robot control. We also hope this will be an unprecedented teaching material for learning programming.

8 Related Works

The endeavors to utilize autonomous robots for school education have been studied since 1990s[4]. The most innovative work in this area is Programmable Bricks[5]. It was developed from experiences of the autonomous robot contest at MIT. It is presented as a large LEGO block, and it is combined with the body and other mechanisms built with conventional LEGO parts to make an autonomous robot. It was commercialized as LEGO MindStorms by the LEGO Group and has been used widely in schools. In LEGO MindStorms, a unit compatible with Programmable Brick is called RCX. It is equipped with 3 inputs for sensors, 3 outputs for motors, simple LCD displays and an infrared ray interface. RCX programming is conducted in RCX Code or ROBOLAB. RCX Code is a visual programming environment which was improved from LogoBlocks[6]. ROBOLAB is another visual programming environment based on LabView.

In lessons using LEGO MindStorms, students can construct robots easily with LEGO blocks. However, it is difficult to construct small and light weight robots. Furthermore, because RCX is a sealed package, students would not see or touch the electronic circuits or IC chips inside of it.

Several years later, Cricket[7] was developed by MIT. Cricket has two inputs for sensors, two outputs for motors, an infrared ray interface and two general bus ports. Programs can be written in Logo languages using Cricket Logo (text language) or LogoBlocks (visual language).

Cricket is also used for constructing autonomous robots. However, as the capacity of its battery is small and it has only two motors by default (without expansion with general bus devices), it is not suitable for complex robots.

RCX and Cricket have in common with our controller boards that they are used for building robots, but there is one notable difference. Most of RCX educators are interested in the type of robots they create. Cricket is often used in a wider range of use of education such as science. Our objective is that students understand computer through constructing and programming of robots. We adopted the Dolittle programming language and students write program by text and input. Because the syntax of Dolittle is simple and it allows to express programs in local languages, students can prepare programs without difficulty. Additionally, because our robots are simple, even controller boards can be constructed by soldering. We designed robots that are simple and not expensive so that students can buy and bring them to their home.

Conclusion

Learning programming by using Dolittle and robot control has been introduced. Ideal learning which students think "it is hard but fun" can be realized by incorporating robot control into learning programming. Furthermore, learning that links the program implemented inside the computer with the real motion of a robot was made possible. Self-produced robots induced a sense of accomplishment in students learning programming.

In addition, students good at learning enjoyed learning programming with Dolittle and tended to have interest in a computer. Low-performing students found pleasure of learning programming with robot control and tended to have interest in a computer. Thus we are convinced of the effect to strengthen motivation by introducing robots to learning programming.

Acknowledgement

We would like to thank Mr. Masayoshi Okada and Mr. Shuji Inoue for their kind cooperation to develop controller boards for autonomous robots we use. We also deeply appreciate the contribution of the teachers and the students of Nishimashizu Junior High School.

References

1. Kanemune, S. and Kuno, Y. : Dolittle : An Object-Oriented Language for K12 Education. *EuroLogo2005*, Warszawa, Poland, pp. 144–153, 2005.
2. Kanemune, S., Nakatani, T., Mitarai, R., Fukui, S. and Kuno, Y. : Dolittle — Experiences in Teaching Programming at K12 Schools. *Proc. of the Second International Conference on Creating, Connecting and Collaborating through Computing (C5)*, IEEE, pp. 177–184, Kyoto, Japan, 2004.
3. Dolittle Programming Language. `http://dolittle.eplang.jp/`
4. Resnick, M. : Behavior Construction Kits. *Communications of the ACM*, Vol. 36, No. 7, pp. 64–71, 1993.
5. Resnick, M., Martin, F., Sargent, R. and Silverman, B. : Programmable Bricks : Toys to think with. *IBM Systems Journal*, Vol. 35, No. 3–4, pp. 443–452, 1996.
6. Begel, A. : LogoBlocks : A Graphical Programming Language for Interacting with the World. Advanced Undergraduate Project, MIT, 1996.
7. Martin, F., Mikhak B., Resnick, M., Silverman B. and Berg, R. : To Mindstorms and Beyond : Evolution of a Construction Kit for Magical Machines. *Robots for Kids*, Morgan Kaufman, 2000.