

# Easy Robot Programming for Beginners and Kids Using Command and Instruction Marker Card

Masahiro Nishiguchi, Kunio Sakamoto

► **To cite this version:**

Masahiro Nishiguchi, Kunio Sakamoto. Easy Robot Programming for Beginners and Kids Using Command and Instruction Marker Card. Hyun Seung Yang; Rainer Malaka; Junichi Hoshino; Jung Hyun Han. 9th International Conference on Entertainment Computing (ICEC), Sep 2010, Seoul, South Korea. Springer, Lecture Notes in Computer Science, LNCS-6243, pp.460-463, 2010, Entertainment Computing - ICEC 2010. <10.1007/978-3-642-15399-0\_60>. <hal-01055601>

**HAL Id: hal-01055601**

**<https://hal.inria.fr/hal-01055601>**

Submitted on 13 Aug 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Easy Robot Programming for Beginners and Kids Using Command and Instruction Marker Card

Masahiro Nishiguchi and Kunio Sakamoto

Department of Intelligence and Informatics, Konan University  
8-9-1 Okamoto, Higashinada, Kobe 658-8501, Japan  
kunio@konan-u.ac.jp

**Abstract.** Robots usually have multiple components, such as motors, sensors, microcontrollers and embedded computers. A robot programming to control motors and measure the output of sensors is complicated. Therefore it is troublesome for beginners to write the program of a robot control. To solve this problem, this paper describes a card programming for controlling a robot.

**Keywords:** card programming, mobile robot control, structured programming

## 1 Introduction

The authors have developed the mobile robot which can be programmed by command and instruction cards. All you have to do is to arrange cards on a table and to shot the programming stage by a camera. Our card programming system recognizes instruction cards and translates icon commands into the motor driver program. This card programming environment also provides low-level structure programming.

## 2 Instruction Sets for Robot Programming

Fig. 1 shows basic instruction sets to control a mobile robot. The mobile robot has wheels to move itself. 'Go', 'Back', 'Left' and 'Right' commands enables to transfer the robot anywhere you like. The wheels are rotated by motors. In order to move the robot, a motor driver needs to control the motors. Fig.2 shows the programming language for our developed motor driver and input/output controller. You must actually indicate which motor to work and how much it to rotate. However it is difficult for beginners to program and troublesome for all users to indicate in detail. So these basic commands are automatically translated into the program for the motor driver in our system. Moreover the authors developed the card programming environment using command and instruction marker which enables beginners or kids to enjoy lightheartedly the mobile robot operation. Fig. 3 shows examples of command and instruction cards.

Our developed motor driver can control six servo motors and have eight digital input/output ports. The motor driver has the PIC(Peripheral Interface Controller)

microcomputer manufactured by Microchip Technology Inc. as a command service module and connects with a Windows PC through the RS-232C serial communication line. In our driver, the PIC microcomputer receives 26 kinds of instructions from the PC. One capital alphabet letter and an additional data constitute this instruction form as shown in Fig. 2. In order to control the mobile robot, the motor driver operates motors and turns a light on or off by the instruction, which is directly inputted or translated on the card programming environment.

Fig. 4 shows a situation of the card programming for our mobile robot control. As shown in Fig. 4, a camera shots the programming stage where you arrange the instruction cards on. On our card programming environment, you can instruct eight commands on the table at the same time. Moreover this card programming system supports a 'for loop' statement and a subroutine statement as the universal programming language. The 'for loop' allows command to be repeatedly executed. The subroutine also involves some commands to build a portion of instruction within a larger program. Then 'for loop' and subroutine have a series of instructions, and these statements behave in much the same way as a program that is used as one step in a larger program or another subprogram. The authors introduced and demonstrated the feasibility of a new concept in packaging. The packaging is to unify some instruction cards into one card as shown in Fig. 1. Thus packaging process can memorize and give a new function to a blank card which involves a series of commands and executes instructions step by step. Then to put a newly generated instruction card on the table means to call and execute the subroutine. The combination of the loop command and packaged instructions card realizes the loop function which executes same commands many times defined by a loop counter.

GO	Move forward
BACK	Move behind
LEFT	Turn left
RIGHT	Turn light
LIGHT	Turn LED light on/off
LOOP	Execute instructions repeatedly
MEMORY	Package instructions (Generate subroutine)
CALL	Execute packaged instructions (Call subroutine)

```

10 U:01 70 #call subroutine
20 H:1 #Port H=high
30 W:100 #wait
40 H:0 #Port H=low
50 U:20 70 #loop 20 times
60 U:99 #program end
70 A:220 #Servo A=220
80 B:90 #Servo B=90
90 U:00 #return

```

Fig. 2. Example text programming

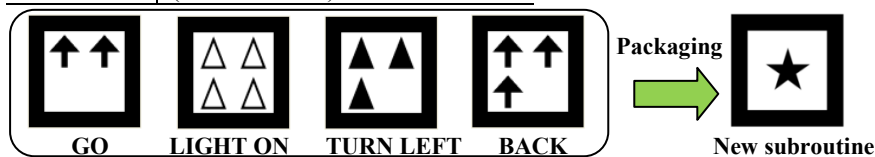


Fig. 1. Example of instructions and generating subroutine

### 3 Easy Programming Using Command Cards

To realize the card programming, our developed easy robot programming system needs to recognize markers in the real world and discriminate command and instruction from drawn patterns on the markers. As shown in Fig. 3, the marker card

consists of a black square frame filled with some kind of a pattern. We shoot makers and take picture of it. The computer vision system firstly finds the black square frames from video streams and then recognizes patterns within the interesting regions. On our developed system, the authors make use of a mixed reality software in order to recognize a kind of the marker and measure its position. This software library is called ARToolkit. ARToolkit involves a video capture, a 3D graphics generator, a spatial measuring and an overlay imaging for the creation of augmented reality applications. Fig. 4 shows the result of card recognition. In Fig.4, the application window area consists of three regions; the upper screen shot shows the recognition result, the lower left shot is an original video stream and the lower right shows recognized instructions. The card programming stage has eight slots and the center is an origin marker for measuring card positions. A user puts the instruction cards onto these slots by turns of execution. At the example shown in Fig. 4, the slot number 2, 6 and 8 are vacant. The robot programming system finds the all instruction markers and overlays cubes on the markers as flags in case that the recognition process is correctly finished. As shown in lower right of Fig. 4, the six instructions are loaded excluding vacant slots. Fig. 5 shows the appearance of our mobile robot. You can control this robot to move anywhere you like and turn head lights on or off using the command and instruction cards.

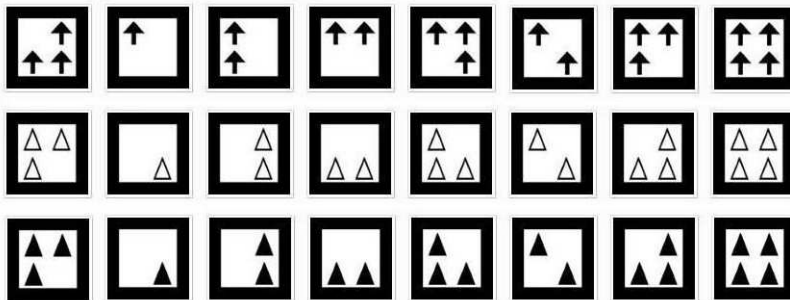


Fig. 3. Example of command and instruction cards

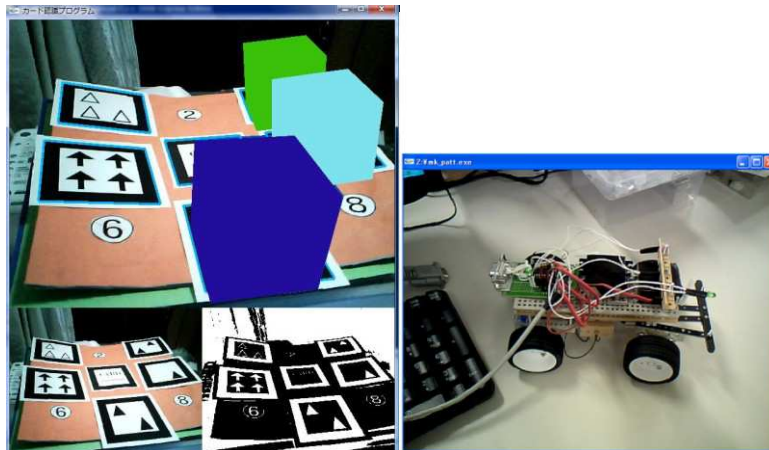


Fig. 4. Result of card recognition

Fig. 5. Appearance of mobile robot

**Acknowledgments.** This research is partially supported by “Grant-in-Aid for Young Scientists(B)” #20700112 and “Scientific Research (C) (General)” #20500481 from Ministry of Education, Culture, Sports, Science and Technology Japan(MEXT).