Development of Physical Blocks with Communication Device for Visual Programming Study

Akihiro Iida, Tokyo Denki University, Japan
Tatsuyuki Takano, Kanto Gakuin University, Japan
Osamu Miyakawa, Tokyo Denki University, Japan
Takashi Kohama, Tokyo Denki University, Japan

The Asian Conference on Education 2017
Official Conference Proceedings

Abstract
The visual programming language Scratch is widely used to learn problem-solving skills and logical thinking; However, it is not easy for elementary school students to learn how to use Scratch language. Conventional Scratch learning has two different aspects on how to use the PCs to carry out various operations and programming. For students who are unfamiliar with how to use the PC such as how to use the keyboard and mouse. For these students, it will difficult for them to proceed with learning how to use the PCs and at the same time how to do programming. The purpose of this research is to introduce students on how to program without having to learn how to operate the PCs. In this research, the authors have developed their own program using real physical blocks instead of the blocks that must be used on the Scratch's screen. By connecting these real blocks together, the students can create the Scratch programs. This created program can then be used to execute on a PC. The authors have created seven different types of physical items which we call "blocks." These "blocks" are as follow: "start," "display," "move," "repeat," "conditional branch," "conditional expression" and "sensor." The "Start block" function is to connect the other blocks to the PC. These physical blocks are then connected in the same way as the Scratch program. Overall, the authors developed real blocks to confirm that the operations worked.

Keywords: Programming, Scratch, Real physical block, Education for elementary school

Introduction

The Ministry of Education, Culture, Sports, Science, and Technology-Japan decided that compulsory programming in elementary school education in 2020 (The Ministry of Education, Culture, Sports, Science, and Technology-Japan, 2017). The aim is to teach problem-solving skills and logical thinking to students.

The visual programming language is said to be easy to understand for beginners, and the programming language has been widely used in programming education especially the Scratch language. There is a practical report which actually carried out lessons using the Scratch language at elementary school (Hideki et al., 2010).

However, learning Scratch has two different aspects on how to use the PCs to carry out various operations and programming. Some students who are unfamiliar with how to use the PC such as how to use the keyboard and mouse. For these students, it will difficult for them to proceed with learning how to use the PCs and at the same time how to do programming.

The purpose of this research is to introduce students on how to program without having to learn how to operate the PCs.
By connecting these physical blocks together, the students can create the Scratch programs.

Scratch

Scratch is a visual programming language developed by MIT (Scratch, n.d.). It is mainly designed for 8 to 16 years old. Scratch programming is done by combining Scratch blocks representing commands on the PC's screen. The execution result is displayed through the character (sprite) on the PC's screen.

Figure 1: A sample of Scratch Program.

An example of the Scratch program is shown in Figure 1. This Scratch program repeats the operation of moving the sprite 10 steps 10 times when the green flag button is clicked.

Movement of the sprite is the basis of Scratch programming. In actual classes, the learner learns the meaning and operation method represented by the Scratch block and progresses learning by programming according to the theme given by the teacher.

System

Teacher demonstrates on how to program Scratch by using physical blocks to students. Teacher executes the assembled program and shows students the result of programming and teaches the meaning of each physical block. Students program the Scratch program by using physical blocks based on the exemplified how to connect physical blocks by the teacher.

The system targets Scratch2.0. The system consists of a PC which executes Scratch, and physical blocks are referring to Scratch blocks on the screen. Connecting physical blocks together represents the Scratch programming, which looks like Scratch programming on the PC's screen. The PC perceives connected physical blocks then build a program to execute the program. Physical blocks include a microcontroller and communicate each other and PC. Seven types of physical block were made: "start," "display," "move," "repeat," "conditional branch," "conditional expression," and "sensor." The authors created the program which was able to communicate to the physical blocks and construct a Scratch program.

Figure 2: Overview of The System

System constitution

Figure 2 is an overview of the system. The system consists of the PC which executes Scratch, and physical block imitating Scratch block. Represent Scratch program by joining physical blocks. The PC creates and executes the Scratch program by reading the alignment in a sequence of the physical block.

Real physical blocks

Table 1 shows the role of the physical blocks were created. The string described on the physical block follows the "Hiragana" expression of the Scratch development environment. "Nihongo"("Hiragana" style) is selected from Scratch's language menu. The shape of the physical block also reduces the difference from the Scratch block.

Type of Physical Block	Name of Scratch Block	Meaning of Command
Start	when Green Flag clicked	Conect other physical block from this physical block
Display	say (var)	Display characters through blooming from a sprite
Move	move (var) steps	Move a sprite 1 pixel to facing way
Repeat	repeat (var)	Repeat the methods inside this block
Conditional Branch	if (var) then	Decide next executing method whether the specified condition is satisfied or not
Conditional Expression	$<$, $>$, $=$, not	Evaluate the value of the sensor and the value set in the "conditional expression" physical block using the comparison operator
Sensor	(var) sensor value	Use the value of specified sensor in the program

Table 1: Role of Each Physical Block

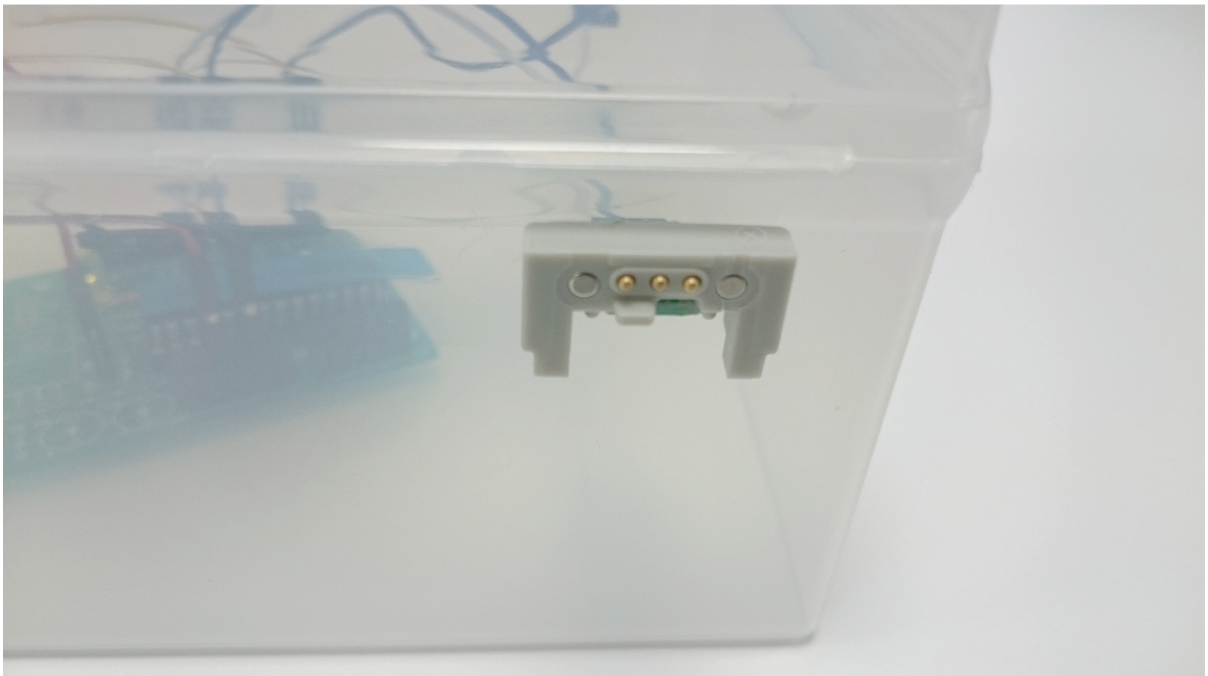

Figure 3: A Connector of the Physical Block

Figure 3 is a connector of the physical block to connect another physical block. Magnets are attached to the connectors, and misconnection can be prevented by using the shape of the connectors and the polarity of the magnets. "Start" physical block is connected to the PC and serves as a starting point of other physical blocks. USB connection is used for connection to the PC. "repeat (var)" and "if (var) then" Scratch blocks have a beginning and end, and has a structure that puts other Scratch blocks between the beginning and the end. In the "Repeat" physical block and the "Conditional branch" physical block, an expansion and contraction mechanism was adopted in order to put other physical blocks between the beginning and the end. As a sensor, the system uses "nekoboard 2" which is a compatible machine of "Picoboard" which is a sensor board used in the Scratch environment (Picoboard, n.d.). To grasp the alignment in a sequence between the physical blocks, a microcontroller is included in the physical block. The PC grasps the alignment in a sequence between the physical blocks by using the serial communication of the microcontrollers. A unique ID is set for each microcontroller, and by sending this ID to the PC, the assembling of the physical blocks are recognized.

Physical block combination

Connection Basis	Connection To
Start	Display, Move, Repeat, Conditional Branch
Display	Display, Move, Repeat, Conditional Branch
Move	Display, Move, Repeat, Conditional Branch
Repeat	Display, Move, Repeat, Conditional Branch
Conditional Branch	Display, Move, Repeat, Conditional Branch
Conditional Expression	Sensor
Sensor	Display, Move, Repeat, Conditional Branch

Table 2: Connection Combination of Physical Blocks

Table 2 shows connection combinations of physical blocks. In the "start" physical block, there is no physical block to be connected before, and the PC is connected. The "repeat" physical block connects physical blocks to be repeated between the beginning and the end. The next physical block is connected to the end of the "repeat" physical block. The condition part of the "conditional branch" physical block is connected so that the "conditional expression" physical block is superimposed on top. The process to be executed when the condition is true is connected to the starting end and the terminating end. The "conditional expression" physical block connects only with the "sensor" physical block. The "sensor" physical block is connected so as to overlap the "conditional expression" physical block.

Hardware in physical blocks

Arduino is used as a microcontroller (Arduino, n.d.). Arduino is open source hardware with AVR microcontroller and input / output port.

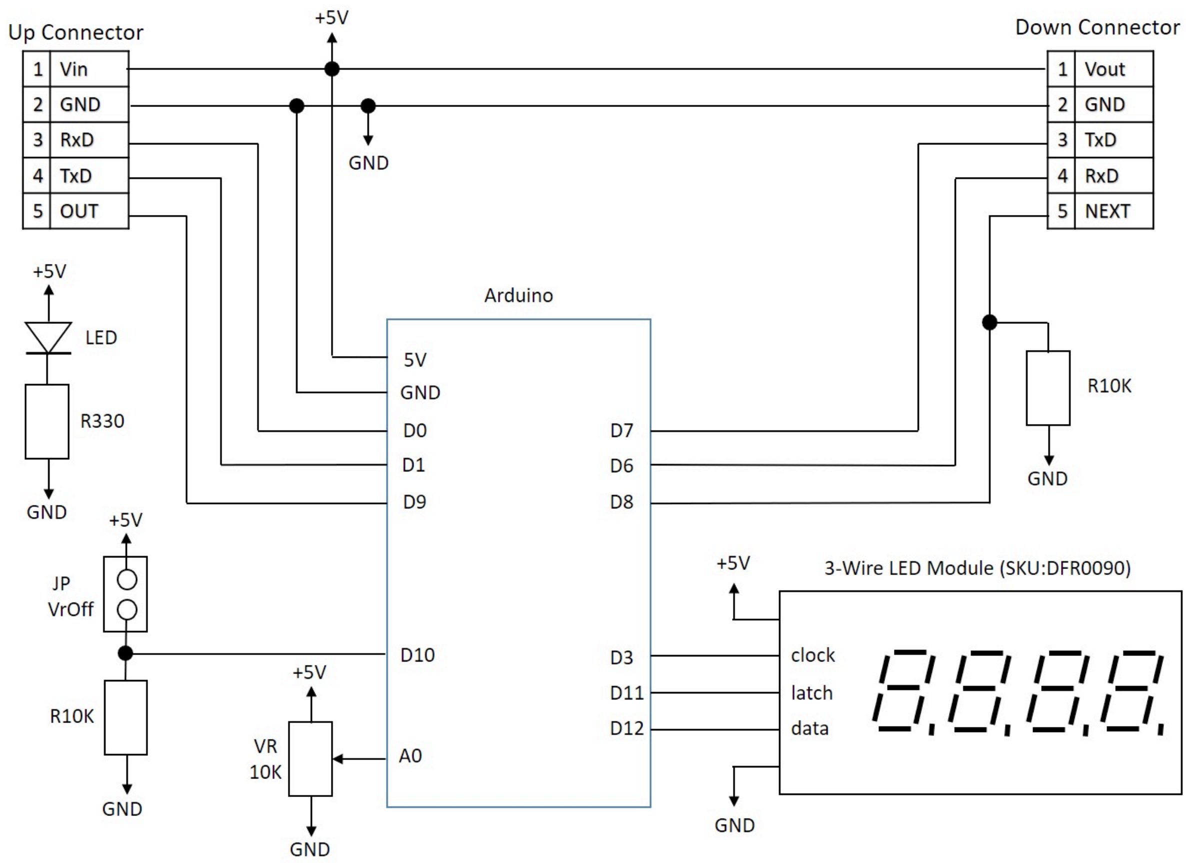

Figure 4: Circuit Diagram inside the physical block

Figure 4 shows the circuit diagram inside the physical block. UART is used for communication between microcontrollers. RxD and TxD represent the reception port and transmission port of the serial communication, respectively. Power is output from Vout and power is input at Vin. This supplies power between the physical blocks. OUT tells that the next connected microcontroller is connected to the previous microcontroller. NEXT grasps the presence or absence of a connection by receiving a signal from the next microcontroller. Use variable resistor and 7 segment LED depending on the role of the physical block. Variable resistance is used by the learner to input numerical values. The 7-segment LED is used to display the value of the input variable resistance.

Software in physical blocks

The software was created by using "the Arduino IDE". The software controls serial communication, receives values from variable resistance, and displays the values to 7-segment LED.

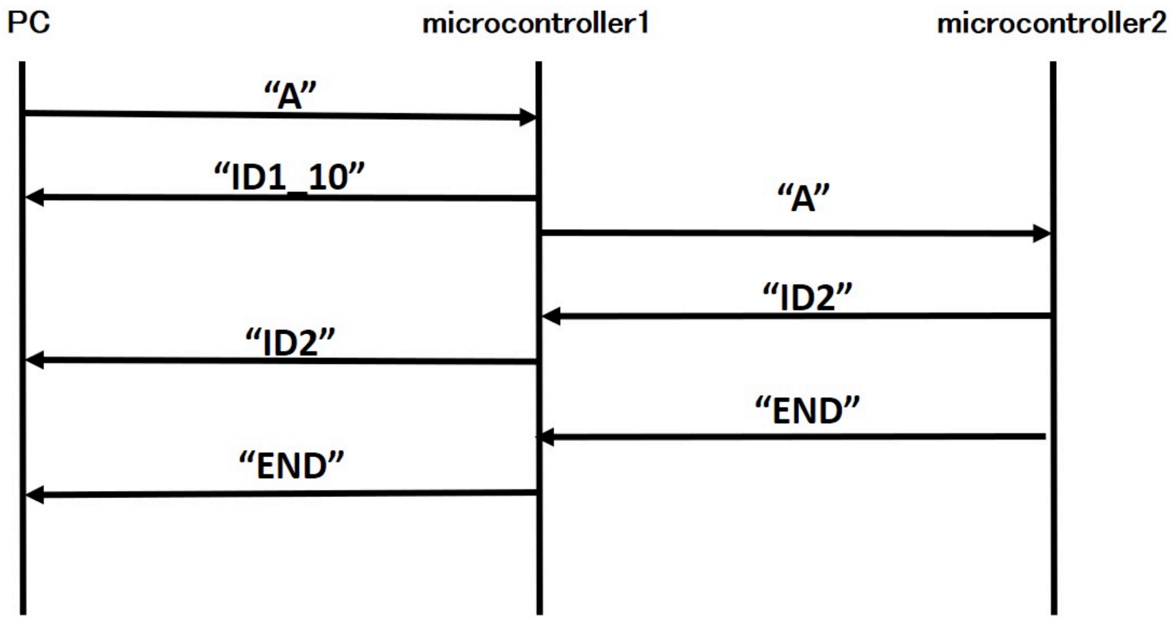

Figure 5: The Communication Procedure of The system

The communication procedure is shown in Figure 5. In order to grasp the role of each microcontroller on PC, ID is set. In order for the PC to recognize the termination of the microcontroller, the PC receives an "END" packet from the microcontroller. "A" sent from the PC represents a request packet. "ID 1 _ 10" returned by the microcontroller 1 to the PC represents a response packet. "ID 1" of "ID 1 _ 10" represents the ID of the microcontroller 1. "10" represents an input value. If there is no input value, omit the input value and send only the ID. "END" is a packet representing the termination. By this communication, it is possible to acquire the connection sequence of the microcontroller and to grasp the connection of the blocks.

Software in PC and how to make a Scratch program from physical blocks

The Software in the PC is created by Java programming language.

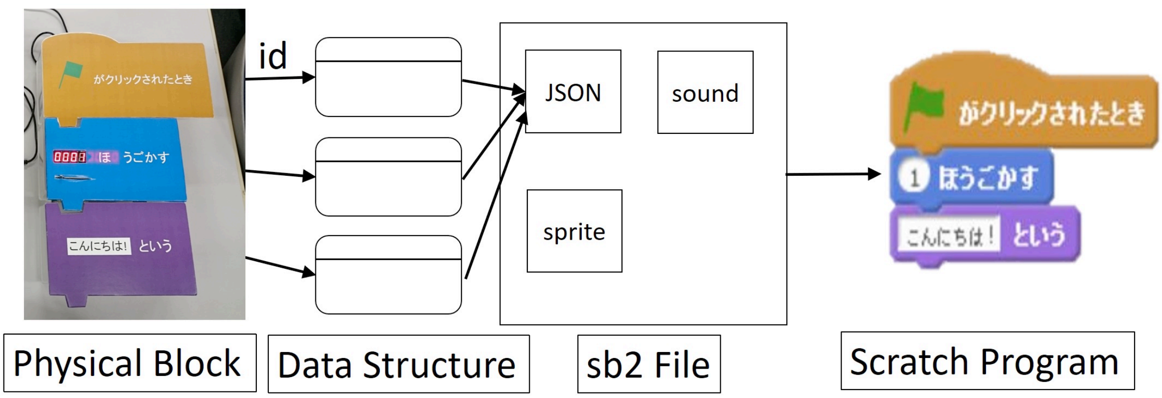

Figure 6: Construct a Scratch Program

Figure 6 shows how to construct a Scratch Program. Construct a Scratch program from the Arduino ID and the alignment in a sequence between physical blocks acquired by the PC. First, convert from the Arduino ID acquired by the PC to the corresponding data structure representing the Scratch block. Second, the structure of the Scratch program is determined based on the alignment in a sequence between the physical blocks. Create a text-based Scratch program from the converted data structure and determined program structure. Text-based Scratch program is output to the file of JSON (JavaScript Object Notation) format where Scratch setting and created program are saved. Along with Scratch's image files and audio files prepared in advance, they are combined into an "sb2" file that is Scratch's executable format. Execute the Scratch program by starting the created the sb2 file with Scratch.

Text-based Scratch program

The source code of Scratch 2.0 is described in the JSON file of the created Scratch project. The source code is described in the script field of the JSON file, and editing is possible. In the source code, text corresponding to the Scratch block is described. For example, "say" in the JSON file corresponds to "say" of the Scratch block, "doRepeat" in the JSON file corresponds to "repeat" of the Scratch block. By editing these texts, it is possible to generate a Scratch program without operating the Scratch block on the PC. This JSON file can be obtained by decompressing an existing Scratch saved file.

System operation check

The authors joined the created physical blocks and confirmed the operation by executing the program on the PC. We created several example programs and confirmed that they could be executed correctly on Scratch.

Figure 7: An Example of Checked Scratch Program

An example of a checked program is shown in Figure 7. For a check of the system, we used following physical blocks: "start," "repeat," "display," "move." The constructed program was executed using Scratch 2.0 Offline Editor. First, connect in order of "start," "display," "repeat," and "move." Second, connect the start to the PC with a USB cable. Third, get the physical block ID and the alignment in a sequence in which the physical block is connected to the PC. Last, construct a Scratch program and check it can be executable.

A program as represented by the physical block is displayed on the editor of Scratch. By pressing the execute button, the Scratch program was executed. Also, by changing the number of iterations, it was possible to check that the number of times on the editor of Scratch also changed, and it was confirmed that the iteration process was performed accordingly.

Experiment

The authors conducted an experiment with physical blocks using by elementary school students at "Kid's seminar" organized by TDU Inzai Innovation Activation Center at Tokyo Denki University, Chiba New Town campus on August 3rd, 2017. The purpose of this experiment is to measure the interface of physical blocks. After explaining how to use the physical blocks, the elementary school students with using the physical blocks. And interviewed the students who used the physical blocks.

Result

Figure 8: An Aspect of Elementary School Student in the Experiment

Figure 8 is an aspect of an elementary school student in the experiment. Students said "It was easy to program." and "The large size physical blocks were easy to use." Some of the students could deal with the other physical blocks which did not explain on how to use the blocks after the students listened to the explanation of how to use the physical blocks only once.

Conclusion

Created real physical blocks for visual programming study. Physical blocks target Scratch2.0. Seven different types of physical blocks were made: "start," "display," "move," "repeat," "conditional branch," "conditional expression," and "sensor." Conducted the experiment and interview with the elementary school students who used the physical blocks.

Acknowledgements

The authors like to thank Professor Niitsu of the Department of Information System Engineering at Tokyo Denki University who permitted conducting the experiment at "Kids seminar", and participants of "Kids seminar" who joined the experiment.

References

Arduino. (n.d.) Retrieved from https://www.arduino.cc/

Hideki, M., Manabu, S., Hai, Z., & Takanori, M., (2010). Practical Study on Scratch Programming Lessons for Elementary School Students. : Rethinking Programming Education at Elementary School. *Journal of Japan Society For Educational Technology*, 34(4), 387-394.

Picoboard. (n.d.) In Scratch Wiki. Retrieved November 21, 2017, from https://wiki.scratch.mit.edu/wiki/PicoBoard

Scratch. (n.d.). Retrieved from https://scratch.mit.edu/

The Ministry of Education, Culture, Sports, Science, and Technology-Japan. (2017). *Syogakkogakusyushidoyoryo* [Elementary school curriculum guideline]. Retrieved from http://www.mext.go.jp/component/a_menu/education/micro_detail/__icsFiles/afieldfile/2017/05/12/1384661_4_2.pdf