International Journal of Advanced Science and Technology
Vol.115 (2018), pp.1-10
http://dx.doi.org/10.14257/ijast.2018.115.01

Block Coding Algorithm Training Examples using Arduino Board for Elementary and Secondary School Students

Kyeong Hur[1], Won-Sung Sohn[1*] and Kil Young Kwon[2†]

[1]Dept. of Computer Education, Gyeongin National University of Education, Gyesan-Dong San 59-12, Gyodae-Gil 45, Gyeyang-Gu, Incheon, 407-753, Korea
[2]Dept. of Family Medicine, Eulji General Hospital, Hagye-Dong 280-1, Hanguelbiseok-street 68, Nowon-Gu, Seoul, 139-872, Korea
Telephone: +82-32-540-1284, Fax: +82-32-548-0288
**sohnws@ginue.ac.kr*

Abstract

In the age of the Fourth Industrial Revolution, software education through physical computing is essential for an understanding of the IoT(Internet of Things) systems. However, there is a lack of effective block coding algorithm training examples for elementary and secondary school students when using EPL(Education Programming Language) like the scratch. Effective physical computing examples should use the Arduino board for scalability. And effective EPL block coding examples should include a problem-solving strategy of the 'top-down' or 'bottom-up' concept. Furthermore, effective EPL block coding examples should include a calculation technique using the 'variable' data structure. In this paper, we propose a hierarchical block coding algorithms to teach 'bottom-up' problem-solving strategy through a physical I/O(Input/Output) system using Arduino. In the proposed algorithms, the calculation technique using the 'variable' data structure is included. We used Scratch for Arduino 1.5 software to teach physical computing programs for elementary and secondary students.

Keywords: *Arduino, Block coding, Physical computing, Scratch for Arduino, Software education*

1. Introduction

The whole society is operated by software, and a new era in which objects are connected to each other by the Internet is coming, and the fields that existed independently became to have the convergence characteristic on the computing base. Naturally, most jobs require computing thinking. As a result of these changes, computational literacy, which is the ability to utilize computational thinking, has begun to be emphasized as a core competence required to live in the future, and to secure professional personnel with an understanding of software principles will soon lead to national competitiveness [1-5].

In Wikipedia, "Computational thinking is the process of generalizing the answer to an unspecified problem. In computational thinking, we break down the entire decision process, taking into account the associated variables and all possible solutions, and taking the corresponding parameters and problem limits into account to make the right decisions.

For this computational thinking, four key thinking skills are needed: First, Decomposition Thinking is the breaking of large, complex problems or systems into smaller pieces. Second, Pattern Recognition is the ability to find similarities within and

---

Received (December 10, 2017), Review Result (January 28, 2018), Accepted (February 4, 2018)
* Corresponding Author
† Co-First Author

ISSN: 2005-4238 IJAST

between cleaved parts. Abstraction Thinking is about focusing only on the important information, ignoring the relevant details. Algorithms are the development of a phased solution of a problem or a set of rules that must be followed to solve a problem [6-7].

The advantages of physical computing education are as follows. First, it can deal with the process of solving real- life problems, so it is possible to integrate with various subjects. Second, the input, processing, and output processes of digital and analog sensors included in Arduino are different from the conventional programming learning, in that programming error detection (debugging) is physically transmitted through the learner's manipulative activities, It is possible to correct the programming error quickly and improve the learning interest and satisfaction of the learners. Third, it provides a process of designing creative prototypes to solve real-life problems, and it can positively influence problem- solving ability by supporting the generation of learner's ideas [5-10].

However, there is a lack of effective physical computing examples for elementary and secondary school students when using EPL(Education Programming Language) like the scratch. In this paper, we propose a hierarchical block coding algorithms to teach 'bottom-up' problem-solving strategy through a physical I/O(Input/Output) system using Arduino [11]. In the proposed algorithms, the calculation technique using the 'variable' data structure is included. We used s4a(Scratch for Arduino) 1.5 software to teach physical computing programs for elementary and secondary students [12].

2. Proposed Hierarchical Block Coding Algorithms using a Physical System

Figure 1 is a schematic diagram of the physical computing circuit used in this paper. Three digital output pins (10, 11, 12) and three analog output pins (9, 6, 5) of the Arduino Uno board shown in *Figure* 2 are used [11-12]. The corresponding color LEDs (Light Emitting Diodes) are connected to each output pin. Figure 3 shows the implemented circuit of power supply on the actual breadboard according to the conceptual diagram of *Figure* 1.

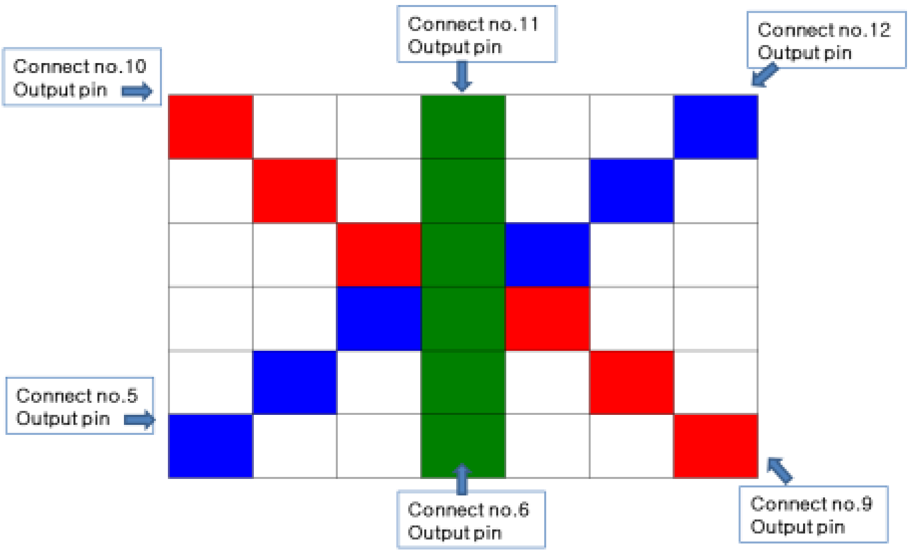

Figure 1. A Schematic Diagram of the Physical Computing Circuit

The analog output command controls the voltage level by dividing the maximum 5V voltage from 0 to 255 steps. The digital output command controls the 5V voltage in two steps, on and off. Here, the step 255 of the analog output is the same as the step of turning on the digital output, and a voltage of 5V is output to the designated pin. Input devices are connected to the A0 to A5 pins of the Arduino Uno board in Figure 2, and the values measured by the input device are represented in analog form from 0 to 1023 steps. It should be noted that the analog output is represented in steps of 0 to 255, and the analog input is represented in steps of 0 to 1023.

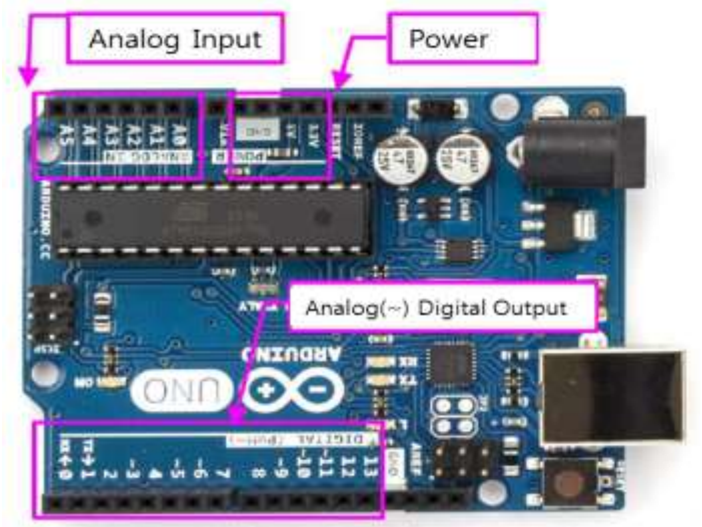

Figure 2. Analog and Digital Input/Output Pins in Arduino Uno Board

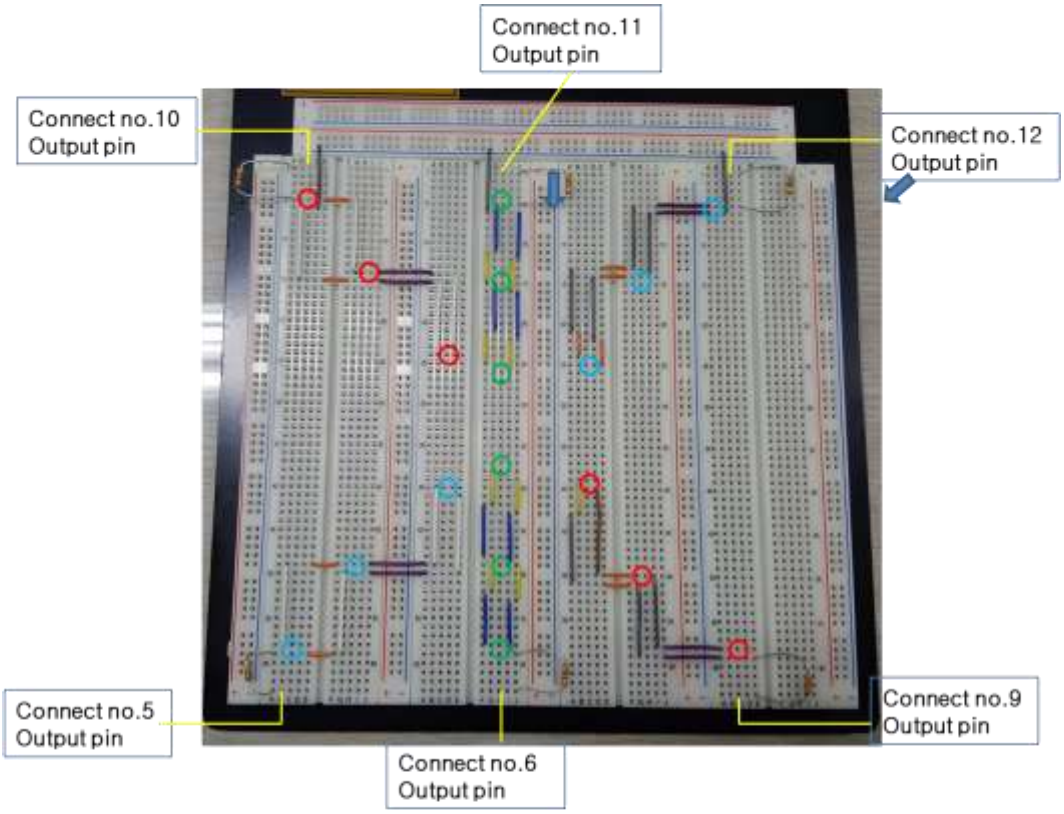

Figure 3. A Physical Circuit Implemented on the Breadboard

2.1. First-layer Hierarchical Function Programming using Global Variables

The order in which the LEDs in Figure 3 are powered clockwise is denoted using the Arduino pin number such as 10, 11, 12, 9, 6 and 5. On the contrary, the anticlockwise order is denoted using the pin number such as 10, 5, 6, 9, 12 and 11. Figure 4 shows a program that turns on LEDs more rapidly in a clockwise direction (dclockwise). In Figure 4, the main function initiated by clicking the green flag button just set the value of DELAY variable and calls six 'x_pin_only_on' functions in a clockwise direction by using 'broadcast and wait' command block. Six 'x_pin_only_on' functions are separated and have different their operation time though 'wait for' command block. A big program can be made efficiently by dividing the whole program into small programs. After those small programs are made with no error, they are combined to work as a whole program through global variables like this DELAY variable. This strategy is called as 'divide and conquer'.

Students can naturally learn how to use the global variable in functions, that is, the use of a DELAY variable, by writing a function that turns on and off three LEDs of each color connected to each output pin. Here, the 'x_pin_only_on' function controls six output pins at the same time in the clockwise order in the circuit, where DELAY variable receives the 0.01- second value. Students can understand the principle of this program by watching the operation of the actual circuit and modify the program by themselves. In the hierarchical function programming, this 'x_pin_only_on' function is defined to play a role of the first and the lowest layer function in the entire program.

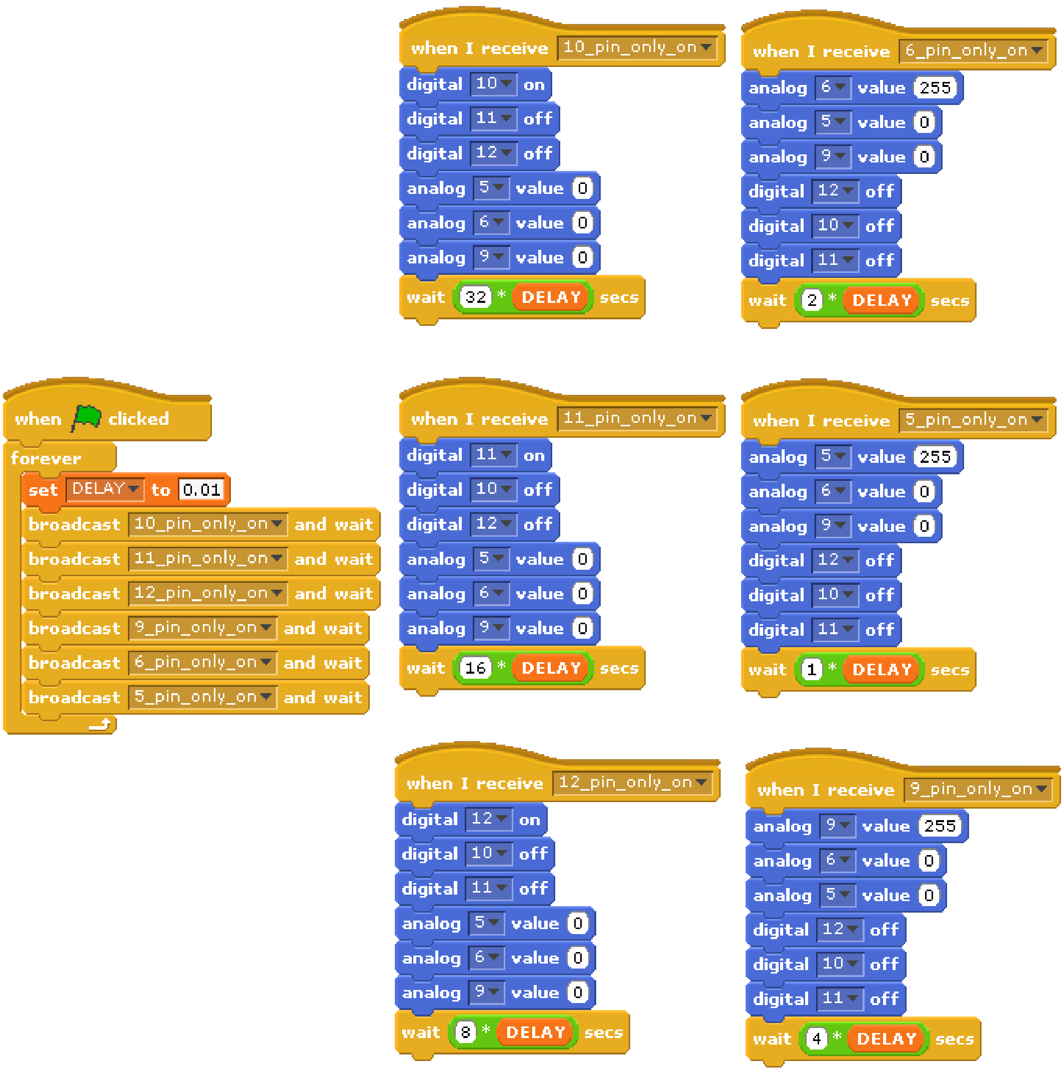

Figure 4. Structure of a 'x_pin_only_on' Function Program to Express a Digital Clockwise Operation (dclockwise)

2.2. Second-layer Hierarchical Function Programming

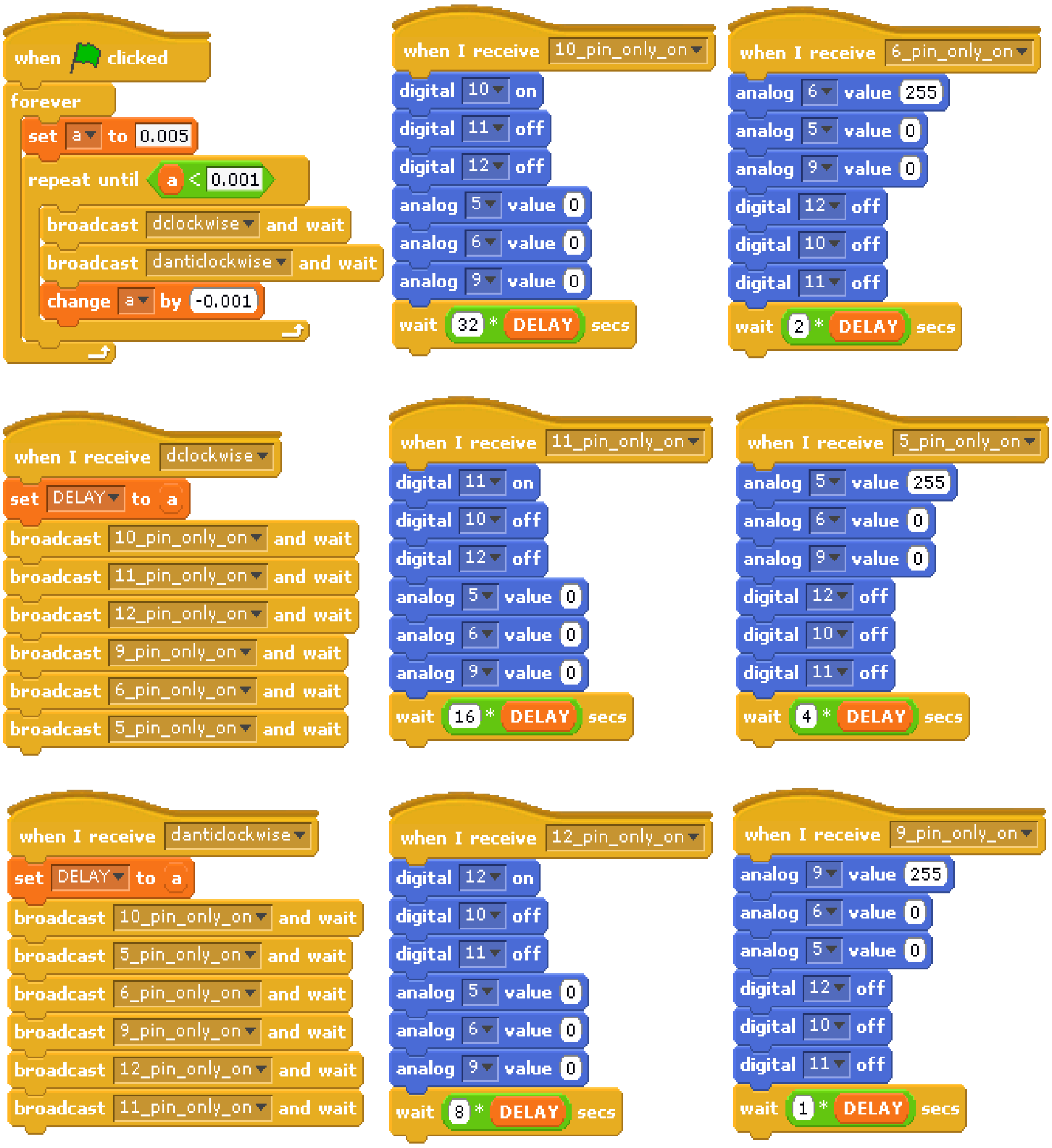

Figure 5. Structure of dclockwise and danticlockwise Function Combined Program

Students can define a dclockwise function that expresses a program that turns on LEDs more rapidly in a clockwise direction as shown in Figure 5 as the second-layer function. In Figure 5, a danticlockwise function is also presented. The dclockwise second-layer function is programmed using several 'x_pin_only_on' first- layer functions. To create a danticlockwise second-layer function whose operation direction is opposite to that of the dclockwise function, it is also defined by the 'x_pin_only_on' first- layer functions. The main function calls dclockwise and danticlockwise functions. And it controls the number of repetition times and whole operation time.

Figure 5 shows a program for expressing dynamic LED lighting by using dclockwise and danticlockwise functions together. That is, after programming the dclockwise and danticlockwise functions and observing the physical results, respectively, students can program the dclockwise and danticlockwise functions together and observe the physical results. After teaching a program like one shown in Figure 5, students should be trained to encourage themselves to express and program various LED lighting patterns. These SW programs are designed without input devices and have sequential and iterative algorithm structure. Even if there is no input device, students can create third- layer hierarchical functions according to students' ideas in top-down or bottom-up style [6-7].

3. Hierarchical Block Coding Algorithms via Input Device Connection

It is possible to express an algorithm that operates selectively according to whether a condition is satisfied or not satisfied. To represent such an algorithm in physical computing education using Arduino devices, an input sensor device must be connected. Figure 6 shows the variable resistor and sound sensor device that can be connected to the Arduino board [11-12]. Figure 7 shows the case where a sound sensor input device is connected to the analog input A2 pin.

And Figure 7 shows a program with LEDs turned on clockwise according to the measured volume value. Here, the time interval for measuring the sound volume value is set to 0.1 seconds. In Figure 7, several ‘x_pin_only_on’ first- layer functions are used hierarchically. Figure 8 shows a program for expressing dynamic LED pattern by using dclockwise and danticlockwise functions together according to current sound volume value. In Figure 8, dclockwise and danticlockwise second-layer function are used hierarchically.

Figure 9 shows a small program of the sixfunction to control the number of repetition times and whole operation time according to measured sound values in Figure 8. In Figure 8, the main function used three global variables of ‘a’, ‘c’ and ‘val’. Through these global variables, the main function controls other nine functions. In Figure 9, ‘b’ variable stores the current sound volume value. And the value of ‘val’ variable is determined conditionally by the value of ‘b’ variable.

Using the program shown in Figure 7, 8 and 9, it is possible for students to perform the programming activities expressing various conditional algorithms by connecting the input devices. Furthermore, the ‘divide and conquer’ strategy should be adopted into the whole problem-solving algorithm. This ‘divide and conquer’ strategy can be implemented into the whole program by constructing the hierarchical functions described above.

While confirming the physical results according to the whole program, students can debug and develop their own program by themselves. It is also possible to design and implement complex operation scenarios by connecting three or four input devices together. This allows a training course to expand the program by combining other functions. In addition, students can create third-layer or fourth-layer functions according to students' ideas in top-down or bottom-up style [6-7].

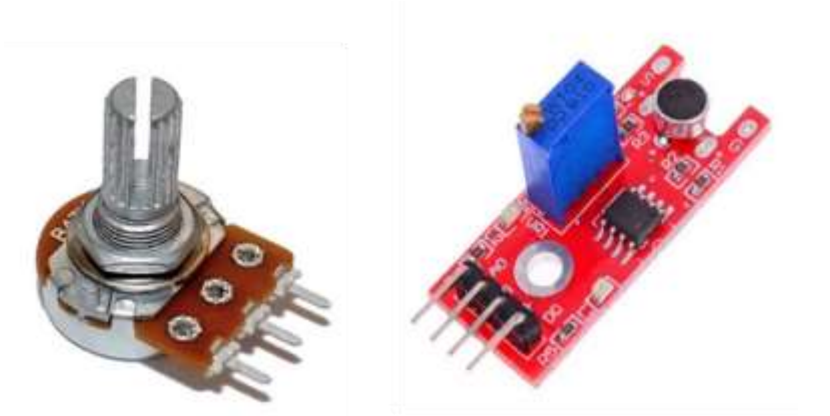

Figure 6. Variable Resistor and Sound Sensor Input Devices

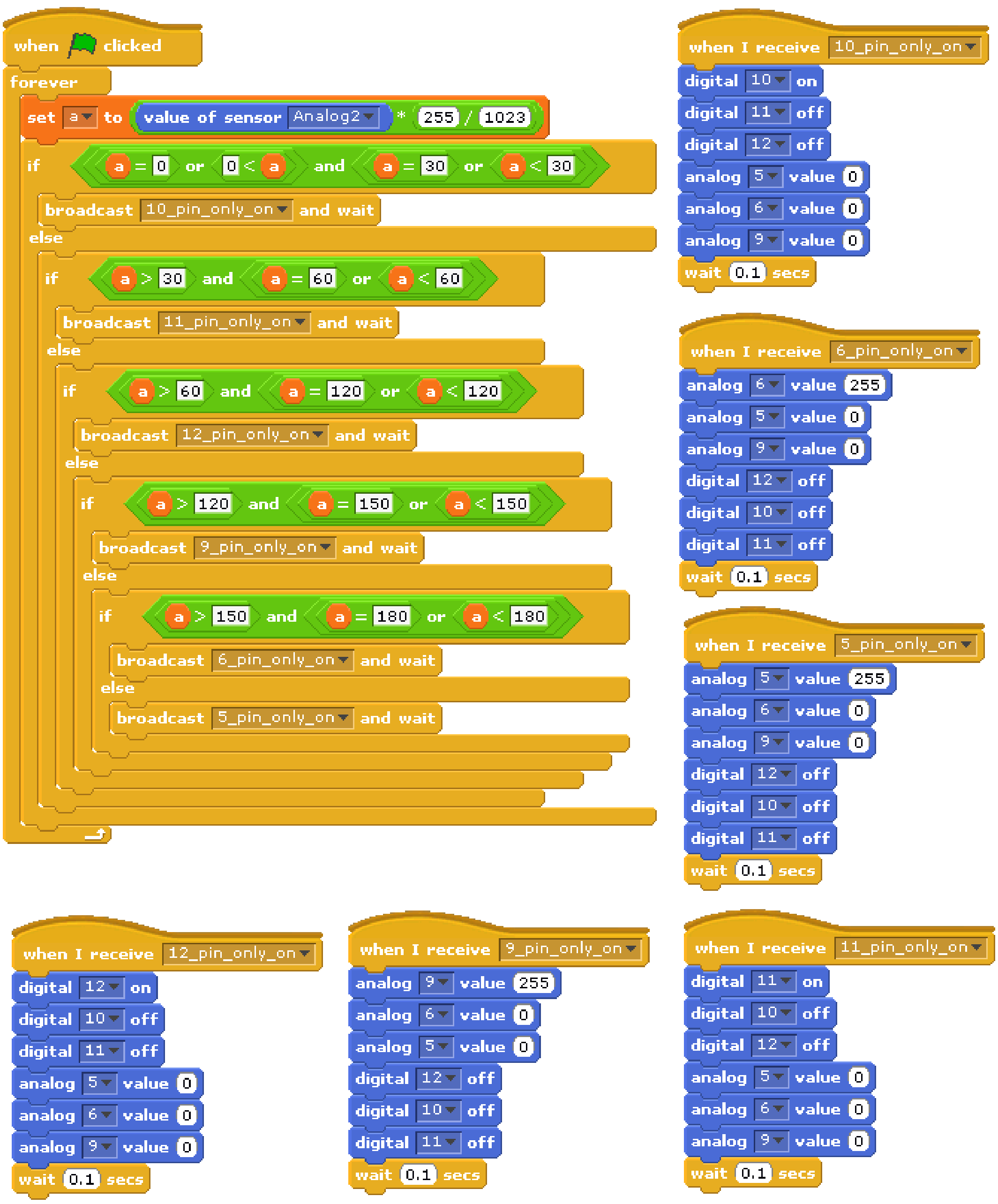

Figure 7. A Program to Express a Digital Clockwise Operation (dclockwise) According to Measured Sound Values

There are anticipated difficulties in teaching with only scratch software, an educational programming language. Students can understand how algorithms are used to represent problem-solving processes for the examples presented in the software textbook.

However, in implementing these algorithms with scratch software, if the students are familiar with the function and usage of each command block, the class will proceed easily. Otherwise, students cannot understand easily the difference between the natural language level algorithm and the implemented real program. In the software textbook, problem-solving processes for the examples are expressed finally into algorithms with natural language codes [1-5].

Therefore, students can understand how to use command blocks used in scratch, especially control blocks such as iterations, conditions, and logic operations, through physical computing class where it is easy to find logical errors. After that, it is effective to learn the software textbook which introduces the solution examples in our real life.

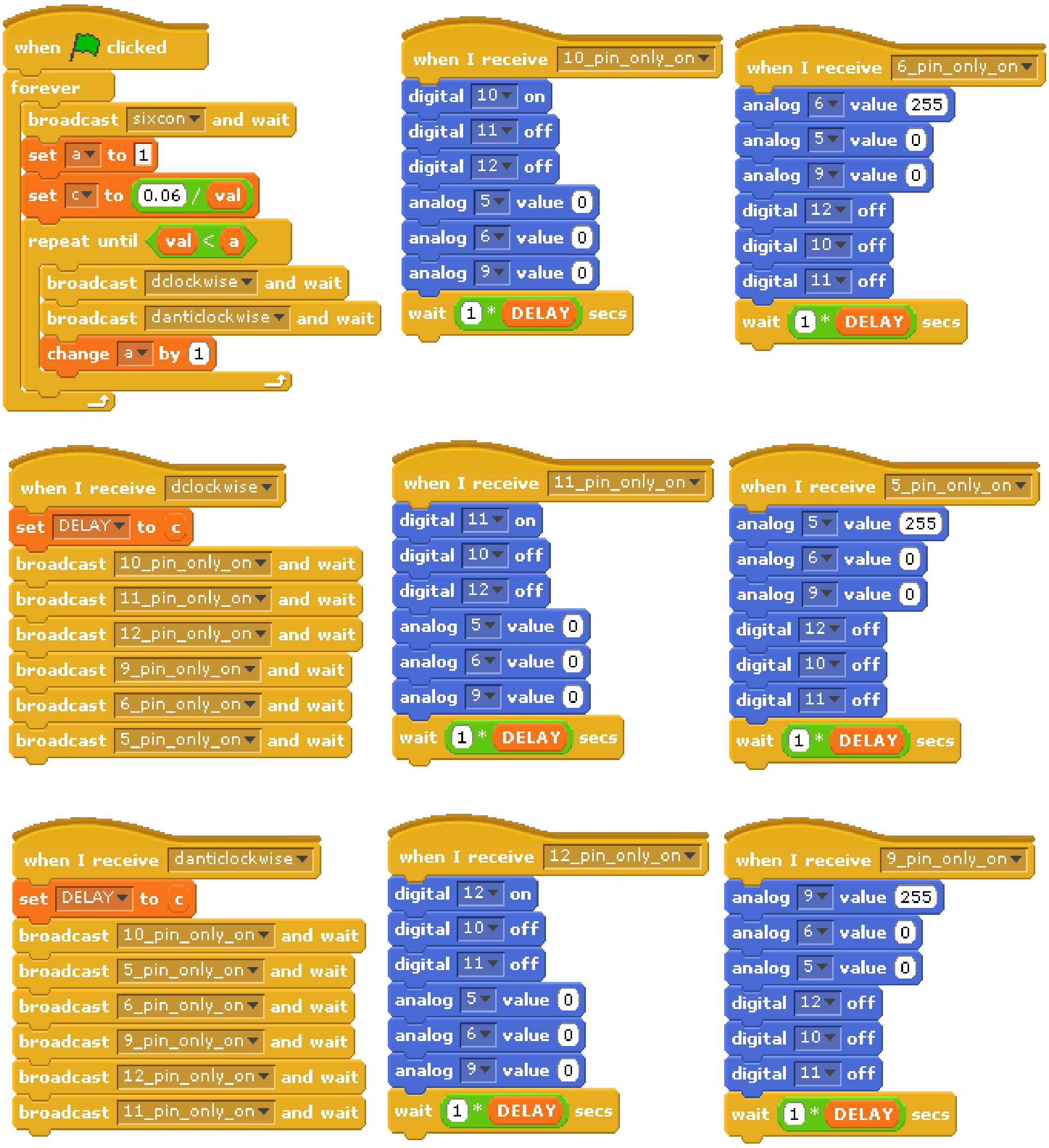

Figure 8. A Digital Clockwise/Anticlockwise Program (dclockwise/danticlockwise) According to Measured Sound Values

The steps required for physical computing education are as follows. Teacher presents several short and simple programs. After executing the program as it is, it is necessary to develop Q&A activity about physical results. Then, the teacher provides experience based on the physical results that will appear when a part of the program is modified. And the teacher trains the ability to solve similar problems and apply them to new problems. The teacher provides programs with various amounts or complexity depending on the level of learners and increases difficulty and complexity to solve problems occurred in real life. The teacher provides an opportunity to complete a single program, either alone or in groups, from start to finish. If an error occurs, the teacher let the learner solve it himself and answers questions.

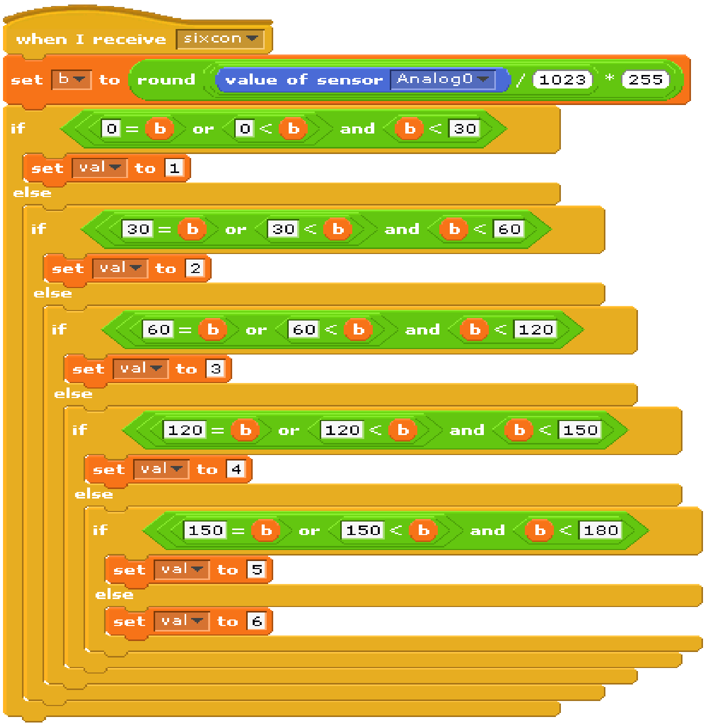

Figure 9. Sixcon Function to Control the Number of Repetition Times and whole Operation Time According to Measured Sound Values

4. Conclusion

The proposed EPL block coding examples included a problem-solving strategy of the top-down or bottom-up concept. Furthermore, proposed EPL block coding examples included a calculation technique using the variable data structure. Furthermore, proposed physical computing examples used the Arduino board for the scalable ability to develop a real system. By using the proposed hierarchical block coding algorithms, it is possible to educate the problem-solving strategy using software and to enhance the understanding of students about data storage and stored data used for the calculation technique. For future works, we will use the sketch C language to teach and develop physical computing programs for secondary and high school students.

Author Contributions

Kyeong Hur wrote the manuscript. Won-Sung Sohn designed experiments. Kil Young Kwon provided technical support.

Acknowledgments

This paper is a revised and expanded version of a paper entitled ‘Software Education using a Physical Circuit’ presented at 2017 1st International Workshop on Cultural and Technological Exchange and Mutual Development of The Pacific Rim Countries (CuTEMD 2017), December 17, 2017, Namseoul University, Cheonan, Korea.

References

[1] A. Elzamly, B. Hussin, S. S. A. Naser and Mohamed Doheir, “Classification of Software Risks with Discriminant Analysis Techniques in Software planning Development Process”, International Journal of Advanced Science and Technology, vol. 81, August, **(2015)**, pp. 35-48.

[2] G. Chandrika, “Study on Software Reliability and Reliability Testing”, Asia-pacific Journal of Convergent Research Interchange, vol. 1, no. 1, **(2015)**, pp. 7-20.

[3] S. H. Sung and P. Z. Khan, "Quantitative and Qualitative Approach for IT Risk Assessment", Asia-pacific Journal of Convergent Research Interchange, vol. 1, no. 1, **(2015)**, pp. 29-35.
[4] S. M. Shin and Sk. Uroosa, "Predicting Software Reliability Using Particle SWARM Optimization Technique", Asia-pacific Journal of Convergent Research Interchange, vol. 1, no. 3, **(2015)**, pp. 17-30.
[5] M. S. Srinivas and G. Pradeepini, "Feature Selection based Neural Networks for Software Defect Prediction", International Journal of Security Technology for Smart Device (IJSTSD), vol. 1, no. 1, **(2014)**, pp. 7-12.
[6] J. Tae, "Development of STEAM Education Program Utilizing IoT Teaching Aid for Middle School Students and Its Application", International Journal of Computer Science and Information Technology for Education (IJCSITE), vol. 2, no. 1, **(2017)**, pp. 15-20.
[7] J.-H. Eom, "Problems and Improvement of the Curriculum for Effective Cyber Security Education and Training", Journal of Security Engineering, vol. 12, no. 4, **(2015)**, pp. 337-350.
[8] Y. J. Lee, J. W. Hwang, D. M. Hwang, K. M. Yun, B. G. Lee, S. T. Yu and I. Y. Moon, "Design and Implementation of Software Program based on Web for Basic Electronic Circuit Practice", Journal of Practical Engineering Education, vol. 7, no. 2, **(2015)**, pp. 113-117.
[9] Y. J. Kim, "Development and Application of STEAM Education Program by Producing UCC", Journal of The Korean Association of information Education, vol.17, no.3, **(2013)**, pp. 339-346.
[10] S. F. Najam, M. L. P. Tan and Y. S. Yu, "General SPICE Modeling Procedure for Double-Gate Tunnel Field-Effect Transistors", Journal of Information and Communication Convergence Engineering, vol. 14, no. 2, **(2016)**, pp. 115-121
[11] Arduino Working Group, Italy, Available at: https://www.arduino.cc/.
[12] Scratch for Arduino, USA, Available at: http:// http://s4a.cat/.

Authors

Kyeong Hur, he is currently a professor in the Department of Computer Education at Gyeongin National University of Education, Republic of Korea. He was senior researcher with Samsung Advanced Institute of Technology (SAIT), Korea from September 2004 to August 2005. He received a M.S. and Ph.D. in Department of Electronics and Computer Engineering from Korea University, Seoul, Korea, in 2000 and 2004, respectively. His research interests include; computer network designs and computer science education.

Won-Sung Sohn, he received the B.S. and M.S. degrees in Computer Engineering from Dongkuk University in 1998 and 2000 and the Ph.D. degree in Computer Science from Yonsei University in 2004. From 2004 to 2006. He was a postdoctoral associate in the Computational Design Laboratory at Carnegie Mellon University. He is currently a professor at Department of Computer Education, Gyeongin National University of Education. His research interests include educational design research, human-computer interaction and computer education.

Kil Young Kwon, she is currently an assistant professor in the Department of Family Medicine at Eulji General Hospital and Eulji University, the Republic of Korea. She served as Internship, Resident & Fellow in the Severance Hospital at Yonsei University College of Medicine, Republic of Korea from March 2001 to February 2006. She received an M.S. International Health Science in the Department of Public Health, Yonsei University and Ph.D. Integrated Medicine in the Department of Medicine, Cha University, Seoul, Korea, in 2008 and 2015, respectively.

