The Journal of Education, 2(1), 25-35
June 2019
http://dx.doi.org/10.25020/JoE.2019.2.1.25

Development of a Physical Computing SW based on Computational Thinking

Hur, Kyeong*
Dept. of Computer Education, Gyeongin National University of Education, Rep. of Korea, professor

ABSTRACT

In physical computing, the input, processing, and output results of digital and analog sensors included in Arduino are different from the conventional programming, so that it is possible to correct programming errors quickly and improve the learning interest and satisfaction of the learners. However, there is a lack of physical computing SW (Software) development examples based on a computational thinking model for elementary and secondary school students. In this paper, we propose an example of SW development required for physical computing with Arduino MIMO (Multiple Input/Multiple Output) systems. This SW was developed in accordance with predefined SW-making steps in a CT (Computational Thinking) model. We used s4a (Scratch for Arduino) 1.5 software by considering elementary and secondary students. This proposal can be applied to various physical computing education classes to enhance the understanding of CT processes.

Keywords: Arduino, Computational Thinking Process, Maker Education, SW Education, Scratch for Arduino

Ⅰ. Introduction

According to the Korean curriculum changed in 2015, SW (Software) education will

* First Author: Hur, Kyeong(khur@ginue.ac.kr)

be executed in elementary, middle and high schools from 2018. SW education focuses not only on acquiring ICT skills but also on effectively solving problems in real life based on the CT (Computational Thinking) (Korea Ministry of Education, 2015; Wing , 2006; Kim, 2018). Furthermore, a new IoT (Internet of Things) era is coming, and the fields that existed independently became to have the convergence characteristic on the computing base. Naturally, most jobs require the CT literacy that has begun to be emphasized as a core competence required to live in the future (Yang, 2017; Wing, 2006; Wing, 2008).

This CT literacy is also required for the maker education in which students demonstrate their creativity by making new things with electronic devices (Maker Education Working Group, 2018; Jung, 2018). Arduino is a representative open-source hardware adequate for physical computing and maker education. According to the Arduino official website (Arduino Working Group, 2018), a variety of devices can be developed by connecting electrical circuits to input and output pins. In physical computing education, students can correct the programming errors quickly. And it provides classes of designing creative prototypes to solve real-life problems.

However, there is a lack of physical computing SW development examples based on a CT model for elementary and secondary school students. CT training projects should include the use of abstraction, decomposition, pattern recognition, and algorithmic thinking techniques along with adequate structures for data storage (Wang, 2018; Li, 2017).

In this paper, we propose an example of SW development required for physical computing with Arduino MIMO (Multiple Input/Multiple Output) systems. This SW was developed in accordance with predefined SW-making steps in a CT (Computational Thinking) model. We used s4a (Scratch for Arduino) 1.5 software by considering elementary and secondary students (Scratch for Arduino, 2018; Malan, 2007). A CT model used in this paper and the physical computing SW-making steps will be explained in the following Sections.

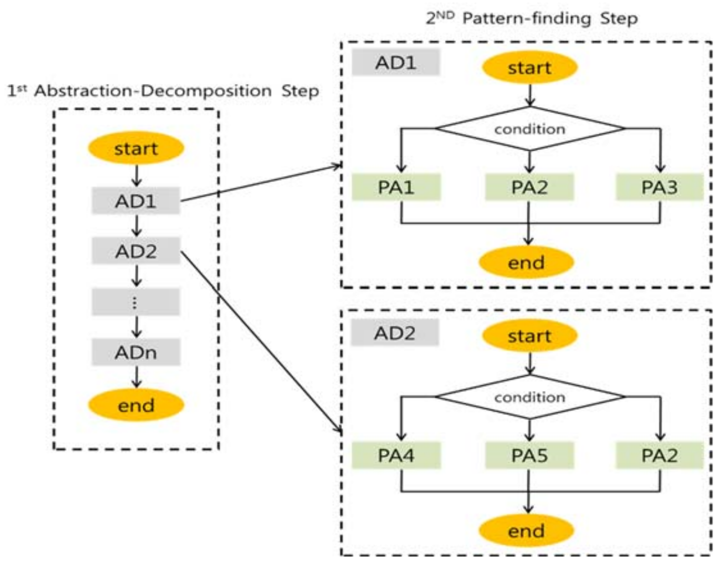
Ⅱ. SW-making model based on CT

For this SW-making based on CT, there are essential four thinking abilities. First, the decomposition ability is to break up a large problem into small problems. Second, pattern recognition ability is to find the similarity between SW processes in small problems. Third, abstraction thinking ability is about focusing only on the important solutions in which the relevant details are included. Finally, algorithmic thinking ability is to complete a sequential, repetitive, and conditional solution, adequately to each cleaved problem (Yang, 2017; Wing, 2006; Wing, 2008).

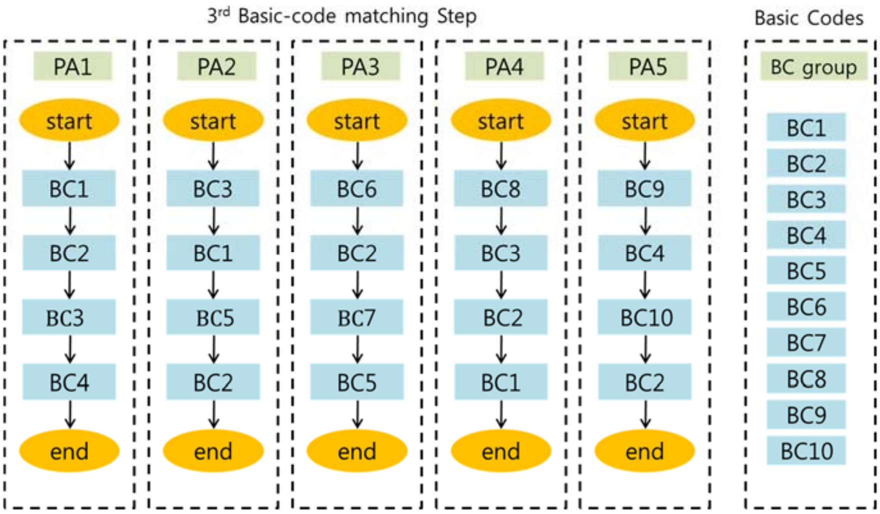
In Figure 1, the 1st Abstraction-Decomposition Step is explained. This model appears in the first problem-solving process. Here, each AD_n block must have a function of receiving an input and making output through conditional processes. If students know the inner functions in the AD_n blocks and complete the first problem-solving process, it proves that they have abstraction thinking and problem-decomposition abilities.

In the 2^{ND} Pattern-finding Step for AD_1 and AD_2 blocks, there are PA_n blocks. This PA_n block is a function execution block without a conditional process. A function execution block denoted by PA_n is defined by collecting BC_n codes that perform corresponding low-level roles in a program. Efficient use of common PA_n functions can reduce duplicate codes and reuse of already created codes. These AD_n blocks are made of PA_n function blocks. In Figure 1, while analyzing the AD_n blocks, some PA_n function blocks, which are commonly used through pattern recognition, can be found to be efficiently reused.

In the 3^{rd} Basic-code matching Step, these PA_n blocks are made of BC_n blocks which are the lowest layer codes. In Figure 2, while analyzing the PA_n blocks, some BC_n blocks, which are commonly used through pattern recognition, also can be found to be efficiently reused. In this order, from creating top-level AD_n blocks to bottom BC_n block group represents a top-down algorithmic thinking process. On the contrary, from creating bottom BC_n blocks to top-level AD_n block group represents a bottom-up algorithmic thinking process. We applied this SW-making model and presented its educational use-case of a physical computing SW development upon an Arduino MIMO system using the s4a 1.5 SW (Scratch for Arduino, 2018).



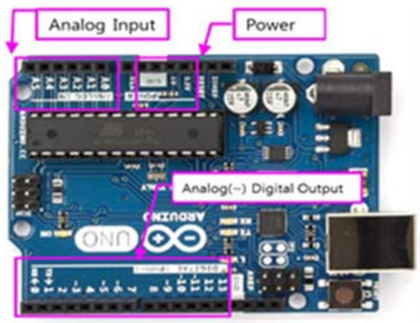
[Figure 1] Relationship between the 1st Abstraction-Decomposition Step and the 2nd Pattern-finding Step



[Figure 2] Relationship between the 2nd Pattern-finding Step and the 3rd Basic-code matching Step

Ⅲ. A physical computing SW development based on CT model

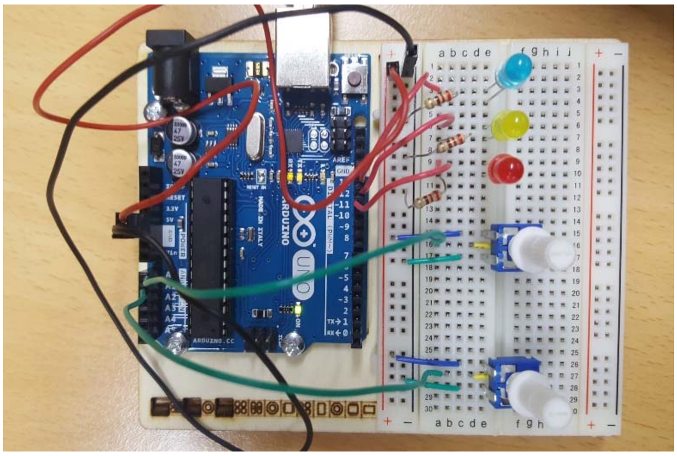
1. Arduino MIMO system hardware Configuration



[Figure 3] Analog and Digital input/output pins in Arduino Uno board

Figure 3 explains analog and digital input/output pins in Arduino Uno board (Arduino Working Group, 2018). Using this information, we configured an Arduino two-input three-output system as shown in Figure 4. In Figure 4, a blue LED (Light Emitting Diode) with a

220Ω resistor is connected to pin 10. A yellow LED is connected to pin 11. And a red LED is connected to pin 12. Therefore, there are three output devices connected to the Arduino board. In s4a SW, 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. In Figure 4, a variable-resistor is connected to the A0 pin, and the other variable-resistor is connected to the A1 pin. Therefore, there are two input devices connected to the Arduino board. The values measured from an analog input device are ranged from 0 to 1023 integers.



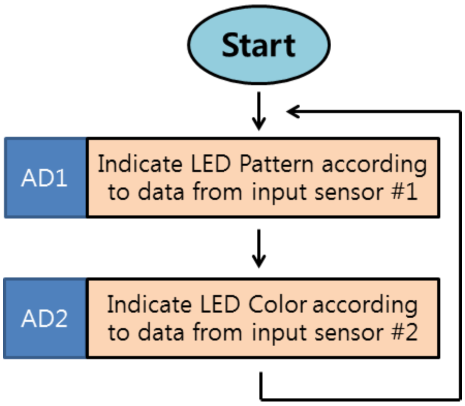
[Figure 4] An Arduino two-input three-output system used in this paper

2. AD_n block group Definition and the 1st Abstraction-Decomposition Step

By using the two-input three-output Arduino system shown in Figure 4, an AD_n block group is set as Table 1. Other AD_n block groups also can be set differently. From this AD_n block set, an example of the 1st Abstraction-Decomposition step is made as in Figure 5. The 1st Abstraction-Decomposition step is always executed in an infinite loop.

<Table 1> Components of an AD_n block group example

Block	Block Name
AD_1	LED Pattern Indicator according to data from input sensor #1
AD_2	LED Color Indicator according to data from input sensor #2



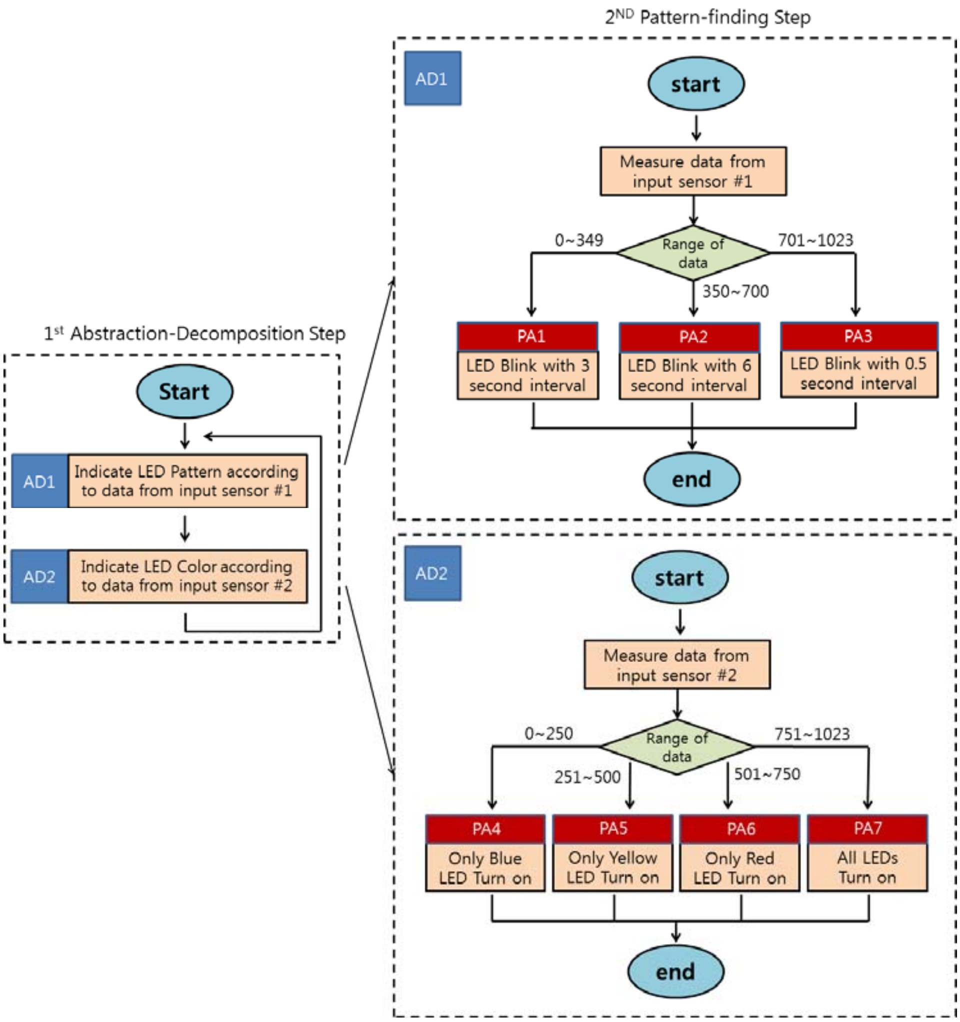
[Figure 5] An example of the 1st Abstraction-Decomposition step

3. PA_n block group Definition and the 2nd Pattern-finding Step

From the example of the 1st Abstraction-Decomposition step shown in Figure 5, an PA_n block group is set as Table 2. Other PA_n block groups also can be set differently. From this PA_n block set, an example of the 2nd Pattern-finding step is made as in Figure 6. In this example, we cannot find the common PA_n block reused to reduce duplicate codes.

<Table 2> Components of a PA_n block group example

Block	Block Name
PA_1	LED Blink with 3 second interval
PA_2	LED Blink with 6 second interval
PA_3	LED Blink with 0.5 second interval
PA_4	Only Blue LED Turn on
PA_5	Only Yellow LED Turn on
PA_6	Only Red LED Turn on
PA_7	All LEDs Turn on



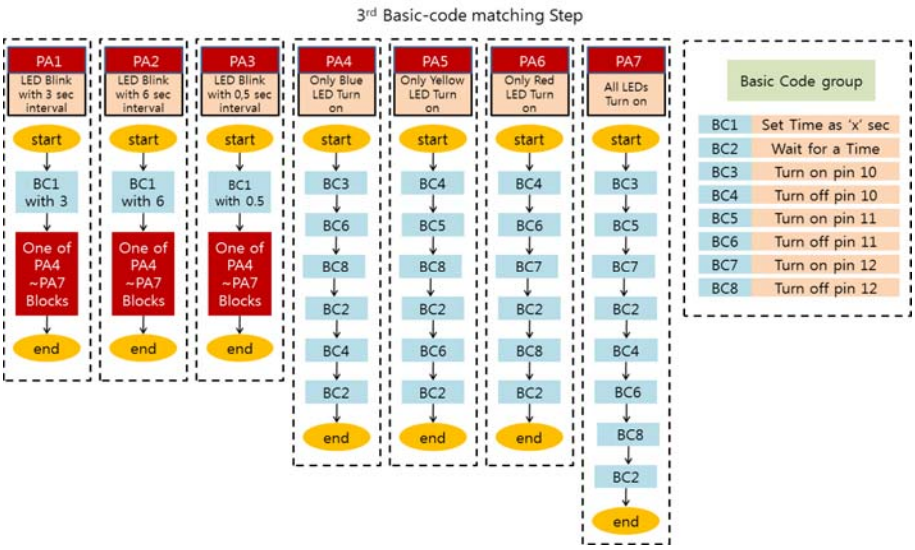
[Figure 6] An example of the 2nd Pattern-finding step

4. Basic Code group Definition and the 3rd Basic-code matching Step

From the example of the 2nd Pattern-finding step shown in Figure 6, an BC_n block group is set as Table 3. Other BC_n code groups also can be set differently. From this BC _n block set, an example of the 3rd Basic-code matching step is made as in Figure 7. In this example, there are many common BC_n codes to compose PA_n blocks. In the codes of PA_1~PA_3 blocks, we can find that PA_4~PA_7 blocks can be reused to reduce duplicate codes through CT pattern recognition.

<Table 3> Components of a BC_n code group example

Code	Role of Code
BC_1	Set Time as ‘x’ sec
BC_2	Wait for a Time
BC_3	Turn on pin 10
BC_4	Turn off pin 10
BC_5	Turn on pin 11
BC_6	Turn off pin 11
BC_7	Turn on pin 12
BC_8	Turn off pin 12

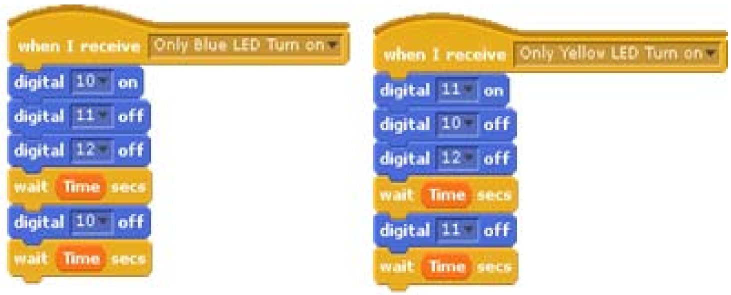
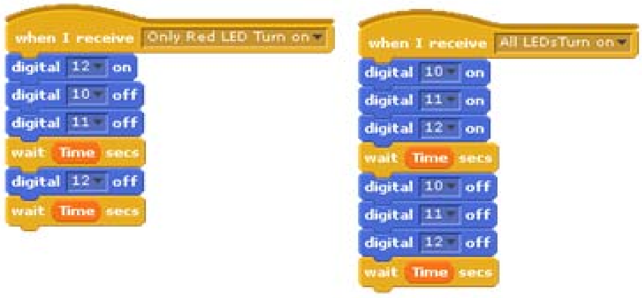
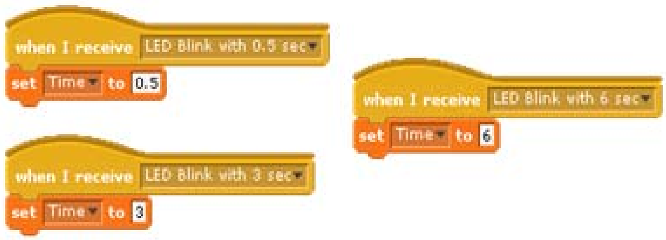


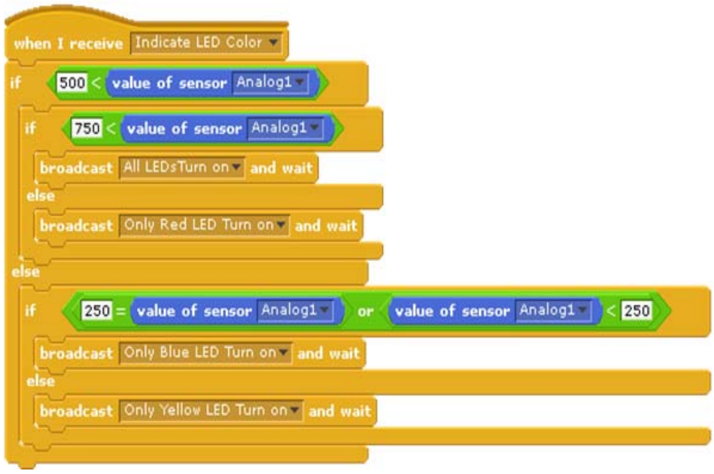
[Figure 7] An example of the 3rd Basic-code matching step

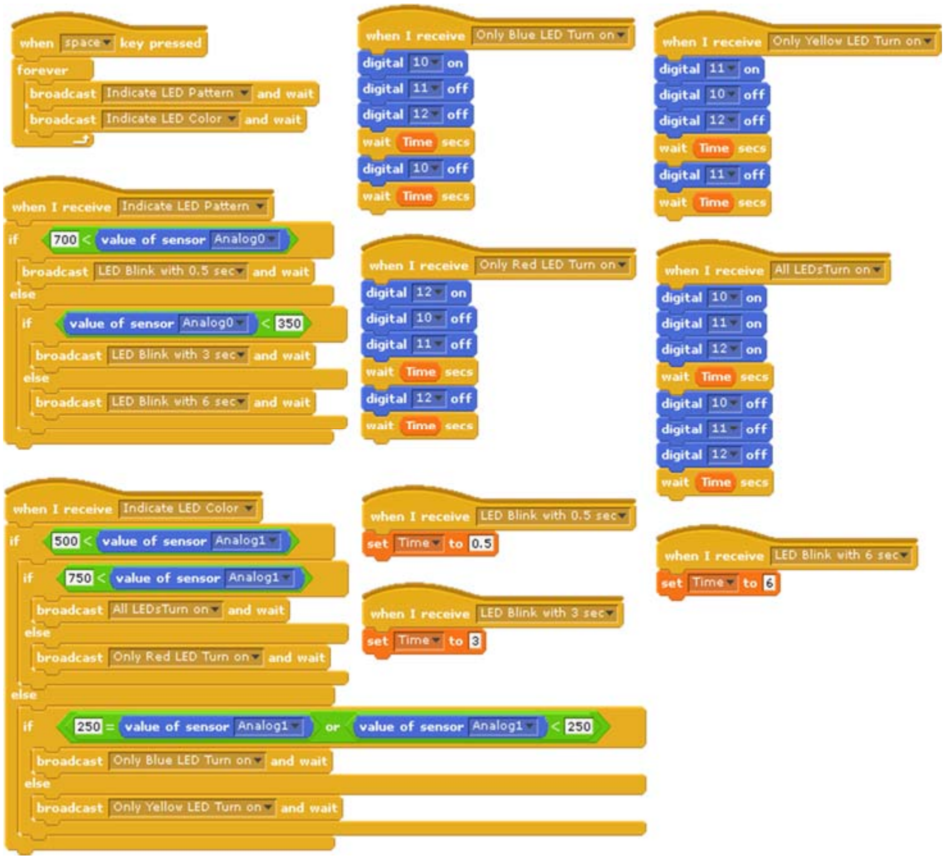
5. Implemented Real s4a Block Codes

Based on the SW-making model based on CT, three SW diagrams are defined for the two-input three-output Arduino system shown in Figure 4. By combining and tracing three SW diagrams, the real s4a scratch block codes are completed. In Table 4, the s4a blocks are arranged according to categories in the model. In these real s4a block codes, we can find the predefined BC_n codes, PA_n blocks, and AD_n blocks. From this phenomenon, it is proved that the presented SW-making model and example based on CT is available and effective materials for SW education. In Figure 8, the all blocks and codes are assembled to show a SW structure based on CT where one main program and nine subprograms operate separately.

<Table 4> the s4a blocks arranged according to categories

Block Model	Real s4a Block codes
PA4, PA5	
PA6, PA7	
PA1, PA2, PA3	
AD1	

AD2	



[Figure 8] overall s4a codes according to the top-down SW-making CT model

Ⅳ. Conclusion

In this paper, we focused on the physical computing with Arduino MIMO (Multiple Input/Multiple Output) systems. By expanding this SW-making example with two-input and three-output Arduino system, students can control easily the more complex Arduino MIMO systems. The explained SW-making model with three steps has the four essential CT components, adequately for elementary and secondary school students. Using this model helps students to understand CT processes. This proposal can be applied to various physical computing education classes. For future works, based on the SW-making CT model, an evaluation method to measure the CT literacy is required.

References

Korea Ministry of Education. (2015). *Curriculum for Informatics Education*.

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM, 49*(3), 33-35.

Kim, H. (2018). A Study on Instructional Model based on the Computational Thinking for Informatics Education, *Journal of The Korean Association of Information Education, 22*(1), 1-8.

Yang, S. (2017). Necessity of Computational Thinking, *Korea Information Processing Society Magazine*, *24*(2), 4-12.

Wing, J. M. (2008). Computational thinking and thinking about computing, *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences, 366*(1881), 3717-3725.

Maker Education Working Group. (2018). USA, available at http://makered.org/

Jung, E.H. (2018). Maker education that all learners become creators, *Seoul Education webzine, 60*(232).

Arduino Working Group. (2018). Italy, available at https://www.arduino.cc/

Wang, X., and Kim, H.-C. (2018) Text Categorization with Improved Deep Learning Methods, *Journal of information and communication convergence engineering, 16*(2), 106–113.

Li, J., Shin, S.Y., and Lee, H.C. (2017) Text Mining and Visualization of Papers Reviews Using R Language, *Journal of Information and Communication Convergence Engineering*, *15*(3), 170-174.

Scratch for Arduino. (2018). USA, available at http://s4a.cat/

Malan, D.J., and Leitner, H.H. (2007). Scratch for budding computer scientists, *SIGCSE Bulletin*, *39*(1), 223-227.