Ilkogretim Online - Elementary Education Online, Year 2021; Vol 20 (Issue 4) pp. 1856-1864
http://ilkogretim-online.org.tr
doi 10.17051/ilkonline.2021.04.208

The effects of algorithm-based software education using micro:bit on elementary school students' creativity

***Yong Min Kim,** *Jeju City Office of Education, Jeju, Korea, mega11@korea.kr* ORCID: 0000-0003-1570-0926
Dong-Wan Kang, *Jeju National Univercity, Jeju, Korea, gengsenge@hanmail.net*
**corresponding author*

Abstract. This study analyzed the effects after developing and applying a software education program using microbits based on the ADDIE development model as an educational plan for improving the creativity of elementary school students. The educational program was developed based on the results of pre-requirement analysis conducted with 40 elementary school teachers. In order to verify the effectiveness of the developed education program, 42 participants of the educational donation program conducted by J University were held for 6 days, 7 sessions per day, for 24 students. As a result of conducting pre and post-tests using Torrance's TTCT test and analyzing educational effects, it was found that the software education program developed in this study had a positive effect on the creativity of elementary school students.

Keywords: Creativity, Micro:bit, Software Education, Physical Computing

Received: Accepted: Published:

INTRODUCTION

A "software-oriented society" has emerged, a society in which software is widely used in individuals, companies, and governments to improve the quality of life and to continuously improve the competitiveness of companies and governments.

The "software-oriented society" is a core value declared by the government, and the government emphasizes the importance of software and promotes the promotion and development of software (Software Policy and Research Institute, 2014). Overseas, we are rapidly promoting software education to respond to these changes and nurture talents who will lead a software-oriented society. Software education is supported through national policy in many countries, including the UK, Finland, Estonia, the United States, and India. Although there are differences between countries, most of them are applied from the elementary school level, and the existing information and communication technology (ICT) education has been converted to programming education (Wells, 2012). In Korea, the curriculum was revised so that students in grades 5-6 are required to complete software education from 2018 in middle schools and from 2019 in elementary schools(Ministry of Education, 2015).

Software education means to creatively solve a given problem based on the theoretical basis of computer science, and learners develop creativity through the process of creating projects by imagining what they want through software education and sharing ideas with others Can be made (Brennan & Resnick, 2013).

Such software education is being conducted in conjunction with various tools. In particular, it is a physical that collects data by directly connecting microcomputers such as 'Arduino', 'Raspberry Pi', and 'Micro:bit' with various sensors using various types of programming languages such as 'Scratch' and 'Python'. Computing has unlimited potential as a visual and convenient learning tool.

Micro:bit, which was developed by the BBC in the UK for learning with several companies around the world, provides 25 external connection pins while equipped with basic IO devices, providing expandability for connection with various IO devices. It is also possible to simply connect them using tongs. In addition, it is possible to configure an Arduino-like environment by using an extension pin connector(Kim, 2020).

As such, micro:bit is a teaching tool that enables physical computing without connection of special tools or sensors, and is widely used all over the world because of its infinite scalability, easy and simple accessibility, and low price(Yoon *et al.,* 2020). However, there are not many domestic and international research cases on how micro:bits affect learners' creativity in software education activities compared to their utilization. Therefore, this study attempts to analyze what changes in learners' creativity occur after utilizing microbits in educational activities, which have various advantages and can be handled relatively easily by elementary school students.

First, through the pre-requirement analysis, the subject and educational content were selected in the direction of using micro:bit, and an educational program was developed. Among the 4th, 5th, and 6th grade elementary school students, 24 volunteer samples were used. After conducting an education program using micro:bits, a creativity test was conducted, and as a test tool, Torrance's Torrance Tests of Creative Thinking (TTCT) test paper figure A type was used.

Theoretical Background

Software education

Ability to solve real-life problems with computational thinking based on information ethics awareness and attitude through three areas: life and software, algorithms and programming, and computing and problem solving in order to nurture'creative and convergent talents with computational thinking ability' It refers to education that fosters talented people (Kim, 2016).

Software education in elementary school focuses on solving real life problems with computational thinking based on information ethics consciousness and attitude rather than program development competency. The educational method is designed so that learners can recognize the meaning and importance of computational thinking skills and check the value through practice-oriented education, not knowledge transfer-oriented education.

Micro:bit

Micro:bit is an ARM-based embedded system developed in 2015 for software education in the UK. Immediately after development, it was distributed free of charge to students aged 7-12, and is currently spreading to the world as a public broadcasting BBC and educational embedded system (Schmidt, 2016).

micro:bit is a small embedded system of about 40*50mm with 256KB flash memory and 16KB RAM based on ARM Coretex-M0. It was developed with an emphasis on software education and supports various hardware devices and various development tools necessary for programming (Ball *et al.,* 2016).

micro:bit appears to show optimal performance when writing programs of less than 100 lines, and accordingly, the main targets for use are elementary school students who have not been in contact with Software education for a long time (Bradley & Gibson, 2017).

Creativity

The most representative way to classify the perspectives of many scholars on creativity is to classify them into cognitive, affective, and integrated perspectives (Hong, 2006).

Among them, the perspective that sees creativity as a cognitive aspect is a perspective that sees the basic elements of creative activity as problem-solving ability and knowledge, and representative scholars include Guilford and Tornace. Guilford (1959) explained that creativity is the ability to produce new thoughts, and considered it as one of human intellectual abilities (Guilford, 1959).

Torrance (1978) explained that it is a process of combining creative thinking, recognizing insufficient factors, obstructions, etc., establishing hypotheses for this, verifying, correcting, or reverifying, and delivering the final result. In his developed and widely used Torrance Tests of Creative Thinking (TTCT), cognitive factors such as Fluency, Originality, and Elaboration are regarded as constituent factors of creativity (Torrance, 1978).

Programming education

Programming refers to the work of arranging formulas or tasks appropriately in a computer, determining the order, and rewriting them with computer commands, and writing commands (codes) is also called coding (Tugun *et al.,* 2017).

Programming can be compared to the process of building a building. Just as a well-designed blueprint is needed to build a sturdy and safe building, algorithm learning is essential to solve problems that occur in the programming process. Therefore, programming education is more effective in learning algorithms and learning strategies for problem solving (Choi, 2015).

Analysis of prior research

The value of lessons using robot programming can be summarized into five categories. First, since robot programming learning provides a collaborative learning environment among learners, it can provide an interactive and experiential learning environment for learners. Second, it provides a learning environment in which learners can reflect and think on their own. Third, robot programming is advantageous in providing a convenient and integrated environment to provide a play-oriented learning environment. Fourth, the nature of interdisciplinary robots shows the possibility of linking with various subjects, which leads to a positive attitude toward computer science and programming and is effective in inducing intellectual curiosity. Fifth, even from the perspective of convergence education, robot programming becomes a useful learning method (Seo & Lee, 2010).

Noh(2017)'s study, it was found that software education using robots improves students' computing thinking, creativity, and immersion. In addition, it was confirmed that SW education using robots is a necessary class for elementary school students and is useful for enhancing high school thinking ability of elementary school students. In addition, if a robot is used for SW education, the abstract problem-solving process can be supplemented with specific tools, and the result of programming can be checked directly through the robot. Therefore, learners can be immersed in solving problems in reality, and ultimately, learning effects can be improved (Noh, 2017).

Park(2016)'s study, the following conclusions were drawn through a study using an embedded computing electronic kit. First, the teaching topics and activities using the embedded computing electronic kit are highly useful for learning in the creative and subject areas in the elementary school course, and are suitable for practical classes. Second, the 2015 revised curriculum can be viewed as a new approach and alternative to software education applied to practical subjects (Park, 2016).

Jang(2017) 's study, the following effects were confirmed through education using physical computing. First, as a result of using physical computing in education, it was possible to observe significant improvement in fluency, originality, flexibility, and elaboration ability among the sub-elements of creativity. Second, it was found that students who participated in education thought that both cognitive improvement and social competency for their own collaboration skills had grown. Third, most of the students who participated in the education responded that their creativity was improved and they made efforts to solve problems in life in various ways.

Recently, educational cases using microbits have also been announced in Korea. In particular, cases in which students' computing thinking skills and coding skills have improved in education using microbits have been reported, and cases supporting ease of use compared to other physical computing tools have also been reported(Yoon *et al.,* 2020).

In addition, when looking at comparative studies with other physical computing devices, education using microbits was more efficient in problem-solving than Arduino-based education. As an advantage, it has been reported that microbit-based IoT coding education has a positive effect compared to Arduino-based(Kim, 2020).

From these responses, it can be seen that education using physical computing is helpful in improving the creativity and collaboration capabilities of elementary school students (Jang, 2017).

METHODS

In this study, an educational program was developed according to the ADDIE model, which is a general model of educational program development. The stages of analysis, design, development, execution, and evaluation were carried out in order, and educational programs were designed, developed, and executed according to the needs analysis, and the results were evaluated through a creativity test.

Analysis:
- Learner analysis
- Needs analysis

Design:
- Teaching Strategy and Media Selection
  - ▶ Individual project learning
  - ▶ Micro:bit
- Evaluation tool design
  - ▶ Learner's creativity test sheet (TTCT figure A type)

Development:
- Teaching·Learning Course Proposal (42 sessions)
- Student activity sheet (42 sessions)

Implementation:
- Conducting classes

Evaluation:
- Learner Creativity Test

Analysis

In this study, Rossett's needs analysis model was used. The Rossett model is an education needs analysis model used in corporate education, and provides a guide that is easy to apply to the implementers of the needs analysis. The needs analysis was conducted on 40 incumbent elementary school teachers.

Table 1. *Preferred software education area*

Educational programming language	Unplugged	Pysical computing
2(5.1%)	17(43.6%)	21(51.3%)

As a result of the priority survey on the preferred software education area, it was found that the preference for physical computing was the highest, followed by unplugged.

Table 2. *Expected effects through software education*

Computational thinking	Problem solving	Creativity	Etc
23(58%)	10(25%)	6(15%)	1(2%)

The expected effects from software education were in the order of computing thinking ability, problem solving ability, and creativity.

Table 3. *Learning method for software education*

Lecture/Practical	Team project	Individual project
15(36.8%)	19(47.4%)	6(15.8%)

As a result of research on learning methods for software education, they prefer team project type, lecture/practice type, and individual project type in order.

As a result of analyzing the demand analysis results, we were able to derive the following demands. First, the interest in physical computing is high, and it is believed that using physical computing tools will be effective. However, as Kim(2016)'s study, when a variety of physical computing tools use sensors to construct circuits, a lot of time is wasted in circuit configuration, so it takes a lot of time only in the assembly process and preparation process. Therefore, the circuit configuration is simple and easy-to-program micro:bit is selected as an application device and applied to the program so that the improvement of creativity through software education becomes the priority purpose. Second, in this study, the lecture/practice type is the basic form of education, but the program is configured so that students can plan and execute individual projects on their own.

Design

In order to design a software education program using micro:bits, an evaluation tool to verify the effectiveness of learning activities after selecting the learning content in consideration of the learning level according to the learner's grade and the difficulty of programming was designed.

The learning contents are shown in Table 4.

Table 4. *Learning content*

Hour	Learning content
1-5	Learning micro:bit fundamentals, micro:bit basic functions practice
6-10	Repeated practice using LED light, expression of algorithm using loop
11-15	Understanding operators, simplify algorithms using logical operators
16-20	Learning create an algorithm with conditions and operators Diversification of algorithms using variables
21-25	Learning creating a project plan for individual projects Creating individual project objects Creating a project plan for individual projects Creating individual project projects
26-30	Present your individual project work Input of post-test

Development & Implementation

The educational program developed in this study was put into a total of 42 lectures and practice for 6 days. The first day was focused on pre-test and orientation, and the last day was a post-test and final presentation of the project created by the learners to students and parents.

Seven classes, breaks, and lunch breaks were held for 40 minutes each day from 9 am to 3 pm, and tasks that could be solved were presented daily by utilizing the learned contents to maximize the learning effect. The ‘selection structure’ of the 2nd day education materials is as follows.

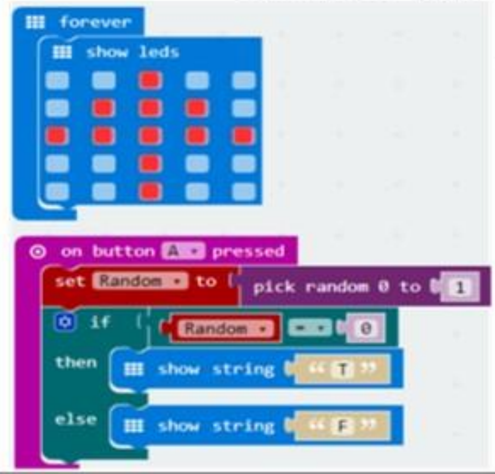
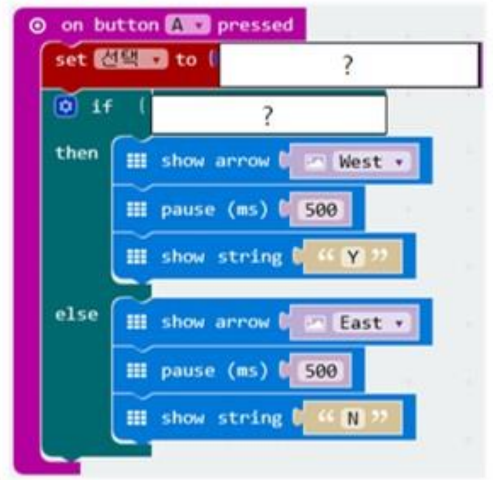
Solving Problems with Micro:Bit 4

● **Learning Objectives**

→ Problem situations related to micro-bit can be solved by using selections and variables.

1. Selection structure

Let’s use the Logic tab. ’If not A, the choice is B. Or, if A is correct, B can also be used.

Examples of programs using the selection structure	
	<Choose a winner> It is a program that shows the winners as T, F when the arrow direction comes out and the A button is pressed after pointing at another person.
	<Cham Cham Cham Game> This program prints Y if the user selects one of the A button and the B button, and if the computer matches the randomly selected direction, it prints N.

Think more
1) Using the selection structure, let's write a program that prints out one of the rock paper scissors by pressing the A button.
2) Let's find a situation that can be expressed as a selection structure among problems in life and express it in microbits.

FIGURE 1. *‘Selection structure” of the 2nd day educational materials*

Evaluation

Before conducting software education using micro:bits, a pre-test of creativity was performed on the experimental group, and a post-test on creativity was performed to verify the effectiveness of education after applying the education program.

Table 5. *Experimental design*

	Pre-test	Treatment	Post-test
class	O_1	x	O_2

X : Software education program
O_1 : Pre test(creativity)
⇒ normality test(Shapiro-Wilks test)
O_1, O_2 : Pre-Post test(creativity)
⇒ After the normality test, Paired sample T-test or Wilcoxon's signed rank test

RESULTS

Creativity change

In order to analyze whether algorithm-oriented software education using micro:bits influenced creativity, it was checked whether the results of the pre-test of creativity of the experimental group had a normal distribution.

Table 6. *Normality test(Creativity)*

Subscales	N	Mean	Std.Dev.	Max	Min	t value	p value
Fluency	24	112.17	16.426	138	68	.959	.411
Originality	24	104.33	19.544	146	51	.937	.589
Titles	24	91.04	32.792	146	0	.939	.159
Elaboration	24	81.04	12.882	105	65	.910	**.035***
Closure	24	76.21	26.145	118	0	.944	.195
Average	24	93.08	15.592	114	37	.791	**.000****
Index	24	94.58	16.487	118	37	.815	**.001****

*p<.05, **p<.01

As a result of conducting the Shapiro-Wilks normality test for the creativity pre-test, the significance of Elaboration, Average, and Index was .035, .000, and .001, respectively, indicating that the normality was not satisfied. In the remaining areas, the significance level was greater than the significance level of .05, confirming the normal distribution.

For 'Fluency', 'Originality', 'Titles', and 'Closure' that did not secure normality as a result of the normality test for the pre-test, 'Wilcoxon's signed rank test' was performed as shown in Table 8. As shown in 7, 'Paired sample T-test' was conducted.

Table 7. *Paired sample T-test(Creativity)*

Subscales	Test	N	Mean	Std.Dev.	t value	p value
Fluency	Pre	24	112.17	16.426	-2.981	**.007****
	Post	24	114.26	14.404		
Originality	Pre	24	104.33	19.544	-3.557	**.002***
	Post	24	107.61	17.453		
Titles	Pre	24	91.04	32.792	-1.543	.137

	Post	24	95.65	24.107		
Closure	Pre	24	76.21	26.145	-2.408	**.025****
	Post	24	79.48	24.755		

*p<.05, **p<.01

As a result of 'Paired sample T-test' in Table 7, 'Fluency' increased significantly at the significance probability of .007. In addition, 'Originality' was significantly increased at the significance probability of .002, and 'Closure' was significantly higher at the significance probability of .025, indicating that it was significantly improved in the three sub-factors excluding 'Titles'.

Table 8. *Wilcoxon's signed rank test(Creativity)*

Subscales	Test	N	Mean	Std.Dev.	t value	p value
Elaboration	Pre	24	81.04	12.882	-2.245	**.025***
	Post	24	91.04	13.766		
Average	Pre	24	93.08	15.592	-2.135	**.033***
	Post	24	97.65	14.093		
Index	Pre	24	94.58	16.487	-1.919	.055
	Post	24	96.13	25.770		

*p<.05

As a result of 'Wilcoxon's signed rank test' in Table 8, the significance probability of 'Elaboration' increased significantly from .025, and the significance probability of 'Average' was .033, indicating that there was a significant increase in score. The average score of 'Index' increased by 1.55 points from 94.58 beforehand to 96.13 after death, but the significance probability was analyzed to be .055, which did not improve significantly.

DISCUSSION and CONCLUSIONS

In this study, to improve creativity for elementary school students, an algorithm education program using microbits was proposed, and an education program was developed and applied according to the development stage of the ADDIE model. The program was applied for a total of 6 days in the form of intensive education during the vacation period, and the results of the pre- and post-test creativity were verified. As a result of the verification, among the sub-elements of creativity, 'Fluency', 'Originality', 'Closure', 'Elaboration' and 'Average' showed significant improvement.

However, there is a limit to generalization of the experimental group of this study because it cannot secure more than 30 participants required for correlation research. In addition, the educational program developed in this study has a problem in that it is impossible to analyze the correlation between whether the improvement of creativity is due to the influence of the educational program developed in this study by putting it in only the experimental group without the comparative group. In future studies, it is necessary to analyze the correlation between each factor of the study results by composing an experimental group and a comparative group for a large number of participants.

REFERENCES

Ball, T., Protzenko, J., bISHOP, J., Moskal M., & Halleux, P. D. (2016). Microsoft Touch Develop and the BBC micro:bit. 2*016 IEEE/ACM 28th International Conference on Software Engineering Companion (ICSE-C)*, 637-640.

Bradley, S. & Gibson, S. (2017). A study of Northern Ireland Key Stage 2 pupil's perceptions of using the BBC Micro:bit in STEM education. *The STeP Journal: Student Teacher Perspectives, 4(1),* 15-41.

Brennan, K., & Resnick, M. (2013). Imagining, Creating, Playing, Sharing, Reflecting: How Online Community Supports Young People as Designers of Interactive Media. In C. Mouza and N. Lavigne (eds.), *Emerging Technologies for Classroom,* Springer*. 253-268.*

Choi, J. W. (2015). *Puzzle-Based Algorithm Learning Model for Improving Computational Thinking for Informatics Gifted Students*. doctoral dissertation. Korean National University.

Guilford, J. P. (1959). Three faces of intellect. *American Psychologist, 14*, 469-479.

Hong S. J. (2006). *Intelligence and creativity : Psychological inquiry For Human intelligence*. Seoul: Yangseowon.

Jang J. S. (2017). *The Effect of STEAM program using physical computing for elementary students creativity and collaboration ability*. Master's Thesis. Gwangju National University of Education.

Noh. J. Y. (2017). *The Effects of SW Education Using Robot*. doctoral dissertation. Ewha Women's University.

Kim, H. S. (2016). A Study of the Direction for Developing Software Education Operating Guide. *Korean Association for Learner-centered Curriculum and Instruction, 16(8)*, 529-548.

Kim. S. Y. (2020). A Study on the Effectiveness of IoT Coding Education Using Micro:bit. *The Journal of the Korea Institute of Electronic Communication Sciences, 15(2)*, 363-370.

Ministry of Education (2015). Software Education Guideline.

Park, K. R. (2016). An Approach for Creative Activity Using of Embedded Computer and Software Developing Tools. *Journal of Korean Practical Arts Education, 22(1)*, 221-239.

Schmidt, A. (2016). Increasing Computer Literacy with the BBC micro:bit. *IIEEE Pervasive Computing, 15(2)*, 5-7.

Seo, Y. M. & Lee, Y. J. (2010). A Subject Integration Robot Programming Instruction Model to Enhance the Creativity of Information Gifted Students. *The Journal of Korean Association of Computer, 13(1)*, 19-26.

Software Policy and Research Institute (2014), Software Oriented Society: The Meaning and the Response, I*ssue Report 2014-003.*

Torrance, E. P. (1978). Giftedness in solving furture problems. Journal of Creative Behavior, 12(2), 75-86.

Tugun, V., Uzunboylu., & Ozdamli, F. (2017). Coding Education in a Flipped Classroom.. *Technology Education Management Infomatics (TEM Journal), 6(3)*, 599-606.

Yoon. J. A., Nam.Y. K., & Cho. W. S. (2020), Development and Application of a Science-SW Convergence Education Program Using Micro:Bit. *The Journal of Korean Association of Computer Education 23(6),* 77-87.

Kim. S. Y. (2020). A Study on the Effectiveness of IoT Coding Education Using Micro:bit. *The Journal of the Korea Institute of Electronic Communication Sciences, 15(2)*, 363-370.

Wells, D. (2012). Computing in Schools: Time to Move Beyond ICT? *Research in Secondary Teacher Education, 2(1)*, 8-13.

Williams, A. B. (2003). The qualitative impact of using LEGO MINDSTROMS robots to teach computer enginnering. *IEEE Transactions on Education*, 46(1), 206.