JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 23, No. 10, pp. 1959-1968, Oct. 2022

초등학생의 컴퓨팅 사고력을 높이기 위한 피지컬 컴퓨팅을 활용한 소프트웨어 교육에 관한 연구

조 은 진[1] · 성 연 옥[2] · 서 영 건[3*]
[1]경상국립대학교 교육대학원 컴퓨터교육전공 석사과정
[2]경상국립대학교 농업생명과학대학 행정실
[3]*경상국립대학교 컴퓨터과학과 교수

A Study on Software Education Using Physical Computing to Increase Computational Thinking in Elementary School Students

Eun-Jin Cho[1] · Yeon-Ok Seong[2] · Yeong Geon Seo[3*]
[1]Master's Course, Department of Computer Education, Graduate School of Education, Gyeongsang Nat'l Univ.
[2]Administration Office, College of Agriculture & Life Science, Gyeongsang Nat'l Univ.
[3]*Professor, Department of Computer Science, Gyeongsang Nat'l University, 501 Jinju-daero, Jinju, Korea

[요 약]

2015 개정 교육과정 정보교과에서는 소프트웨어 교육에서 추구하는 핵심역량으로 컴퓨팅 사고력을 강조하였다. 이는 피지컬 컴퓨팅을 활용한 소프트웨어 교육은 현실과 컴퓨팅 환경을 연결해 주기 때문에 실생활 문제해결에 가깝게 다가갈 수 있도록 하여 학습자의 창의적 문제해결력 향상과 소프트웨어 역량 향상을 기대할 수 있다. 본 연구에서는 피지컬 컴퓨팅을 활용하여 코딩 수업을 한 A 초등학교와 EPL만을 이용해 코딩 수업을 한 B 초등학교를 비교하여, 피지컬 컴퓨팅을 활용한 코딩 수업의 양상과 컴퓨팅 사고력 향상에 도움이 되는지를 연구하였다. 실험 결과로써 피지컬 컴퓨팅을 활용한 소프트웨어 교육에서 학습자가 더 흥미로워하며, 적극적으로 수업에 참여하는 경향이 있는 것으로 관찰되었으며, 피지컬 컴퓨팅을 활용한 소프트웨어 교육이 초등학생의 컴퓨팅 사고력 향상에 긍정적인 영향을 미치는 것으로 분석되었다.

[Abstract]

In the information subject of the 2015 revised curriculum, computational thinking was emphasized as a core competency pursued in software education. This is because software education using physical computing connects reality with the computing environment, so it can approach real-life problem solving, so that learners can expect to improve their creative problem solving ability and software competency. In this study, by comparing elementary school A, which took coding class using physical computing, and elementary school B, which took coding class using only EPL, the aspect of coding class using physical computing and whether it helps to improve computational thinking was studied. As a result of the experiment, it was observed that learners were more interested in software education using physical computing and tended to actively participate in class. analyzed.

색인어 : 정보교과, 컴퓨팅 사고력, 피지컬 컴퓨팅, 소프트웨어 교육, 문제 해결
Keyword : Information subject, Computational thinking, Physical computing, Software education, Problem solving

http://dx.doi.org/10.9728/dcs.2022.23.10.1959

Received 24 August 2022; **Revised** 13 September 2022
Accepted 07 October 2022

***Corresponding Author; Yeong Geon Seo**
Tel: +82 55-772-1392
E-mail: young@gnu.ac.kr

Ⅰ. Introduction

Today, we live in a knowledge and information society in which new technologies and digital environments are developing rapidly and rapidly changing. Terms such as artificial intelligence (AI), Internet of Things (IoT), big data, drones, 3D printers, autonomous vehicles, and virtual reality (VR), which are representative technologies of the 4th industrial revolution, have become part of our daily lives. In this era of the 4th industrial revolution, interest in cutting-edge science and technology and software education to cope with changes in future jobs and lives has increased. According to a study on software education teaching and learning model development, the knowledge information society is moving from the first IT revolution centered on infrastructure and hardware to the second IT revolution centered on software and content based on creative IT talent. To the extent that software is expressed as “Software is eating the world” and “Software is everywhere”, the impact on almost all industries such as the automobile industry, aviation industry, financial industry, and shopping industry is enormous [1].

As software becomes a software-centered society that determines the competitiveness of individuals, companies, and governments, major countries around the world are pursuing national strategies to actively respond to these changes[2]. In major developed countries, software education is provided from the lower grades of elementary school to develop learners' computational thinking(CT) skills, and the software is actively used to achieve sustainable growth of the national economy and solve social problems. Therefore, in Korea, it is required to prepare a foundation for a curriculum so that all students can be provided with software education opportunities that are suitable for their abilities and aptitudes[1]. It is necessary to think about what kind of education the learners who will live in the changing times should receive and what kind of education should be provided to nurture them into creative convergence talents demanded by our society. In this regard, the 2015 revised information department emphasized CT as a core competency pursued in software education. To understand and build a concrete and practical solution, it is necessary to devise an algorithm and make it a reality through programming[3].

In this study, by comparing elementary school A, which took coding classes using educational programming language(EPL) and physical computing, and elementary school B, which took coding classes using only EPL, for 3rd and 4th graders who applied for coding classes after school at elementary school, we studied whether it helps to improve the class patterns and CT. After-school education can design a software curriculum that meets the needs of students and their parents in a variety of subjects and methods without being bound by the curriculum[4]. Therefore, students' opinions were reflected or students decided on their own topics so that they could actively participate in the class. In addition, among SW education teaching and learning models for CT enhancement, we designed a software class using physical computing, focusing on the demonstration-centered model (DMM model) and the reconstruction-oriented model (UMC model), and examined the appearance in the class. The results of the analysis on the effect on the CT of elementary school students are presented after analyzing the Beaver challenge problem for the evaluation of CT before and after class, and the CT test of Codemos in the last session.

Ⅱ. Related Studies

2-1 Computational thinking

CT refers to the ability to simplify complex problems, such as the way computers solve problems, and to solve them logically and efficiently. CT was first used in 1980 by MIT professor Seymour Papert. In 2006, Jeannette Wing said that CT is a basic thinking ability that everyone living in the 21st century should have, just like reading, writing, and counting, and presented it as an essential element in children's education, playing a decisive role in disseminating CT. Scholars have slightly different opinions on the definition or elements of CT. The International Association for Educational Engineering (ISTE) and the Computer Science Teachers Association (CSTA) define the operational definition of CT and the 9 elements are presented: data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, simulation, and parallelization[5].

2-2 Software education and physical computing

According to the official blog of the Ministry of Education, software education is defined as education that develops CT to solve various problems creatively and efficiently based on the basic concepts and principles of computer science. The purpose of software education is to expand the perspective of ICT literacy and utilization education performed in the existing ICT education, and it is defined as the purpose of cultivating the ability of learners to solve problems based on CT skills necessary for living in the future society and software education in elementary and middle schools focuses on solving real-life problems through CT based on information ethics awareness and attitude rather than program development competency[1].

In Korea, 17 hours were allocated for practical subjects in elementary school, and 34 hours were allocated to middle school students in grades 1 to 3. Since the grade in which software education started is relatively late compared to other countries, and 34 hours are operated in any grade for three years of middle school, there may be a disconnection between grades and school level[6]. As a result, it is a situation that is significantly lacking compared to other countries, and it is identified as a situation in which solutions and countermeasures need to be prepared.

Physical computing refers to downloading data in the real world to a digital device, processing it in the form of software, and outputting the result to a monitor, LED, or other devices. Physical computing, which has the characteristics of exchanging information between the virtual world and the real world, becomes the basis of the IOT field that recognizes the surrounding environment and processes necessary information. It is expected that it will play an important role in the software industry in the future thanks to being able to go. The use of physical computing tools can achieve the achievement standards of the curriculum, maintain learners' interest, improve CT, and provide experience and competency as a future maker, is expected to increase the likelihood that it will be used as a tool for learners' active attitude change[7]. The various tool types used in physical computing education are classified and analyzed into robot type, modular type, board type, and sensor type as in table 1[8].

2-3 Software education teaching and learning models to improve CT Skills

표 1. 피지컬 컴퓨팅에 활용되는 도구

Table 1. Tools using for physical computing

Category	Characteristics, pros and cons
Robot type	- Complete form of physical output devices such as motors - Ready to use without assembly - Expensive and low usability
Module type	- Combining input and output devices and connecting them to the controller - Various input and output devices can be installed - Expensive and low compatibility
Board type	- Form board with microcontroller on electronic board - Various input and output devices can be installed - Electric circuits, electronic knowledge required and elementary school students are less useful
Sensor type	- Sensors are mounted on the microcontroller with both input and output devices - Various sensors are pre-installed, so no circuit configuration is required - Cannot be used other than the installed sensor

Studies related to teaching and learning design focusing on CT have been conducted in a variety of ways, and according to the SW education teaching and learning model development study[1], the model for enhancing CT has been suggested. The model was developed with a focus on functional areas among the learning target areas - knowledge, skills, and attitudes, in addition to functions, behaviorism, cognitivism, and constructivism were considered. Also, a new model was developed by focusing on a model frequently dealt with in various existing teaching methods such as problem-solving learning, and reconfiguration was made possible in consideration of the characteristics of the learner and the school environment.

표 2. 컴퓨팅 사고력 향상을 위한 SW 교육학습 모델

Table 3. SW teaching and learning models to improve CT

model	teaching method	procedure	description
DMM model	direct teaching	Demonstration	teacher's explanation, demonstration, and standard model presentation
		Modeling	student imitation, questions and answers
		Making	demonstration of skill acquisition through phased, independent practice, and repetitive activities
UMC model	discovery learning	Use	learner experience activities, observation and exploration
		Modify	intentionally modified modules and algorithms presented by teacher
		reCreate	design and create their own programs by extending play and revision activities
DDD model	discovery learning	Discovery	construction of knowledge through exploration and discovery
		Design	algorithm planning and design
		Development	implementation and feedback in programming language
NDIS model	project learning	Needs	consideration of a given problem and user-centered needs analysis
		Design	decomposition, pattern finding, and algorithm design
		Implementation	implementing products through programming and physical computing
		Share	self-reflection through product sharing and feedback
DPAAP model	problem solving learning	Decomposition	decomposing the problem into solvable units
		Pattern Recognition	searching for constant recurring trends and rules
		Abstraction	simplifying the problem, formulating the principles discovered by pattern recognition
		Algorithm	procedurally composing abstract core principles
		Programming	implementing and executing in a language that computers can understand

The developed instructional model set the basic direction of including CT components (decomposition, pattern recognition, abstraction, algorithm, programming) within step-by-step activities to achieve the goal of increasing CT, and proposed all five models that are a demonstration-oriented(DMM) model, a reconstruction-oriented(UMC) model, a development-oriented(DDD) model, a design-oriented(NDIS) model, and a CT element-oriented(DPAAP) model in that the components of CT should be the goal and basis of all learning as shown in table 2.

2-4 Existing studies analysis and problems

Recently, while emphasizing CT as a core competency of software education, research on methods for cultivating CT is increasing. Various aspects of research, such as problem solving methods and changes in personal characteristics, are active[9]. In addition, research on the effect of software education using various physical computing on CT is being emphasized. [10] described the results of learning the concepts and necessity of abstraction, problem decomposition, pattern recognition and algorithms for learners as a basic problem-solving course education method after providing software education using physical teaching tools for 6th graders of elementary school.

[11] conducted 20 hours of physical computing education using hamster robots for 6th graders of elementary school, and determined that a robot education program applying scientific principles had a positive effect on improving CT ability of elementary school students. [12] conducted software education using micro:bit for 6th graders of elementary school, and physical computing education can increase learners' divergent thinking and critical and logical thinking, and can inspire learners' learning motivation. In the process of identifying problems and creating their own programs, learners arouse interest in software classes and enable active learning.

[13] stated that physical computing is a necessary process because it connects reality and the computing environment, so that it can approach real-life problem solving, and claimed to be effective in improving learning immersion and SW awareness. [14] developed a data science education program using microbits, centering on elementary science textbooks, and applied it to 6th graders of gifted students to compare the growth of CT skills and verify the effectiveness. In the process of learning a series of data science processes and reinforcing the ability to solve problems based on data, data science education will have an effect on the improvement of elementary school students' CT, and have a significant effect on enhancing elementary school students' CT skills.

Looking at the previous studies, there have been many studies on software education using physical computing for elementary school students, but most of the research is focused on gifted learning or upper elementary school students (5th and 6th graders). However, as interest in software education has increased in society as a whole, there have been many arguments that education should start from the low graders of the elementary school, as in other countries. Therefore, it is necessary to study how physical computing education should be presented to young learners. In addition, software education using physical computing is economically burdensome because there are many expensive teaching tools such as robots, and it acts as a stumbling block for practical application in the educational field. In this regard, it seems that research on the appropriate selection of teaching tools that can bring out economical and effective educational effects is also needed.

Ⅲ. Research Approach and Procedure

3-1 Research questions and subjects

To study the change in CT of students who participated in a software class using physical computing, the following research questions were set.

First, how will the software class using physical computing be designed in consideration of the elementary and lower grades?

Second, when a software class using physical computing is applied to elementary school students, how does the class appear?

Third, comparing elementary school students who experienced software classes using physical computing and students who took general software classes, does it have a significant effect on improving CT skills?

The subjects of this study were 3rd and 4th grade students who applied for after-school coding classes at elementary school A and elementary school B, and consisted of 15 students each in an experimental group and a comparison group. In both schools, the experimental group, A elementary school and the comparative group, B elementary school, coding classes were first opened after-school, and in the preliminary survey, they consisted of students who had no experience in software classes and were new to educational programming languages(Entry or Scratch).

3-2 Research design

Software training based on physical computing was conducted once a week during an after-school coding class, and the training lasted a total of 30 weeks. A simple preliminary questionnaire and a preliminary test were conducted in the first class. Coding education of both groups was proceeded focusing on the demonstration-centered model (DMM) and the reconstruction-centered model (UMC) among the models proposed in the SW education teaching and learning model development study. After the completion of the software training in the 27th session, in the 29th and 30th sessions, a beaver challenge problem for CT evaluation, a CT test of Codemos and a brief interview were conducted, respectively. The procedure of this study is as follows.

① Selection of research topics and research subjects
② Research-related theories and previous research studies
③ Development of physical computing-based software curriculum
④ Conducting a preliminary investigation
⑤ Application of physical computing-based software curriculum (27th session)
⑥ Post inspection (2 times) and brief interview
⑦ Statistical processing and data analysis
⑧ Summary of research results

In the 1st class, a preliminary test was conducted, and in the 2nd to 4th classes, the contents of the entry menu and basic blocks were taught in the same way as the experimental group and the comparison group. From the 5th to the 28th classes, the learning topic and basic concepts are taught with the same content, The experimental group used physical computing to modify and extend the program and the comparison group conducted a class to modify and expand the program using only the EPL (Entry).

3-3 Test tools

1) Beaver challenge

In many previous studies, the Beaver challenge has been widely used as a tool for measuring software-related core competencies. Besides, the Beaver challenge was implemented in software education through a study on the development of a tool for measuring the effectiveness of software education in 2017 conducted by the KERIS[15]. Therefore, in this study, a pre-test was conducted with the Beaver challenge problem for grades 3 and 4 in 2017, and a post-test of primary CT was conducted with the Beaver challenge problem for grades 3 and 4 in 2020. The Beaver challenge was divided into 6 groups in consideration of the age of the participants, its difficulty level and questions were different and group 2 level (primary) problems were used for grades 3 and 4. The pre-test consists of 9 items and the post-test consists of 10 items. In the difficulty level the upper level is 12 points, the middle level is 9 points, and the lower level is 6 points. The pre-test is out of 81 points and the post-test is out of 90 points.

2) CT test of Codmos

For the secondary CT post-test, it was conducted using the CT test provided by Codemos free of charge at the 'Online Coding Party 2021' hosted by the Ministry of Science and Technology, the Ministry of Education, and the Korea Foundation for Science and Creativity. Logibrothers, which provides Codemos, an elementary coding education service, collaborates with representative educational institutions such as EBS, Woongjin, and Ice Cream Homerun, and provides educational programs to major private elementary and middle schools. It is evaluated as an online solution optimized to improve CT as well as learning essential coding concepts for elementary school students, out of the framework of programming education. Based on the learning data of 1 million people, the Codemos CT pre-diagnostic evaluation, which diagnoses the five areas of CT, automation, reasoning, generalization, abstraction, and data processing, consists of 5 questions, and there is no distinction based on grade level and difficulty level.

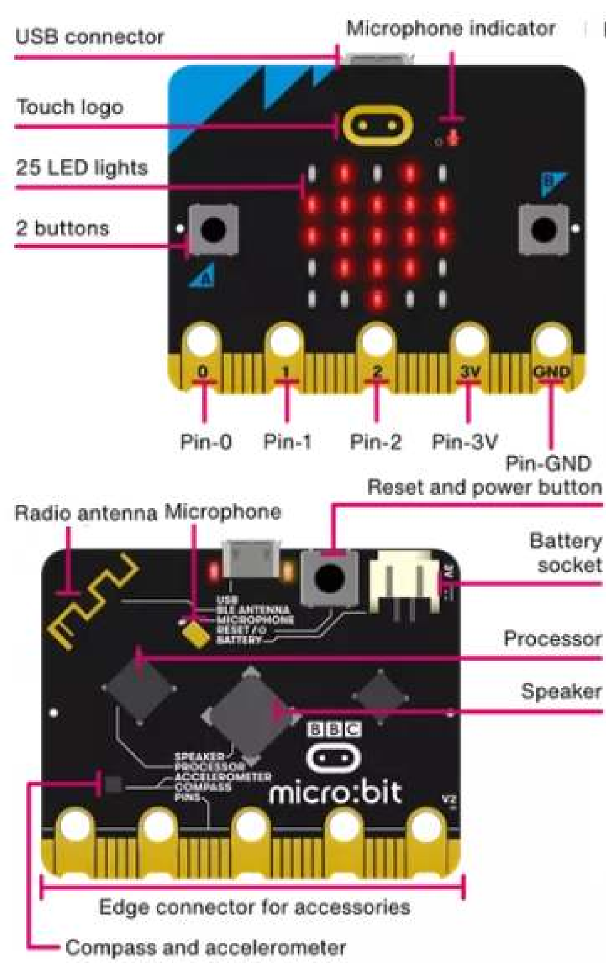

그림 1. 마이크로비트 전면부 및 후면부
Fig. 1. Micro:bit front and back view

표 3. 차시별 학습 내용

Table 3. Learning content for each class

class	learning subject	concept	physical computing	sensor
1	pre-test	2017 3~4th grade Beaver challenge questions		
2	talking	sequential	-	-
3	light on	condition	-	-
4	rabbit running	repeat	-	-
5	cleaning	condition, repeat	Funboard	button sensor
6	stage change	condition, repeat	Funboard	button sensor
7	snowing day	condition, repeat	Funboard	button sensor
8	leaves falling	sequential condition, repeat	Funboard	mic sensor
9	balloon raising	sequential repeat, operator	Funboard	mic sensor
10	planet coming	sequential condition, repeat	Funboard	LED bulb
11	fishing	variable, condition	Funboard	buzzer
12	bat appear	selection, repeat	Funboard	light sensor
13	obstacle jumping	variable, coordinate	Funboard	button sensor
14	random box making	list	Funboard	dot matrix
15	bug catching	AI	Funboard	touch sensor
16	spaceship making	selection, repeat, function	Funboard	slide sensor
17	dessert crossing	selection, function	Funboard	slide sensor
18	bomb passing	condition, operation comparison,	Micro:bit	LED screen
19	rock paper scissors	condition, random number	Micro:bit	button sensor
20	manual counter	condition, selection, repeat	Micro:bit	button sensor
21	automatic counter	condition, repeat, function	Micro:bit	slope sensor
22	compass	variable, condition	Micro:bit	magnetic sensor
23	halloween lamp	condition, selection, repeat	Micro:bit	LED module
24	bridge making	condition, repeat, operation	Micro:bit	logo switch
25	zombie run game	condition, repeat, function	Micro:bit	button sensor
26	star avoiding	condition, repeat, selection	Micro:bit	slope sensor LED module
27	dragon making	condition, selection	Micro:bit	variable resistance module, servo motor
28	sound meter	condition, repeat, operation	Micro:bit	mic
29	post test 1	2020 3~4th grade Beaver challenge questions		
30	post test 2	Codmos' CT test and interview		

3-4 Selection of education tools

In this study, a block-type EPL, Entry was selected as a tool to analyze the effectiveness of EPL education and software education using physical computing. In addition, Funboard and Microbit, which are physical computing tools that support hardware blocks in the entry, were selected.

Entry is structured based on a block-type programming language, so elementary school students can use it easily and fun. It provides an environment suitable for understanding the basic principles of software, such as interworking with such as Python and JavaScript, and providing various class materials and management functions. One of the characteristics of Entry is the support of various hardware. It easily connects more than 100 different hardware, and the available blocks are provided differently depending on the type of hardware.

Funboard is a physical computing education board made with a single microcontroller that combines several sensors. Considering the suitability of the curriculum, the appropriateness of the learning and development stage, performance and convenience, and safety, which are the evaluation criteria for the selection of physical computing tools as argued by [16], Funboard can be operated without separate circuit or electrical knowledge. It can be said that it is safe because it has a housing that protects the board. In addition, since it is a sensor integrated type, it is convenient to use because it does not use a breadboard or jumper wire, and various sensors are included in Funboard, so students can easily check the results of programming and it was judged to be suitable for elementary school level.

Micro:bit board is 4×5cm in size and consists of an ARM-based processor, 25 LEDs, 2 buttons, pins for connecting external devices, light, temperature, acceleration, compass sensors, wireless, Bluetooth, and micro USB and can be used independently by connecting the battery to USB. After writing the program, they can easily transfer the code to the hardware by simply connecting Micro:bit to PC as a USB connection device and downloading the written code. Compared to existing physical tools, Micro:bit is cheap, easy to use, highly scalable, and a physical computing teaching tool equipped with various and powerful functions[12].

3-5 Software curriculum development

The teaching and learning process consisted of 27 sessions centered on DMM model and UMC model. At the beginning of the class, it is mainly the basic stage of programming and the function of the command block is not well understood, so the teacher explains the function and demonstrates the function, and then the students follow along and ask questions. After that, the students organize the class with a DMM model that allows free project activities according to the teacher's explanation, and uses the UMC model according to the topic and content to motivate the students to explore the learning modules learned through play, and through the modification process of the module prepared in advance, the students were able to understand the functions and concepts. Finally, by expanding the play and modification activities, it is possible to design and produce their own programs. Table 3 shows the concept of learning content for each class.

Ⅳ. Research Results and Evaluation

Funboard was easy to handle for young learners who are new to physical computing because the sensors are already installed on the board and can be used just by connecting it to a computer USB port. In addition, for physical computing, only the functions of the blocks and sensors that can be used were briefly explained and the class was conducted, so the class went smoothly without any difficulties without providing additional time. However, as the class progressed, it was difficult to understand the sensor because it was not possible to see the installed sensor, and since it was not possible to use other sensors attached, the expandability to various activities was disappointing.

Micro:bit has various sensors installed on the board itself and has the advantage of being able to do various activities by attaching the necessary sensors, but it was difficult for 3rd and 4th graders of elementary school to connect directly to the sensor and the board. For this reason, an expansion board that can be easily connected to the sensor was purchased separately. In addition, in order for software education using physical computing to proceed smoothly, it is important to provide teaching tools suitable for the learner's level in stages, and changes according to the learning situation must be made appropriately.

Both groups said that after-school coding classes were opened for the first time, and in a preliminary survey, they were exposed to software education for the first time. It consisted of 7 3rd graders and 8 4th graders with similar grades, respectively.

This study investigated how software education using physical computing affects the improvement of CT of elementary school students and what differences it has with EPL-based software education. Table 4 shows the results of the independent sample t-test to check the prior homogeneity of the experimental group and the control group. In the pre-test, the two groups had a t-value of -0.164 and a significance P-value of 0.871 under the assumption of equal variance, which was greater than the significance level of 0.05 and the homogeneity of the two groups was confirmed.

Pre-test and post-test on CT were conducted using Beaver challenge problem for 15 students in 3rd~4th grades of elementary school B, who received EPL-based software education as a control group. As shown in table 5, as a result of CT, post-test score was 51.11, which was higher in post-test compared to pre-test. To check whether the corresponding sample t-test result is statistically significant, t-value was -3.937 under the assumption of equal variance, and the value of the probability P was 0.0005, which was less than the significance level of 0.05, which was analyzed to be significant. Therefore, it is evaluated that EPL-based software education is meaningful in improving CT of elementary school B learners.

Pre-test and post-test of CT were conducted using Beaver challenge problem for 15 students in 3rd~4th grades of elementary school A, who were trained in software using physical computing on experimental group. As shown in table 6, post-test scored 56 points, which was higher in post-test compared to pre-test. To find out whether the corresponding sample t-test result is statistically significant, t-value is -6.173 and probability P value is 0.00000115 under the assumption of equal variance, which is less than the significance level of 0.05. It was analyzed that it had a significant effect on the change of CT. Therefore, it is evaluated that software education using physical computing is meaningful in improving CT of elementary school A students.

Table 7 shows 1st post-test results to confirm the effect on the improvement of CT of learners in experimental group A elementary school and control group B elementary school. One of the experimental group had a higher mean than one of the control group. To find out whether the independent sample t-test result is statistically significant, t-value was -2.117 and probability P-value was 0.0433 under the assumption of equal variance, which was less than the significance level of 0.05, which was analyzed to be significant. Therefore, it is evaluated that the effect of software education using physical computing on the improvement of learners' CT is significant.

표 4. 사전 동질성 검사
Table 4. Pre-test homogeneity test

class	N	M	SD	t-value	P-value
experimental group	15	32.35	10.52	-0.164	0.871
control group	16	31.60	13.99		

표 5. 비교집단 t-검정 결과(B학교 EPL 수업)
Table 5. Control group t-test results(B school EPL class)

class	N	M	SD	t-value	P-value
pre-test	15	31.60	13.99	-3.937	0.0005
post-test	15	51.11	13.13		

표 6. 실험집단 t-검정 결과(A학교 피지컬 컴퓨팅 수업)
Table 6. Experimental group t-test results(A school physical computing class)

class	N	M	SD	t-value	P-value
pre-test	15	32.35	10.52	-6.173	0.00000115
post-test	15	62.22	15.52		

표 7. 두 그룹의 1차 사후 테스트 t-검정 결과
Table 7. 1st post-test t-test results of two groups

class	N	M	SD	t-value	P-value
experimental group	15	62.22	15.52	-2.117	0.0433
control group	15	51.11	13.13		

표 8. 두 그룹의 2차 사후 테스트 t-검정 결과
Table 8. 2nd post-test t-test results of two groups

class	N	M	SD	t-value	P-value
experimental group	15	1232.2	223.42	3.269	0.0028
control group	15	977.81	202.25		

Also, in the results of 2nd post-test Codemos CT test of the two groups conducted at the 30th class, the average of the experimental group was 1232.2 points and the average of the control group was 977.81 points, indicating that the average of the experimental group was higher. To find out whether the independent sample t-test result is statistically significant, t- value was 3.269 and probability P-value was 0.0028 under the assumption of equal variance, which was less than the significance level of 0.05, which was analyzed to be significant. Therefore, it is evaluated that the effect of software education using physical computing on the improvement of learners' CT is significant.

The purpose of this study is to investigate the effect of software education using physical computing on CT of elementary school students. However, since there are no clear standards and questions for software education using physical computing, or a credible test tool that can evaluate CT, the effect on CT indirectly was verified by using the previous questions of the Beaver challenge and CT test of Codemos. Therefore, it was not possible to directly verify the results.

It is difficult to quantitatively measure the interest of the learners in the experimental group, but when the reaction was observed, it was found that they wanted to continue their activities without going home after class. Therefore, additional time was provided using the break time, and it was composed of individual activities and paired activities of two people so that they could solve problems together. They actively participated in classes, showed a high level of interest, and did not give up even if they felt difficulties during activities and developed the program.

Ⅴ. Conclusion

In this study, to investigate the effect of software education using physical computing on the improvement of CT in elementary school students, 27 class programs were developed and practiced. The 27 classes were applied to the experimental group and the control group, respectively. In 30 classes, the improvement of CT was tested using Beaver challenge problem for 3rd~4th graders in the 1st and 29th classes of the 30 classes, and Codemos CT test and interview were conducted at the 30th class and the conclusions obtained are as follows

First, it was analyzed that software education using physical computing had a positive effect on the improvement of CT of elementary school students.

Second, it was found that it is necessary to select a physical computing tool considering the age of the learner, and effective software education can be achieved when it is provided according to the educational environment and economic situation.

Third, it was observed that learners were more interested in software education using physical computing and tended to actively participate in class.

Taken together, it can be said that the improvement of elementary school students' CT is more effective when getting software education using physical computing than when doing it using only EPL.

As for the effectiveness verification, the study is conducted for a limited period of time with limited content, group, and so there are limitations to generalize the results of the study to elementary school students nationwide. In addition, there is a limit to generalizing the results of the study as an effect because various variables such as the study period, various teaching and learning methods, teacher variables, and environmental variables were not taken into account. Therefore, for follow-up studies, it is necessary to study the implementation of software education using physical computing in consideration of various variables, to select a more extensive research area and subject, and to follow up with a longer-term and systematic study.

References

[1] Jinsook Kim, Software Education Teaching and Learning Model Development Research, *2015 KERIS Issue Report*, 2015.

[2] Yura Lee, “The Effect of Creative Convergence Lessons of Information-Science using Physical Computing on SW Education Awareness and Convergence Capability Enhancement,” *Korea National Teacher’s University, Master’s Thesis*, 2020.

[3] Jaehwi Kim and Dongho Kim, “Development of Physical Computing Curriculum in Elementary Schools for CT,” *Journal of The Korea Association of Information Education,* Vol. 20, No. 1, pp. 69-82, Feb. 2016, http://dx.doi.org/10.14352/jkaie.2016.20.1.69

[4] Byungdoo Kim, “Development of Arduino Coding-based Convergence Talent Education Program and Application of Elementary School After-school Classes,” *Daegu National Teacher’s College, Master’s Thesis,* 2019.

[5] Gini Coding Lab, ISTE & CSTA(2011) Computational Thinking Leadership Toolkit 1st edition, Computational Thinking 9 Components, *http://jinicoding.net/computational–thinking-understanding*, 2022.

[6] Sinchun Kang, Overseas Software Education Operation Status Empirical Data Analysis Report, *2019 KERIS Issue Report*, 2019.

[7] Youngjae Lee and etc, “Development and Applyment Selection Standards of Physical Computing Teaching Aids for Elementary SW Education According to the 2015 Revised Curriculum,” *Journal of The Korean Association of Information Education,* Vol. 21, No. 4, pp. 437-450, Aug. 2017. http://dx.doi.org/10.14352/jkaie.2017.21.4.437

[8] Myeonghee Lee, “A Study on the Effects of Coding Education Using Funboards on Computational Thinking of Elementary School,”, *Kumoh National Institute of Technology, Master’s Thesis*, 2019.

[9] Sunhee Min, “A Study on Development and Application of Physical Computing Lessons to Promote Computational Thinking in Elementary School Students,” *Proceedings of The Korean Association of Computer Education*. Vol. 23, No. 1, pp. 79-82, Dec. 2018.

[10] Chanwoong Kim and Kyeong Hur, “Application of Physical Computing Education using Arduino Devices in Elementary School Subjects,” *The Journal of Education,* Vol.37, No.1, pp. 183-197, 2017. http://dx.doi.org/10.25020/je.2017.37.1.183

[11] Jaeho Lee and Hakmin Kim, “The Influence of Robot Education Using Magnetic Force on the Computational Thinking,” *Journal of Creative Information Culture*, Vol.5, No.3, pp. 275–283, 2019. http://dx.doi.org/10.32823/jcic.5.3.201912.275

[12] Kyeong Tae Kim and Chul Hyun Lee, “The Effect of Physical Computing Education Using Micro:bit on Creative Problem Solving Ability of Elementary School Students,” *Journal of KPAE*, Vol. 34, No. 1, pp. 85-111, 2021. http://dx.doi.org/10.24062/kpae.2021.34.1.85

[13] Jieun Kim, “The Effects of Physical Computing-Based Programming Education on Computational Thinking, Learning Flow, and Recognition Concerning SW,” *Aju University, Master’s Thesis*. 2017.

[14] Jaejoon Kim, “The Effect of Data Science Education Using Micro:bit on Computational Thinking : Focusing on Elementary Science Subject,” *Cheju University, Master’s Thesis*, 2021.

[15] Okji Song, “The Effect of Software Education Using Micro:bit on Computational Thinking of Elementary School Students,” *Journal of Knowledge Information Technology and System,* Vol.15, No.1, pp.37–46, 2020. http://dx.doi.org/10.34163/jkits.2020.15.1.004

[16] Youngjae Lee and etc, “Development and Applyment Selection Standards of Physical Computing Teaching Aids for Elementary SW Education According to The 2015 Revised Curriculum,” *Journal of The Korean Association of Information Education*, Vol. 21. No. 4, pp. 437-450, 2017. http://dx.doi.org/10.14352/jkaie.2017.21.4.437

조은진(Eun-Jin Cho)

2014년 : 한국방송통신대학교 컴퓨터과학과 (이학사)
2022년 : 경상국립대학교 교육대학원 컴퓨터교육전공 (교육학석사)
2002년~현 재 : 초등학교 방과후 강사

※관심분야 : FL, BL, 컴퓨팅 사고, 피지컬 컴퓨팅

성연옥(Yeon-Ok Seong)

2000년 : 한국방송통신대학교 경영학과 (경영학사)
2011년 : 경상대학교 경영학과 박사 (경영학박사)
1988년~현 재 : 경상국립대학교 농업생명과학대학 행정실 재직

관심분야 : 플립 러닝, PBL, HR, HRD

서영건(Yeong Geon Seo)

1987년 : 경상대학교 전산과 (이학사)
1997년 : 숭실대학교 전산과 (공학박사)
1989년~1992년 : 삼보컴퓨터
1997년~현 재 : 경상국립대학교 컴퓨터과학과 교수

※관심분야 : PBL, 컴퓨팅 사고, 의료 영상 처리, SLAM, 영상 인식, 컴퓨터 네트워크