Cracking the Code: The Effects of Using Microcontrollers to Code on Students' Interest in Computer and Electrical Engineering

Ali Bicer[††], Yujin Lee[†], Robert M. Capraro*[‡], Mary M. Capraro*[‡], Luciana R. Barroso*[§], *[‡], Daniella Bevan*[‡], and Katherine Vela*[‡]
College of Education; University of Wyoming, Laramie, WY, U.S.A.
Aggie STEM*; Dept. Teaching Learning and Culture[‡]; Dept. Mathematics[†]; Dept. Civil Engineering[§]
Texas A&M University , College Station, TX, U.S.A.
alicanerbicer@gmail.com, yujinlee@tamu.edu, rcapraro@tamu.edu, mmcapraro@tamu.edu lbarroso@tamu.edu, dbevan114@tamu.edu, kvela07@tamu.edu

***Abstract*—Professors who have integrated computer programing into their college courses have faced a complex challenge as students often have had difficulties learning the fundamental concepts of coding due to a lack of prior knowledge and interest. There are few existing studies in which researchers have investigated how to prepare 6th-12th grade students for coding-oriented college level STEM courses using physical computing materials. The purpose of this study was to first design and implement a microcontroller coding course during a two week STEM summer camp for middle and high school students (*N*=86) in order to prepare such students for coding-oriented STEM courses in college. The results from the quantitative data (surveys) and qualitative data (interviews and observations) indicated that students, after taking the course, felt that they were more likely to be successful in courses requiring coding tasks because they had developed an initial understanding of coding. Most students showed an increased interest ($p < .05$) in majoring in computer and electric engineering after engaging in the microcontroller course includes coding. Incorporating coding into middle and high school STEM courses may have far reaching implications for student preparation in programming and interest in computer and electrical engineering.**

Keywords—informal STEM learning, coding, engineering interest, and microcontroller.

I. Introduction

A. Coding Needs for STEM majors

The ever-growing shortage of students who are interested in pursuing engineering and technology related college majors has been a major concern for both educators and policy makers [1]. A myriad of strategies and modifications have been implemented in existing science, technology, engineering, and mathematics (STEM) programs and curriculum in an attempt to increase middle and high school students' interest in engineering and technology related career paths [2]. For example, today's formal STEM programs in schools often include engineering-focused curriculum (e.g., Project Lead the Way), engineering-based clubs (e.g., robotics club), and opportunities for students to meet and, on occasion, work alongside professional engineers. In addition, recently (developed/modified) informal STEM programs frequently include engineering- and technology-focused STEM camps, trips to STEM museums, and opportunities for students to participate in engineering fairs [1]. Researchers have already noted that attending such STEM programs has the potential to increase students' motivation toward engineering disciplines, which in turn influences the likelihood that those students will pursue engineering-related majors [2].

The positive effects of such STEM programs in motivating students to pursue post-secondary academic and career pathways in STEM fields has a critical influence in the U.S.'s ability to maintain its status as a global economic power and scientific leader. However, there is further cause for concern in the retention rate for students in STEM majors. Researchers have found that 62% of students who initially major in STEM disciplines during college eventually change their major to a non-STEM discipline. One significant factor that may influence the low retention rate in STEM majors is students' lack of coding knowledge and skills. Most college freshman have expressed discomfort and insecurity when first introduced to coding in their college courses [3]. In fact, professors whose courses require coding skills encounter a significant challenge as a number of students, due to a lack of prior knowledge and interest, enroll in their courses believing that the class content will be unrelated to coding [4]. Unfortunately, this lack of prior experience and understanding of coding has caused many students who have initially majored in STEM fields to struggle with learning basic coding concepts, which in turn has led some of them to believe that engineering is not one of their core interests [5]. Thus, there is a dire need for innovative teaching and learning practices in secondary school classrooms that will better prepare students for the coding material they will encounter in college.

B. The Importance of Creativity & Problem Solving in the 21st century

It is equally critical that innovative coding-oriented practices and curriculum in middle and high school foster students' abilities to derive creative solutions for contemporary issues. In the 21st century, engineers and scientists are expected to both effectively employ available cutting-edge technology and to produce innovative technology that will advance that which already exists. To meet these expectations, the current and future STEM workforce requires exceptional problem-solving skills and creative abilities. In fact, most employers in the STEM sector now seek to hire divergent thinkers with the ability to produce creative and innovative thought and solve problems [6]. Divergent thinking has been defined as a collective process that includes the different ways that a person suggests a solution, the originality of the person's thought, and the various ways that the person solves the problem [7]. Creativity within this context has been defined as a "set of rules and practices that must be transmitted from the domain to the individual. The individual must then produce a novel variation in the content of the domain, and this variation must be selected by the field from inclusion in the domain" [8] Researchers have found that students recognize the relevance of creativity in their learning process and its relation to STEM fields. They determined that high school students believed using creativity was necessary for a career in STEM and that it helped to enhance their problem-solving skills [9]. Proficient problem-solving skills are equally important as such skills help to develop students' critical-thinking skills, which are also equally essential for success in the 21st century [10]. Critical-thinking skills develop over time to assist with solving a variety of problems, from academic work and research one may engage in while pursuing a STEM degree, to those found in STEM occupations, to more general problems encountered in one's personal life. The ability to harness problem-solving and critical-thinking skills to creatively approach a problem from a wide variety of perspectives within any discipline allows for the development of more well-rounded and comprehensive solutions. Students who have opportunities to develop such abilities within a STEM context prior to college will be equipped with essential skills required to excel in engineering-related post-secondary academic and career pathways.

C. *Coding in middle and high school (6th-12th grade)*

Several scholars have proposed new teaching and learning practices targeting 6th-12th grade students to mitigate the difficulties college students have experienced in coding and to improve students' creative abilities and problem-solving skills [11-12]. Incorporating robotics classes in STEM programs for secondary students, for example, is one method that may build students' creativity and problem-solving skills and enable them to understand programming concepts prior to college [13]. For the United States (U.S.) to maintain its status as a global leader, researchers have noted a need for innovative instructional practices that aim to prepare K-12 students with the knowledge and skills to 1) become future innovators 2) enable them to adopt and create new technological developments [14]. Findings from prior research indicated that the innovative STEM experiences of secondary students positively affected their STEM-related learning and interest in STEM careers [15]. Therefore, making changes to the education system in ways that integrate computers along with cutting-edge technology and the tools to harness and manipulate such technology, like coding, into classrooms to generate a technologically literate society is vitally important.

Many countries have already updated their computer science curriculum to create both a technologically literate society and producers of future technologies [16]. As being born into digital age requires today's children to be more dynamic, flexible, and innovative, having and updating 6th-12th computer science curriculum based on new technological developments is beneficial in order to produce a creative, higher-order thinking, and problem-solving generation equipped to generate creative solutions for a given problem by adopting and producing new technologies. Researchers have suggested creating formal and informal learning opportunities for middle and high school students that include the use of computer and engineering applications (e.g., robotics, 3D printers, and an unmanned aerial vehicle (UAV) in order to improve their coding proficiency [17]. However, the process of teaching and learning coding might also pose a challenge for 6th-12th grade teachers similar to that faced by college students as a number of teachers may have no prior coding knowledge and skills [5]. Once coding is introduced to students in middle and high school, issues such as such as algorithmic rules of logic and syntax of programming languages may arise [4]. Therefore, due to the complex nature of coding, it is essential to develop innovative instructional models that are comprehensible for teachers and that enable those teachers to coherently and thoroughly teach 6th-12th grade students how a computer [17] and electronic function through engaging engineering activities.

Effective instructional models that incorporate coding into engineering activities should consist of extensive multi-sensory and self-guided learning opportunities. Research has shown that students felt more comfortable when they learned coding through visual representations (videos, animations, and figures), verbal explanations, and discovering codes by themselves [18]. Receiving computer science related intervention improved students' programming skills, higher order thinking, and algorithmic problem-solving skills [19]. Findings from another study indicated that K-12 students statistically significantly developed their geometric thinking and metacognitive tasks after receiving computer science related intervention [20]. Additionally, middle and high school students who applied the trial-error method to solve issues related to programming developed their analytical reasoning and became better problem solvers [21]. Although the positive effects of coding intervention on K-12 students' metacognition, problem-solving skills, and creativity have been reported, researchers have yet to address how students' perceptions of their own problem-solving skills, creativity, and aspirations to major in computer and electrical engineering have changed after they received coding-focused microcontroller intervention. Moreover, none have examined the experiences of students who have engaged with coding and electricity concepts through the use of microcontrollers in a project-based learning class. Therefore, we developed a course that incorporated the use of Arduino-Uno, a collection of microcontroller physical computing materials, and then implemented this course during a two-week STEM summer camp for middle and high school students. This study investigated the effects of participating in the coding-focused microcontroller course on middle and high school students' perception about their creativity and problem-solving skills along with their experiences and attitudes toward coding tasks and majoring in computer and electrical engineering.

II. PURPOSE AND RESEARCH QUESTIONS

The purpose of this study was to determine the influences of coding-focused microcontroller intervention on students' perceptions of their creativity, problem solving, and attitudes toward pursuing computer/electrical engineering degrees. Based on this objective, we posed the following research questions:

1. How do students' self-perceptions about their creativity in STEM disciplines change after receiving coding-focused microcontroller intervention?
2. How do students' self-perceptions about their problem-solving skills in STEM disciplines change after receiving coding-focused microcontroller intervention?
3. How do students' self-perceptions about majoring in computer and electrical engineering disciplines change after receiving coding-focused microcontroller intervention?

III. METHODOLOGY

Middle and high school students (N=86) enrolled in a two-week STEM camp held on a university campus in central Texas during the summer of 2017. Students were accepted on a first come first served basis regardless of their demographic and academic backgrounds; however, a diverse student body was served in terms of their gender and ethnicity. The camp consisted of students who would be entering grades 8 to 12 in the upcoming Fall 2017 Semester.

A. Intervention

We designed the microcontroller lesson for the present study to provide an extensive, hands-on learning experience addressing advanced high-school level coding concepts. The lessons were taught on a daily basis over the course of 10 days, during which time the students received guided assistance in utilizing Arduino Uno, which is a microcontroller tool, and Arduino IDE, which is a coding program employed to run Arduino Uno. The lesson activities provided students with continuous opportunities to practice and better understand coding and its practical application in the programming of a microcontroller. Moreover, we designed the activities to encourage students to self-construct their knowledge and refine their approaches to thinking about coding and engineering processes in direct relation to other subject areas such as mathematics, music, and science (see Table 1 for interdisciplinary activities). At the start of the 10-day intervention, students learned the basic concepts of how code and electricity function in a microcontroller and then gradually learned more advanced concepts of coding and programing as the intervention continued. For instance, during the first four days of intervention, the students first learned about basic electrical knowledge involved in engaging with microcontrollers by creating a simple circuit with a LED. After acquiring this basic electrical knowledge, they then learned how to control various features of the LED circuit to adjust the brightness and change the color of the LED. The students then progressed to using several LEDs to form integrated LED circuits, which required them to consider the possible ways of creating various LED color sequences. They tried various sequence combinations of coding they had created, and connected this idea with physics (e.g., electricity and circuits) and mathematics concept (e.g., permutation and combination). In addition, at this advanced level in the activities, students had the opportunity to use other materials such as a temperature sensor, a servo, a motor, a buzzer, and a liquid crystal display (LCD), allowing them to approach diverse types of electronical products and learn how to make these products work. During the intervention, each activity was progressively more advanced and required knowledge of more complex electronical concepts than the previous activity. Through engaging in these activities, we expected that the students would have an opportunity to grasp the engineering knowledge and skills necessary to execute basic coding and programming, as well as to increase their creativity. An outline of each lesson's content is listed in Table 1.

Table 1. *Coding-Focused Microcontroller Lesson Contents*

Day 1	Task: Learning:	Create a simple circuit with LED to make it blink Students understand basic electrical knowledge.
Day 2	Task: Learning:	Control the brightness of an LED using a potentiometer Students learn that a potentiometer, which is known as a variable resistor, demonstrates a variable voltage divider circuit.
Day 3	Task: Learning:	Change colors with one LED Students understand how to use a RGB LED to create unique color combinations and to think about the number of possible sequences of those combinations (the concept of permutation, combination).
Day 4	Task: Learning:	Change colors with multiple LEDs Along with controlling the RGB LEDs, students learn about several programming tricks that keep the code neat and tidy and the larger number of color combination sequences (more complex concept of permutation, combination).
Day 5	Task: Learning:	Measure temperature with a linear temperature sensor Students learn how to use the breadboard to measure temperature and use the Arduino IDE's serial monitor to display the temperature.
Day 6	Task: Learning:	Move a servo to various positions at several speeds Students learn how to use pulse width modulation to control and rotate a servo.
Day 7	Task: Learning:	Make a motor spin Students assemble the motor circuit's components and create a code to spin the motor.
Day 8	Task: Learning:	Connect integrated LED circuits Students practice using a shift register, which is called a serial-parallel converter, to control eight LEDs. They understand the concept of integrated circuits and how predefined codes works in output.
Day 9	Task: Learning:	Compose music Students create their own functions using more codes to create and play the music.

Day 10	Task:	Display commands on a LCD
	Learning:	Students learn how to use a LCD that can display commands or information depending on how students program their breadboard. They can formulate their own program to display their commands on a LCD.

B. Assessment

We employed a triangulation method and design when conducting the present study in order to utilize a qualitative paradigm to support our quantitative findings. "This design is used when a researcher wants to directly compare and contrast quantitative findings or to validate or expand quantitative results with qualitative data" [22]. (Creswell & Clark, 2007, p. 62). The quantitative portion of the study, which consisted of comparing students' responses to a pre-post Qualtrics survey, was quasi-experimental in design. The survey questions consisted of Likert-scale items on a scale from 1 to 5 or from strongly disagree to strongly agree. Students took the pre-survey a week before the intervention started and took the post-survey immediately after the intervention concluded. Three items were employed to measure differences in students' perceptions of their own creativity, problem-solving skills, and majoring in computer/electrical engineering. The pre-post survey consisted of the following questions: 1) I believe I can come up with divergent ideas when I am working within STEM disciplines; 2) I believe I can solve problems in STEM disciplines; and 3) I want to major in computer/electrical engineering fields. SPSS 24 was used for statistical analyses. To determine pre-to-post differences on survey items, a paired-sample *t*-test was applied. Hedge's *g* effect sizes and confidence intervals were reported.

IV. Results

Results from the present study revealed that students' perception about their own creativity in STEM fields underwent a statistically significant ($p < .05$, see Figure 1) change after students engaged with a coding/microcontroller intervention. The Hedge's *g* effect size for this difference was 0.60, and the confidence interval associated with this effect size was (0.48, 0.72). These results indicated an improvement in the students' perceptions of their creativity after the intervention.

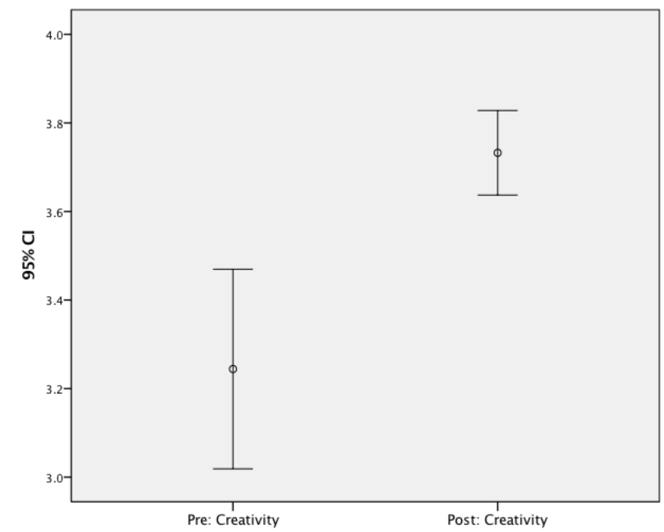

Fig. 1. Perceptions about the need for creativity in STEM fields

The quantitative results, when supplemented with qualitative data gathered from the students, further suggested that the students were better able to conceptualize and articulate the ways in which creativity may be expressed in STEM disciplines after the intervention. For example, one student said "We did have an option to use our creativity in coding and microcontroller classes. While we were working on our Arduino project, we were free to choose and utilize the patterns we liked to follow. For example, I created a linear model when coding to make four LEDs on meaning the LEDs on as it is ordered. I mean first LED turns on and off, and the second, the third, and the fourth. However, my friend did her coding in a way that that the first and last LED turned on and off, and then the second and the third one turned on and off. Since we had freedom to choose any sequential patterns to do on our coding, we challenged ourselves with each of us trying to be more creative to produce something different than the other". This student's response suggested that the course allowed him freedom of creative expression in an engineering context, which helped to develop his existing creativity. Other students expressed that their understanding of the relationship between creativity and STEM disciplines had both broadened and deepened. One student noted that there are multiples ways that creativity and artistic expression can be incorporated in a coding and microcontroller context, providing the following observation: "After taking the coding and microcontroller class, I understand that creativity doesn't mean solely that I got to draw, but it means I did something that required my brain to think divergently, like using my knowledge of coding and electric circuits to create something new, like new coding sequences that produced LED new color combinations. So, I would say definitely the in coding and microcontroller class, I got use my brain in artistic ways in all of the activities". While student in the previous response described the use of creativity in the coding and microcontroller course as an artistic process, another student felt creativity in this course was not as related to art as it was to divergent thinking. This student shared the following perspective after the intervention, "After taking the coding and microcontroller class, creativity to me isn't about being artistic in STEM but more about thinking outside the box; except on a dramatic level. The world couldn't have advanced without daring risks and that to me is creative. So, during the course, I would try to think of the best and fastest way to go about creating a product in a coding and microcontroller project while also giving it the most unique aspects I could". This response suggests that the student was able to define creativity's relation to STEM within the course, but more importantly, it also suggests that the course helped the student to conceptualize the innovations throughout the history of mankind with the ability to take intellectual risks and harness the ability think creatively to create solutions. Despite the slight variation in responses, for example, on the question of how related creativity is to artistic expression, the students provided considerably articulate, insightful, and broad perspectives on the relation between creativity and STEM disciplines.

Results for students' self-perceptions about their problem-solving skills in STEM disciplines statistically significantly ($p < .05$, See Figure 2) changed from pre-survey to post-survey. After receiving coding-focused microcontroller intervention, a majority of the students identified themselves better problem solvers in STEM disciplines. The Hedge's *g* effect size associated with the pre- and post-difference of students' perception about their own problem-solving skills in STEM disciplines was 0.68, and the confidence interval associated with this effect size was (0.56, 0.8).

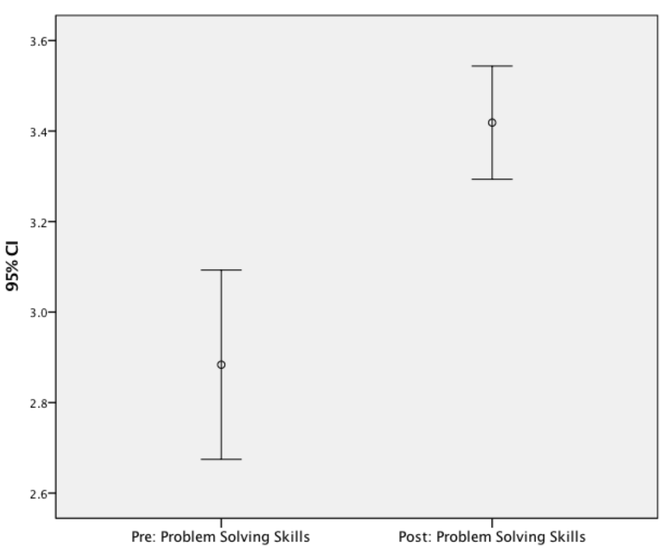

Fig. 2. Students' self-perception about their problem-solving skills

Qualitative findings again supported the quantitative results. Students felt that they became better problem solvers after participating in the coding-focused microcontroller class, and most of them attributed this change with the nature of the coding/microcontroller class itself. They expressed that learning coding through the trial-and-error method, visual representations, verbal explanation, and discovering the codes by themselves enabled them to understand that a problem can be creatively solved in more than one way. For example, one student said, "While coding, we were repeatedly guessing and checking to create the codes, and this makes me think that if you spend enough time and effort, there is no problem you cannot solve, and with this method, I think it's a lot easier to create code on my own. It did help though to have some guidance, like instructional videos that show demonstrations to help me get the basic idea". Another student shared his experiences by saying, "Staying with problems during our coding/microcontroller projects and applying trial-and-error strategy helped to have a better awareness of my mistakes and how to fix those mistakes by myself. I feel that I am now better in finding creative solutions to problems through trial-and error strategy in coding." Hearing these experiences from students lead us to conclude that students felt that they were more likely to be successful in courses requiring coding tasks because they had developed an initial understanding of coding.

Students' perceptions of majoring in computer and electric engineering also changed after they received coding/microcontroller intervention. Results indicated that their interest in majoring in computer and electrical engineering statistically significantly increased ($p < .05$). The Hedge's *g* effect size associated with this increase was 0.45, and the confidence interval associated with this effect size was (0.32, 0.58).

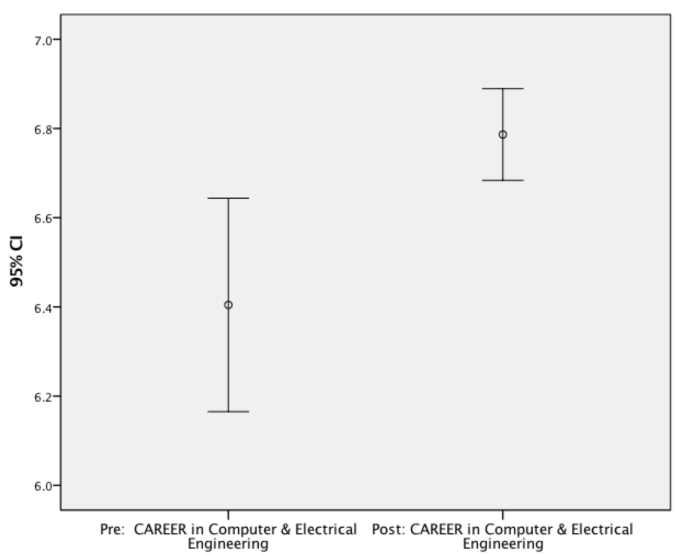

Fig. 3. Students' interest in pursuing computer and electrical engineering degrees

After participating in the coding/microcontroller course, students appeared to be more interested in pursuing academic and career opportunities in the fields of computer and electrical engineering. For example, one student wanted to pursue a computer science degree after realizing the great utility of computers when one has the tools and knowledge to use them to their full potential. He said, "After I took the class, I realized how many things I could do by using computers. I definitely think I will be majoring in computer engineering." Another student also emphasized the idea that simply learning about the coding and microcontroller materials and how to use them heighted her interest in pursuing a related academic and career path. She said, "The microcontroller course did help me to see the application of my coding as Arduino visualized the code I created on breadboard. Arduino also helped me to observe the basic foundational concepts and uses of electricity and electronics such as loop concepts, circuits, sensors, servo motor, and LEDs. I think I am much more interested in pursuing an electronic engineering degree after having opportunity to engage with these materials." Likewise, one student mentioned that "I was always hearing about materials such as breadboard and sensors but had never physically touched them. Taking the microcontroller class gave me opportunities to engage with these materials. Before the class, I was more interested in the theory of physics, but now I am also more interested in application of the coding and electronics." This student expressed a similar sentiment to the other two, namely, that exposure to and engaging with the materials was key to his interest in further investigation into the many applications of both coding and electronics. Hands-on experience with the materials in the coding/microcontroller course was a key factor in their heightened interest in further pursuing computer and electrical engineering.

V. Conclusions

The primary purpose of this study was to explore the effects of Arduino-based coding/microcontroller intervention on middle and high school students' major and career interest in computer and electronic engineering and their self-perceptions about their creativity and problem-solving skills. The findings indicated that using the Arduino-based microcontroller/coding activities improved students' self-perceptions about their creative ability and problem-solving skills. The intervention also increased students' major and career interest in computer and electronic engineering. Students perceived their time spent engaged in the coding/microcontroller class as a valuable learning experience that afforded them the opportunity to create code which was then used to program tangible objects, thereby providing students allowing students to see the practical application of coding.

The results of the present study are consistent with previous research findings, in which researchers advocated teaching abstract problems by using tangible objects [2, 23, 24, 5]. One possible explanation for the improvement found in students' self-perception about their problem-solving skills and creativity might be engagement with hands-on activities while coding. Working with tangible objects to solve problems related to coding may enable students to meaningfully understand the function of each command in syntax. Furthermore, working with hands-on activities such wiring wires or placing LEDs or sensors on breadboard may have helped students understand the roles of each material for successful data transmission from computer to microcontroller. After students meaningfully understand the foundation of abstract concepts such as coding, they may be better prepared to generate more creative solutions for given problems. In addition, it appears that the creative potential inherent within the coding process may have prompted students to consistently attempt to solve problems related to the coding/microcontroller activities during the intervention in a creative way. This persistent effort to find creative solutions and codes may be one of the reasons why we observed an improvement in students' self-perceptions about their creativity and problem-solving skills after the intervention. This explanation can be supported by one of the most popular quotes of Albert Einstein; "It's not that I am so smart. It's just that I stay with problems longer". Since staying with problems longer, or demonstrating dedication to finding solutions for given problems, can lead students be more productive in terms of finding creatively divergent solutions, they may also have fun while learning. As Albert Einstein said "Creativity is intelligence having fun".

The present study also noted that students became more interested in pursuing computer and electronic engineering degrees and career paths. This finding can be explained by Lent, Brown, and Hacket's socio cognitive career theory [25], in which they posit that a person's career interest is shaped by the practices he/she experienced. The underlying concept of this theory is that students' increased self-efficacy and interest in a particular subject through early experiences can predict their future career interest. For example, a student who was exposed to computer and electronic related activities at an early age is more likely to plan to seek out computer and electronic engineering-related activities to engage in, like a robotics club or computer AP class. In sum, the implications of implementing coding-focused microcontroller activities within middle and high school level curriculum consist of preparing a technologically literate and advanced group of future STEM leaders in the fields of computer and electrical engineering who have the skills and knowledge to successfully pursue academic and vocational pathways in STEM using creative and innovative problem-solving methods.

References

[1] W. W. Lau, G. Ngai, S. C. Chan, and J. C.Cheung, "Learning programming through fashion and design: a pilot summer course in wearable computing for middle school students," In *ACM SIGCSE Bulletin* vol. 41, no. 1, pp. 504-508), 2009.

[2] A. Bicer, P. Boedeker, R. M. Capraro, and M. M. Capraro, "The effects of STEM PBL on students' mathematical and scientific vocabulary

knowledge," *International Journal of Contemporary Educational Research,* vol. 2, no. 2, pp. 69-75, 2015.

[3] H. Qin, "Teaching computational thinking through bioinformatics to biology students," In *ACM SIGCSE Bulletin*, vol. 41, no. 1, pp. 188-191, 2009.

[4] A. Robins, J. Rountree, and N. Rountree, "Learning and teaching programming: A review and discussion," *Computer science education*, vol. *13*, no. 2, pp. 137-172, 2003.

[5] M. A. Rubio, C. M. Hierro, and A. P. D. M. Pablo, "Using arduino to enhance computer programming courses in science and engineering," In *Proceedings of EDULEARN13 conference*, pp. 1-3, 2013.

[6] M. H. Land, "Full STEAM ahead: The benefits of integrating the arts into STEM," *Procedia Computer Science*, vol. 20, pp. 547-552, 2013.

[7] C. G. Ferrándiz, M. Ferrando, G. Soto, M. Sainz, and M. D. Prieto, "Divergent thinking and its dimensions: What we talk about and what we evaluate?," Annals of Psychology, vol. 33, no. 1. 2017.

[8] B. Sriraman,"The characteristics of mathematical creativity," *Mathematics Educator*, vol. *14 no.* 1, pp. 19-34, 2004.

[9] A. T. Oner, S. B. Nite, R. M. Capraro, and M. M. Capraro, "From STEM to STEAM: Students' beliefs about the use of their creativity. , 2(2), Article 6.," *The STEAM Journal,* vol. 2, no. 2, 2016.

[10] R. Cooper, and C. Heaverlo, "Problem Solving And Creativity And Design: What Influence Do They Have On Girls' Interest In STEM Subject Areas?,".*American Journal of Engineering Education*, vol. 4, no. 1, pp. 27, 2013.

[11] K.E. A. Merrick, "A comparative study of value systems for self-motivated exploration and learning by robots," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 2, pp. 119-131, 2010.

[12] J. Wells, R. M. Barry, and A. Spence, "Using video tutorials as a carrot-and-stick approach to learning," *IEEE Transactions on Education*, vol. 55, no. 4, pp. 453-458, 2012.

[13] G. T. Richard, "Employing physical computing in education: how teachers and students utilized physical computing to develop embodied and tangible learning objects," *The International Journal of Technology, Knowledge and Society*, vol. 4, no. 3, pp. 93-102, 2008.

[14] V. Grout, N. Houlden, "Taking computer science and programming into schools: The Glyndŵr/BCS Turing project," *Procedia-Social and Behavioral Sciences*, vol. 141, pp. 680-685, 2014.

[15] R. V. Small, "The motivational and information needs of young innovators: Stimulating student creativity and inventive thinking," *School Library Research,* vol. 17, pp. 1-36, 2014.

[16] M. U. Bers, L. Flannery, E. R. Kazakoff, and A. Sullivan, "Computational thinking and tinkering: Exploration of an early childhood robotics curriculum," *Computers & Education*, vol. 72, pp. 145-157, 2014.

[17] F. Kalelioğlu, "A new way of teaching programming skills to K-12 students: Code. org.," *Computers in Human Behavior*, vol. 52, pp. 200-210, 2015.

[18] J. X. Zhang, L. Liu, P.O. de Pablos, and J. She, "The auxiliary role of information technology in teaching: Enhancing programming course using Alice, *"International Journal of Engineering Education*, vol. 30, no. 3, pp. 560-565, 2014.

[19] G. Fessakis, E. Gouli, and E. Mavroudi, "Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study," *Computers & Education*, vol. 63, pp. 87-97, 2013.

[20] G. Keren, and M. Fridin, "Kindergarten Social Assistive Robot (KindSAR) for children's geometric thinking and metacognitive development in preschool education: A pilot study," *Computers in Human Behavior*, vol. *35*, pp. 400-412, 2014.

[21] C. C. Liu, Y. B. Cheng, and C. W. Huang, "The effect of simulation games on the learning of computational problem solving," *Computers and Education*, vol. *57*, no. 3, pp. 1907-1918, 2011.

[22] J. W. Creswell, and V. L. P. Clark, "Designing and conducting mixed methods research", 2007.

[23] C. H. Crouch, A. P. Fagen, J. P. Callan, and E. Mazur, "Classroom demonstrations: Learning tools or entertainment?,"*American journal of physics*, vol. *72*, no. 6, pp. 835, 2004.

[24] J. Piaget, "*The origin of intelligence in the child*," Routledge and Kegan Paul, 1953.

[25] R. W. Lent, S. D. Brown, and G. Hackett, G., "Toward a unifying social cognitive theory of career and academic interest, choice, and performance," *Journal of vocational behavior*, vol. *45, no.* 1, pp.79-122, 1994.