International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:14, No:4, 2020

Investigation of the Physical Computing in Computational Thinking Practices, Computer Programming Concepts and Self-Efficacy for Crosscutting Ideas in STEM Content Environments

Sarantos Psycharis

***Abstract*—**Physical Computing, as an instructional model, is applied in the framework of the Engineering Pedagogy to teach "transversal/cross-cutting ideas" in a STEM content approach. Labview and Arduino were used in order to connect the physical world with real data in the framework of the so called Computational Experiment. Tertiary prospective engineering educators were engaged during their course and Computational Thinking (CT) concepts were registered before and after the intervention across didactic activities using validated questionnaires for the relationship between self-efficacy, computer programming, and CT concepts when STEM content epistemology is implemented in alignment with the Computational Pedagogy model. Results show a significant change in students' responses for self-efficacy for CT before and after the instruction. Results also indicate a significant relation between the responses in the different CT concepts/practices. According to the findings, STEM content epistemology combined with Physical Computing should be a good candidate as a learning and teaching approach in university settings that enhances students' engagement in CT concepts/practices.

***Keywords*—**STEM, computational thinking, physical computing, Arduino, Labview, self-efficacy.

I. Introduction

THERE is a lot of discussions for the ideas/concepts and pedagogical practices that should be included in Science and Engineering Education in order to engage students in Scientific and Engineering practices and acquisition of relevant skills. For the implementation of these practices in the teaching and learning sequences we should take into account the form of engagement of learners as well as the types of concepts that they should be included in the curriculum, as well as the scientific practices involved. Research suggests that this integration can be implemented by engagement in the following dimensions:

- "Scientific and engineering practices
- Crosscutting concepts that unify the study of Science and Engineering through their common application across fields
- Core ideas in four disciplinary areas: Physical Sciences; Life sciences; Earth and Space Sciences; and Engineering, Technology, and applications" [1].

Research also suggests the Computational Pedagogy -as a model for computational learning-provides a methodology to effectively provide the methodology to implement the Scientific and Engineering practices, the core ideas and the cross-cutting concepts during the development of educational scenario [2]. "The term "practices" is used instead of a term such as "skills" to emphasize that engaging in scientific investigation requires not only skill but also knowledge that is specific to each practice [1].

Sarantos Psycharis is with the Sarantos Psycharis, Greece (e-mail: spsycharis@gmail.com).

II. CT

CT is "the thought process involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information processing agent" [3]. CT can be included in various disciplines engaging learners in the development of models of simulations using the Computational Experiment methodology [2].

In the absence of an agreement for the definition of CT, a set of core concepts/dimensions and practices is continuously developing for a more concise and complete definition for CT. These include: abstraction (considering a problem at different layers of detail using the inductive process and decision of the variables to use), modeling (restriction of theory, selection of variables and the relation between them), algorithmic thinking, automation, decomposition of a problem as a set of simpler problems, debugging, pattern recognition ,metacognition and generalization [4] as well as design-based thinking.

III. Engineering Pedagogy and STEM Epistemology

The discipline of engineering includes the engineering content and the engineering design. Engineering content includes the integration of cognitive areas of Science and Mathematics as well as the collection of methodologies and s and practices used by the engineers in order to design solutions which obey the laws of science and specific restrictions/ constraints [5]-[7]. Engineering design is the fundamental engineering approach for solving problems and engagement in this process enhances students' analytical and synthetic skills [7]. CT practices are related to engineering design through for example pattern recognition, abstraction and the construction of design-based computational artifacts, that can be either a computer program or a physical construction.

Engineering Education/Pedagogy is based on the integration of the engineering epistemology (justification of knowledge), the engineering design process and the application of proper instructional/pedagogical strategies that enhance student-centered learning of mathematical and scientific core and crosscutting concepts [8].

Research also suggests that CT concepts can be diffused and mapped in Engineering Pedagogy, when inquiry based environments are implemented through the development of models that will be simulated in alignment with the use of computational problem-solving practices, and systems thinking practices [2], [9]. Engineering Pedagogy is a fundamental component of STEM content education. Integrated approaches of STEM Education are based on interdisciplinary or/and trans-disciplinary methods and have been suggested for applying integrated STEM as an holistc approach to the curriculum [2]. Such approaches are expected to enhance students' capacity to "solve real-life problems by applying concepts that cut across disciplines" [10].

Integrated STEM content education is "an effort to combine some or all of the four disciplines of science, technology, engineering, and mathematics into one class, unit or lesson that is based on connections between the subjects and real world problems" [11]. STEM content approach follows the so-called transdisciplinary approach and it "focuses on the merging of the content fields into a single curricular activity or unit to highlight "big ideas" from multiple content areas" [12].

IV. Physical Computing

In educational literature, Physical Computing constitutes a teaching strategy that integrates "computing" and CT to the real-physical world data and phenomena [13]. Physical Computing can be implemented in the curriculum as a Scientific and Engineering practice enabling learners to construct computational models, collect and analyze data and construct a computational artifact while their models will be tested against real data from literature.

Physical Computing "integrates" digital elements with real-world phenomena, by creating an interface that connects the physical world (e.g. using sensors to collect data) with the digital/virtual world of the computer (i.e. the computing) [14].

"Physical computing can be implemented in two ways: either to teach concepts of computer science using physical computing or to use selectively physical computing as an entry point to different topic areas of computer science" [15]. Our argument is that Physical Computing can serve as a proper medium to implement the practices of CT and the Computational Experiment methodology in education settings. In addition, physical computing enhances engineering design practices as it facilitates the construction of artifacts that receive data from the real world, while students have the chance to test their prototype model against real data.

Within the framework of Computational Pedagogy model we implement Physical Computing using the Arduino platform and the Labview software. Arduino is an open-source electronics platform that includes digital and analog inputs and outputs (www.arduino.cc).

Labview (http://www.ni.com/en-us/shop/labview.html) offers a graphical programming interface that provides the tools to easily receive and visualize data on a diagram, to engage in data analysis, algorithms, and design interfaces. Labview is suitable for measuring physical data with sensors or actuators and it is compatible with the Arduino platform.

V. The Computational Pedagogy Model

Computational Science is a scientific area that provides the methodology for the construction of computational models. It follows the interdisciplinary epistemology that uses advanced computing and data analysis to explore and solve complex problems [2], [16]-[19]. One of the fundamental components of Computational Science is the abstraction of a phenomenon and its implementation as a computational model that can be tested by comparing with real data [2], [18].

Computational Science can be integrated with many concepts of Computational Thinking. For example, [20] stated that "the ability to think computationally is essential to conceptual understanding in every field, through the processes of problem solving and algorithmic thinking".

According to [2], [19] during the Computational Science epistemology, the following three spaces are included:

1. "The hypotheses space, where the instructor guides the students to create their hypotheses according to prior knowledge and decide about the model that should be used. Misconceptions and cognitive conflicts should be explained by the instructor during this phase.
2. The experimental space, which includes the method and the simulation of the model for the phenomenon under study, or better, for the crosscutting concept that should be explored. In this space, students collect the data from their model and analyze them, while they try to connect them with the theory they have been taught. Physical computing is applied in this space enabling students to design the system and take measures by controlling the variables of their model
3. The prediction space, where the results, solutions or conclusions formulated in the experimental space, are compared with the data presented in the textbook or other sources provided by the instructor. This space is very important for the metacognitive awareness of students".

Computational Science "can be an effective methodology to support learners to solve a STEM problem using models that encourage learners to be engaged in different didactic strategies, such as: formulating the problem in a way suitable for simulations using models, choosing an efficient computational algorithm, running the simulations and collecting numerical data, analyzing the data obtained, finding patterns in order to generalize the method to other equivalent problems , extracting the solution of the problem in a form that can lead to the creation of artifacts" [2], [19].

Science Inquiry based processes, supported by different computational environments, that could be used in STEM content learning approaches, include: orienting and asking questions; generating hypotheses; planning; investigating; analyzing and interpreting; exploring and creating models; evaluating and concluding; communicating; and predicting [21], [22].

The nine inquiry tools of [21] are related to the essential features of The Inquiry-based Teaching and Learning Approach [22], namely:

- Question (learner engages in scientifically oriented questions)
- Evidence (learner gives priority to evidence)
- Analyse (learner analyses evidence)
- Explain (learner formulates explanations from evidence)
- Connect (learner connects explanations to scientific knowledge)
- Communicate (learner communicates and justifies explanations)
- Reflect (learner reflects on the inquiry process, respond to his/her work, develops metacognitive experience).

In Table I, the connection between the three spaces of the Computational Science Experiment with the dimensions of CT and the essential features of inquiry dimensions is presented [2], [19]. Physical Computing can be a proper "medium" – through the potentiality of its tools and its interface - to implement the computational experiment in inquiry based environments by transforming concepts included in a model to real world concepts that can be measured in the experimental space.

The term Computational Pedagogy was introduced by [17] as an extension of TPACK, and was called Computational Pedagogical Content Knowledge. In this article the aforementioned model is adopted [17] with the addition of computational spaces [2], [19], [23] and practices related to the engineering design and CT practices.

TABLE I
MODEL FOR THE CONNECTION BETWEEN CT, COMPUTATIONAL EXPERIMENT AND PHYSICAL COMPUTING

Spaces of the Computational Experiment	Essential Features of Inquiry Physical Computing Dimensions of CT	Inquiry tools
Hypotheses space	Essential Features of Inquiry: Question Physical Computing: Unplugged activities Dimensions of CT: Abstraction, Decomposition	Orienting and asking questions Generating hypotheses
Experimental space	Essential Features of Inquiry Evidence, Analyze, Explain Physical Computing Design, Making, Creation of Code Dimensions of CT Abstraction, Algorithmic thinking	Planning-Investigating Analysis and interpretation Modelling
Prediction Space	Essential Features of Inquiry Connect, Communicate, Reflect Physical Computing Remixing Dimensions of CT Debugging, Generalization	Conclusion Evaluation Prediction

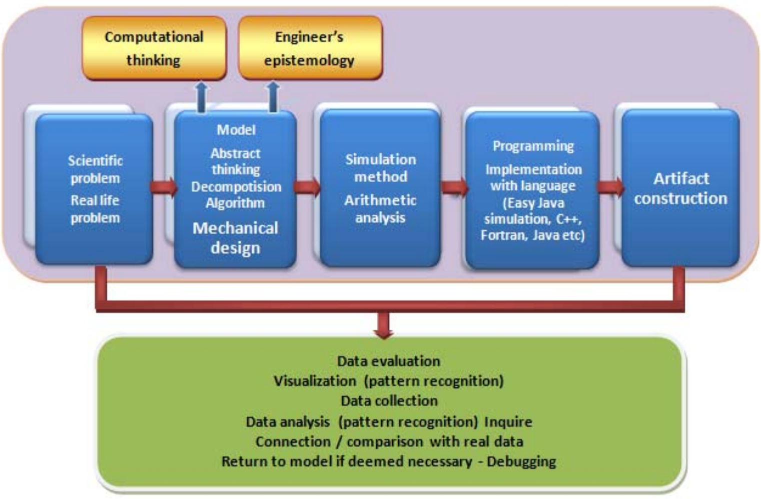

Fig. 1 The Computational Science Experiment (CE experiment) –Computational Pedagogy

In Table I we present the connection between the computational spaces with the Inquiry based teaching and learning features and the physical computing components while in Fig 1 our model (Computational Pedagogy) is presented and includes the engineering epistemology and the STEM content epistemology.

VI. METHODOLOGY

The main purpose of this study is to examine the impact of teaching using the Computational Pedagogy model -by the utilization of Labview and Arduino- on learners' self-efficacy for CT practices when crosscutting ideas are included in a STEM content course.

In this work we applied the Computational Pedagogy model [2], [19], [23], using physical computing environments. Self-confidence constitutes the self-awareness of capacity [24], and is considered as a main component of the Inquiry teaching and learning process [25].

Data were collected from students at a Higher Education Institute in Athens Greece, while the questionnaire for self-efficacy in CT concepts [4] was employed. 35 students worked on interdisciplinary core ideas (like the idea of periodicity) using the Labview and the Arduino in order to control actions like the rotation of a servo or the control of a LED in order to study periodicity. Other examples used the above mentioned software for exploring crosscutting concepts, like the conservation of energy, the exponential decay etc.

The questionnaire for self-efficacy in CT concepts [4] contains 23 questions and each of these questions measured self-efficacy on a five-value Likert scale: strongly disagree, somewhat disagree, not sure, somewhat agree and strongly agree.

The different concepts included in the questionnaire are:

- Four for algorithmic thinking (ALG)
- Three for abstraction (ABS)
- Two questions for problem decomposition(DEC)
- Two questions for data (DAT)
- Three questions for parallelization
- Five questions for control flow (CON)
- Two questions for incremental and iterative(IAI)
- One question for testing and debugging(TAD)
- One question for questioning (QUE).

Learners' scores were compared before and after the instruction using Labview and Arduino which lasted for six weeks. The whole course lasted for 13 weeks and during the course, students were engaged in other tools too, like Scratch, App inventor and Easy Java Simulations. In particular, the pre-test scores learners' self-efficacy for CT were recorded and compared with the corresponding post-test scores. A paired t-test was employed to compare the means of the same group in the pre-test and post-test since the data were normally distributed. Questionnaire has Cronbach's alpha internal consistency reliability 0.85.

VII. LABVIEW AND ARDUINO APPLICATIONS

We present some of the activities implemented during the instruction. After working with basic LEDs, SERVO motors (Fig. 2), digital signals and connections, we introduced students to crosscutting ideas like periodicity and how we can measure the period and frequency of physical phenomena.

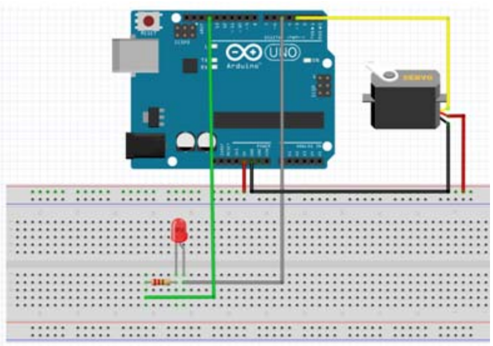

Fig. 2 Connection of LED and SERVO in Arduino

At the beginning, students connected a LED and a SERVO motor, with the necessary wires and resistors in the Arduino microcontroller and the breadboard (Fig. 2). Finally, students developed the interface using numeric controls, indicators, waveform chart, stop button and the led controller in the LABVIEW. In Fig. 3 the Front Panels' design of the model in Labview is presented, while in Fig. 4 the code in Labview is presented. Using this activity students were engaged mainly in the practices of CT and the engineering design process.

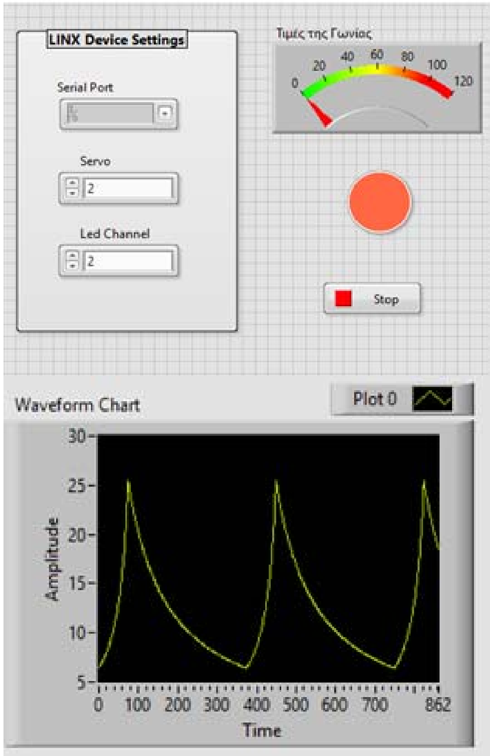

Fig. 3 Design of the model in LABVIEW to measure the periodicity- the outcome of the experiment

VIII. RESULTS

Results show a significant change in students' responses for self-efficacy for CT before and after the instruction. Results also indicate a significant relation between the responses in the different CT concepts.

Students completed the self-efficacy in CT concepts [3] which contains 23 questions related to the different practices of CT. Examples of questions were:

- "When solving a problem I look how information can be collected, stored, and analyzed to help solve the problem" (practice of CT (collection, representation, and analysis of data)).
- "I can write a computer program which runs a step-by-step sequence of commands" (practice of CT-Algorithms).
- "When creating a computer program I run my program frequently to make sure it does what I want and fix any problems found" (practice of CT (Testing and Debugging)).
- "When creating a computer program I break my program into multiple parts to carry out different actions" (concept of CT (Problem Decomposition)).

A paired t-test was applied for 23 questions ($p = 0.002$), indicating a significant difference in self-efficacy for CT concepts. We also observed significant differences for all the different concepts except for the concepts TAD and QUE. In all other cases the p-value was less than 0.05.

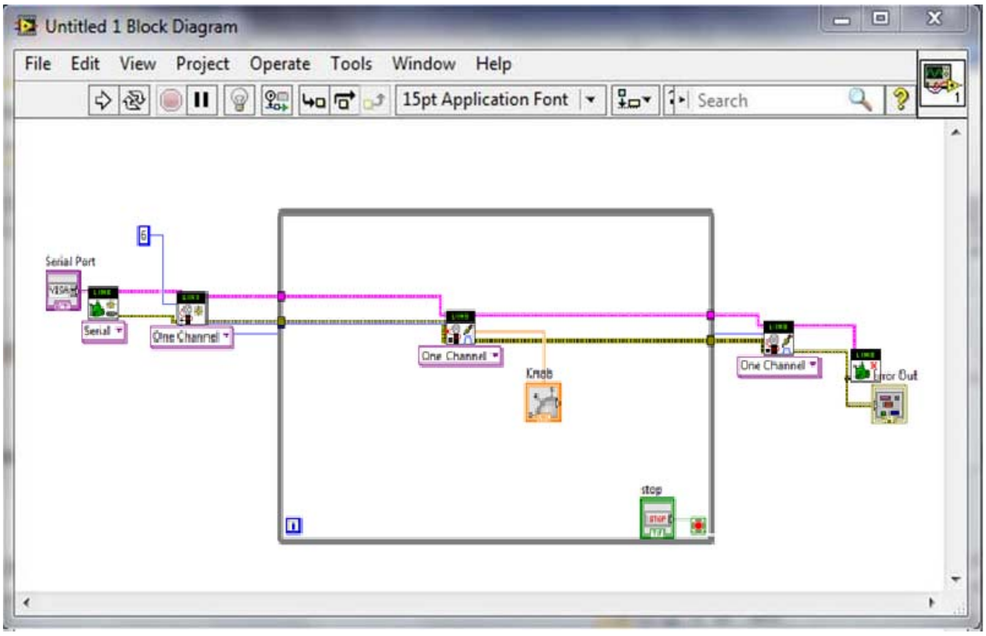

Fig. 4 Block Diagram in LABVIEW with servo motor connection-A design process

In addition to the questionnaire for the self-efficacy for the practices of CT, an additional questionnaire was given in order to register students' views for the "Computer Programming Self-Efficacy Scale (CPSES)" [26]. We changed some of the questions included questionnaire in order to be in alignment with Physical Computing. For example, instead of "I can write syntactically correct C/C++ statements" we used "I can write syntactically correct LABVIEW statements" and instead of "I can write logically correct blocks of code using C/C++", we used "I can write logically correct blocks of code using LABVIEW".

The use of this questionnaire is justified by its content validity to measure computer programming in relation to self-efficacy. The inclusion of the questionnaire for the Computer Programming Self-Efficacy is justified by the relation of Physical Computing with "computation" and "computing". According to [3], [5]-[7], "computing includes computer science, computer engineering". According to [7], computing is related to engineering design. The term computation is also a fundamental component of STEM disciplines as they are practiced in the professional world [27].

According to [28], "computing" includes "computation". Our motivation for examining the relation of programming with self-efficacy was based on the argument "if CT concepts are related to self-efficacy should we expect the same for programming and self-efficacy in a physical computing environment which is a computation environment like the LABVIEW?"

The results of applying the CPSES questionnaire verify our hypothesis. A pair t-test applied gives ($p = 0.003$), indicating a significant change when the physical computing-inquiry based model is applied. In addition, we noticed a strong correlation between the answers of CPSES and the answers in the CT questionnaire [3], after the instruction, which indicates a strong relationship between the CT practices and computer programming.

IX. Conclusions

Our results are in alignment with research findings that indicate that physical computing is an effective tool of teaching scientific concepts and can be the entrance to computer programming concept(like e.g. the loop concept used in our example) [15], [29].

As it can be assumed, physical computing has a dual role on motivating students to explore a scientific concept and as an instructional strategy which can trigger students' interest to be engaged in CT practices and computer programming concepts, in alignment with the scientific and engineering practices and the engineering design process [1].

Data collection, in a form of tables or graphs in LABVIEW, triggered students' interest -according to a preliminary quantitative analysis- for exploration of hidden patterns and decomposition of a problem and the Computational Science epistemology.

Physical computing is strongly related to real-time experiments, which follow the computational science method in the experimental step (see above for the spaces of the computational experiment) and can be easily considered as an inquiry tool according to [21], [22] while it is correlated with CT practices, as presented in Table I. The results of the current study indicate a strong effect of physical computing regarding introductory computer concepts and crosscutting ideas for teaching.

Physical computing can serve as a platform to reveal CT practices when the STEM content epistemology is applied in alignment with the computational pedagogy model. Physical computing provides the tools to communicate with the real world and test students' model according to the engineering design cycle.

Before the instruction, students were presented with some ready examples in LABVIEW and Arduino and they were taught about this type of optical programming, with focus on concepts they had met during their courses in engineering. They had also attended a course in C++, so they already knew the basic programming concepts (control structures, loop control etc.). They also knew the concept of motor (so they understood the operation of servo).

At the beginning of the course, during the hypotheses step of the computational experiment, they were asked to describe phenomena that exhibit periodicity and how periodicity appears in mechanics, electromagnetism etc. During this stage, it was apparent that they could connect text based commands with the measurements of physical quantities and they expressed the opinion that the language of LABVIEW is more proper to "take numbers" for a quantity because they have a visual representation.

Students considered -after some presentations by the instructor- that the interface and the blocks of LABVIEW were easily understood and operational. Severe difficulties were expressed for the understanding of connection of elements inside and outside the loop structure. These were attributed to the fact that these connections are related to the procession of data and students were not familiar with these processes. Another issue recorded by discussions with students was their difficulty to recognize which is the "system" which is the surroundings. This was reflected in programming since they had difficulties to include some inputs in Labview.

Students expressed the issue that the interface of Labview operates as a scaffolded computational tool which connects the digital world with the physical environment.

Results for TAD (Testing and Debugging (i.e. performing intermediate testing and fixing problems while developing the code)) and QUE (i.e. Questioning – working to understand each part of the code instead of using code that is not understood well) are not statistically significant ($p > 0.05$). After the intervention, we discussed with students about these dimensions and we tried to analyze their responses.

Students consider that debugging needs further teaching about the kind of issues included to data processes and the way they are connected inside and outside the loop.

Despite the fact that CT is something beyond computer programming, our results indicate that for the self-efficacy structure, CT and computer programming are strongly related.

During the course, we applied the computational pedagogy model as a theoretical framework for model development and collection and analysis of data for system design (an engineering design process).

During the intervention, a series of didactic scenario were presented using certain repositories with simulated models. The scenario was based on the constructivism theories but in some of them the educational material was provided –without students' intervention- from well-known repositories.

Students expressed the view that despite the fact that these repositories contain very good material, they are restrictive for the developing real-time simulations, and they can serve mainly as virtual experiments. At the last space of the computational experiment (the generalization/metacognitive phase) students started to think in a more abstract ways in order to find common factors that govern a phenomenon that exhibits periodicity.

One of the scenarios presented was about the periodicity in RC circuits and the exponential decay (one core idea) appeared in many phenomena. The intervention was mainly quantitative but a preliminary qualitative analysis was also implemented with the purpose to be extended.

The aims of further research include the development of more computational pedagogy education scenario with didactic activities for improving students' self-efficacy in CT and programming practices and concepts which will not be restricted to specific disciplines but focus will be given to cross-cutting ideas following the STEM content epistemology

Finally, we recognize that some of our results may rely on the small sample size of our experiment, and larger scale studies will be needed to fully investigate the effectiveness of the suggested tools to self-efficacy for CT.

REFERENCES

[1] NGSS, "A Framework for K-12 Science Education Practices, Crosscutting Concepts, and Core Ideas". Washington, DC: The National Academies Press, 2012.

[2] Psycharis, S. Computational Thinking, Engineering Epistemology and STEM Epistemology: A primary approach to Computational Pedagogy. International Conference on Interactive Collaborative Learning, ICL 2018: The Challenges of the Digital Transformation in Education pp 689-698 https://link.springer.com/chapter/10.1007/978-3-030-11935-5_65

[3] J. Wing, "Research Notebook: Computational Thinking--What and Why?", 2011, March 6,. (J. Togyer, Ed.) The Link Magazine. Retrieved March 29, 2016, from http://www.cs.cmu.edu/link/researchnotebook-computational-thinking-what-and-why.

[4] J. L. Weese, R. Feldhausen, and N.H. Bean, "The Impact of STEM Experiences on Student Self-Efficacy in Computational Thinking," Proceedings of the 123rd American Society for Engineering Education Annual Conference and Exposition (ASEE 2016). New Orleans, LA, USA.

[5] K. Shirey, "Teacher Productive Resources for Engineering Design Integration in High School Physics Instruction (Fundamental)". In Proceedings of the 2017 ASEE Annual Conference, Columbus, OH, June 2017.

[6] A. Rugarcia, R.M. Felder, D.R. Woods and J.E. Stice, "The future of engineering education: I. A vision for a new century". Chemical Engineering Education. 34, 16–25. 2000.

[7] L. Katehi, G. Pearson and M. Feder, "Engineering in K-12 education: Understanding the status and improving the prospects". Washington, DC: National Academy of Engineering and National Research Council. 2009.

[8] National Academy of Engineering (NAE) & National Research Council (NRC). (2014). STEM integration in K-12 education: status, prospects, and an agenda for research. The National Academies Press, Washington.

[9] D. Weintrop, E. Beheshti, M. Horn, K. Orton, K. Jona, L. Trouille and U. Wilensky, "Defining Computational Thinking for Mathematics and Science Classrooms". Journal of Science Education and Technology, 2016, 25(1), 127-147. DOI: 10.1007/s10956-015-9581-5.

[10] A . Burrows and T. Slater, “A proposed integrated STEM framework for contemporary teacher preparation,” Teacher Education and Practice, 28(2/3), 2015, pp. 318–330.
[11] TJ. Moore, MS. Stohlmann, HH. Wang, KM. Tank, A. Glancy and RH. Roehrig, “Implementation and integration of engineering in K-12 STEM education.” In JStrobel, S Purzer, M Cardella (Eds.), Engineering in precollege settings: Researchinto practice, 2014, (pp. 35–59). West Lafayette: Purdue University Press.
[12] T. J. Moore, “STEM integration: Crossing disciplinary borders to promote learning and engagement”. Invited presentation to the faculty and graduate students of the UTeach Engineering, UTeach Natural Sciences, and STEM Education program area at University of Texas at Austin, December 15, 2008.
[13] S. Libow Martinez and G. Stager, “Invent to Learn - Making, Tinkering, and Engineering in the Classroom,” Torrance CA: Constructing Modern Knowledge Press. 2013.
[14] S. Schulz and N. Pinkwart, “Physical computing in stem education”. In Proceedings of the Workshop in Primary and Secondary Computing Education (pp.134-135). ACM.November2015.
[15] M. Przybylla, and R. Romeike, “Physical computing and its scope-towards a constructionist computer science curriculum with physical computing”. Informatics in Education, 13(2), 225. 2014.
[16] R. Taub, M.Armoni and M. Ben-Ari, M. “The Contribution of Computer Science to Learning Computational Physics”. In I. Diethelm & R. Mittermeir (Eds.), Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages (Vol. 7780, pp. 127-137): Springer Berlin Heidelberg. 2013.
[17] O. Yasar, P. Veronesi, J. Maliekal, J. LittleL. S.E. Vattana and I.H. Yeter, “Computational Pedagogy Q Fostering a new method of teaching”. Presented at: ASEE Annual Conference and Exposition. Presented: June 2016. Project: SCOLLARCIT
[18] S. Psycharis and Kalia, M. “The Effects of Computer Programming on high school students' problem solving, reasoning skills and self-efficacy in Mathematics”; Instructional Science, 45(5), 2017, 583-602 10.1007/s11251-017-9421-5,
[19] S. Psycharis. “STEAM in Education: A Literature review on the role of Computational Thinking, Engineering Epistemology and Computational Science. Computational STEAM Pedagogy (CSP)” 2018. Scientific Culture, Vol.4, No.2, 51-72. https://sci-cult.com
[20] Bundy, “Computational Thinking is Pervasive”. Journal of Scientific and Practical Computing, 1(2), pp.67-69. 2007.
[21] T. Bell, D. Urhahne, D., S. Schanze and R. Ploetzner, “Collaborative inquiry learning: models, tools and challenges”. International Journal of Science Education,2010, 32(3), pp. 349-377
[22] L. D. Asay and M.K. Orgill, “Analysis of essential features of inquiry found in articles published in The Science Teacher”, 1998-2007. Journal of Science Teacher Education, 2010, 21, pp. 57-79.
[23] S. Psycharis, K Kalovrektis, E. Sakelalridi, K Korres and D. Mastorodimos. Unfolding the Curriculum: Physical Computing, Computational Thinking and Computational Experiment in STEM’s Transdisciplinary Approach. European Journal of Engineering Research and Science (EJERS) http://www.ejers.org/index.php/ejers/article/view/639/
[24] B.A. Weinberg, B.“A model of over-confidence”. Pacific Economic Review, 14, 502–515. 2009
[25] J. M.Greenwald, “Antecedents of core confidence latent construct: Direct and reciprocal links”. Dissertation Abstracts International Section A, 72,270. 2010.
[26] O.Kormaz and H. Altun, “Adapting Computer Programming Self-Efficacy Scale and Engineering Students’ Self-Efficacy Perceptions”. Participatory Educational research, PER, Vol 1(1), pp2-31,June 2014
[27] K., Jona,, U. Wilensky., L Trouille, M. S., Horn, K. Orton,, D., Weintrop & E. Beheshti, (2014). “Embedding computational thinking in science, technology, engineering, and math (CT-STEM). In future directions in computer science education summit meeting, Orlando, FL”.
[28] S. Chande (2015). “A Conceptual Framework for Computational Thinking as a Pedagogical Device. International Journal of Innovative Research in Computer and Communication Engineering” vol. 3, Issue 11, November 2015
[29] R. Mason and G. Cooper, “Mindrostrms Robots and the Application of Cognitive Load Theory in Introductory Programming”. Computer Science Education, 23(4), pp.296-314. 2013.