International Journal of Engineering Education Vol. 37, No. 4, pp. 1071–1079, 2021
Printed in Great Britain

0949-149X/91 $3.00+0.00

Interactive and Collaborative Experimental Platforms for Teaching Introductory Internet of Things Concepts*

MARKO SARAC
Faculty of Computing and Informatics, Singidunum University, Danijelova 32 Street, 11000 Belgrade, Serbia.
E-mail: msarac@singidunum.ac.rs

SASA ADAMOVIC
Faculty of Computing and Informatics, Sinergija University, Raje Banicica bb Street, 76 300 Bijeljina, Bosnia and Herzegovina.
E-mail: sadamovic@sinergija.edu.ba

MUZAFER SARACEVIC
Department of Computer Sciences, University of Novi Pazar, Serbia. E-mail: muzafers@uninp.edu.rs

The traditional approach to teaching computer system and IOT involves the presentation of a large amount of theoretical and mathematical data and materials to students. The common approach is a theoretical explanation of the principles, on the whiteboard and if possible, with the use of graphical animations. The main issue in this area is raising interactivity and getting collaboration between students on project assignments. In this paper, authors are proposing more interactive approach that could be beneficial to teachers of related subjects. Presented are the results of using software and simulating tools (in combination with Arduino hardware development solution) for teaching collaboratively and interactively courses related to IoT. The paper presents ideas for implementation of practical and experimental project assignments, student engagement and collaboration. The paper also presents the results of authors after applying this approach to one generation of students. The application access with practical assignments gave markedly better results, measured through student engagement, classroom attendance and distribution of student results. In this manner, the students had a better understanding of the basics of the microcontrollers, sensors, IoT and were willing to engage in programming of microcontrollers at the lower level. Students have also shown better results in closely related courses such as the courses of operating systems. In addition, some students continued to use IoT platforms for future projects (in college or practice).

Keywords: interactive approach; IoT computer science teaching; interactive teaching; computer science

1. Introduction

The course entitled Fundamentals of Computer Science belongs to a group of basic subjects at the Faculties of Information Technology and Electrical Engineering, it covers a wide range of areas. Course is aimed at gaining knowledge about the functioning of the computers, introduction to mathematical and logical fundamentals that underpin the work of digital systems, as well as the study of the basic hardware components of computers, their characteristics and principles of microcontrollers and IoT functioning.

The authors have identified that the main problem in the knowledge transfer from the field of computer science is different foreknowledge and student motivation in the field. Thus, the primary objective of the course is to introduce students to the mathematical foundations of computer techniques and point to the significance of the binary number system possibilities as the most appropriate for modern electronic technology. The course involves students in the practical work with IoT controllers on the basics of computer organization through the Intel x86 family and ATmega328 chips and their functioning. It also involves the presentation of hardware organization in the personal computer through the demonstration and implementation of the most important components. Due to the previously mentioned main goals of the course, it is necessary to actualize the course annually. To that end, it is necessary for the teacher and the course to support the appropriate educational approaches, materials and learning conditions.

This paper describes a model for interactive teaching of IoT and computer techniques, used in the course Fundamentals of Computer Engineering during the 2017/2018 and 2018/2019 school year. The course was held with a group of 120 students in each study year. The model is based on the use of mTutor (on-line testing platform), Tinkercad (on-line IoT simulation platform) and Arduino software simulating tools in combination with Arduino hardware development solution. The paper shows the evaluation of the proposed model in terms of student engagement, classroom attendance and final scores of students in terms of progress compared to the previous generation.

* Accepted 6 March 2021.

2. Related Work

In this section, the authors discuss other existing models for interactive and collaborative approaches to learning the fundamentals of computer science [1–7].

Authors in [1] describe the challenges faced in the implementation of active learning methods and Collaborative Teaching and Learning Strategies for Communication Networks. The experiences resulted in a blended methodology, which combines collaborative and problem-based learning with a learning management system.

In paper [2] Jin presents Collaborative Instructional Models for Teaching Community Service to Engineering Students. This study aims to develop instructional models for service learning in engineering education and verify their effectiveness using a formative research methodology. This study examined the effects and improvements of instructional models for service learning by applying the models to the service-learning courses.

In their paper [3] Igor Zubrycki and Grzegorz Granosik also dealt with the problem of introducing modern robotics and computer components to their students. They believe that the benefits of using ROS (Robot Operating System) are vast and worth our work of finding skillful methods, easy to use tools and appropriate knowledge, to involve even less "computer science type" students in using this modern robotic tool. They have also reported the results of students' projects and concluded that the best way to introduce this framework is to use simplified solutions. Their students are not experts in computer science and have little experience in typical software development. Yet they have the broad knowledge of other disciplines, which can introduce them to ROS. Making use of physical devices such as Arduino boards with sensors makes ROS functionality easier to understand and gives more motivation than just a simulation. The authors Igor Zubrycki and Grzegorz Granosik confirmed our assumption that experiments with online tools convinced them that this approach is also attractive.

Also, in [4] Valdivia and Nussbaum describe the application and effects of technological support for collaboration in a computer science course for engineering students. The technology in question is based on a wireless network of PDAs that implements a classroom dynamic to stimulate communication, discussion while agreeing on questions put to students.

"Problem-Based Learning for Foundation Computer Science Courses" [5] by authors J. Kay, M. Barg, A. Fekete, T Greening, O Hollands, J. H. Kingston and K. Crawford paper describes some of the challenges and how the authors have designed problem-based learning (PBL) courses to address them. The authors discuss that the problems were keen to overcome: the purely technical focus of many courses; the problems of individual learning and the need to establish foundations in a range of areas which are essential for computer science graduates. The authors conclude with a summary of experience over three years of PBL teaching and discuss some of the pragmatic issues around introducing the radical change in teaching, maintaining staff support, and continuing refinement of our PBL teaching.

Author M. Ben-Ari in his article "Situated Learning in Computer Science Education" [6] examines situated learning within the context of computer science (CS) education. Situated learning accurately describes some CS communities like open-source software development, but it is not directly applicable to other CS communities, especially those that deal with non-CS application areas. Nevertheless, situated learning can inform CS education by analyzing debates on curriculum and pedagogy within this framework. CS educators should closely examine professional CS communities of practice and design educational activities to model the actual activities of those communities.

The courseware engine as a virtual classroom for active and collaborative teaching present in paper [7]. This paper focuses on explaining how TELD is used as a virtual classroom for active and collaborative teaching and learning. TELD represents a method of 'teaching by examples and learning by doing' that unifies several contemporary methods such as problem-based learning, project-based learning, and case method in medical, engineering, and business education respectively.

Paper [8] explores a novel approach to harnessing the Internet-of-Things (IoT) as a teaching and research vehicle in education. The IoT is the latest innovation and increasingly growing area to be implemented in all areas of life especially in higher education [9].

Paper [10] presents a review of traditional education (previous classroom education) and online education. In conclusion, IoT has overcome the disadvantages of online education; moreover, IoT removes the traditional barriers of teaching and learning. On a different note, understanding IoT with its advantages and disadvantages will help reach its aimed vision, and here we can benefit from it greatly. This paper discusses the usefulness and applications of IoT in the field of education. Moreover, it tries to present the recent research works, challenges, and impact of IoT in future education [11]. IoT stands to change dramatically the way universities work and enhance student learning in many disciplines and at any level. It has huge potential for universities or any other educational institutions; if well prepared to ensure a widespread and successful implementation by leadership, staff, and students. IoT needs development where universities can lead. Academics, researchers, and students are in a unique place to lead the discovery and development of IoT systems, devices, applications, and services [12].

3. Proposed Solution

The first step towards improving the teaching of the course entitled Fundamentals of Computer Science meant to increase the level of interactivity between students and presented material. The traditional approach involved the implementation of instructor lectures presentation with the help of PowerPoint presentations with the passive participation of students as listeners, and oral discussion. While the exercises were implemented with the classical demonstration method of implementation using PowerPoint materials and tasks solving with the help of the traditional table, paper, pens and written tasks. The main drawback of this approach is the low level of abstraction and the lack of interaction during lectures and exercises. Besides, a considerable amount of time was needed for the realization of complex tasks that demonstrate the specific circuits and components. Also, as the realization of the course progressed, the authors observed that it was hard for the students, who did not continuously attend and master the lecture and exercise parts, to understand and master the material as well as the colleagues who attended all the classes. It should also be borne in mind that the level of students' prior knowledge of computing in each group differs, which is reflected in different possibilities for implementing complex tasks and asynchronous group work.

To facilitate the cooperation among students, the problem-solving methodology has been developed that uses a hardware solution, a software simulator and competition groups through electronic tests. The model enables the cooperation of students in problem-solving and smoother progress at the group level.

The application of the proposed model in the course with the specified extensions to the traditional model has shown significantly better results for students (engagement, grades, attendance, activity, creativity), which are discussed in the section related to the evaluation of solutions.

3.1 The Use of Modern Solutions

The authors have been teaching the course entitled Fundamentals of Computer Science since 2011. The first generations of students attended the course relying on the previously described traditional teaching methods. Since 2015, the implementation of the teaching method includes the mini metrics tests within solutions for testing and adopting knowledge (mTutor) developed at the University. The solution mTutor allowed the verification of mastering the course material within a relatively short period. The student was committed to mini tests at the end of each week. Each mini test lasted 15 minutes and the students were able to solve problems in areas presented in the teaching week. The implementation of this solution significantly improved the course, since the professors were able to receive considerable feedback on the areas that students mastered, but also those they found problematic. By implementing the solution such as mTutor, it was possible to test their knowledge and to transfer knowledge from one student to another in a large group in a relatively short time frame.

During the period from 2015 until 2017, based on the results obtained from students, each generation of the authors was working on the course syllabus,

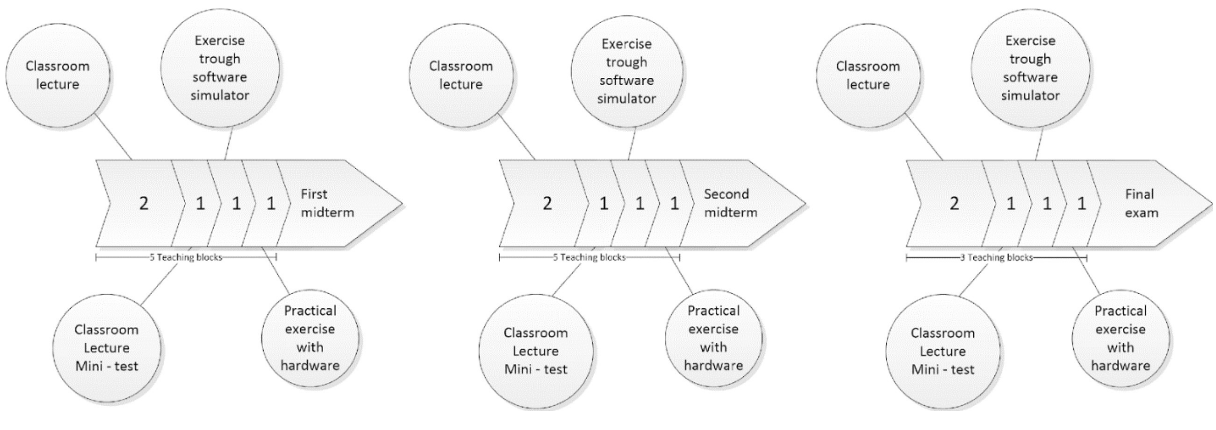

Diagram 1. Graphic model of course implementation with mTutor mini tests, tests, Thinkercad software package and Arduino Uno Starter kit HARDWARE solution

continually improving it. Also, considerable resources were used to develop better presentations and teaching materials. The mTutor solution has significantly improved feedback and allowed better preparation of the students' assessment. However, the mTutor solution was not enough, as it could not make the lectures more interactive and provide a better way of knowledge adoption and transfer. Besides good feedback, the solution had its limitations, which were reflected in the fact that the basis of knowledge transfer remained the same.

The next step was the implementation of a software simulator in the course syllabus. The first implementation included the author's solution that has been developed in Java. Already in the first implementations, it turned out that these solutions were not attractive to students and did not develop an interest in further interaction or an increasing level of interest towards the subject. The next attempt was by the convergence of hardware (tangible solutions). The choice of hardware solution that would offer adequate simulators for independent work and experiments was adopted in favor of Arduino. The platform was designed and developed for learning and knowledge transfer, and therefore the authors expected positive feedback by the students.

As the authors will show in this paper, it turned out that a good choice of hardware solutions and a software simulator with a constant level of knowledge adoption in the transferred knowledge is the winning combination of quality. The authors believe that this combination can achieve significant results.

To check the quality of the transferred knowledge and the knowledge reception, a system for electronic testing mTutor was developed. The system mTutor supports the teaching process with a series of weekly mini tests after the lectures and exercises. Professors and assistants use these tests to get feedback on the areas that students have not mastered well so that they could devote more time to those areas in the upcoming period.

The next challenge of the lecturing process was the more significant involvement of students. Relying on their previous experience with the self-developed solution, the teachers chose a proven and extremely user-friendly software package Tinkercad by Autodesk. This online software package allows both beginners and experienced users to create IoT simple and complex electronic schemes using generally accepted and easy-shelf electronic components. Additionally, this solution has the possibility of simulating the programming work of ATmega328P microcontroller, which is implemented on one of the most popular development boards Arduino Uno.

The implementation of hardware parts for students to work with was the next logical step of the previously mentioned solutions. It was done so that students got even closer and more involved in the field of IoT, microcontrollers, sensors, and electronics. The teaching staff decided to implement innovative hardware through the Arduino Uno Starter Kit [16] plus additional sensors like ultrasonic, infra-red, electromotor, etc. The package itself was designed for future users through over 15 exercises of varying difficulty hardware and software implementations.

Due to the introduction of new elements into the teaching syllabus, the course syllabus needed to be modified so that it could correspond to the actual model being used for teaching modified course with new lecture materials and equipment.

3.2 Labs examples

This part of the paper contains examples of laboratory exercises based on the use of previously described solutions [12–14]. The laboratory exercise is one of the introductory laboratory exercises in the course. It refers to the use and development of electronic elevator circuits as one of the examples whose usage students encounter daily. Familiarity with the case is identified as a key factor for students interested in solving the problem. Another laboratory practice refers to something more complex, implementation of acceleration, temperature, and light sensors to control devices custom settings. This exercise is more complex than the exercise related to the elevator circuit and involves the participation of more students when it comes to collaborative learning. In addition to selecting the right elements, it requires the right program on a controller and the appropriate parameters for simulation of the processor chip itself.

3.3 Elevator Circuit

The elevator circuit is one of the easiest ways to move a student's attention closer to the base of electronic components and electronic schemes. In today's elevators, more complex elements are being used than those realized in laboratory exercises. Nonetheless, elevator circuit example logic and implementation through logic gates are very suitable for the demonstration of the basic concepts and can be applied in practice. Fig. 1 represents a simple model of realization of elevator circuit via 74HC series of integrated chips and via 8-bit programmable controller ATmega328 that can be found on Arduino Uno simulation bord.

The observed parameters include three positions of micro switches that are realistic representations of the microswitch being used on an external door, inner door, and pressure sensors in the cabin floor

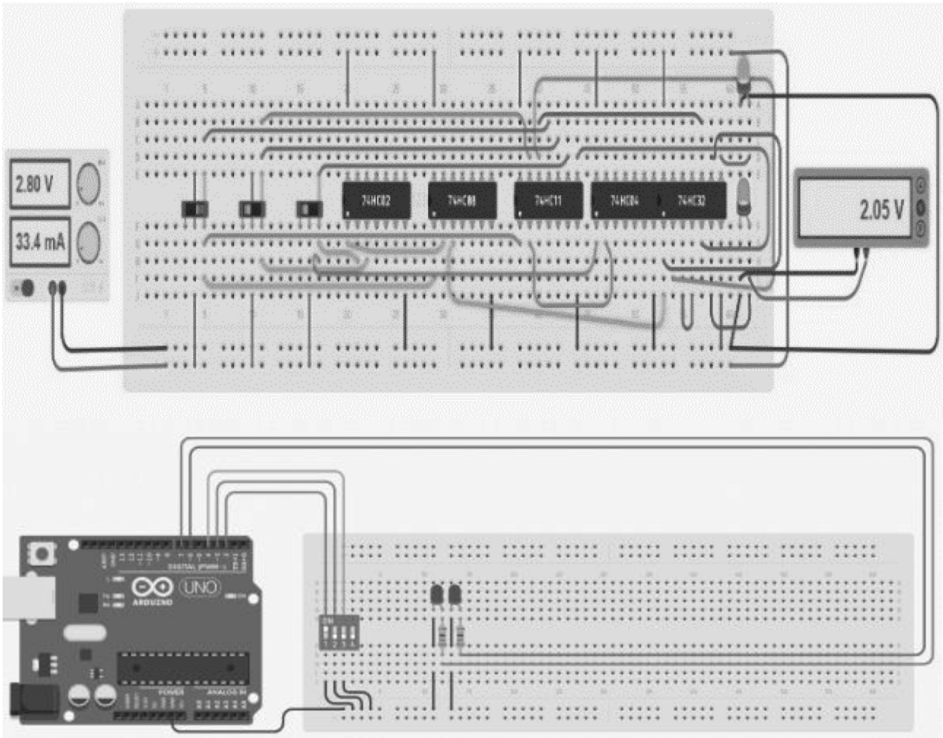

Fig. 1. Visual comparison between 74HC logic gates integrated circuits vs programmable microcontroller, simulated realization in Tinkercad software.

of the elevator. The students receive a task (previously described) in the form of textual problems to which they adopt the electronic elements and propose solutions in groups or independently.

3.4 Adjusting Electronic Devices to Physical Ambient

While the first scenario could have been successfully simulated in the Tinkercad solution, and software switches used could successfully represent the status of physical microswitches in the elevator, problems for the simulation environment appeared with the sensors used in real-world scenarios. The problems in working with the simulator occurred during examples with temperature and light sensors. The values of temperature and light sensor could successfully be represented as a digital value in a simulator. Nonetheless, the teachers noticed that students better accept information and knowledge when they can try and see what these values and parameters mean in the physical world. The values represented by the elevator micro switches that have exclusive state 0 or 1 are easily conceivable, however, with conditions such as temperature or the amount of light, it is not possible to define such exclusive parameters. These conditions have weighted values in microcontrollers that are converted from the analogue world to their digital form. It is possible to present these values via digital representations of the analogue world in the simulator, but it turned out that this way of presenting a sensor to the students involves poor student motivation. With this poor motivation for using software-simulated sensors, the desire to solve complex tasks was lacking, and it was necessary to overcome the given problem.

From this point, it has become evident that the use of Arduino Starter Kit will be necessary for the students to get involved and familiar with physical sensors that acquire data from the physical world and programming of microcontrollers that convert this information with digital systems. Although the first exercises were simple and used only a few electronic components, they managed to get students attention and were also able to adequately present the possibility of implementation of electronic components to collect analogue information from the physical world and transforming them into digital form. The TMP36 analog temperature sensor has been used. It outputs an analog value that is proportional to the ambient temperature. For the exercise with an ambient light sensor, three photoresistors and RGB LED have been used in combination with RGB foils represented in Fig. 2.

As expected, it turned out that these exercises motivate students to implement project tasks as described in project documentation and to develop their ideas and modify the existing ones. The

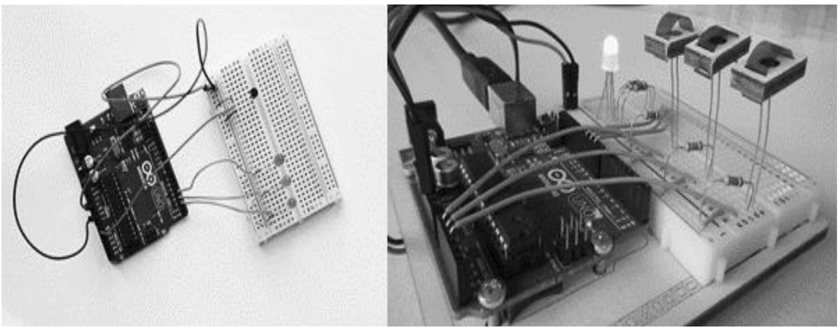

Fig. 2. The physical representation of described exercises of acquiring analogue data into the digital world.

aforementioned solution has triggered creativeness and teamwork in solving problems and tasks that (in previous years teaching) were often problematic and monotonous, and therefore difficult to understand.

3.5 Application of Mini Tests

Along with the introduction of simulation software for electronic devices and simulation of the environment in Tinkercad software, a mini test was introduced for monitoring the student progress in the observed area. Mini tests were formed through the electronic test and were represented as tasks with multiple-choice answers with a self-developed software system named mTutor. The questions in the context of lectures and exercises were formed solely based on the processed material in the given teaching week. This test enabled students to review how well they mastered the course material and provided the teaching staff with feedback on areas that need more detailed processing.

The tests included between 15 and 20 questions, depending on the area being covered in the lecture week. The tests were based on multiple choice questions with predefined answers. The tests were formed from a predefined database of questions, so each student received a unique test. It guarantees independent preparatory work of students for the test. The tests showed that the individual work was a better solution for the mini test because it developed competitive spirit among students, which enhanced student attentiveness during the presentation of study material and better results in after class tests. In Arduino and Tinkercad simulations, a better solution was to provide a group work task, since it led to faster and more efficient solving of complex tasks and assignments. It also led to achieving self-developed solutions not mentioned in textbooks and manuals.

In each of the tests, students, student data, the results previously achieved in the mini test and the computer on which the test was performed were monitored. After the test, the student would have received the notice of the percentage performance on mini tests but could also see their tests and correct and wrong answers for each question.

Similarly, the teachers also received individual as well as group test results. There is also the teaching part of the application, where teaching staff can define each of the individual tests, change or adapt the representation of questions that appear in the test. Besides, after doing a group mini test teacher can determine how specific issues were represented in the tests, but also how certain issues are resolved successfully or unsuccessfully.

The example generated for teaching staff upon the test is shown in the figure below, where the first column presents the number of students who received a particular question within the mini test, while the next column presents the percentage of successful response rates. Fig. 3 shows the possibility of analyzing the percentage of successful response rates for the question in the second course week.

4. Evaluation

The introduction of the described tools resulted in good student reaction and apparent positive effects. One of the most essential observed effects is the exchange of knowledge and understanding between students. The students who adopted the knowledge more quickly understood the concepts positively and influenced the students who needed more time, primarily in the desire to help their colleagues and to foster competitiveness and provide themselves with aides to perform scenarios that required more participants. In this way, teachers avoided student polarization and separation of groups depending on the speed of mastering the material, and the described method enabled faster progress of the entire group. The teachers experienced similar student activities and behavior in their previous work with CrypTool software [15].

Additionally, the course attendance has significantly increased so that more than half of the teaching class in 2018/19 attended by more than 75% of students, compared with 59% of students

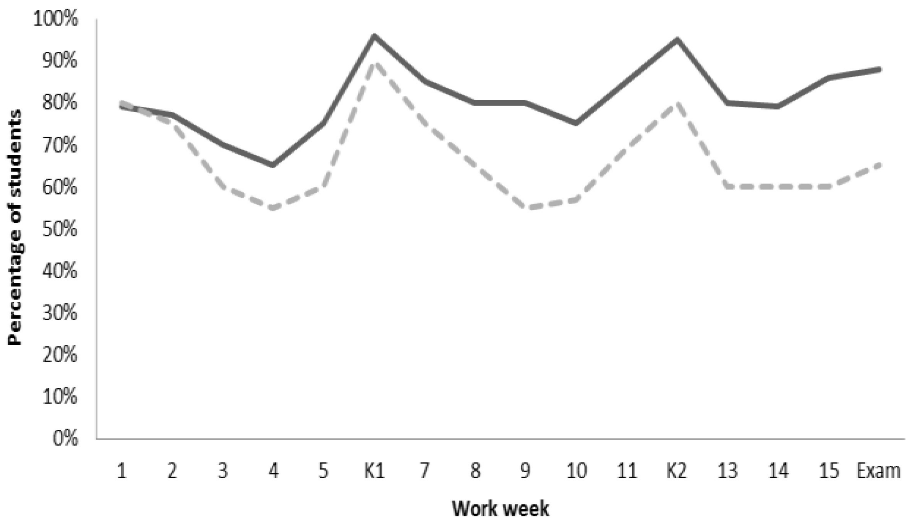

Fig. 3. Percentage of student classroom attendance through 15 workweeks, comparation of attendance between two generations through the weeks.

in the previous year. Fig. 3 provides a comparative analysis of course attendance for student generation 2017/18 (dotted line) and generation 2018/19 (full line). The first and the second midterm causes peaks in course attendance in week 6 and week 12.

The mean of the control group results was 7.34, the standard deviation was 1.53 and the variance was 0.09. For the results of the treated group, the mean was 8.12, the standard deviation was 1.43 and variance was 0.10. Based on the comparison of results with the corresponding values in T – table (for a statistically acceptable p-value of 0.0001 calculated based on degrees of freedom for both groups), a statistical difference could be observed.

The improvements could also be observed in the final exam student results and grade distribution in the first three exam terms upon the course completion. In the following figure, Gaussian distribution was used for ideal measurement. A comparative analysis was performed for student generation 2017/18, which did not have the course with paper described methods, and generation 2018/19 with paper described methods implemented in the course syllabus. As represented in Fig. 4, generation 2017/18 had much lower average grade and much higher numbers of a minimum passing grade (grade 6), and the number of maximum grades (grade 10) for this generation was trivial.

After applying paper described methods, next generation, generation 2018/19 had a much better average grade, a much lower number of minimum grades, and a higher number of maximum grades. It is worth mentioning that generation 2018/19 also did not fulfil ideal Gaussian distribution, but it was much closer to it than generation 2017/18.

Table 1. T-Test result and intermediate values

Intermediate values used in the calculation			
Control Group		Treated Group	
Mean	7.34	Mean	8.12
SD	1.53	SD	1.43
SEM	0.09	SEM	0.10
N	115	N	126
The standard error of the difference			0.138
Degrees of freedom			473
T value			5.5888
Confidence interval			
CG mean – TG mean			–0.77
Confidence interval (95%)			–1.05 to –0.50
P value and statistical significance			
P value			0.0001

5. Conclusion

This paper presents a solution for interactive and collaborative work of students in mastering the material basis in the field of computer science and IOT related courses. The proposed solution is based on engaging students with the use of self-evaluation through electronic testing, simulations of realistic scenarios with online IoT electronic components in Tinkercad, and real-world implementations with Arduino Uno Starter Kit hardware solution.

The use of proposed solutions ensures the appropriate level of abstraction in working with IoT electronic components and principles and basic algorithms of programming microcontrollers. The presented methods are not time-consuming regarding the realization of theoretical procedures in implementation. The focus was moved from understanding the mathematical foundations and algo-

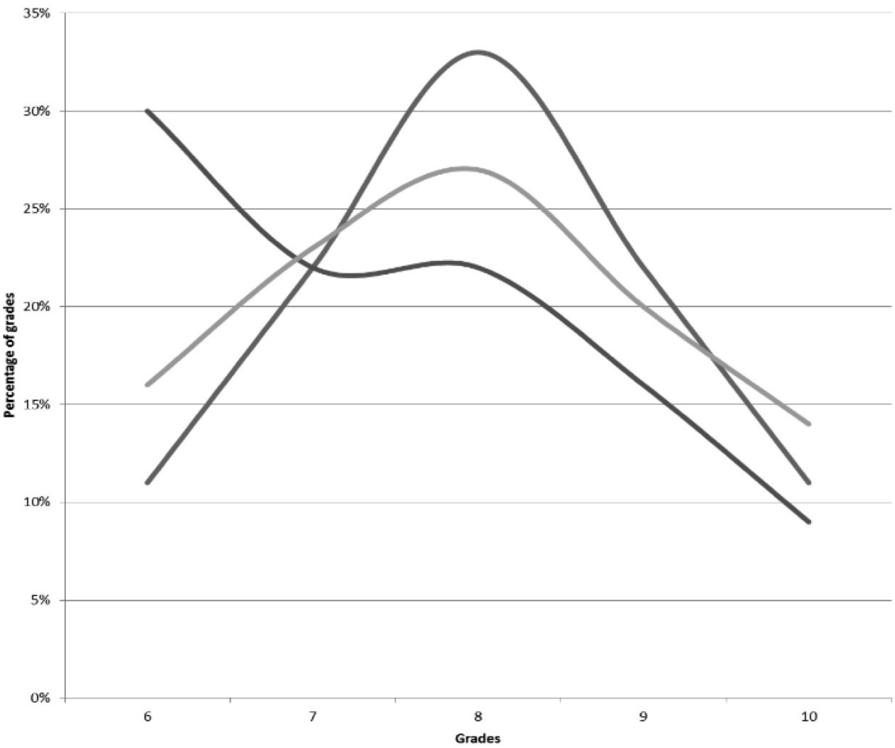

Fig. 4. The grade distribution for students after the final exam, Ideal Gaussian distribution (dark gray line), distribution for generation 2017/18 (full black line), and distribution for generation 2018/19 (light grey line).

rithms to understanding the architecture and complex systems and the use of best practices for their implementation, which was a more suitable approach for the first-year students.

The paper presents and discusses the results of the approach being implemented. Also, a better final grade of students, a significantly higher interest of students in this field could be observed, which is manifested in the form of greater classroom attendance and more active and creative participation in lectures and exercises. Besides, there have been better continuity in studying at both individual and group level. Based on a good experience within the evaluated group, further experiments will be made. But the current model is already promising and can be recommended to teachers working in the field of computer science and IOT related courses.

For further work, the teacher will continue to promote their teaching model and course syllabus with a new software and hardware solution to enable future generations of students to assimilate course materials more efficiently.

References

1. F. N. Leite, E. S. Hoji and H. Abdala, Collaborative Teaching and Learning Strategies for Communication Networks, *International Journal of Engineering Education*, **34**(2), pp. 527–536, 2018.
2. S. H. Jin, Collaborative Instructional Models for Teaching Community Service to Engineering Students, *International Journal of Engineering Education*, **34**(6), pp. 1897–1909, 2018
3. I. Zubrycki and G. Granosik, Introducing modern robotics with ROS and Arduino, including case studies, *Journal of Automation, Mobile Robotics & Intelligent Systems*, **8**(1), 89–493, 2014.
4. R. Valdivia and M. Nussbaum, Face-to-face collaborative learning in computer science classes, *International Journal of Engineering Education*, **23**(3), pp. 434–440, 2007
5. J. Kay, M. Barg, A. Fekete, T. Greening, O. Hollands, H. Kingston and K. Crawford, Problem-Based Learning for Foundation Computer Science Courses, *Computer Science Education*, **10**(2), pp. 109–128, 2000.
6. M. Ben-Ari, Situated Learning in Computer Science Education, *Computer Science Education*, **14**(2), pp. 85–100, 2004.
7. G. Q. Huang, B. Shen and K. L. Mak, TELD: Courseware engine as a virtual classroom for active and collaborative teaching, *International Journal of Engineering Education*, **17**(2), pp. 164–175, 2001
8. J. Chin and V. Callaghan, Educational Living Labs: A Novel Internet-of-Things Based Approach to Teaching and Research, In: *9th International Conference on Intelligent Environments*, Athens, 18–19 July, pp. 92–99, 2013.
9. H. Shaikh, M. S. Khan, Z. A. Mahar, M. Anwar, A. Raza and A. Shah, A Conceptual Framework for Determining Acceptance of Internet of Things (IoT) in Higher Education Institutions of Pakistan, In: *International Conference on Information Science and Communication Technology (ICISCT)*, Karachi, Pakistan, 16-17 November, pp. 1–5, 2019.

10. L. Banica, E. Burtescu and F. Enescu, The Impact of Internet-of-Things in Higher Education, *Sci. Bull. Sci.*, **16**(1), pp. 53–59 2017.
11. S. Gul, M. Asif, S. Ahmad, M. Yasir, M. Majid and M. S. Arshad, A Survey on Role of Internet of Things in Education, *International Journal of Computer Science and Network Security*, **17**(5), pp. 159–165, 2017.
12. H. Aldowah, S. U. Rehman, S. Ghazal and I. N. Umar, Internet of Things in Higher Education: A Study on Future Learnin, *IOP Conf. Ser. J. Phys.*, **892**, pp. 1–10, 2017.
13. S. Monk, *Programming Arduino Getting Started with Sketches: vol. 2*, McGraw-Hill Education TAB, NY, USA, 2011.
14. J. H. Au and E. Gertz, *3D CAD with Autodesk 123D: Designing for 3D Printing, Laser Cutting, and Personal Fabricatio*, **1**, Make Community, LLC, 2016.
15. S. Adamović, M. Šarac, M. Veinović, M. Milosavljević and A. Jevremović, An Interactive and Collaborative Approach to Teaching Cryptology, *Journal of Educational Technology & Society*, **17**(1), pp. 197–205, 2014.
16. Arduino Starter Kit part list available on link, https://store.arduino.cc/genuino-starter-kit, Accessed 17 October 2020.

Marko Šarac, PhD degrees in 2009. and in 2013. respectively, from the Faculty of Informatics, Singidunum University, Serbia. He was born on November 16, 1981. He is Head of the IT department at Singidunum University from November of 2006. His areas of expertise are Informatics and computing, Electrical engineering and computing, E-Business. He is certified in the administration and database field by many major IT companies such as Cisco, Microsoft, Mikrotik, Google, SAP, IBM, HP, Solidworks, Oracle. He is the author of three books and several journals and conference scientific papers. His current research interests include security, computer networks, databases, and IoT.

Sasa Z. Adamovic, PhD is an Associate Professor at Singidunum University. He received his PhD title with a doctoral dissertation titled: ‘‘An class of systems for generating cryptographic keys based on biometric data’’, from the Singidunum University in Belgrade, in 2013. Currently, he is the Dean of the Faculty of Informatics and Computing at the University of Sinergija in Bijeljina, BiH. The research areas of professor Adamović are Cryptography and Security, Digital forensics, Biometrics, Information theory and coding, Java programming, Data Science, and Artificial Intelligence. He is certified in the administration and database field by many major IT companies such as Google, SAP, IBM, HP, and Oracle. He is the author of more than 80 scientific and professional papers printed in eminent international and national journals and proceedings of international and national scientific conferences.

Muzafer H. Saračević, PhD is an Associate Professor at the University of Novi Pazar, Serbia. He graduated in computer sciences (cryptography) at the Faculty of Informatics and Computing in Belgrade, obtained his Master degree from the University of Kragujevac, Faculty of Technical Sciences, and completed his PhD (field of computational geometry) at the University of Niš, Faculty of Science and Mathematics in 2013. Muzafer’s research areas are programming, software engineering, computational geometry, and cryptography. He authored and co-authored several university textbooks and over 150 scientific papers printed in international and national journals, and proceedings of international and national scientific conferences. He is a member of the editorial board for 8 journals. He worked on reviews for 10 international journals, 3 domestic journals, and many conferences.