

Article

HYDRA: Introducing a Low-Cost Framework for STEM Education Using Open Tools

Georgios Tsalmpouris ¹, George Tsinarakis ² , Nikolaos Gertsakis ³ , Savvas A. Chatzichristofis ⁴ 
and Lefteris Doitsidis ^{2,*} 

- ¹ Department of Mechanical Engineering, Hellenic Mediterranean University, 71140 Heraklion, Greece; tsal81@yahoo.com
² School of Production Engineering & Management, Technical University of Crete, 73100 Chania, Greece; tsinar@dpem.tuc.gr
³ Psychology Department, University of Crete, 74150 Rethymno, Greece; gernick2@gmail.com
⁴ Intelligent Systems Laboratory, Department of Computer Science, Neapolis University Pafos, Pafos 8042, Cyprus; s.chatzichristofis@nup.ac.cy
* Correspondence: ldoitsidis@dpem.tuc.gr

Abstract: STEM education is of paramount importance, especially in the lower levels of education, and it has been proven beneficial for students in many ways. Although there are various tools available, there are significant drawbacks mainly related to the cost and the ease of use. In this study, we introduce a new low-cost educational framework oriented toward elementary and secondary educational needs. The proposed system exploits open tools and low-cost devices. The system's core is based on the popular Arduino microcontroller, a low-cost device supported by a large community. The overall system was designed and developed, providing an expandable, modular system of low complexity suitable for students with no or low prior knowledge in related subjects, among others, to programming, embedded devices, sensors and actuators, as well as robotics. Our scope was to provide a system with a small learning curve. Practically, this makes it possible in a short amount of time for the students to perform appealing yet straightforward tasks which will boost their self-confidence and creativity, improve their technical skills and simultaneously provide a system with several capabilities usable in different kinds of projects. The introduced system was tested through a preliminary study using flow theory in a team of 68 students of the three last grades in an elementary school in Greece.

Keywords: STEM educational tool; Arduino; Petri Nets



Citation: Tsalmpouris, G.; Tsinarakis, G.; Gertsakis, N.; Chatzichristofis, S.A.; Doitsidis, L. HYDRA: Introducing a Low-Cost Framework for STEM Education Using Open Tools. *Electronics* **2021**, *10*, 3056. <https://doi.org/10.3390/electronics10243056>

Academic Editor: Carlos J. Bernardos

Received: 25 October 2021

Accepted: 1 December 2021

Published: 8 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The use of Science, Technology, Engineering and Mathematics (STEM) tools has been proven useful for both teachers and students in many learning contexts. These have been widely used in closely related fields, among others, for programming, automation, and robotics. In [1], the authors introduced conceptual frameworks regarding the practical implementation of STEM education concepts in different countries, while in [2], the authors compared the conceptualization and teaching practices in geographically proximate countries, since no homogeneous standards regarding this issue exist. However, even when students are not interested in STEM-oriented disciplines, they are motivated by them when they are involved as a way of teaching in other fields such as arts and music [3–5]. Reviews of the use of STEM education in early childhood [6] and on the status and trends in STEM education research [7] summarize the majority of the latest publications in this field and discuss a number of issues concerning its increased popularity. Based on the aforementioned publications, STEM tools are considered to be important and innovative and can shape education from kindergarten to university in a variety of fields [8,9].

The core idea behind STEM tools is the constructionism learning theory developed by Papert [10], based on concepts initially introduced in the constructivism theory by Pi-

aget [8,11]. Based on this approach, by manipulating and constructing objects, the students interact with their environment, continually adding new knowledge and building upon existing experiences by adapting previously held ideas to create new information [12].

STEM tools offer hands-on activities that encourage students to become active learners and create an interactive environment where they can investigate and work with complex real-world problems. In this sense, they construct and constantly reconstruct their knowledge and meaning through personal experience as they become active learners [3,8,13,14]. Physical devices can transform the procedure of learning into a fun activity that attracts and keeps students interested in learning, therefore enhancing students' interest and curiosity [9]. Through this procedure, students have the opportunity to improve their critical thinking and problem-solving skills [4,11]. At the same time, many studies report a positive impact on personal development, including collaboration, social and communication skills, cognitive, meta-cognitive, and social responsibilities [4,15]. By developing their self-confidence and self-direction, students increase their creativity, innovation and motivation [4,8,11,16]. Exposure to the use of STEM tools has also been proven useful in assisting students making career choices towards STEM domains and as it was highlighted in [17], a significant, positive, strong correlation between interpersonal skills, STEM exposure, career choice, family and school support, and external motivation exists.

Programming is perceived as an essential skill for everyone to learn with the potential of fostering computational thinking and problem-solving competence. Hence, a variety of different curriculum and different approaches to teaching programming have been proposed [18]. Despite its importance, programming is perceived by students as a difficult task to perform [5,19]. Students in programming courses must familiarize themselves with the fundamental programming concepts while they must simultaneously learn each programming language's rigid syntax and commands [20]. Learning to program becomes more difficult when learners are not native English speakers, as most programming languages use simple English keywords to represent syntactic and semantic rules. At the same time, students must learn how to solve a problem and transform their solution into a textual representation. Studies also show that the acquisition of programming skills is considered more challenging for female and younger students [14]. In this notion, learners perform a heavy cognitive effort during the programming learning process, leading to decreased motivation and satisfaction. Accordingly, students can adopt a negative attitude towards programming [20].

On these grounds, extensive research has been performed regarding the learning methods and how the educational environment can be transformed when STEM tools, i.e., educational robots, are used as a tool [21,22]. In a programming context, robots as tangible devices can provide a physical environment where students can manipulate physical objects to solve problems through innovative play. With the use of sensors and actuators, robots allow learners to explore and interact with the real world's complex problems while programming. They have the opportunity to constantly design and test their ideas while they receive immediate feedback on their solutions. Through experimentation, students improve their motivation and interest, which leads to easier knowledge acquisition and retention. When they reflect on and correlate problem solving strategies with authentic contexts, students are equipped with the confidence to successfully solve problems in real situations [5,20,21]. Additionally, tangible robots assist the student in solely concentrating on solving a problem and finding the algorithmic solution of a given exercise instead of dealing with each programming language's features. Thus, students perceive programming as a fun and challenging activity instead of a painful procedure [23]. Because of its simplicity, the robot as a means of teaching enables students of different ages, intellectual backgrounds, or with learning disorders to access learning [24–26].

Another aspect of educational robots that can affect the quality and effectiveness of learning is the game factor. Many researchers have noted different game features such as challenge, fantasy, complexity, rules, strategy, and goals can make a game an engaging educational tool [14,21,27]. By using game activities in the learning process, a student's

motivation and interest are increased as they are actively involved in an entertaining procedure [21]. At the same time, game activities introduce competition and cooperation, factors that encourage learners to immerse themselves in learning. Given a specific challenge, students tend to discover solutions and new strategies in order to increase their performance and win [14]. Furthermore, games are fun, turning programming tasks perceived by learners into a source of enjoyment. Essentially, users' attention is exclusively devoted to the programming and acquisition or improvement of their algorithmic skills [11,14,20]. However, for a game activity to be effective in education, certain features must be attained. The game cycle is the key component, consisting of the loop triplet judgment–behavior–feedback triggered by specific game features such as the degree of guidance and difficulty. Consequently, integrating the playful aspect of robotics in learning must be done with caution [27].

The application of a tangible user interface has been widely studied with different robotics kits in many different group ages [5]. Several tangible programming projects have a seam to influence the development of tangible programming. AlgoBlock was one of the first tangible programming tools developed by Suzuki and Kato in which they introduced interlocking blocks representing the commands of a language similar to Logo [18,20].

Tangible programming Brick developed by McNerney added the use of parameters and variables. More precisely, this tangible programming interface used Lego bricks with embedded electronics to program, through the combination of bricks in the correct order, the graphical user interface of Logo Block and Lego Mindstorms [5,20,22].

Lego Mindstorms is a programming and engineering-oriented system consisting of graphical software and handy hardware. The Lego kit includes a control unit in the form of a brick that controls the system, a set of motors, modular sensors (e.g., touch sensors and ultrasonic sensors), and parts from the technic line such as gears and Lego bricks based on the traditional Lego design for the building part. It also includes a graphical programming language where students can manipulate the available picturized commands and connect them just like puzzle pieces to create a program. Learners can build the instructed models included in the kit by following building instructions or experimenting with customizing their robots with different sensors, motor placement, and gearing [13].

More programming concepts were added with systems such as FlowBlock, that enabled students to see the changes in variables based on the movement of lights on a series of arrows-blocks and comprehend the structure of the sequence, repetition, and branch with the use of probes, by counting the times the light passes [14]. Other studies followed, proposing innovating platforms such as TurTan by Gallardo where instructions appear as figures, the Electronic Blocks by Wyeth, which allowed students to build and program robots and mechanisms with blocks and Quetzal–Tern by Horn, which could identify the connected commands with the use of a scanning system [18,20,28,29].

Acknowledging that tangible interfaces enhance learning experiences and increase students' algorithmic/programming thinking skills, a lot of robotics kits were developed to assist students in learning [5]. The most common educational robotic kits used today for teaching programming to younger students are Lego Education (including Mindstorms and Wedo), Engino Robotics, Bee-Bot, and Arduino-based kits. Most of the proposed systems convert written programming to graphical, and others support palpable code over using a computer keyboard, or mouse [30,31]. Even though this is suitable for young novice programmers, students seek more from a tangible programming interface after learning basic programming or reaching a certain age.

Summary of Contribution

Motivated by the wide adoption of STEM approaches in elementary and secondary education and the limited availability of open low-cost tools, we introduce a new educational framework oriented towards the needs mentioned above. The proposed system is based on a two-fold approach. On the one hand, we have the hardware and software based on the popular Arduino microcontroller, a low-cost device supported by a large community,

and Ardublockly, a visual programming editor for Arduino. On the other hand, the second part refers to the model of the education procedure using Petri Nets, a typical discrete event-based modeling and simulation method. This enables their use in any STEM-related activity concerning the interacting entities and state-changing events. This, combined with the open nature of hardware and software, can accommodate different kinds of STEM activities. To the best of our knowledge, this is the first formal approach, which combines the tools mentioned above towards developing a working STEM system for elementary and secondary education students.

The overall system is designed and developed to provide an expandable, modular system of low complexity suitable for students with no or low prior knowledge in related subjects, among others, to programming, embedded devices, sensors and actuator, robotics, etc. Furthermore, our scope is to provide a system with a small learning curve. Therefore, the students would be able to perform appealing yet straightforward tasks in a short time period, which will boost their self-confidence and creativity. Simultaneously, it would provide a system with several features that may be adopted in different projects. Furthermore, we decided not to constrain our approach with a robotic device as the majority of the methods in the literature but instead focus on an open architecture that will allow the students to compose their working prototypes/systems.

The rest of the paper is organized as follows. In Section 2, we describe our proposed system in full detail, starting from the overall motivation, the hardware which was developed, and the related software which was adopted and enhanced to accommodate our needs. In Section 3, we present the modeling of the educational process, which was performed using the Petri Nets theory, for the monitoring and observation of the educational procedure. The proposed model ensures the consistency of the followed procedure for all groups of students and that the training process is teacher independent. It also provides a valuable tool to the teacher to detect and solve possible problems during the education process. In Section 4, we describe in detail the procedure followed to preliminary validate our approach using flow theory in an elementary school in Greece. Our main goal was to identify the students' acceptance of the proposed system and how the system managed to put the students in a "flow state". To achieve the goal mentioned above, we accompanied the proposed system with a series of sample courses, with an open-ended structure based on the principles of problem-based learning. Finally, in Section 5, we offer some concluding remarks and some thoughts for future research.

2. The Proposed System

2.1. Motivation

The proposed system was designed, with some key characteristics in mind essential for its proper adaptation from the target group, which consists of 9–15-year-old students. The overall approach is presented in Figure 1. Our design philosophy, was based on an effort to increase access to STEM education, a necessity which was identified in [32], by using low cost, user friendly, modular devices. At the same, the educational goals, align with the approach presented in [33] and the learning outcomes as they were identified in educational robotics, which are applicable to other STEM-related activities [23].

During the design and development phase, it was identified that the user-friendliness of the proposed system was of paramount importance. Creating even simple circuits using the conventional approach requires significant effort and understanding from the students and skills that they may lack, especially during the introductory phase with the equipment at hand. The proposed system minimizes the complexity since the student is required to make simple connections.

Crucial factors for the success of the proposed system were its expandability and modularity since these are the key elements that affect the functionality and adaptability in different educational scenarios. Another key issue was the hardware and software complexity of the proposed system since it has been shown that high levels of complexity create frustration and a lack of focus for the students.

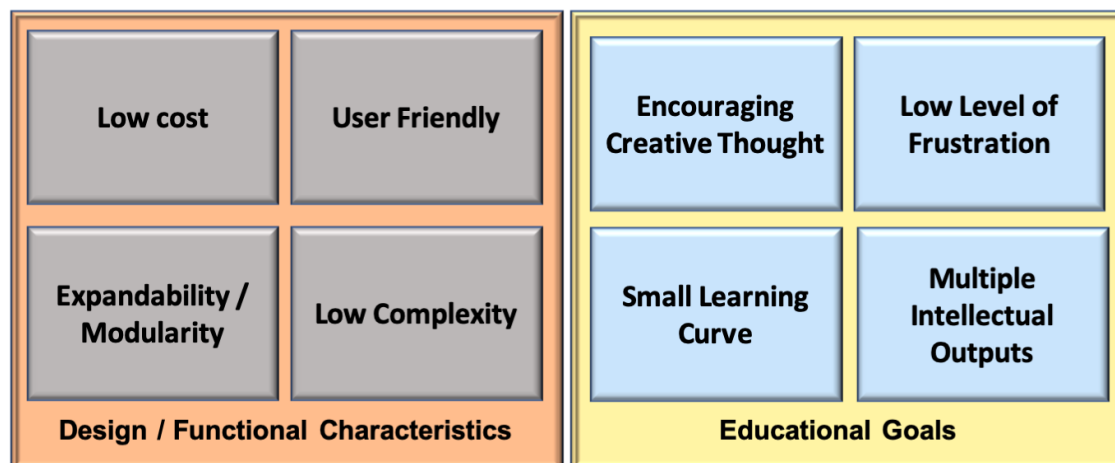


Figure 1. Design philosophy.

The aforementioned design and functional characteristics were considered to meet a diverse set of educational goals. The main one was the encouragement of creative thinking through the simultaneous development of software and the interaction with the environment through different sensors and actuators. In addition, the overall design considered the minimization of the learning curve since the educator has limited time in the classroom to demonstrate the proposed system's functionality and highlight the educational goals of the lecture. By keeping the learning curve small, the student has more time to actually focus on the actual content of the course.

We decided to adopt open tools both for hardware and software during the design process and the maximization of the impact of the proposed system and the minimization of the cost. The core of the hardware system was based on the Arduino family of microcontrollers, which is an extremely popular solution for different kinds of educational projects and the microcontroller of choice is the Arduino Mega 2560. The proposed system's software was based on the popular platform Blockly, which was developed by Google [34], and Ardublockly [35] which is a visual programming editor for Arduino.

2.2. Hardware

Hydra's core module is based on the popular Arduino Mega 2560, which was chosen based on its excellent cost-to-capabilities ratio. It currently consists of eight blocks as presented in Figure 2.

These are divided into four output modules (highlighted with purple): (a) a four LED module; (b) a seven-segment display; (c) a red, green, and blue (RGB) LED; and (d) a DC motor, three input modules (highlighted with cyan): (a) a potentiometer, (b) sonar sensor, and (c) a four-button interaction module and the main control board (highlighted in yellow). All the modules were designed using the Autodesk Eagle PCB.

The Arduino Mega 2560 was adapted to the main control board through proper connectors. All the external modules were connected to the mainboard using RJ45 connectors, which were properly aligned around the mainboard. The usage of the RJ45 connection assures that all the external modules can be connected in a unique way to the mainboard, therefore minimizing the potential errors that the user might make. Overall, the mainboard consists of a USB type-B port (mainly for powering the motors) and nine RJ45 connections grouped in three categories: (a) digital, (b) analog, and (c) PWM, as presented in the schematic in Figure 3.

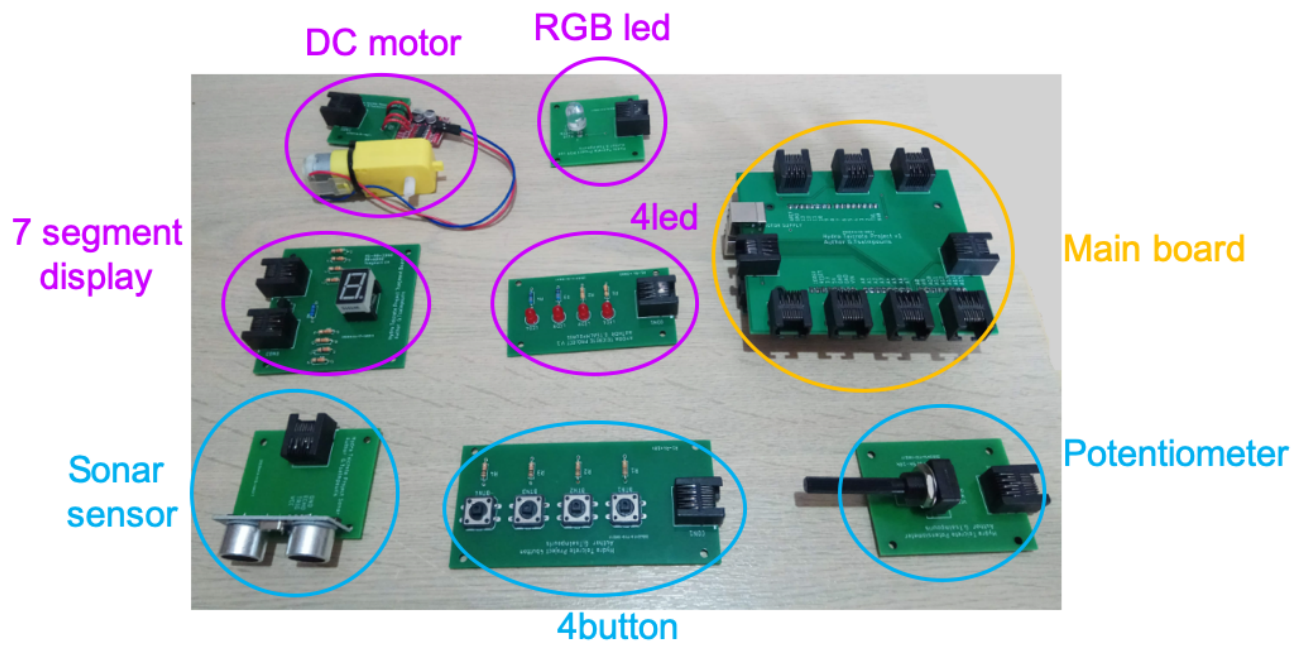


Figure 2. The complete set of Hydra modules.

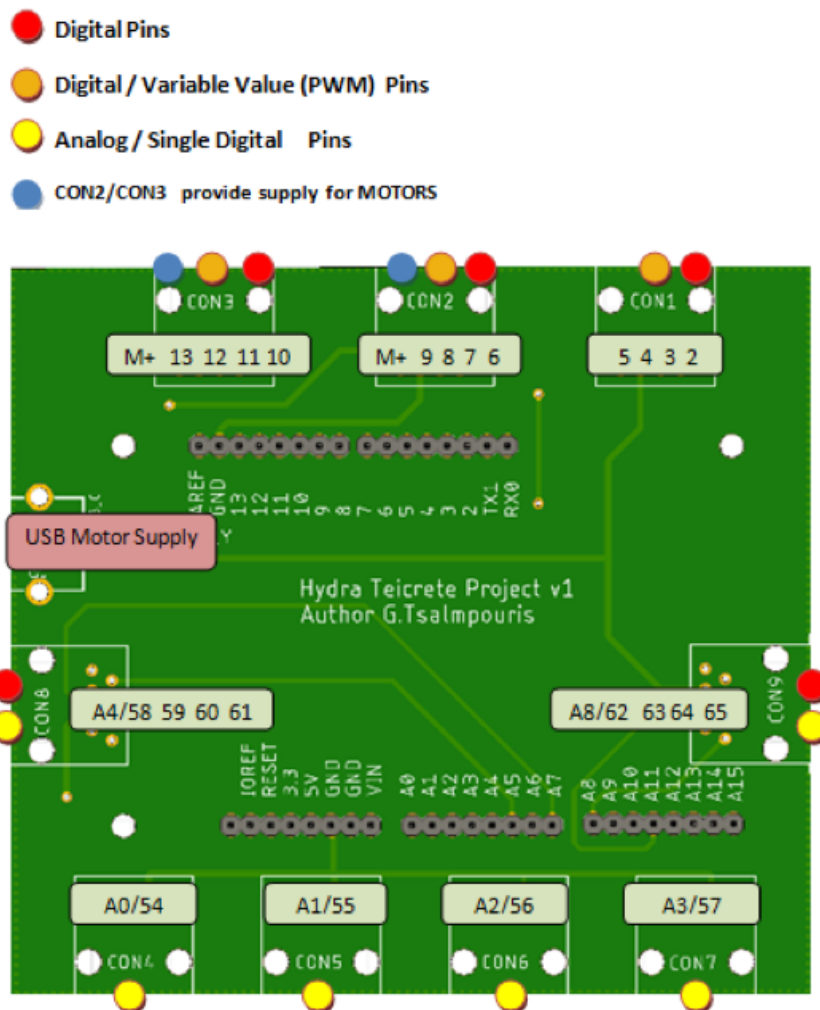
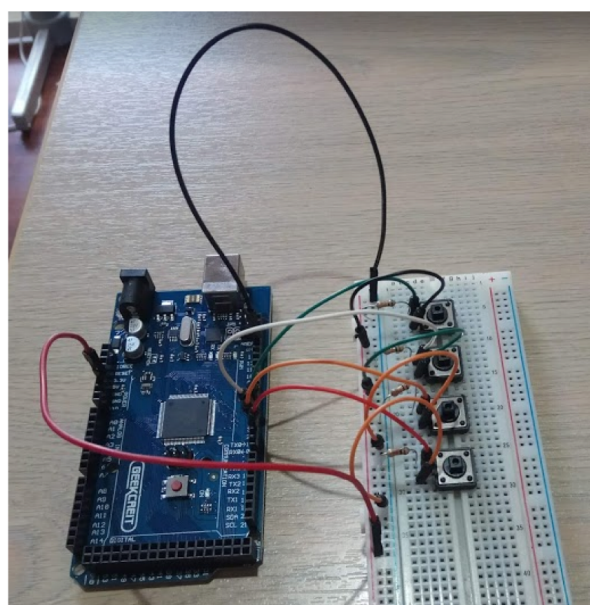


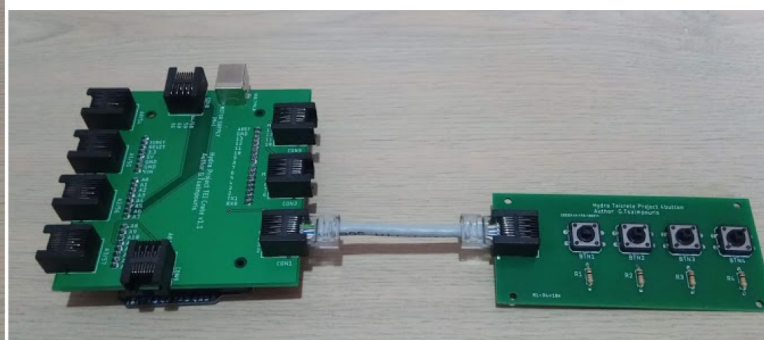
Figure 3. Main board schematic of Hydra.

The digital ports can have two different values: HIGH, which corresponds to 5 V; and LOW, which corresponds to 0 V and can be used as input or output ports, while the analog ports can only be used as input ports for the analog input ranging from 0 to 5 V which corresponds to values ranging from 0 to 1023. Finally, the PWMs can be used as outputs to create rectangular voltage pulses with a variable duty cycle which can control the rotation speed of a motor in the respective module or control the brightness of the LEDs.

The overall approach is modular and allows the students to effortlessly make all the necessary connections without focusing on the time-consuming and sometimes frustrating procedure—especially for non-experienced users—of creating a working circuit. An example of how the proposed system differs from the conventional approach is presented in Figure 4.



(a)



(b)

Figure 4. Comparison between (a) the traditional approach and (b) the Hydra approach.

As we mentioned earlier, our goal was to provide an affordable platform compared to commercial solutions. In Table 1, the detailed bill of materials used per set is presented, and in this cost, we have to add the cost of printing the PCBs in an external vendor. The overall cost is approximately EUR 35 per set without including an external power supply since most of the commercial USB chargers can be used.

Table 1. Bill of materials (per Hydra set).

Type of Electronic Part	Quantity	Cost (EUR)
Arduino Mega 2560	1	8
PCB RJ45 Connectors	17	4
Male 8pin header 2.54 mm	5	0.5
Red led 5 mm	4	0.1
Tact switch 12 × 12 × 7.3 mm	4	0.2
Linear potentiometer 5–10 kΩ	1	1.5

Table 1. Cont.

Type of Electronic Part	Quantity	Cost (EUR)
7-segment display CK	1	0.5
Sonar sensor HC-SR04	1	2
Motor shield MX1508	1	1
Dual axis Geared DC Motor 5 V	1	1.5
RGB led 10 mm CK	1	0.2
Resistors and transistors	21	0.5
USB type b pcb connector 90 degree	1	0.2
USB cable type b	1	2

2.3. Software

The software of the Hydra platform is an enhanced and modified version of Ardublockly [35], which is a visual programming editor for Arduino based on Google's Blockly, which has been properly modified to generate Arduino code. The main features of Ardublockly, therefore, inherited by Hydra, is that it can generate an Arduino code with visual drag-and-drop blocks, which it can load to an Arduino Board, and it produces useful code block warnings whilst being compatible with a wide range of official Arduino boards. Another key advantage is that the students can access to the Arduino IDE. Therefore, the code as used by the micro-controller is accessible.

Hydra builds upon the original version and provides support for all the hardware modules, which were described in detail in Section 2.2. In order to optimally address the needs of the targeted audience, the main menu of the Ardublockly environment was modified in a minimal manner, and an extra tab for the Hydra modules was added. For every single hardware module, there is a corresponding software module which can access every function. Apart from the single software module for the cases in which there are more than one controllable element in a hardware module, simplified versions that can access single elements were also developed, i.e., the LED case where there is a module responsible for controlling the whole hardware module and a software module responsible for controlling a single LED, as presented in Figure 5.

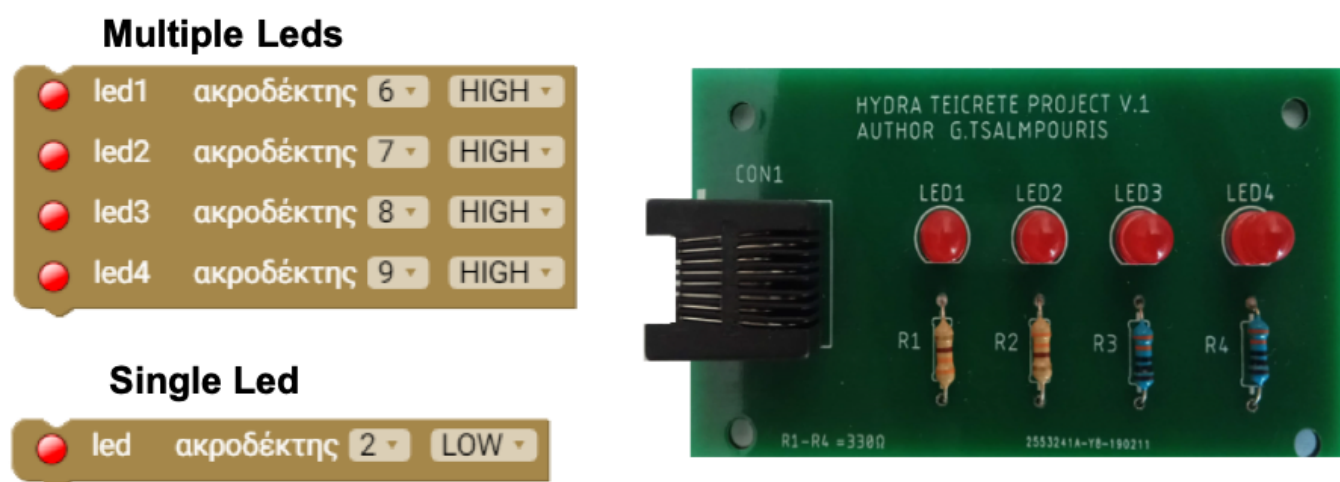


Figure 5. Ardublockly software modules for 4 LED boards.

A sample Ardublockly program for controlling the rotation direction of a DC motor using two buttons is presented in Figure 6. The Hydra DC motor module and the four-button interaction module were used in the configuration presented in Figure 7.



Figure 6. Sample Hydra configuration for DC motor control.

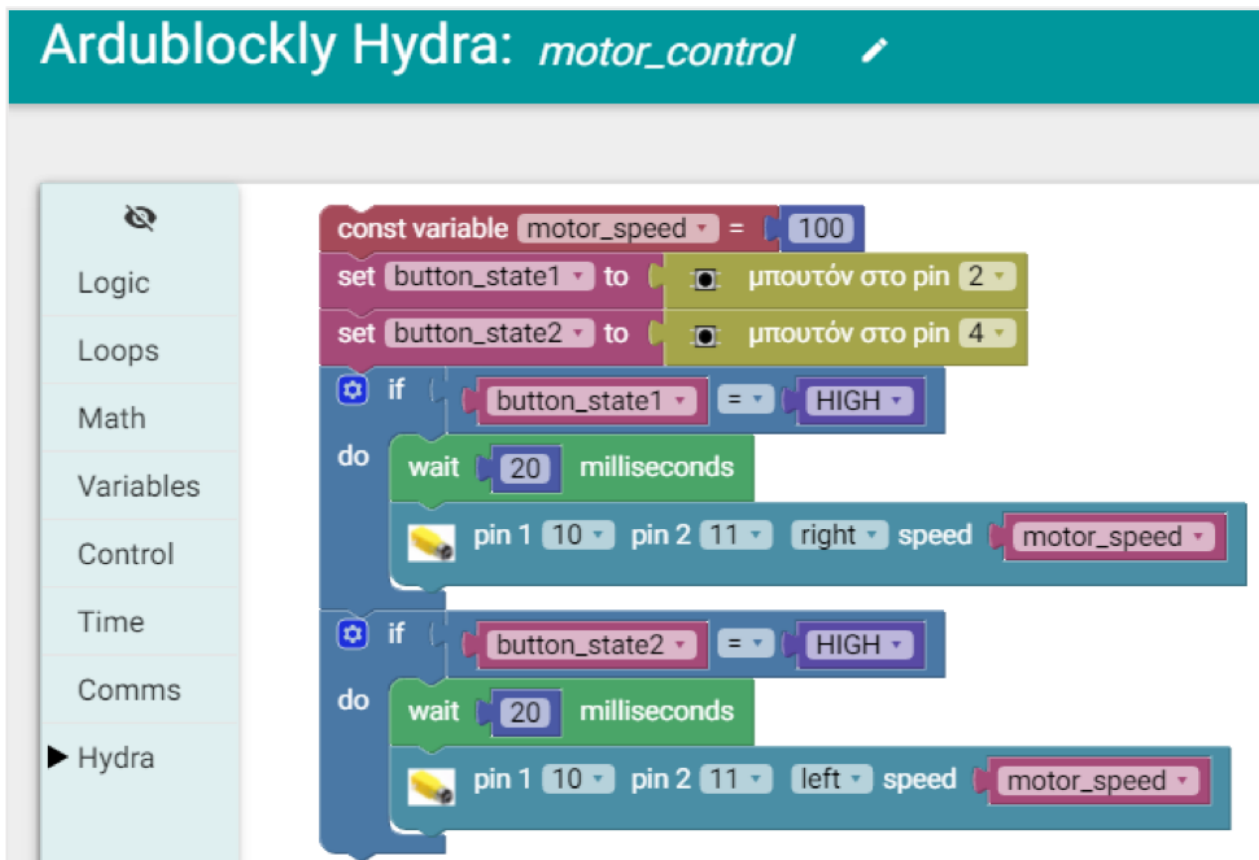


Figure 7. Sample Ardublockly program for controlling Hydra DC motor module.

3. Modeling of the Educational Procedure

Petri Nets (PNs) are a popular mathematical and graphical tool widely used for the modeling, analysis, synthesis, performance evaluation, simulation, and control of processes and systems typically considered as discrete events. They allow the representation and study of the structure as well as of the dynamic behavior of systems and processes and have been proven to be a powerful tool for studying system concurrency, sequential, parallel, asynchronous, distributed deterministic or stochastic behavior, resource allocation, mutual exclusion, and conflicts [36–38]. Popular fields of Petri Nets use include production and manufacturing systems, project management, computer networks, software development and engineering, traffic monitoring and control, power systems, and robotic tasks. However, apart from the typical engineering applications, Petri Nets have been used for studying chemical and biochemical processes, medical and healthcare tasks, and cognitive, educational, and learning procedures.

In [39], the authors used Petri Nets to model the beginning of one lecture and the alternative routes, containing different types of source materials, that a university student could follow to successfully finish the final test of this lecture. In [40], Fuzzy Petri Nets are used to create a concrete model for the adaptation of web-based teaching processes to the individual users of different profiles. In contrast, in [41], the authors introduced a Petri Net-based methodology used to verify the intelligent tutoring system for the English language in Taiwan.

In [42], the authors used Petri Nets to create a model of student behavior in an LMS e-course. In particular, using the available Moodle's log files, the authors were able to see the parts of the course that the students visited, the route that they followed through the different individual parts of the course (Learning part), the parts that were ignored or repeated and the time that they spent in each part. The results from this procedure were used to modify the e-course in order to become more efficient and to compare the actual students' behavior with the respective behavior considered during course design. In [43], the authors introduced a learning evaluation model which applies a high-level fuzzy Petri net (HLFPN) and infers via a fuzzy reasoning method the different answering performances generated by different students' abilities corresponding to the test items with different degrees of difficulty. The results of the test were used to evaluate the overall performance of students not only by considering scores but also by comparing the students' performance. From this study, significant conclusions concerning students' performance as well as indicators for the teachers for the students that need more concern and more efficient guidance were extracted. In [44], the authors introduced a Petri Net-based intelligent tutoring system, used for teaching English courses. The proposed system consists of different parts for teachers and students that can communicate and interact. From all the previous works, it is obvious that Petri nets comprise a valuable tool for a number of applications in education.

In this paper, Petri Nets were used for the monitoring and observation of the educational procedure described. In particular, the implemented model ensures the consistency of the followed procedure for all the groups of students as well as that the training process is teacher independent. Furthermore, monitoring the educational process ensures that the trainer can detect and solve the possible problems such as the timing of the process due to constraints of the overall educational process following the necessary actions.

3.1. Petri Net Fundamentals

Ordinary Petri Nets (OPNs) are bipartite directed graphs formally defined as five-tuple: $PN = \{P, T, I, O, m_0\}$. The respective sets for the two types of nodes are $P = \{p_1, p_2, \dots, p_{np}\}$ which is a finite set of places and $T = \{t_1, t_2, \dots, t_{nt}\}$ which is a finite set of transitions. $P \cup T = V$ where V is the set of vertices and $P \cap T = \emptyset$.

In Petri Nets, places describe conditions (e.g., for control purposes) or resource availability. Transitions represent events or actions and arcs (that may have weight equal or greater than one), direct connections, access rights, or logical connections between places

and transitions. Thus, places are the passive element of the PN, while transitions are the active one. I represents the input function, the output function by O and m_0 is the PN initial token distribution referred to in the literature as marking. Transitions become enabled when all their input places contain several tokens at least equal to the weight of the arc connecting place to transition and fire by removing tokens equal to these weights from all the input places and adding tokens to all the output to the transition places according to the respective arc weights. PN properties (reachability, coverability, safeness, k-boundedness, conflicts, liveness, reversibility, persistency, deadlock-freeness, P- and T-invariants) capture the precedence relations and structural interactions between system components. More analytically, PN theory was described in [36,45].

The inclusion of time delays (constant, following distribution, or random according to the actions) in the transitions of the initial formalism implements T-timed PNs (TPNs). TPNs are defined as $\{P, T, I, O, m_0, D\}$ with the first five variables responding exactly to the same features as in the case of OPNs and D representing time delay that is a function of the set of non-negative real numbers $\{0, \mathbb{R}^+\}$. TPNs have advanced use compared to OPNs as, except for modeling purposes, they can be used for the simulation and calculation of the time duration of sequences of events.

The use of arc extensions increases the modeling power of the initial model as this makes possible the representation of more sophisticated concepts implemented with more compact net structures. Arc extensions were used to activate or deactivate the executions of parts of the PN as long as certain conditions are active. In the literature, three types of arcs are usually used, the standard arcs (\rightarrow), inhibitor arcs that are represented by arcs whose end is marked with a small circle ($-O$), and activator arcs that are drawn as dashed vectors [44].

3.2. Application of Petri Nets for Modeling of the Educational Procedure under Study

The main steps of the followed educational procedure include the introduction of the students to the basic concepts of the Hydra by the teacher, and them interacting with the different hardware modules and the programming environment. The students were then divided into groups, and the teacher assigned different roles, namely “programmer”, “electronic”, and “manager/secretary”. The manager/secretary coordinates the efforts of the team based on the educational material which is available and takes appropriate notes during the experimentation; the electronic interacts with the hardware; and the programmer develops the code modules based on the feedback from the team members and the material that accompanies the different modules. The educational material which is distributed to the students is briefly and concisely written, highlighting the key concepts by simultaneously giving several degrees of freedom to the students to experiment and develop their own working paradigms using the available devices. The goal of each session is that the students will develop small projects of their own using the material at hand. First, the students briefly describe the small projects to the teachers, and afterward, they implement them using the Hydra modules.

Based on the aforementioned concept, we modeled the educational procedure that the student will follow, as presented in Figure 8, using PN models. The overall Petri Net model consists of 15 places and 12 transitions, is conflict-free, and live. Its execution is mainly sequential, with the exception of t_5 that models parallelism and t_6 that models concurrency. The exact meaning of the places in the PN model was presented in Table 2, while the transitions of the model are presented in Table 3.

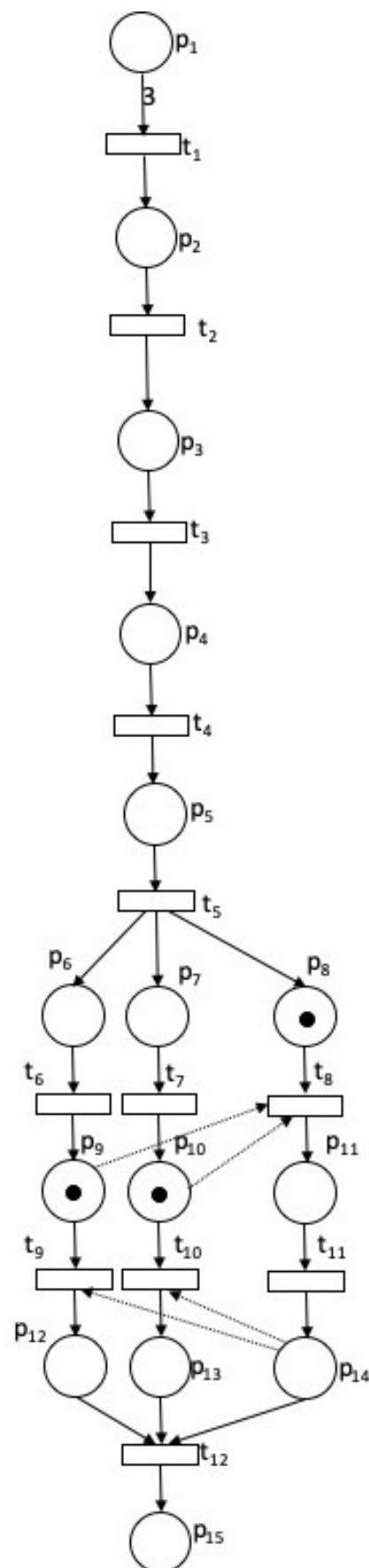


Figure 8. PN model of the educational procedure.

Table 2. Places of the Petri Net model.

Place	Meaning
p_1	Available students
p_2	Team of three students has been created
p_3	Completed predefined laboratory examples
p_4	Team has no more questions
p_5	Open-ended project is defined
p_6	Programmer
p_7	Electronic
p_8	Secretary
p_9	Finished coding
p_{10}	Connections have been implemented
p_{11}	Code and connections are checked
p_{12}	Coding has been approved by secretary
p_{13}	Connections have been approved by secretary
p_{14}	Secretary has finished their tasks
p_{15}	Finished educational procedure

Table 3. Transitions of the Petri Net model.

Transition	Meaning
t_1	Initiation of the educational procedure for the team
t_2	Design of laboratory sheets
t_3	Answer of questions and explanations
t_4	Definition of open ended project
t_5	Role assignment
t_6	Code writing (or coding)
t_7	Implementation of connections
t_8	Secretary checks code and connections
t_9	Coding has been terminated
t_{10}	Connections have been finalized
t_{11}	Approval of code and connections
t_{12}	Measurement recording and operation validation

In the Petri Net model of Figure 8, all the connection arcs have unitary weight apart from the initial ones (from p_1 to t_1). This happens because the educational procedure cannot start if three students are unavailable, as each of these will take a specific role in the following steps. In addition, after the firing of t_5 (role assignment), each of the three students has to do their own tasks which are independent, especially for electronic and programmer, while the secretary interacts with both in a manner. For the electronic and programmer, these tasks are independent in their first stage while the secretary interacts with both in the second stage, giving their final approval. For this reason, in this part of the PN, test arcs were added. In particular, the two test arcs from p_9 and p_{10} to t_8 show that when the programmer and electronic finish their tasks, the secretary will check them and two test arcs from p_{14} to t_9 and

t_{10} to show that when the secretary approves code and connection, the respective tasks can be completed. However, the whole process cannot be terminated unless all three students have finished their tasks. In addition, tokens were added in places p_8 , p_9 and p_{10} to show a state of the modeled system. In this state, the programmer finished coding, the electronic finished the connections, and the secretary will start the checking and approval procedure. Since all preconditions are satisfied, t_8 will launch the next step of PN execution.

The presented Petri net model can be used in its present form for the educational process monitoring and observation from the trainer. The simplicity of representation makes it easy to understand even for trainers with reduced technical skills. In addition, it can be used to ensure that different trainers follow exactly the same steps defined during educational process design. Finally, the implemented model can be used for a what-if scenario simulation to distribute the available time between the different stages in the most efficient way with respect to the performance indicators used for assessment.

The advanced capabilities arising from the extensive use of sensors nowadays, as well as the implementation of technologies closely related to the Internet of Things and Industry 4.0, such as digital twins, cyber-physical systems, machine learning, and analytics, increase the significance, the fields of possible application as well as the efficiency of the implemented models. Furthermore, the exploitation of these advances and tools will improve the real-time monitoring, execution, evaluation, design, and update of the educational procedure concerning the feedback received from the students. Thus, the final target is the optimization of the learning procedure and the personal development of the students through an always up-to-date, interactive, and non-static educational procedure.

4. Proof of Concept and Experimental Evaluation

Informatics, robotics, automation, and other STEM-related courses are relatively new in the curriculum of secondary education in Greece and are only integrated at an elementary level in primary education. A detailed study about the courses currently offered and their content can be found in [46]. The tools that are used vary from Lego devices to popular Arduino kits. We decided to perform a preliminary pilot study in primary education to identify the feasibility of our proposed approach. The assessment was based on the basic principles of flow theory that will be described in the following subsection. The main goal was to identify the students' acceptance of the proposed system and how this system managed to put the students in a "flow state". To proceed, we developed a series of sample courses with an open-ended structure based on the principles of problem-based learning [47].

4.1. Flow Theory

Flow experience (flow) is defined as the state in which an individual feels completely absorbed and fully engaged in an activity [48]. Flow is a concept initially introduced by Csikszentmihalyi [49] in their book 'Beyond Boredom and Anxiety'. Flow experience is an extremely rewarding experience, balancing challenge and skill. Flow is often associated with high levels of performance and is a positive psychological experience [50]. It allows the individual to perform at an optimum level. Flow has been used and researched in various fields of everyday life, including but not limited to sports, work-related environments, creative arts, media, and related educational activities where the high state of performance could be beneficial. Flow occurs when one is totally involved in the task at hand. It can occur at different levels of complexity, however, by definition, flow is intrinsically rewarding, regardless of whether it involves a simple game of throw and catch or a complicated and dangerous gymnastics routine [50]. When in flow, one feels strong and positive, not worried about themselves or of failure. To be found in such a psychological state, they must meet two factors which play the most important role: (a) their perception of the difficulty of the challenges they have to face; and (b) their perception of their own skill and ability to deal with this challenge [51].

There are nine key factors that contribute to the appearance of flow which are described in detail in [50,52]. Namely, these nine factors are: (1) challenge–skill balance; (2) action–

awareness merging; (3) clear goals; (4) unambiguous feedback; (5) total concentration on the task at hand; (6) sense of control; (7) loss of self-consciousness; (8) transformation of time; and (9) autotelic experience. All the factors mentioned above contribute to the optimal psychological state of flow.

4.2. Pilot Study

For the pilot study of the proposed approach, we distributed the system to 68 students with their ages ranging from 9 to 12 years old (21 students of the 4th, 26 students of the 5th and 21 students of the 6th grade of the Greek Elementary school). The participants were a mixed group consisting of 35 boys (51.5%) and 33 girls (48.5%). All the students from the three classes participated in the pilot study without considering any prior experience in similar tools or programming. The experiments were conducted inside their school premises, and more specifically, in the school's computer room. Each session with the proposed system lasted approximately 45 min, and at each session, nine students, using one full system per three persons, participated.

The students during their experimentation were monitored by a teacher who was familiar with the system and the related processes. They were initially introduced to the basic concepts of Hydra by the teacher, and afterward, they followed the procedure as described in Section 3.2. The students were only allowed to interact with their team members, while the teacher had minimum intervention, encouraging the students to provide their solutions to any problems that they may have.

Upon the completion of the process, as described in Section 3.2, they answered an anonymous questionnaire consisting of a set of closed-type questions, divided into two main categories. The first category had seven questions concerning the system and its functionality, and the second category had ten questions related to the key factors of flow theory. The structure of the questionnaire and the related questions/statements regarding flow experience were similar to the ones presented in [53]. All the answers were based on a seven values scale where one corresponds to easy/low while seven corresponds to high/hard.

As far as the first category of questions is concerned, the key findings show that 92.6% of students found the overall procedure easy and only 7.4% found it to be of medium difficulty (Figure 9).

None of the students expressed the opinion that the proposed system was difficult to use. The students were self-confident with regard to their skills in the related activity since 89.7% declared that they were highly confident in their skills while only 10.3% perceived that they had medium skills. On the other hand, 69.1% answered that they did not have any previous experience related to programming or automation-related projects, and only 30.9% declared that they had some previous experience with the activities mentioned above. It is worth mentioning that 69.1% stated that they did not have any previous experience with similar systems, highlighting how easy it is to adopt the proposed approach if we consider the answers to the previous question too. Furthermore, the students' opinion was in favor of "Hydra" per se, since they found that: (a) the course demands were medium (79.4%) or low (20.6%); (b) that the software was easy to use (70.6%); and (c) that the hardware was very handy (88.2%).

Flow experience was measured using the flow short scale which measures all components of flow experience with ten items and was used to measure the flow during all activities [53]. The statistical analysis was performed using the SPSS statistical package. The internal consistency for the flow score was $\alpha = 0.92$.

The mean of the answers in the ten questions was combined in a new variable named FLOW. The flow experience reported by the students which participated in the aforementioned study ($t = 19.24$) statistically differed from the average of the scale ($MD = 2.32$, $df = 67$, $p < 0.001$). The high average of the flow experience indicates that the proposed system is appealing in all cases of students independently from factors including sex and age (Figure 10).

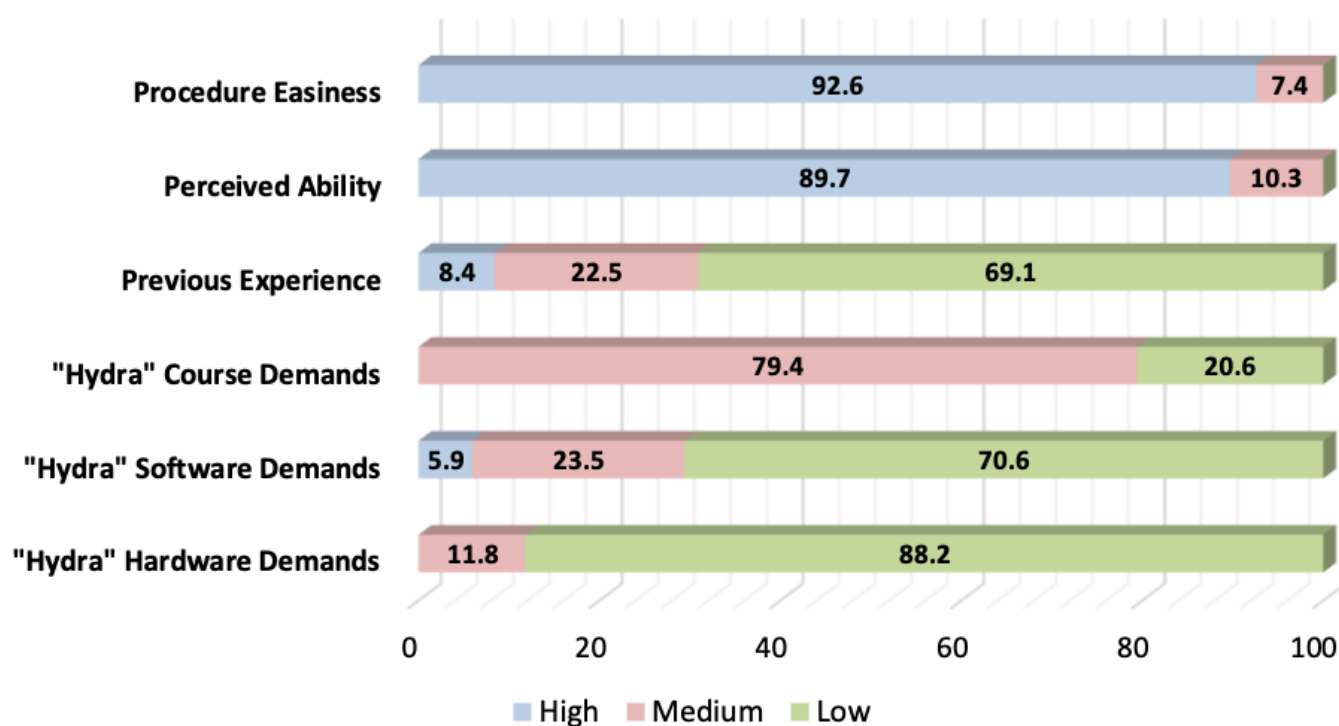


Figure 9. Evaluation findings based on the first set of questions.

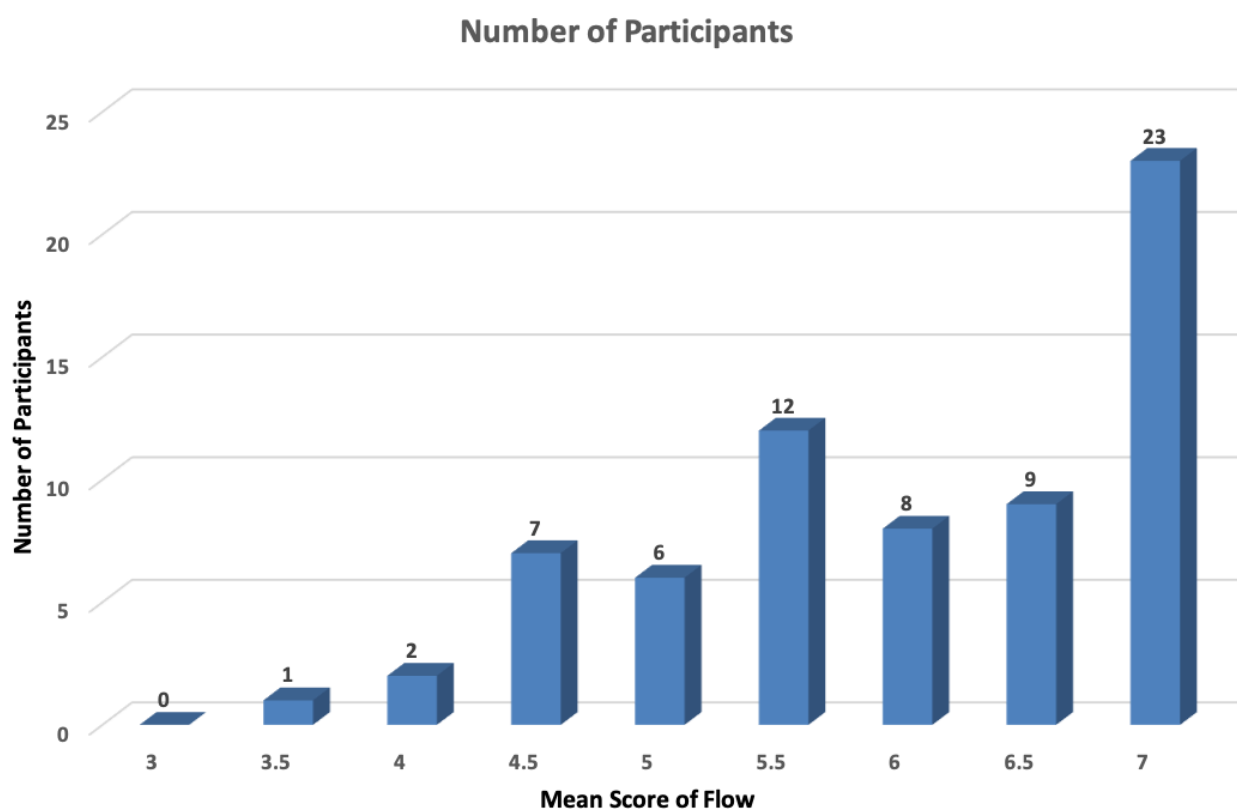


Figure 10. Flow experience distribution.

5. Conclusions

In the proposed manuscript, we present a new low-cost framework for STEM education based on open tools. We present in detail the hardware and software developed in addition to the modeling of the overall educational process using Petri nets. As a proof

of concept, we present a preliminary evaluation based on the testing of the proposed system by 68 students of three different grades of a Greek elementary school, using flow theory. The overall adoption rate of the proposed system was high among the participants, highlighting the approach's potential. Although our experiments were conducted using a small sample of students, the reported results are promising. Further research is required to detect the significant effects, confirm our intuitions, and further generalize the findings. In future work, the proposed framework will be tested on a larger sample of participants.

We are currently working on the second prototype of the HYDRA system, which will include a new design aiming to increase the robustness of the system and increase its capabilities by adding new modules. Furthermore, we are developing new structural components which will enclose the electronic components, to allow the users to create complex working prototypes. We also aim to develop an augmented reality application, which will work in tandem with the proposed system. This will significantly enhance the education process by providing an appealing visual tool that will guide the education process using the concept described in detail in Section 3.2. At the same time, it will provide the students with a visual guide that will be accessible using an intuitive interface.

Our vision is to offer the solutions mentioned above through a web-based system, accompanied by the related educational material and detailed electronic and 3D drawings. Thus, the users—both students and teachers—will have the opportunity to interact and simultaneously participate by uploading their material in the aim of a sustainable, low-cost approach to education.

Author Contributions: Conceptualization, G.T. (Georgios Tsalmopoulos) and L.D.; investigation, G.T. (Georgios Tsalmopoulos), S.A.C. and L.D.; methodology, G.T. (Georgios Tsalmopoulos), G.T. (Georgios Tsinarakis), N.G. and L.D.; software, G.T. (Georgios Tsalmopoulos); supervision, L.D.; validation, G.T. (Georgios Tsalmopoulos) and N.G.; writing—original draft, L.D.; writing—review and editing, G.T. (Georgios Tsinarakis), N.G., S.A.C. and L.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Publicly not available nor exploitable by other researchers so as to be shared.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Hasani, A.L.; Juansah, D.; El Islami, R. Conceptual Frameworks on How to Teach STEM Concepts in Bahasa Indonesia Subject as Integrated Learning in Grades 1–3 at Elementary School in the Curriculum 2013 to Contribute to Sustainability Education. *Sustainability* **2021**, *13*, 173. [\[CrossRef\]](#)
- Tawbush, R.L.; Stanley, S.D.; Campbell, T.G.; Webb, M.A. International comparison of K-12 STEM teaching practices. *Int. J. STEM Educ.* **2020**, *13*, 115–128. [\[CrossRef\]](#)
- Benitti, F.B.V. Exploring the educational potential of robotics in schools: A systematic review. *Comput. Educ.* **2012**, *58*, 978–988. [\[CrossRef\]](#)
- De Vries, M.J. *Handbook of Technology Education*; Springer: Berlin/Heidelberg, Germany, 2018.
- Kwon, D.Y.; Kim, H.S.; Shim, J.K.; Lee, W.G. Algorithmic bricks: A tangible robot programming tool for elementary school students. *IEEE Trans. Educ.* **2012**, *55*, 474–479. [\[CrossRef\]](#)
- Han, Z.H.; Jiang, Y.; Zhan, Y. STEM Education in Early Childhood: A Review of Empirical Studies. *Int. J. STEM Educ.* **2020**, *32*, 940–962.
- Li, Y.; Wang, K.; Xiao, Y.; Froyd, E.J. Research and trends in STEM education: A systematic review of journal publications. *Int. J. STEM Educ.* **2020**, *7*. [\[CrossRef\]](#)
- Alimisis, D. Educational robotics: Open questions and new challenges. *Themes Sci. Technol. Educ.* **2013**, *6*, 63–71.
- Eguchi, A.; Shen, J. Student learning experience through CoSpace educational robotics. In *Society for Information Technology & Teacher Education International Conference*; Association for the Advancement of Computing in Education (AACE): Waynesville, NC, USA, 2012; pp. 19–24.
- Papert, S.; Harel, I. *Constructionism*; Ablex Publishing Corporation: New York, NY, USA, 1991.
- Atmatzidou, S.; Demetriadis, S.; Nika, P. How Does the Degree of Guidance Support Students' Metacognitive and Problem Solving Skills in Educational Robotics? *J. Sci. Educ. Technol.* **2018**, *27*, 70–85. [\[CrossRef\]](#)
- Papert, S. *Mindstorms: Children, Computers, and Powerful Ideas*; Basic Books, Inc.: New York, NY, USA, 1980.

13. Miglino, O.; Lund, H.H.; Cardaci, M. Robotics as an educational tool. *J. Interact. Learn. Res.* **1999**, *10*, 25–47.
14. Shim, J.; Kwon, D.; Lee, W. The effects of a robot game environment on computer programming education for elementary school students. *IEEE Trans. Educ.* **2016**, *60*, 164–172. [\[CrossRef\]](#)
15. Chin, K.Y.; Hong, Z.W.; Chen, Y.L. Impact of using an educational robot-based learning system on students' motivation in elementary education. *IEEE Trans. Learn. Technol.* **2014**, *7*, 333–345. [\[CrossRef\]](#)
16. Tuomi, P.; Multisilta, J.; Saarikoski, P.; Suominen, J. Coding skills as a success factor for a society. *Educ. Inf. Technol.* **2018**, *23*, 419–434. [\[CrossRef\]](#)
17. Yoel, S.R.; Dori, Y.J. FIRST High-School Students and FIRST Graduates: STEM Exposure and Career Choices. *IEEE Trans. Educ.* **2021**, 1–10. [\[CrossRef\]](#)
18. Caceres, P.C.; Venero, R.P.; Cordova, F.C. Tangible programming mechatronic interface for basic induction in programming. In Proceedings of the 2018 IEEE Global Engineering Education Conference (EDUCON), Santa Cruz de Tenerife, Spain, 17–20 April 2018; pp. 183–190.
19. Ali, A.; Mensch, S. Issues and challenges for selecting a programming language in a technology update course. In Proceedings of the Information Systems Education Conference, Phoenix, AZ, USA, 2008. Available online: https://www.researchgate.net/publication/228651088_Issues_and_challenges_for_selecting_a_programming_language_in_a_technology_update_course (accessed on 25 September 2021).
20. Sapounidis, T.; Demetriadis, S.; Stamelos, I. Evaluating children performance with graphical and tangible robot programming tools. *Pers. Ubiquitous Comput.* **2015**, *19*, 225–237. [\[CrossRef\]](#)
21. Bers, M.U.; Flannery, L.; Kazakoff, E.R.; Sullivan, A. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Comput. Educ.* **2014**, *72*, 145–157. [\[CrossRef\]](#)
22. McGill, M.M. Learning to program with personal robots: Influences on student motivation. *ACM Trans. Comput. Educ. (TOCE)* **2012**, *12*, 4. [\[CrossRef\]](#)
23. Evripidou, S.; Georgiou, K.; Doitsidis, L.; Amanatiadis, A.A.; Zinonos, Z.; Chatzichristofis, S.A. Educational Robotics: Platforms, Competitions and Expected Learning Outcomes. *IEEE Access* **2020**, *8*, 219534–219562. [\[CrossRef\]](#)
24. Kaburlasos, V.G.; Dardani, C.; Dimitrova, M.; Amanatiadis, A. Multi-robot engagement in special education: A preliminary study in autism. In Proceedings of the 2018 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 12–14 January 2018; pp. 1–2.
25. Amanatiadis, A.; Kaburlasos, V.G.; Dardani, C.; Chatzichristofis, S.A. Interactive social robots in special education. In Proceedings of the 2017 IEEE 7th International Conference on Consumer Electronics-Berlin (ICCE-Berlin), Berlin, Germany, 3–6 September 2017; pp. 126–129.
26. Amanatiadis, A.; Kaburlasos, V.G.; Dardani, C.; Chatzichristofis, S.A.; Mitropoulos, A.C. Social Robots in Special Education: Creating Dynamic Interactions for Optimal Experience. *IEEE Consum. Electron. Mag.* **2020**, *9*, 39–45. [\[CrossRef\]](#)
27. Garris, R.; Ahlers, R.; Driskell, J.E. Games, motivation, and learning: A research and practice model. *Simul. Gaming* **2002**, *33*, 441–467. [\[CrossRef\]](#)
28. Gallardo, D.; Julia, C.F.; Jorda, S. TurTan: A tangible programming language for creative exploration. In Proceedings of the 2008 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems, Amsterdam, The Netherlands, 1–3 October 2008; pp. 89–92.
29. Horn, M.S.; Jacob, R.J. Tangible programming in the classroom with tern. In *CHI'07 Extended Abstracts on Human Factors in Computing Systems*; ACM: New York, NY, USA, 2007; pp. 1965–1970.
30. Mondada, F.; Bonani, M.; Riedo, F.; Briod, M.; Pereyre, L.; Rétornaz, P.; Magnenat, S. Bringing robotics to formal education: The thymio open-source hardware robot. *IEEE Robot. Autom. Mag.* **2017**, *24*, 77–85. [\[CrossRef\]](#)
31. De Cristoforis, P.; Pedre, S.; Nitsche, M.; Fischer, T.; Pessacq, F.; Di Pietro, C. A behavior-based approach for educational robotics activities. *IEEE Trans. Educ.* **2012**, *56*, 61–66. [\[CrossRef\]](#)
32. English, L.D. Advancing Elementary and Middle School STEM Education. *Int. J. Sci. Math. Educ.* **2017**, *15*, 5–24. [\[CrossRef\]](#)
33. Fallon, G.; Hatzigianni, M.; Bower, M.; Forbes, A.; Stevenson, M. Understanding K-12 STEM Education: A Framework for Developing STEM Literacy. *J. Sci. Educ. Technol.* **2020**, *29*, 369–385. [\[CrossRef\]](#)
34. Fraser, N. Ten things we have learned from Blockly. In Proceedings of the 2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond), Atlanta, GA, USA, 22 October 2015; pp. 49–50. [\[CrossRef\]](#)
35. Ardublockly's Web Page. Available online: <https://ardublockly.embeddedlog.com> (accessed on 29 January 2019).
36. Murata, T. Petri nets: Properties, analysis and applications. *Proc. IEEE* **2014**, *77*, 541–580. [\[CrossRef\]](#)
37. Desrochers, A.; Al-Jaar, R. *Applications of Petri Nets in Manufacturing Systems-Modeling, Control and Performance Analysis*; IEEE Press: New York, NY, USA, 1995.
38. Proth, J.; Sauer, N. Scheduling of piecewise constant product flows: A Petri net approach. *Eur. J. Oper. Res.* **1998**, *106*, 45–56. [\[CrossRef\]](#)
39. Kuchárik, M.; Balogh, Z. Student Learning Simulation Process with Petri Nets. In *Recent Developments in Intelligent Computing, Communication and Devices*; Patnaik, S., Jain, V., Eds.; Advances in Intelligent Systems and Computing; Springer: Singapore, 2018; Volume 752, pp. 1115–1124.
40. Balogh, Z.; Turóáni, M. Possibilities of Modelling Web-Based Education Using IF-THEN Rules and Fuzzy Petri Nets in LMS. In *Informatics Engineering and Information Science: International Conference*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 93–106.

41. Wang, Y.Y.; Lai, A.F.; Shen, R.K.; Yang, V.; Chu, Y.H. Modeling and verification of an intelligent tutoring system based on Petri net theory. *Math. Biosci. Eng.* **2019**, *16*, 4947–4975. [CrossRef] [PubMed]
42. Balogh, Z.; Drozda, M.; Kuchárik, M. Petri net model of student choices in a LMS moddle e-course. In Proceedings of the 12th International Scientific Conference on Distance Learning in Applied Informatics, Sturovo, Slovakia, March 2019. Available online: https://www.researchgate.net/publication/331546741_PETRI_NET_MODEL_OF_STUDENT_CHOICES_IN_A_LMS_MOODLE_E-COURSE (accessed on 25 September 2021).
43. Shen, V.R.L.; Yang, C.Y.; Wang, Y.Y.; Lin, Y.H. Application of high-level fuzzy Petri nets to educational grading system. *Expert Syst. Appl.* **2012**, *39*, 12935–12946. [CrossRef]
44. Wang, J. *Timed Petri Nets: Theory and Application*; Kluwer: New York, NY, USA, 1998.
45. Zurawski, R.; Zhou, M.C. Petri nets and Industrial Applications: A Tutorial. *IEEE Trans. Ind. Electron.* **1994**, *41*, 567–583. [CrossRef]
46. Tsalmpouris, G. Design and Development of a STEM Platform for the Needs of Secondary Education. Master's Thesis, Hellenic Meditteranean Univeristy, Heraklion, Greece, 2019.
47. Hmelo-Silver, C.E. Problem-Based Learning: What and How Do Students Learn? *Educ. Psychol. Rev.* **2004**, *16*, 235–266. [CrossRef]
48. Vasiliou, C.; Ioannou, A.; Zaphiris, P. Measuring Students' Flow Experience in a Multimodal Learning Environment: A Case Study. In *Lecture Notes in Computer Science*; Zaphiris, P., Ioannou, A., Eds.; Springer International Publishing: Cham, Switzerland, 2014; Volume 8523.
49. Csikszentmihalyi, M. *Beyond Boredom and Anxiety: Experiencing Flow in Work and Play*; Jossey-Bass: San Francisco, CA, USA, 1975.
50. Jackson, S.A.; Eklund, R.C. *The Flow Scales Manual*; Fitness Information Technology: Morgantown, WV, USA, 2004.
51. Giasiranis, S.; Sofos, L. Flow Experience and Educational Effectiveness of Teaching Informatics using AR. *J. Educ. Technol. Soc.* **2017**, *24*, 78–88.
52. Jackson, S.A.; Csikszentmihalyi, M. *Flow in Sports*; Human Kinetics: Champaign, IL, USA, 1999.
53. Engeser, S.; Rheinberg, E. Flow, performance and moderators of challenge-skill balance. *Motiv. Emot.* **2008**, *32*, 158–172. [CrossRef]