# An Iterative Learning Approach for Novice Engineering Students to Grasp Important Concepts in Autonomous Systems

Peeranut Pongpakatien
*Department of Computer Engineering*
*Chiang Mai University*
Chiang Mai, Thailand
peeranut_p@cmu.ac.th

Arnan Sipitakiat
*Department of Computer Engineering*
*Chiang Mai University*
Chiang Mai, Thailand
arnan.s@cmu.ac.th

***Abstract*—This work describes the design of an iterative learning approach to teach basic concepts in autonomous and control systems to novice engineering students. The Learning Continuum approach builds on three theoretical principles: Zone of Proximal Development, Selective Exposure, and Concrete before Abstract. The developed curriculum unit splits the lesson into two iterations. The first focuses more on the high-level concepts using tools that avoid low-level complexities of typical microcontrollers. The second iteration then uses a more traditional microcontroller setup to explore the learned ideas. We tested this framework with 217 novice engineering students against a control group consisting of 111 students. Results from a range of data collected show that this approach helps students during the learning process. Instructors in the study have adopted our learning continuum for regular use in their course.**

***Keywords*—*education, microcontrollers, autonomous systems, iterative learning***

## I. Introduction

Embedded Systems and microcontrollers are important tools in learning about autonomous and basic control systems. In recent years, DIY tools led by the Arduino Platform [9] became ubiquitous in technical classes and student projects [7][8]. Arduino offer extensive resources with references and extensions covering most technical requirements of engineering courses, all while being relatively affordable. However, this work argues that popularity and ubiquity do not automatically make the tool a universally good learning platform. This is especially true when the learning objectives are not only about electronics but also about higher level concepts.

This work highlights a large freshmen course that teaches embedded systems and basic automatic control concepts such sensor sensitivity, noise rejection, and hysteresis. While a learning goal is to teach embedded systems, novice learners often find that the technical details of putting together a circuit and coding it obstruct learning about the higher level concepts [1] and affect learner's attitude towards the subject [10]. This research offers a novel learning model that facilitates learning of important control ideas while still being able to offer fluency with the microcontroller platform.

The designed curricula unit uses a multi-iteration approach, where students perform the same activities twice, each with a different learning objective. The first iteration focuses on the high-level automation concept while the second drills down to microcontroller usage. To make this approach viable given the course's time constraint, each learning iteration must only take only half as long as the usual unit duration. Conducting multiple runs of the same activity can also be perceived as more work to instructors. Therefore, the benefits must be clear to get the necessary buy-in. These are challenges that this paper has overcome as elaborated in later sections.

## II. Design Principles

### A. Zone of Proximal Development

Developed by the Russian psychologist Lev Vygotsky [3], Zone of Proximal Development (ZPD) describes tasks that are within the learning ability of a student. If the task is within the "learnable zone", the learner can accomplish the lesson with guidance. However, if the task is too complex, putting it outside the learnable zone, the student will struggle even with facilitation. We use ZPD is an important guideline in designing activities with varying complexity in each learning iteration geared towards the readiness of learners.

### B. Selective Exposure

Selective Exposure [5] is a design framework that emphasizes the importance of deciding what to expose and what to hide when creating a tool. This choice depends on the objective of the tool. Blikstein illustrates this principle by comparing the LEGO robotics kit "Mindstorms" with microcontrollers. Although both Mindstorms and microcontrollers can connect to sensors and motors, the former offers a "port level" approach where sensors and motor modules are plug-and-play while microcontrollers require some knowledge about supply voltage, ground, pull-up resistors, motor drivers, etc. The two tools are suitable for different learning goals. Mindstorm's high level abstraction is suitable for activities that focuses on inspiring new learners and teaching high-level concepts while microcontrollers are suitable for training a more experienced group.

Using a tool that abstracts too much or too little has a significant impact on the learnability of that tool. On one hand, using a microcontroller with novice learners could strain them by requiring an understanding about low-level concepts both in hardware and software. On the other hand, using Mindstorms with experienced electrical engineering students may underwhelm them by offering too little access to the needed functionalities.

In this project, we follow this guideline by using a LEGO-like tool for our first learning iteration to focus more on the control engineering concepts and less on the lower-level circuits. Then in the second iteration, we switch to microcontrollers to show learners more about circuits and microcontroller programming.

### C. *Concrete before Abstract*

Learning about autonomous and control systems involves abstract concepts that are not straight forward to comprehend. Wilensky [4] describes an educator's common learning assumption that learners must understand the abstract concept before they can comprehend the concrete uses in the real world. He argues that although abstract concepts are essential, better learning of those concepts often takes place through concrete experiences.

We use this concept to design labs containing activities that show learners situations where the concepts being taught are useful. For example, we create a situation where a light bulb would flicker and show how hysteresis can be useful in fixing the problem. This concrete before abstract approach combined with selective exposure allows for learning though hands-on activities, teamwork, and problem solving [6].

## III. Designing the Learning Continuum Curriculum

The learning continuum curriculum has two major learning objectives. First, it teaches basic concepts in autonomous and control systems. Second, the curriculum offers basic fluency with microcontrollers (in this case the Arduino platform).

### A. *Course Content: Basic Concepts in Autonomous and Control Systems*

The following list describes the high-level concepts included in this curriculum.

1) *Sensor Sensitivity:* Students should be able to define the threshold of a sensor, through code, to yeild the desired behavior. For example, how much motion from the sensor before triggering an alarm or how much brightness before turning off the room light. The system's sensitivity depends on what threshold is used.

2) *Hysteresis:* Students should be able to prevent rapid activation and deactivation of actuators when the sensor value fluctuates near the threshold. Students should be able to demonstrat the ability to set separate activation and deactivation thresholds.

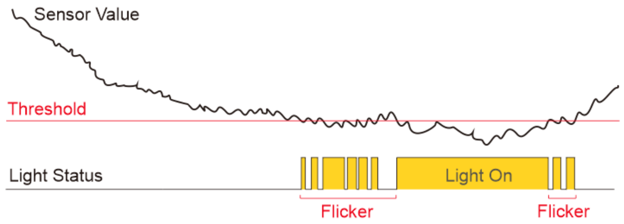


Fig. 1. The sensor fluctuation impacts the actuator activation that cause light flickering.

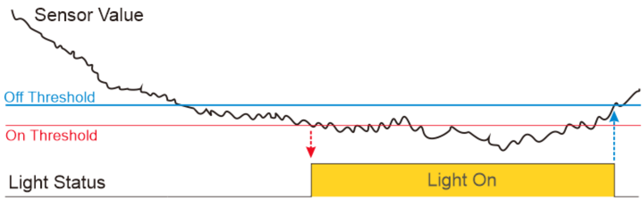


Fig. 2. Separated activation threshold can prevent the light flickering.

3) *State and State-Change:* Students should be able to detect the state (e.g. whether a button is being pressed or not) as well as detect state changes (e.g. detect a button press or release event).

### B. *The Learning Activities*

Following the concrete-before-abstract framework, we designed hands-on situations that would demonstrate the need and usefulness of the concepts being taught. This design resulted in two learning activities.

1) *Activity 1 Automatic Night Light*: This activity is used to teach about sensor sensitivity and hysteresis. Students are asked to create a lamp that turns on automatically when dark and turns off otherwise. This simple setup uses a light-dependant resistor to measure brightness and a relay to control a lamp. We created a contraption with a motorized lid that controlled the amount of light being exposed to the light sensor. This contraption can simulate the gradual change from daytime to nighttime and vice versa. Students are asked to alter the light threshold to incease or decrease the light sensitivity of the lamp. There are also challenges like "Find a threshold value that will never turn on the lamp" to stimulate students as well.

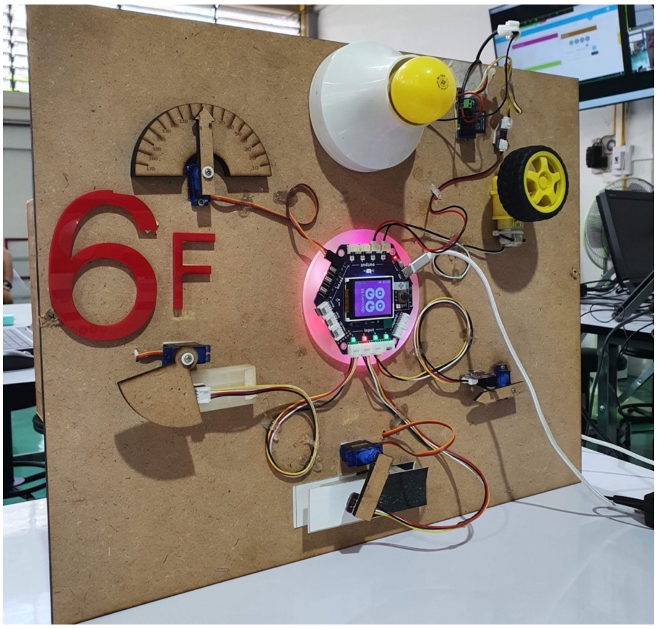


Fig. 3. An experimental playground woodboard for students to learn autonomous and control systems.

We also use this setup to demonstrate hysteresis. Given that sensor values naturally fluctuate, students will see the lamp flicker when the sensor value is near the threshold. "Why does the lamp flicker? And how do we fix it?" are stimulating questions. Students then learn how to use separate thresholds to turn on and turn off the lamp.

2) *Activity 2 Robot Car Control:* This activity is designed to teach about state and state-change. Students are asked to control a two-wheel floor robot usijng a control pad with four buttons consisting of "forward", "backward", "left" and "right". Students control the direction of two DC motors to move the robot car acconding to the button pressed. This task is done by detecting the state of each button.

Once students are comfortable controlling the robot car, we created a situation to highlight the importance of state-change by asking students to use only two buttons. The first button was to toggle between forward and backward and the second to toggle between turning left and right. We want students to discover that writing code to detect the button state does not work anymore as the robot will erratically switch back and forth between the two actions during the time a button is depressed. To make this work properly, students must detect a button depress event (state changes from released to pressed).

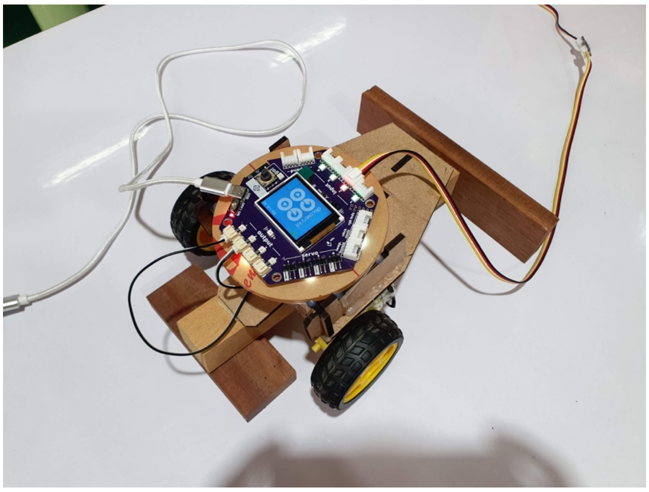

Fig. 4. A two-wheel floor robot car.

### C. *The Learning Continuum Steps*

Guided by the ZPD principle, the curriculum is divided into two sequential steps. Each step provides a learnable "zone" we believe is suitable for novice learners.

1) *Step I: Grasping the Basic Concepts in Autonomous and Control Systems*

2) *Step II: Reitterating with microcontrollers*

Step I created learnable activities that highlighted basic control ideas in autonomous systems without exposing students to the low-level circuits and offered a high-level programming language instead of the C-based languages commonly used with microcontrollers. We used a platform called the GoGo Board [2] instead of the Arduino in this step. The assumption is that novice learners will be able to grasp the main ideas without being distracted by low-level technical details.

Once the learner has grasped the concepts, Step II asked students to reiterate their understanding of autonomous systems. But this time students used microcontrollers and the C-based programming language. We used the Arduino platform in this step.

### D. *The Tool: The GoGo Board Platform*

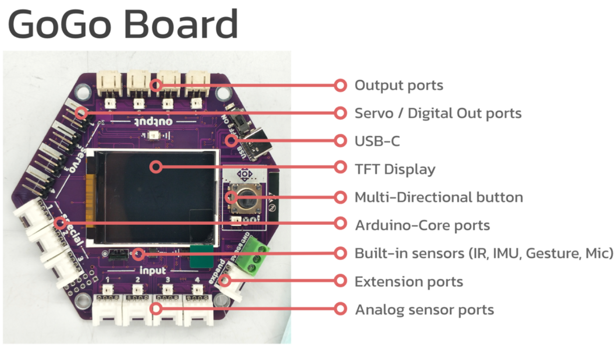


Fig. 5. The GoGo Board: An educational robotics learning toolkit.

The GoGo Board [2] is a robotics kit that offers plug-and-play ports for sensors and actuators (See a photo of the GoGo Board in Fig. 5). It has built-in circuitry to interface with analog sensors and contains built-in drivers for DC loads. It also has a built-in screen that shows graphs of sensor values, which allow learners to easily observe the behavior of sensors. The GoGo Board has an operating system that drives the screen and other components. Actuators can also be controlled directly from the screen.

Programming is done using a drag-and-drop blocks environment based on the Logo Programming language which was first developed by Seymour Papert at MIT [11] and is known for its learnability. Fig. 6 shows an example of a program written for the GoGo Board.

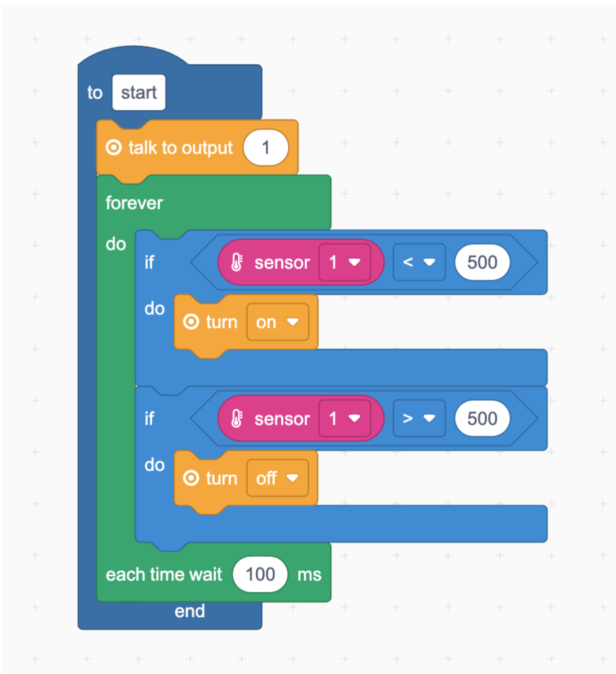


Fig. 6. An automatic night light program written for the GoGo Board.

The GoGo Board platform is suitable for the first learning step because it hides many complexities of the circuit and program allowing learners to spend most of their time on the autonomous control ideas being learned.

## IV. Method

### A. The Population and Learning Activities

The work was conducted as part of a curriculum design process for the 259106 Workshop Technology course offered to all freshmen engineering students at Chiang Mai University in Thailand. There were 402 students taking the course during the experiment. The course was divided into 10 sections, with 40-50 students per section. Each section divided students into 5-6 groups with 6-8 students per group. There were a team of 5 instructors teaching in every section. We randomly divided the sections into experimental and control groups.

- The experimental group consisted of 7 sections. In this group, we used our two-step curricular design.
- The control group consisted of 3 sections. In this group, we taught the control concepts using only microcontrollers (Arduinos).

We conducted a pre-class survey and found that both groups had similar experiences with coding and microcontrollers. The experimental group has scored on average 2.69 out of 5, SD of 1.08 while the control group has scored on average 2.49 out of 5, SD of 1.04.

The lessons were conducted over 12 hours divided into 4 weeks of 3 hours each. The learning activities were divided as shown in the following table.

TABLE I. Activities of the Experimental and Control Groups

| Week | Learning Units | | | |
|---|---|---|---|---|
| | *Experimental group* | | *Control group* | |
| | *Topic* | *Tool* | *Topic* | *Tool* |
| 1 | **Activity 1, Iteration I** Sensitivity & Hysteresis | GoGo Board | **Activity 1 Part 1** Control the automatic lamp. | Arduino |
| 2 | **Activity 1, Iteration II** Sensitivity & Hysteresis (repeat) | Arduino | **Activity 1 Part 2** Sensitivity & Hysteresis | Arduino |
| 3 | **Activity 2, Iteration I** State and State-Change | GoGo Board | **Activity 2 Part 1** Control the robot car. | Arduino |
| 4 | **Activity 2, Iteration II** State and State-Change (repeat) | Arduino | **Activity 2 Part 2** State and State-Change | Arduino |

For the experimental group students would study each activity twice (over two weeks). The first week they use the GoGo Board which allows students to focus mainly on the concepts. The second week the students try to accomplish the same task but by using the Arduino, which has more technical details about the circuit and code than the GoGo Board. The control group would study each activity once using only the Arduino, but each topic was spread across two weeks. The first week was to get started with the activity while the second week focuses more on the concepts.

### B. Data Collection

The data used in the analysis of this work were collected as follows:

1) *Time spent on each activity:* Students were asked to log their in-class progress in a shared online spreadsheet. We used this data to calculated the total time spent on each activity using a custom made script. This data can offer an insight to how much work the activities were to students.

2) *Formal Test:* All students in the study were asked to take a multiple choice quize covering the course contents.

3) *Survey:* Students volentarily filled out a survey sent out after the completion of the learning activities. We had 328 students (81.6%) who participated, 217 out of 277 from the experimental group (78.3%) and 111 out of 125 from the control group (88.8%).

4) *Focus Group:* We conducted two focus groups one with the experimental and the other with the control group. Students were selected randomly and then asked to participate. We had 18 students participated from the experimental group and 11 students from the control group. Each session lasted 30 minutes.

5) *Instructor interview:* We interviewed all 5 instructors about their experience conducting both the experimental and control groups.

## V. Results

### A. Time Spent on Learning Activities

One concern that was raised during the design process was whether the two-step process will require too much additional time as the students have to repeat the activities twice. This issue turns out to be manageable. In fact, the total time used in the second activity (robot car) was lower when using the two-step process as shown in the graph below (Fig. 7).

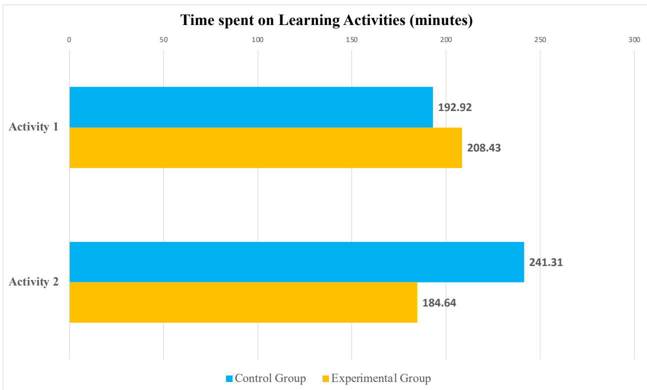


Fig. 7. Time spent at designed activities across two student groups.

In week 1, students using the GoGo Board spent on average 86.58 minutes to both create the automatic lamp and experiment with sensitivity and hysteresis. In contrast, the control group spent on average 113.54 minutes just creating the automatic lamp using the Arduino (no experiment with sensitivity and hysteresis). In week 2, the experiment group remade the automatic lamp and recreated the experiment with sensitivity and hysteresis using the Arduino. The students spent on average 121.85 minutes. The control group spent on average 79.38 minutes extending their lamp from week 1 to experiment with sensitivity and hysteresis. In total, the experimental and the control groups spent 208.43 and 192.92 minutes on activity 1 respectively. The time of the experimental group was not significantly higher than the control group and fits within the available class time.

For activity 2 in week 3 and 4, the average total time spent for the experimental group was 184.65 minutes while the average time for the control group was 241.31 minutes. This finding is quite remarkable as the experimental group, despite having more work, spent almost an hour (56.66 minutes) less to finish the lessons than the control group. This data may suggest that when students gain understanding of the higher-level principles prior to learning the technical details of microcontrollers, they are able to perform better. However, not all results point in this direction as shown in the next section.

### B. *Formal Test Comparison*

We conducted a multiple-choice formal test on the learned topics to compare academic performances of both groups. The results show that there were no significant differences between the two groups. The experimental group scored on average 7.36 out of 11 (66.88%), SD=2.15. The control group scored on average 7.06 out of 11 (64.16%), SD=2.03. This result was unexpected as we had anticipated that the experimental group would score better and reflect a better academic performance. However, multiple choice exams may not capture all the dimensions needed to appreciate the benefits of our approach.

### C. *Surveys*

Most students in the experimental group found that the two-step approach helped them understand the concepts and tools better (Score=4.04 out of 5, SD=0.84). We had 122 students who further elaborated their answer. We grouped this feedback and ranked them by frequency. Positive feedback samples from the sorted list are *"the GoGo Board allows for easy understanding before implementing the real system in Arduino", "the lesson goes from easy to complex", "I understand the GoGo Board but still struggle with the Arduino", "Once we can do it with the GoGo Board, doing it with the Arduino becomes not too difficult"*. Negative feedback samples from the sorted list are *"I still don't understand even after doing it twice", "I prefer writing text code more than using blocks"*. "I like the Arduino better. It is more useful for future work".

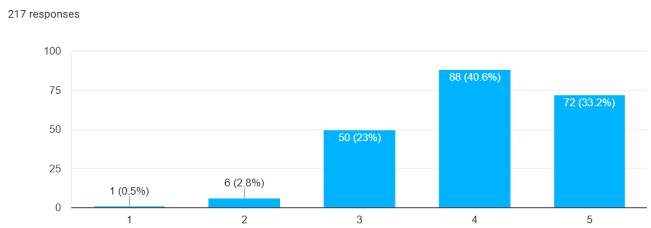


Fig. 8. The responses from the question: How much do you think learning in 2 iterations helped you understand the concepts and tools in this course.

Both student groups were asked how competent they think they are in using microcontrollers in the future. The experimental group had a slightly higher score compared to the control group. The average score was 3.52, SD=0.93 compared to 3.09, SD=1.07 respectively. Representative elaborations from groupings sorted by frequency in the experimental group includes *"The details [of using the tools] might have to be refreshed but I will remember the general idea", "I did the experiment twice using different tools. I think I get it.", "I can choose the right tool for the job"*. Elaborations from the control group includes *"I might need someone to teach me a bit more", "I think I can do it, but I will find a tool that I'm comfortable with for the job"*.

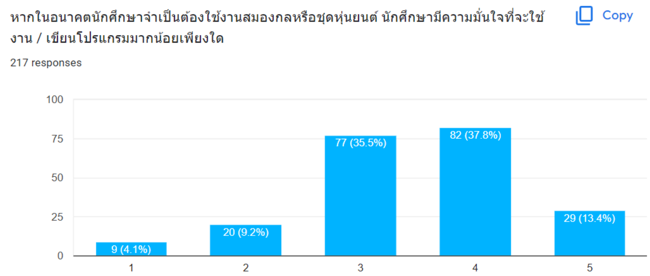


Fig. 9. The responses from the experimental group to the question: How competent do you think you are if you need to use microcontrollers or robotics tools in the future.

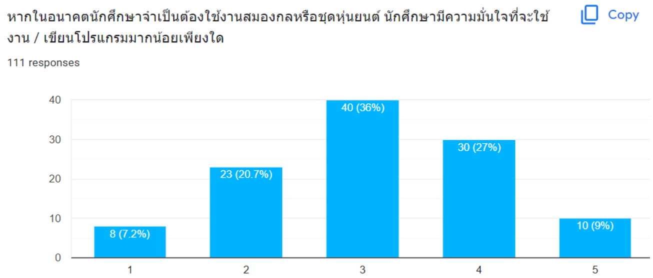


Fig. 10. The responses from the control group to the question: How competent do you think you are if you need to use microcontrollers or robotics tools in the future.

### D. *Focus Group*

The 18 volunteers from the experimental group were asked to describe their learning experience and how they felt about the multi-iteration approach. Students commented that the Arduino requires more technical understanding so starting with the GoGo Board allowed them to focus on the lesson. The Arduino were still perceived as technically challenging. Students commented that they need more training to fully understand it.

Students in the focus group consisted of both students who are completely new to microcontrollers and those with some experience. Students with experience mentioned that they preferred the Arduino because it is more powerful and flexible, but they also acknowledged that starting with the GoGo Board is beneficial to their less experienced friends.

We had 11 volunteers from the controlled group. The consensus was that they were not so confident about using the Arduino platform. They have limited experience from the 4-week course. There were so much to learn about the platform. Being able to use Arduinos in future projects will need more practice.

The fear of not being able to identify and fix problems is a particular concern. Several students referred to activity 2 where they were asked to detect the button state-change. They did not know if the problem at hand was with the hardware circuit, the code, or their thinking.

### E. Instructor Interview

Feedback from the 5 instructors involved in this course was extremely positive. We asked them to evaluate both approaches and choose the one they wish to continue using.

All instructors preferred using the two-step method. Representational comments include *"Even though using both the GoGo Board and the Arduino requires more setup work, it is easier to conduct the classes.", "The ideas we need to teach are simpler to comprehend when demonstrated using the GoGo Board. When we shift to Arduinos, it is mostly just about the technical setup. It is so much easier [to use Arduinos] when the students already understand the concept."* The interview shows that the instructors see the two-step approach as more effective.

One instructor further elaborated that when using only the Arduino, there are countless ways a student can make mistakes algorithmically. Sometimes the instructors struggle to help students when the approach taken is far off from the correct path, and they end up spending most of the time fixing a problem that is a result of not understanding the underlying concepts. The instructor mentioned that *"sometimes it is better to step back and make sure the student understands [the concept] before coding it. This is why I think using the GoGo Board first is a much better way."*

The two-step approach has been selected as the method used in this course in subsequent semesters. This is important evidence that the instructors genuinely prefer this approach. So far, since this experiment, more than 1,500 engineering students spanning 4 semesters have taken this course using this two-step approach.

## VI. Conclusions

This work has presented a curricular design approach that emphasizes learnability by creating sub-units that take into consideration the Zone of Proximal Development, which defines what is learnable by the target group. We show a two-step approach for teaching novice engineering students the basic concepts of autonomous and control systems. The steps separate learning of the underlying control concepts from applying the concepts using microcontrollers.

We used the GoGo Board as the tool to convey the important control concepts. This is an example of "selective exposure" where choosing a tool with the right level of abstraction is beneficial to accomplishing the learning objective.

We described a learning activity that offers hands-on experience with the desired concepts. This concrete-before-abstract approach also helps students accomplish the desired learning objectives.

Results have shown that this two-step approach is beneficial. Although there were no significant benefits in formal testing scores, the attitude towards microcontrollers of the students is more positive with the two-step approach. The learning experience appears to be positive as well. Students in the experimental group clearly expressed that the two-step approach helped them with the learning process. We have also proven that students can perform the required tasks within the given time constraints even though they must repeat the tasks twice using different tools. In fact, there is evidence that the proposed approach could end up using less time. Further studies can investigate to better understand this finding. One possible study is to see if the multi-iteration approach is more effective with more complex topics. Lastly, we have shown that instructors prefer this two-step approach using both GoGo Boards and Arduinos over using Arduinos alone. Our proposed approach has been adopted by this course and has been used with subsequent students since.

Future work can observe the long-term benefit of the proposed approach especially how well students later use the tools and concepts learned from the course in their own works. It would be beneficial to learn how the positive attitude towards microcontrollers and the concepts play a role in later stages of a student learning process.


## Acknowledgment

This research supported by Learning Inventions Laboratory under Department of Computer Engineering, Faculty of Engineering, Chiang Mai University, Thailand.


## References


[1] P. Blikstein, "Gears of Our Childhood: Constructionist Toolkits, Robotics, and Physical Computing, Past and Future," In Proc. of International Conference on Interaction Design and Children, 2013.

[2] A. Sipitakiat, P. Blikstein, and D. Cavallo, "The GoGo Board: Moving towards highly available computational tools in learning environments," In Proc. of Interactive Computer Aided Learning International Workshop, 2002

[3] LS. Vygotsky, and M. Cole, "Mind in society: Development of higher psychological processes." Harvard university press, 1978.

[4] U. Wilensky, "Abstract meditations on the concrete and concrete implications for mathematics education." Cambridge, MA: Epistemology and Learning Group, MIT Media Laboratory, 1991.

[5] P. Blikstein, "Computationally enhanced toolkits for children: historical review and a framework for future design." Foundations and Trends® in Human–Computer Interaction 9.1, 2015, pp. 1-68.

[6] S. Murphy, A. MacDonald, and L. Danaia, "Sustaining STEM: A framework for effective STEM education across the learning continuum." STEM education across the learning continuum: Early childhood to senior secondary, 2020, pp. 9-28.

[7] Benitti, and Fabiane Barreto Vavassori, "Exploring the educational potential of robotics in schools: A systematic review." Computers & Education 58.3, 2012, pp. 978-988.

[8] Alimisis, and Dimitris, "Educational robotics: Open questions and new challenges." Themes in Science and Technology Education 6.1, 2013, pp. 63-71.

[9] Galadima, and Ahmad Adamu, "Arduino as a learning tool." 2014 11th International Conference on Electronics, Computer and Computation (ICECCO). IEEE, 2014.

[10] M. Bers, "Positive technological development: Working with computers, children, and the internet." MassPsych 51.1, 2007, pp. 5-7.

[11] M. Resnick, S. Ocko, and S. Papert, "LEGO/logo–learning through and about design." Cambridge: Epistemology and Learning Group, MIT Media Laboratory, 1990.