Journal of Engineering Education Transformations,
Volume No 36, January 2023, Special issue, eISSN 2394-1707

10.16920/jeet/2023/v36is2/23082

Enhancing Experiential Learning to Address Loss of Hands-on Laboratory Due to Covid-19 Pandemic

K. Supriya, H. Rohini, N. Prabha, Basawaraj, I. Nalini
Electronics and Communication Engineering, KLE Technological University, Karnataka, India.

***Abstract*—All sectors of life have been hit hard by the pandemic caused by covid-19. This impact has been brutal on streams like engineering in the educational sector, where practical or laboratory courses play a vital role in learning. Due to the limitations of online learning, the students could not explore the full extent of hands-on learning in their Fourth semester ARM microcontroller course. In order to compensate for the loss, the laboratory activities in the Fifth semester RTOS (Real-Time Operating System) Laboratory course are enhanced to provide students with a hands-on learning experience in building applications in both the ARM and RTOS environment. The extended activities enhanced student learning in the ARM environment, which they were previously deprived of during online instruction. Also, emphasis is provided on applying optimization techniques to memory and timing requirements for a given problem statement. The results show that the extended activities helped students co-relate and integrate concepts addressed during their 4th and 5th-semester courses and build a small application of the embedded systems.**

***Keywords*—Embedded Real-time systems; Extended Activity; project-based learning;**

***JEET Category*: Practice**

I. INTRODUCTION

We may not realize it, but the microprocessors or CPUs surround us and the computing these do for us. They are present in every aspect of our lives. CPUs are in the desktop or laptop we use, the electronics that enable different functions in our car, and the machines used to check out at the store. The CPUs help scientists and artists create things unimaginable only yesterday. CPUs are everywhere and shape just about everything we do. Such systems can be called embedded systems consisting of hardware and software components. Hence there is a need for the electronics engineer to learn and experiment with the embedded system and solve some problems in the future. To serve this purpose, the course structure in our university supports learning the CPUs and software in the four courses along with the aid of integrated Laboratory experiments for undergraduates (Patil et al., 2016). The Covid-19 pandemic in 2020 and 2021 has adversely affected the laboratory courses, due to online training. This study addresses the loss of experimentation in one such laboratory course. The two courses considered for the study are ARM Microcontroller Lab (the course deprived of detailed investigation during online training) and Operating System and Embedded Systems Design (the course in the analysis in the consecutive semester and experimented in offline mode of exercise) (Pattanashetti et al., 2021). These two courses deal with one kind of CPU and hardware platform to create an embedded system application with two different software approaches (Hongal et al., 2016; Pillai et al., 2020; SalewskiFalk et al., 2005; Sudha et al., 2008). By applying this new pedagogy, the loss of hands-on with hardware using the bare-metal coding technique in one semester is effectively compensated in the subsequent semester by integrating bare-metal and real-time operating system techniques. The study also enhanced the student's qualities required to work in a team.

II. METHODOLOGY

The basic idea of this activity is to connect and correlate the two courses by having a hands-on session for the intended microcontroller and employing peripheral programming with two different approaches (Hohl & Hinds, 2014; Li & Yao, 2003). Figure 1 shows the methodology followed to implement the activity.

It has two phases:

Phase 1: Application development in ARM mode/ bare metal Coding

Phase 2: Application development in RTOS Mode Environment.

Phase 1 begins with the creation of student teams. Each student team follows a process to complete the activity in a stipulated time. The process starts with allocating a unique problem to each group; the team produces different solutions with appropriate assumptions to solve the given situation. They apply ARM microcontroller programming knowledge to develop the firmware (Wray & Crawford, 1984). In the next step, a bare-metal code is implemented to achieve the desired application results both at the simulation and hardware levels. Different code optimization techniques such as modular approach, loop unrolling, data alignment, and others are applied at this stage to analyze the timing and memory usage of the proposed application. The students undergo review at each step of the activity. Parallel to the activity, the students learn operating systems and real-time operating system concepts. The RTOS concepts like scheduling, synchronization, and inter-task communication are dealt with as part RTOS laboratory, which prepares the students for the next phase of the activity. The second phase of the exercise begins with the justification for choosing the scheduling policy and different kernel components/objects to be used in the RTOS-based development of the same application developed in phase 1. The next stage is implementing the application in the RTOS environment, followed by applying proper optimization techniques to analyze the system's memory usage. The completion of the activity with a comparison of both modes of implementation. The evaluation method and rubrics for evaluation are discussed in the following sections

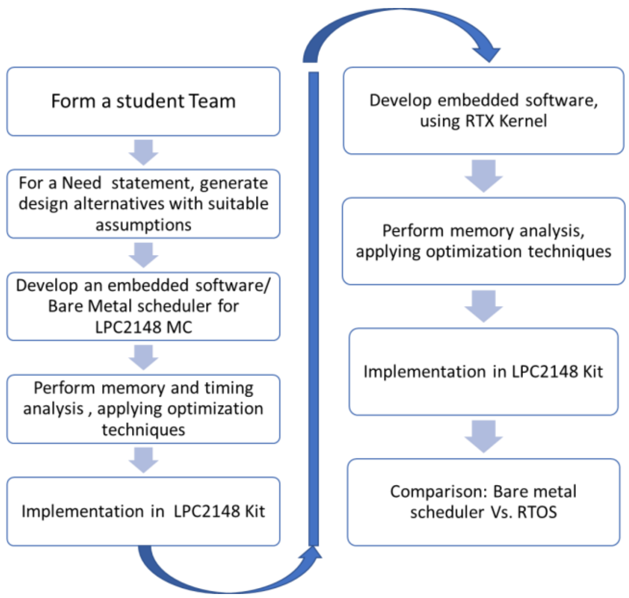

Fig 1: Methodology

III. ASSESSMENT STRATEGY

Evaluation rubrics and performance indicators for outcome-based education are shown in Table I, and Table II shows the Assessment plan for the course. The activity is assessed in two reviews, i.e., review-1, which accounts for 40 marks in cumulative in-semester evaluation (CIE), to evaluate the application in the ARM mode. Furthermore, review -2 to assess the application's assessment in RTOS mode accounts for the 25 marks in the CIE. The post-tests account for five marks and the exercise experiments evaluate for ten marks. The detailed rubrics used for evaluation are discussed in Table III.

A. SAMPLE PROBLEM STATEMENTS

Each student team has been allocated a unique Real-time application-based problem statement. Some examples are:

1. Develop a home intruder detection system. Make reasonable assumptions, and consider possible means of intruder entry and detection. The design should be easy to use.

TABLE I:
Review Parameters

		Parameters	PI's	Marks
Review-1(40M)	Arm Mode	Identifying multiple solutions, selecting the best-suited solution and justifications	4.2.1	5
		Functional block diagram relating input & output(Interfacing Diagram)	4.1.2	5
		Algorithm for the selected application	1.4.5	5
		Optimization Techniques used(min of 2)	1.4.5	5
		Modular Approach implementation for the application code	13.3.1	5
		Execution Profiling(Timing and Memory Analysis)	5.2.2	5
		Demonstration of results in Simulation	4.3.2	5
		Demonstrate the results in Hardware	4.3.2	5
Review-2 (25M)	Rtos Mode	Implementation in RTOS Environment with suitable kernel objects used	1.4.5	5
		Execution Profiling(Timing and Memory Analysis)	5.2.2	5
		Demonstration of results in Simulation	4.3.2	5
		Demonstrate the results in Hardware	4.3.2	5
		Report submission in Latex (as given in the format)	10.1.2	5
			Total Marks	65

TABLE II
Evaluation Scheme

Internal Semester Assessment (ISA)-80%	Assessment	Weightage in Marks
	Review-1	40
	Review-2	25
	Experiments	10
	Post Test	05
End Semester Assessment (ESA)-20%	Write up & viva	10
	Conduction and Result	10
	Total	100

2. A patient has been advised to wear a monitor by the doctor. The device can monitor heart rate and blood pressure. The device stores the data and, at regular intervals, uploads it to a web server, which the doctor can access. Further, if there is a rapid change in either value, the monitor immediately alerts the doctor and an emergency contact.
3. A planter with a GPS sensor and an actuator for controlled drop or seed in the trench exists. The farmer can pre-set the desired crop density for different farm regions. As the planter is pulled along, it should determine its position and appropriately change the rate at which the seeds are dropped. Note that when the tractor reaches the end of the field and is turning, the planter is raised, and the seeds are not dropped. For simplicity, you can assume that the field is a rectangle, and the regions in the farm are rectangular strips

running perpendicular to the direction in which the planter is being pulled. (A planter is farm equipment pulled by a tractor and plants seeds in rows throughout a field.)

4. A toll-road operator has automated toll collection to improve the speed and efficiency of operations at a busy toll plaza. The automated toll collection system can accept electronic payments (RFID tags) and cash. For simplicity, assume that: a) cash payments are made using coin(s) of only two denominations (5, 10); and b) the toll for all vehicles is the same, 25.
5. Checkout counters in stores can be backed up during certain periods. Such long lines can turn away customers, but increasing the number of counters is expensive. Automated billing can help decrease costs. Help design an RFID-based system to automate billing and checkout.
6. Banks would like to serve their customers at a lower cost better. Hence, they would like to install modern Automated Teller Machines (ATMs). Help design a suitable system.

TABLE III
Rubrics

Assessment Factor	**PI**	**Evaluation Parameters**	**Good**	**Average**	**Poor**
Functional block diagram relating input & output	4.1.2	Relate modern engineering experimentation, including experiment design, system calibration, data acquisition, analysis and presentation	Able to identify and relate input and output peripherals	Able to identify, fail to justify	Poor in identifying appropriate input and out peripherals
Programming Skills	1.4.5	Apply Programming Skills	Able to Code in Embedded C with appropriate optimization techniques	Able to code, fail to use proper optimization techniques	Poor in writing embedded code
Debugging logical & syntax errors	4.3.2	Critically analyze data for trends and correlations, stating possible errors and limitations	Aware of identifying logical and syntax errors to debug the program	Able to identify syntax errors, fail to debug the program	Poor in identifying errors and
Discussion of results	4.3.2	Represent data (in tabular and graphical forms) to facilitate analysis and explanation of the data and drawing of conclusions	Able to write observations before the commencement of lab and discuss the results obtained immediately after completion of the experiment.	Not able to write observations before the commencement of lab, but discuss the results.	Unable to write observations and discuss the results obtained immediately after completion of the experiment.
Modern Engineering tools	5.2.2	Demonstrate proficiency in using EDA tools	Able to independently explore and use KEIL IDE to write and debug ARM programs using assembly and c language for a given problem.	Able to explore and use KEIL IDE to write and debug ARM programs using assembly and c language for a given problem with the instructor's help.	Unable to explore and use KEIL IDE to write and debug ARM programs using assembly and c language for a given problem without instructor's help.
Journal and observation book	10.1.2	Well-constructed document Produce clear, well-constructed, and well-supported written engineering documents	Produce clear, well-constructed, and well-supported written engineering documents.	Able to write the report in a specified format but not in time.	Unable to write the report in the specified format and does not report in time.
Developing the application code	13.3.1	Ability to identify design principles for the development of software systems.	Able to identify design principles for the development of software systems.	Partially able to identify design principles for the development of software systems.	Unable to identify design principles for the development of software systems.

IV. RESULT ANALYSIS

The effectiveness of the experimentation done for the RTOS lab is discussed in this segment. The designed problem statements helped the students understand the underlying concepts related to using different on-chip and off-chip peripherals. Students were able to give solutions to the problems. The In Semester Assessment (ISA) results for a set of students have been considered for analysis. The first measure of the course outcome was objective and based on the student performance in their ISA, which accounts for 80% of the final grade. ISA is constituted of two reviews and laboratory exercises. Reviews 1 and 2 accounted for 45% and 25% of the final grades, respectively. The results for students in two different batches, chosen at random from 10, were considered. There were 48 students, 25 in the first and 23 in the second. The analysis shows there is generally an improvement in the marks for all students in both Batch-1 and Batch-2 with respect to review-1. Moreover, the average percentage of marks obtained increased from about 70% to 83% from Review-1 to Review-2 for both batches. The individual scores are shown in Figure 2 and Figure 3 for two batches.

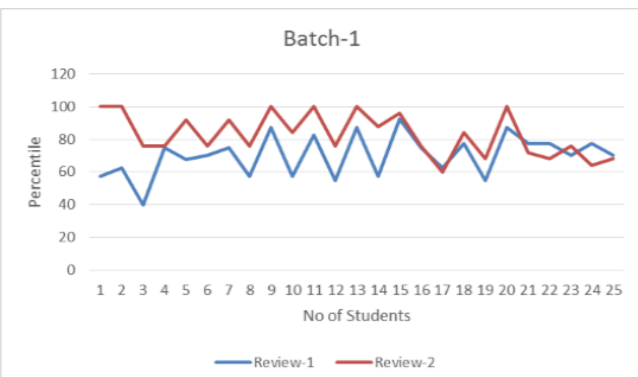

Fig. 2: Result Analysis for Batch-1.

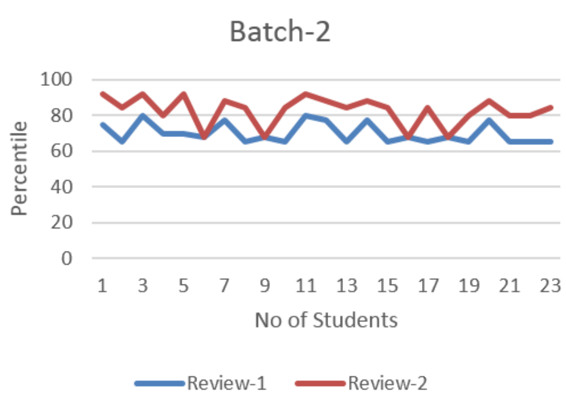

Fig. 3. Result Analysis for Batch-2.

V. IMPACT OF THE ACTIVITY

The impact of extended activities on student learning was also measured through student feedback obtained employing a survey questionnaire shown in Table IV. The input was conducted using the google forms platform. The questionnaire consisted of nine items measured using a five-level Likert scale. Seventy-three students responded to the survey Figure 8 shows the platform and response details.

• Three items, namely "learning objectives were clear," "course content was organized and well planned," and "course workload was appropriate," addressed the structure of the laboratory activities. The results show that over 75% of the students had either a positive or neutral opinion about the clarity of learning objectives and planning organization of course content. A detailed analysis of the responses to the individual questions shows that only under 10% of the respondents had an unfavourable opinion of the clarity of the learning objectives. This is important because a significant goal of course design is clarifying the students' learning objectives. An issue highlighted in the results is the course content organization planning and the resulting workload. The respondents responded more negatively to these two issues, with about 20% and 25% of the students having a negative opinion. The results are shown in Figure4.

• Another three items, namely the ability to "understand the optimization techniques and analysis," "analyze the difference between C code and RTOS," and "understand the hardware peripherals in a better way," addressed the outcome of laboratory activities. The results show that over 80% of the students had a non-negative opinion of all three items. A detailed analysis of the results, similar to the items regarding the structure of the laboratory activities, shows that less than 10% of the students had an unfavourable opinion of the ability to distinguish between C code and RTOS. However, about 15% and 20% had a negative view of understanding optimization techniques, analysis, and hardware peripherals. These results highlight the need for additional activities utilizing hardware peripherals and instruction on optimization techniques. The results are shown in Figure 5.

• The survey also measured the effort the students put into the course. The results show that only a tiny percentage of students were dissatisfied with the amount of effort they put in. The results are shown in Figure 6.

• Finally, the questionnaire addressed the contribution of laboratory activities to learning. Two items, "level of skill/knowledge at the start of course" and "level of skill/knowledge at the end of the course," were used to gauge the students' perception of the change in their skill levels. The results were positive, with most students improving their skill levels. This is an actual result because the primary goal of the course was to enhance student skill/knowledge levels, which was successfully achieved here. Looking at the results in detail shows an increase in the number of students with a positive opinion about their improvement in the level of skill/knowledge and a decrease in both students with negative and neutral views about the same. The results are shown in Figure 7.

TABLE IV.
Activity Feedback Questionaire

Sl No.	Questions	
1	Lab activity structure	Learning objectives were clear
		The course content was organized and well planned
		The course workload was appropriate
2	Lab activity outcomes	Able to understand the optimization techniques and analysis
		Able to analyze the difference between C code and RTOS
		Able to understand the hardware peripherals in a better way
3	Level of effort	Level of effort you put into the course
4	Contribution to learning	Level of skill/knowledge at the start of the course
		Level of skill/knowledge at the end of the course

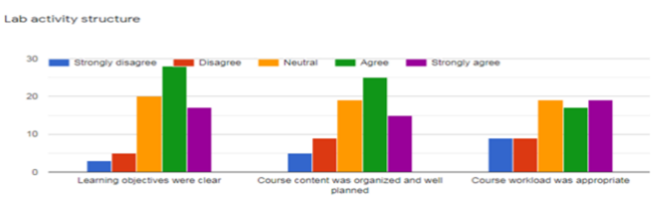

Fig 4: Student's Feedback for Lab activity structure

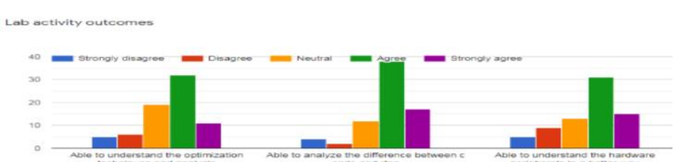

Fig 5:: Student's Feedback for Lab activity outcomes

Fig 6:: Student's Feedback for Level of effort

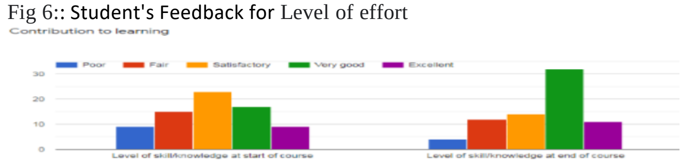

Fig 7:: Student's Feedback for Contribution to learning

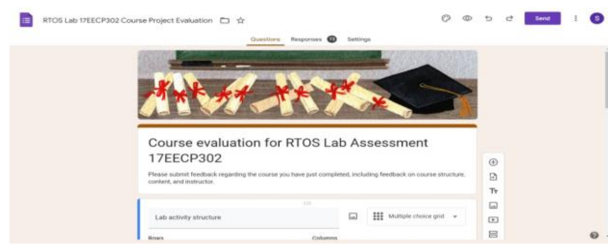

Fig 8:: Students Feedback platform

VI. CONCLUSION

Disruptions caused by the COVID-19 pandemic severely affected in-person instructions and have potentially harmed students' learning outcomes. This work aimed to discuss the consequences of an attempt to address the loss of hands-on learning experience through experiential learning. The approach involved stitching two related courses through implementation assignments/extended problem-solving exercises. Objective and subjective criteria were used to measure the enhanced coursework's effect on student learning. Two in-semester reviews as a part of the evaluation significantly impact students' performance in the exam. A survey of the students was conducted as part of the subjective measure. The results of this survey show that the student's opinion of the course outcomes, i.e., the learning outcomes, was very positive.

Furthermore, their opinion of the course's structure and organization was also positive. The results of this study hold great value to educators striving to address limitations in pre-requisite knowledge among their students. The results show that it is possible to appropriately manage and overcome the limits through a suitably designed course. Note that while the survey results show that the students felt that they had put in more effort, this is expected as the course attempted to stitch concepts learnt in the previous semester and apply them in the current semester. Further studies are required to understand such courses' potential challenges and limitations. In future, we will attempt to build appropriate studies to understand the same.

References

Hohl, W., & Hinds, C. (2014). *ARM Assembly Language: Fundamentals and Techniques*. CRC Press.

Hongal, R. S., Shettar, R. B., Supriya, K., & Shrishail, P. (2016). "Product Development" - a New Learning in Open-Ended Experiment. *Journal of Engineering Education Transformations*, *29*(Special Issue). https://doi.org/10.16920/JEET/2016/V0I0/85572

Li, Q., & Yao, C. (2003). *Real-Time Concepts for Embedded Systems* (1st ed.). CRC Press. https://doi.org/10.1201/9781482280821

Patil, S., Supriya, K., Uma, M., Shettar, R. B., & Kumar, P. (2016). Open Ended Approach to Empirical Learning of IoT with Raspberry Pi in Modeling and Simulation Lab. *2016 IEEE 4th International Conference on MOOCs, Innovation and Technology in Education (MITE)*, 179–183. https://doi.org/10.1109/MITE.2016.044

Pattanashetti, S., Hongal, R. S., Nissimagoudar, P., Katwe, S., Pillai, P., Kinnal, B., & Iyer, N. C. (2021). Enhancing System Development learning by applying Real-Time Operating System concepts through connecting courses. *Proceedings of 2021 World Engineering Education Forum/Global Engineering Deans Council, WEEF/GEDC 2021*, 95–100. https://doi.org/10.1109/WEEF/GEDC53299.2021.9657463

Pillai, P. S., Bhagyashree, K., Hongal, R., Nissimgoudar, P. C., Shrishail, P., & Iyer, N. C. (2020). Enhancing Optimization Skills in Embedded Stream: Integrated Learning. *Procedia Computer Science*, *172*, 229–234. https://doi.org/10.1016/J.PROCS.2020.05.036

SalewskiFalk, WilkingDirk, & KowalewskiStefan. (2005). Diverse hardware platforms in embedded systems lab courses. *ACM SIGBED Review*, *2*(4), 70–74. https://doi.org/10.1145/1121812.1121825

Sudha, G., Kumar, A., Mercado, R., Manimaran, G., & Rover, D. T. (2008). Enhancing Student Learning with Hands-On RTOS Development in Real-Time Systems Course. *2008 38th Annual Frontiers in Education Conference*, S2H-11-S2H-16. https://doi.org/10.1109/FIE.2008.4720667

Wray, B., & Crawford, B. (1984). *What Every Engineer Should Know about Microcomputer Systems Design and Debugging*. Marcell Dekker.