The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
ASEE
Paper ID #44428

Agile Model-Based Systems Engineering Framework to Design a Laboratory Course—Case Study: An Embedded Systems Laboratory Course

Mr. Kishore Kumar Kadari, University of South Florida

Kishore Kadari is a Ph.D. Candidate in the Electrical Engineering (EE) department at the University of South Florida(USF). He is currently working as a Design Engineer at Jabil, focusing on Embedded Software. His vision is to contribute to the advancements in high-level orchestration of education and healthcare services using AI, ML, computer vision, Model-Based Systems Engineering, and embedded development. He finished his master's degree with a thesis under Dr. Arash Takshi on "Electroplated 3D printed conductive track" in the EE department. He worked as a research assistant for Dr. Kwang Cheng Chen on privacy-preserved machine learning, SVM. He is working with Dr. Wilfrido Moreno on a patient monitoring system that can accelerate positive outcomes using AI and MBSE.

Dr. Wilfrido A. Moreno P.E., University of South Florida

Wilfrido A. Moreno received his M.S.E.E & Ph.D. degrees in Electrical Engineering from the University of South Florida (USF), Tampa - Florida in 1985 and 1993 respectively. He is currently a Professor in the Electrical Engineering Department at USF.

Luis Miguel Quevedo, IEEE Educational Activities

Agile Model-Based Systems Engineering Framework to Design a Laboratory Course - Case Study: An Embedded Systems Laboratory Course

1. Introduction

Model-Based Systems Engineering for Embedded Systems is experiencing rapid evolution due to the increasing complexities of modern projects. This era demands heightened integration and collaboration, with a keen focus on AI trust, risk, and security management alongside intelligent applications[1]. Academia plays a crucial role, especially in the area of microprocessor/embedded systems education; as society transitions towards a digital economy, it is important to adapt and evolve. There's a paramount need to strengthen the US workforce and domestic manufacturing capabilities. Academia must lead in workforce development and in STEM education(Science, Technology, Engineering, Mathematics), emphasizing the importance of lifelong learning; it's crucial to leverage innovative AI tools to adapt and thrive in the face of these constantly evolving demands[2].

2. Pedagogical Survey[1]

To keep pace with technological advancements, it is vital to update curricula by integrating cutting-edge tools like AI[3], virtual reality, quantum computing, and metaverse[4] technologies, leveraging industry partnerships. Additionally, inclusive education across all age groups is vital to adapt to rapid changes[5]. Embedded systems engineering is evolving with the proliferation of microcontrollers (MCUs), remaining central to digital transformation and economic growth. This field now requires designing parallel systems using a systematic, model-based approach.

Technological advancements in sensing, computing, and communications have expanded to robotics, IoT, and machine learning-based applications. The increase in popularity of single-board architectures has been fueled by the innovative solutions offered by electronic manufacturers. Training students in using computers and evaluation boards available on the market is essential, especially in using programming languages like C for embedded systems. The industry that designs Embedded Systems seeks professionals with a blend of electrical engineering and microprocessor programming skills. This includes software and hardware proficiency and the ability to interpret schematics and datasheets.

The Model-Based Systems Engineering (MBSE) approach, incorporating structure, behavior, requirements, and parametric, is essential in the context of the developed/proposed Embedded Systems Laboratory(ES-LAB). The literature[6] shows the strategies and best practices for MBSE adoption in the Embedded Systems industry. Local Industry partners play a crucial role in identifying industry needs. The demand for talent in embedded programming throughout the industry is evident. Dassault System's Magic System of System Architect tool for MBSE implementation and training Teaching Assistants in SysML is a step forward in aligning educational programs with industry requirements.

3. Course Specifications

The block definition diagram shown in Figure 1 effectively illustrates the proposed ES-LAB system. In the Microprocessors lab, students engage with the MCUXpresso IDE and the

[1]This material is based upon work supported by the National Science Foundation under Grant No. 2022299

LPC55S69 board, facilitated by the Canvas learning management system. The ‘Student’ is conceptualized as an ‘IO block’, responsible for assimilating inputs from Canvas and various tools, and demonstrating their learning through different types of continuous evaluations, i.e. quizzes, test, etc. This representation underscores the interactive and integrative nature of the learning process in the lab.

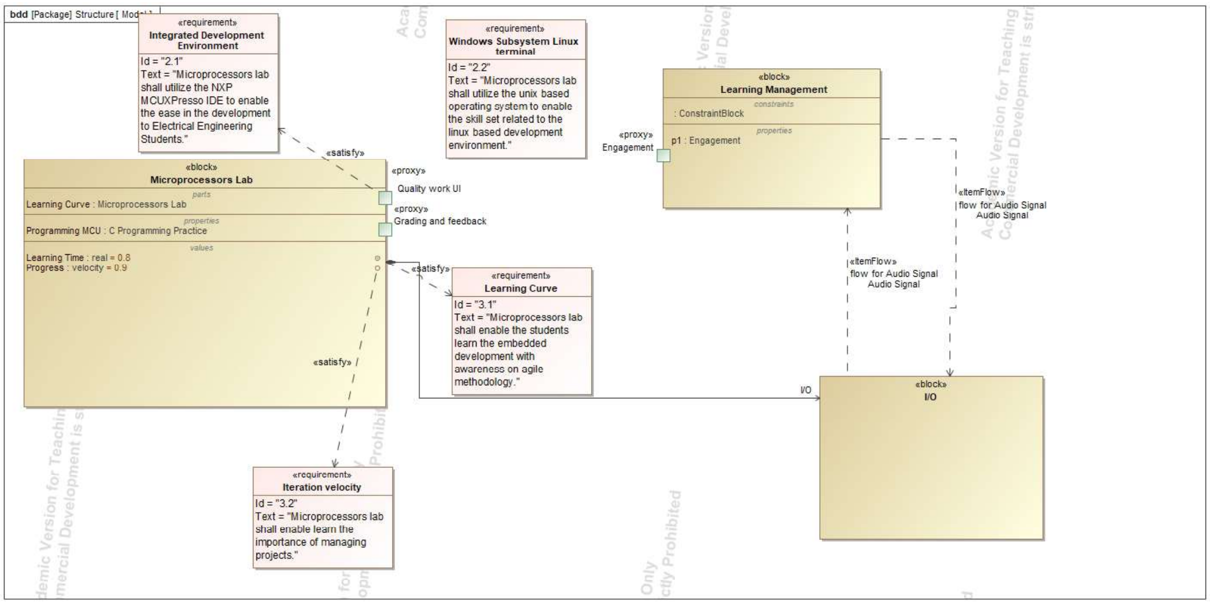


Figure 1: ES-LAB System Structure

In Figure 2, the student utilizes a learning management model to develop their model. This process is enhanced by continuous feedback from the Teaching Assistant (TA) on each assignment, allowing for ongoing adjustments and improvements in performance throughout the semester. This dynamic approach highlights the importance of timely feedback in the learning process.

The learning management model, as depicted in Figure 3, initiates with updating the faculty and teaching assistant skillsets. Subsequently, a comprehensive lab manual was developed, followed by creating a course in Canvas. Students engage with the content provided by the faculty and teaching assistants, conducting experiments as part of their learning process. The teaching assistants’ evaluation plays a crucial role in enabling students to fine-tune their learning curve, thereby mastering the different learning objectives in the course. This model emphasizes the iterative process of learning and assessment.

There are three main categories in the course requirements outlined in Figure 4. The functional requirements encompass all the learning objectives necessary for success in the ES- LAB by utilizing specific tools. The User Interface (UI) requirements are centered around the Integrated Development Environment and debugging tools. Lastly, design constraints pertain to the student’s learning curve and the pace at which they complete assignments, highlighting the balance between educational content and the learner’s progress.

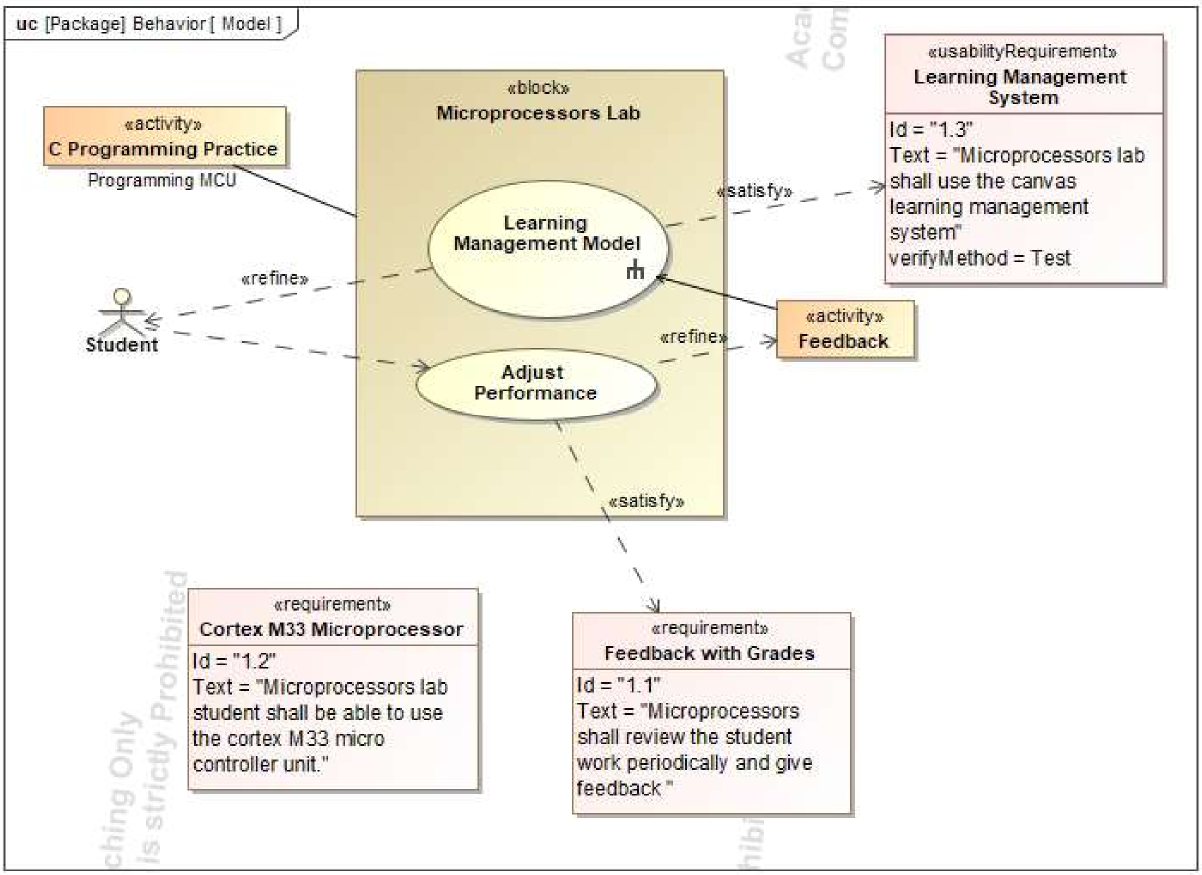


Figure 2: ES-LAB System Overall Design

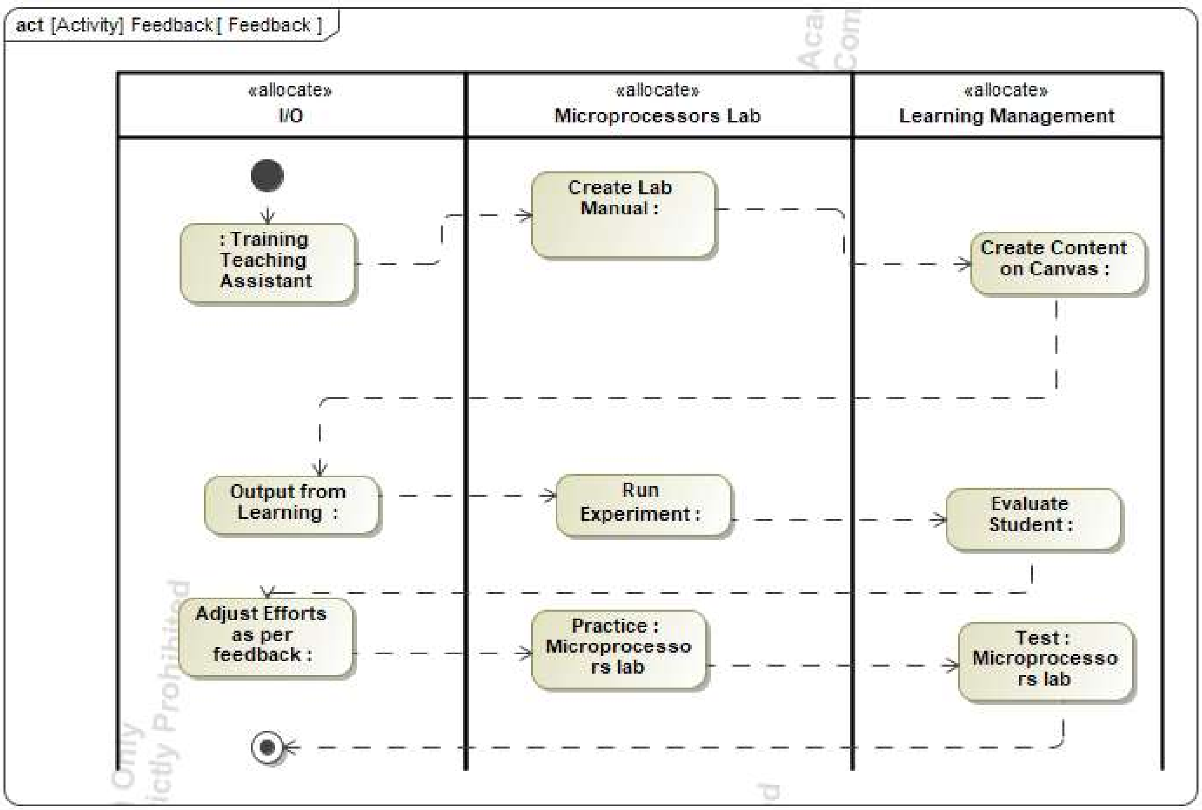


Figure 3: ES-LAB System Detailed Design

#	Name	Text	Satisfied By
1	1 Functional Requirements		
2	1.2 Cortex M33 Microprocessor	Microprocessors lab student shall be able to use the cortex M33 micro controller unit.	LpcXpresso55s69 Documentation
3	1.1 Feedback with Grades	Microprocessors shall review the student work periodically and give feedback	Adjust Performance
5	1.3 Learning Management System	Microprocessors lab shall use the canvas learning management system	Learning Management Model
6	1.5 Microcontroller Board	Microprocessors Lab enables students learning hardware analysis skills using Altium Web viewer.	LPCXpresso55S69 Development boa
7	1.6 Embedded Tool Chain	Microprocessors lab shall teach the compilation tools and techniques and methodologies behind it.	Training Teaching Assistant(context
8	1.7 Programming Skills	Microprocessors lab shall enable students learn C Programming to talk to the microcontroller unit.	C Programming Practice(classifier b
9	1.8 Problem Solving	Microprocessors lab shall facilitate brainstorming to identify the issue, formulate to Teaching Assistant and solve with help of communication skills and being resourceful, i.e, holistic development skill set.	Microprocessors Lab
10	1.9 Verification activities	Microprocessors lab shall develop the capability to verify the functionality peripherals interfaced with the ARM micro controller units	Verification
11	1.10 Peer Programming and Team building	Microprocessors lab shall introduce the peer programming skills while debugging and also using the team building and leading skills in the final projects.	Team Building and Leadership
12	1.11 Interdisciplinary Perspective	Microprocessors lab shall facilitate the development of the interdisciplinary skill which is an instrumental skill in the embedded development.	Interdisciplinary knowledge
13	1.12 Engagement	Microprocessors lab shall facilitate the persistent motivation through subject matter expert videos, the pre lecture reading about the experiments, post lab assignments to reinforce the learning and the quizzes to test the knowledge along the path.	Engagement
14	2 Ui Requirements		
15	2.1 Integrated Development Environment	Microprocessors lab shall utilize the NXP MCUXPresso IDE to enable the ease in the development to Electrical Engineering Students.	Quality work UI
16	2.2 Windows Subsystem Linux terminal	Microprocessors lab shall utilize the unix based operating system to enable the skill set related to the linux based development environment.	
17	2.3 Hands On Debugging Experience	Microprocessors lab shall provide the training for applied engineering skills such as firmware debugging that is very much essential in the design engineering	Grading and feedback
18	3 Design Constraints		
19	3.1 Learning Curve	Microprocessors lab shall enable the students learn the embedded development with awareness on agile methodology.	Learning Time : real = 0.8
20	3.2 Iteration velocity	Microprocessors lab shall enable learn the importance of managing projects.	Progress : ISO80000-3 Space and Tin

Figure 4: System Requirements Table

4. Course Implementation

In direct collaboration with industry stakeholders, the LPCXpresso55S69 board, a product of NXP Semiconductors, has been selected for ES-LAB experiments. The lab instructional platforms were donated by NXP Semiconductors, ensuring that students are exposed to industry-relevant and state-of-the-art equipment for their practical learning.

Table 1: System Comprehensive Overview

Course Content	Instructional Strategy
Canvas Discussions based on Motivational videos	To provide Motivation
Weekly Quizzes /Prelab Assignments	Knowledge test
Lab Activity	Demonstration and verification
Post Lab	Lab activity Validation
Mid-term and Final test	Test on Labs
Project	Problem solving, Project Management

The LPC55S69, an Arm Cortex®-M33 based microcontroller, is designed for embedded applications. It features up to 320 kB SRAM and 640 kB flash memory, supports high-speed and full-speed USB interfaces with crystal-less operation at full speed, and includes an SD/MMC/SDIO interface. The device is equipped with multiple timers i.e.general-purpose, SCTimer/PWM, RTC/alarm, Multi-Rate, and Watchdog, a high-speed SPI, eight versatile communication peripherals, and a 16-bit ADC with 1.0 Msamples/sec speed and temperature sensing capabilities.[7]

The Arm Cortex M33 features TrustZone technology for enhanced security, isolating valuable intellectual property and data. It also streamlines the design and software development of systems requiring digital signal control through built-in DSP instructions. Additionally, the LPC55S69 supports various security requirements, including HASH, AES, RSA, UUID, dynamic encryption and decryption, debug authentication, and TBSA compliance, further bolstering its capabilities in secure applications.[8]

The structure of the CANVAS modules, as depicted in Figure 5 are a result of applying Instructional Design methodologies. This approach begins with an engaging motivational video to introduce the specific module topic, followed by a discussion session for in-depth exploration and understanding. A quiz, drawing from the module notes or initial lab activities, assesses the students' comprehension up to this point. The centerpiece of the module is the hands-on lab activity, providing practical learning experiences. The module concludes with a post-lab report to reinforce and consolidate the learning objectives. This comprehensive and systematic design, rooted in Instructional Design principles, ensures a thorough engagement and assessment of students.

The course begins by emphasizing safety and awareness, focusing on lab equipment safety. It then delves into the evolution of computer architecture, highlighting the latest processor features, including the LPCXpresso55S69 board and MCUXpresso IDE v11.2.0. The curriculum covers an array of topics, such as an introduction to C programming, build systems, MCUXpresso IDE

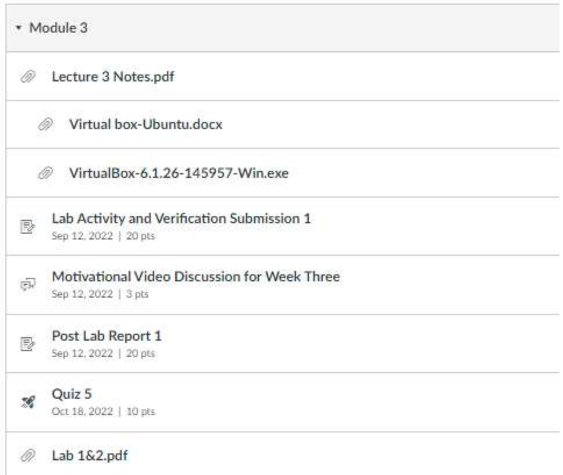
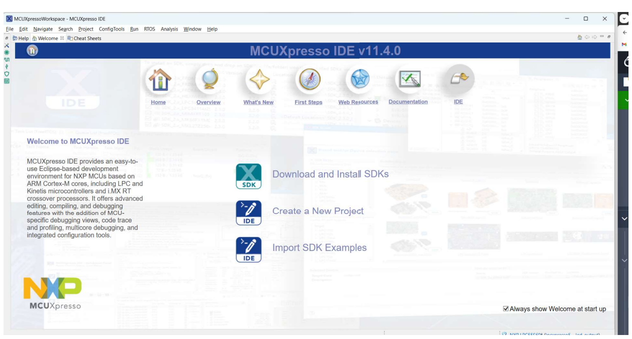
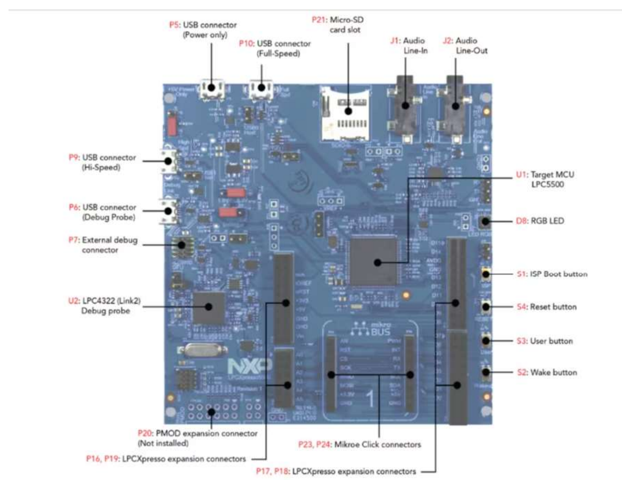


Figure 5: CANVAS Module Context

validation, ARM Architectures, and key hardware and software components like GPIO, ISR, Clock Configurations, ADC, UART, and I2C. Additionally, it integrates IoT applications with AWS services, and TrustZone provides a comprehensive learning experience.



(a) MCUXpresso IDE



(b) LPC55S69 Board

Figure 6: NXP Platform: MCUXpressoIDE and LPC55S69 Board

4.1 C Language for Embedded Systems

C programming language[9], is defined by 32 keywords and features structures for decision-making, including if-else and switch, as well as loops like while, for, and do-while, complemented by an exit mechanism through the break statement. Functions, which are collections of statements for specific tasks, utilize variables for data storage in named memory locations of the processor, classified as either local (function-specific) or global (accessible in any function). A C program is structured around a single main function. The compilation and execution process in the build system is facilitated by the MCUXpresso's toolchain, symbolized by a hammer icon for building and a bug icon for programming the board, as illustrated in Figure 7. Using platforms like GitHub or GitLab, version control is also taught, covering basics from configuration to branching, emphasizing the checkout and merging processes for final project development.

4.2 GPIO, Interrupts, ADC, UART, Timers, I2C

The Lab course, as detailed in our previous 2018 paper[10], encompasses a range of topics including General Purpose Input/Output(GPIO) for basic digital Input/Output (I/O) functions, Universal Asynchronous Receiver/Transmitter (UART) for serial communication, and Inter-Integrated Circuit (I2C) for multi-master, multi-slave communication between devices. Each of these components is essential to microcontroller-based systems. The course also delves into Analog-to-Digital Conversion (ADC), which is key for converting analog signals to digital signals, and interrupt service routines, which are key for handling asynchronous events. Figure 7 illustrates the ES-LAB project setup, showcasing the tools for pin, clock, and peripheral configuration, along with the Trusted Execution Environment (TEE). It also highlights the Software Development Kit (SDK) window, where students can create Board Support Package (BSP) code and build upon it to develop applications. This comprehensive coverage ensures a deep understanding of both hardware components and their integration in software development.

4.3 TrustZone

TrustZone serves as a sophisticated security protocol within embedded systems, comparable to establishing an exclusive enclave for sensitive intellectual property. Conceptualize this enclave as a VIP section within a club—the device in this analogy. Within this secured zone, critical data can mingle without fear of external interference, akin to the crown jewels enjoying a private party.

However, TrustZone is not exclusionary; it acts as a discerning bouncer, ensuring a controlled environment. This metaphorical club delineates boundaries between the secure VIP section and the non-secure areas. While the VIP zone safeguards intellectual property, the non-secure region accommodates everyday software activities without compromising security.

This course additionally offers a hands-on cybersecurity experiment experience, along with accessible laboratory knowledge in embedded systems. A key focus is on TrustZone technology, which safeguards intellectual property by segregating it in a secure zone while simultaneously allowing unsecured software to operate in a non-secure zone. Projects within the course typically involve interaction between secure and non-secure worlds, demonstrating practical applications of these concepts.

4.4 Internet of Things Project

IoT (Internet of Things) applications involve connecting physical devices, sensors, and actuators to the internet to collect and exchange data. These applications enable automation, remote monitoring, and control of various systems, leading to improved efficiency, convenience. In this project, the primary goal is to develop an Alexa skill utilizing AWS services. This involves creating a Lambda function and a ”thing” using the SDK on MCUXpresso. The objective is to establish a system where credentials can be loaded and applications can be downloaded seamlessly. The orchestration involves controlling the board LEDs based on voice signals received through Alexa. Integrating IoT devices with voice assistants like Alexa adds another layer of interaction and convenience for users. It allows users to control devices using natural language commands, which can enhance usability and accessibility.

Here, in ES-lab, accelerometer data refers to the measurements taken by an accelerometer sensor embedded within a microcontroller unit (MCU). An accelerometer is a sensor that measures acceleration forces, including the force of gravity, which allows it to determine the orientation of the device and detect its movement in three-dimensional space. Accelerometer data can be used to detect motion or changes in velocity. This could be leveraged to trigger events or updates in your system. For example, if the device is moved suddenly or experiences a significant change in acceleration, it could initiate a data upload process. The orchestration in this lab also includes updating accelerometer data.

To enhance the user experience, an Android application complements the setup. This application serves as a visualization tool, offering a graphical representation of the controlled LEDs and real-time accelerometer readings. The integration of voice-activated controls via Alexa and the synchronized visualization on the Android app underscores the multidimensional nature of this experiment, showcasing a cohesive and interactive IoT environment[11,12].

4.5 Agile MBSE

In the context of Agile MBSE[13], there has been a continuous effort to enhance the curriculum, as highlighted by Lee’s work on sustainability[14]. The curriculum evolves to meet the dynamic needs of the industry, a process facilitated by faculty members who actively engage in ongoing improvement initiatives and stay abreast of industry technologies through collaborative endeavors[15]. This commitment to continuous improvement aligns academic efforts with industry requirements.

The synergy between academia and industry becomes evident in the realization that academia plays a pivotal role in preparing students to address industry needs. The curriculum serves as a training ground, shaping individuals who are not only cognizant of current scenarios but also adept at tackling real-world engineering challenges. The strategic mapping of academic endeavors to real-world situations significantly contributes to students’ professional growth.

Moreover, the incorporation of experiential learning in engineering education, with a particular focus on integrating a Systems Engineering(SE) framework with the design of embedded systems, accentuates the importance of hands-on experiences in the learning journey following industry tools. This pedagogical approach involves designing courses that seamlessly blend

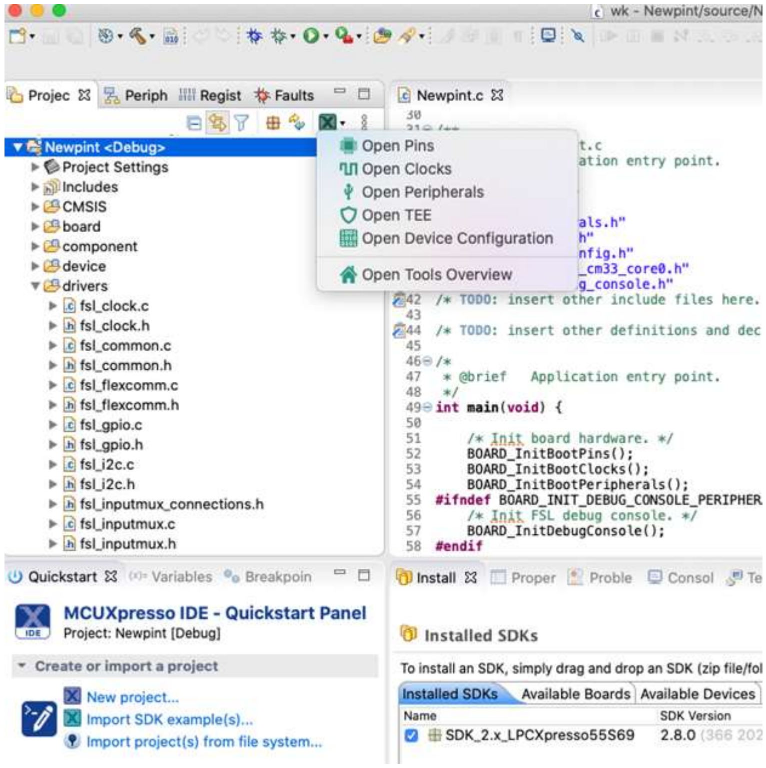


Figure 7: System Context

theoretical knowledge with practical applications. Immersing students via experiential learning achieves a profound comprehension of systems engineering concepts, preparing students to navigate the intricacies of complex engineering challenges driven by industry needs.

The inclusion of embedded systems in the curriculum further enriches the educational experience by providing a comprehensive understanding of both hardware and software aspects. This holistic approach ensures that graduates possess a well-rounded skill set poised for success in the dynamic landscape of engineering. The consolidation of theory and practice equips graduates with the necessary knowledge and skills to excel in their careers, mirroring the adaptability demanded by the ever-evolving field of engineering.

Integrating the Scrum Agile methodology in an embedded systems lab implementation involves forming cross-functional teams, prioritizing security tasks in the product backlog, and breaking down the project into iterative sprints. Daily stand-ups and sprint reviews facilitate communication and feedback on security-related tasks, while continuous integration and testing ensure that security features are thoroughly validated throughout the development process. Documentation and knowledge sharing help maintain a consistent focus on security considerations, resulting in the efficient delivery of secure embedded systems that can adapt to evolving threats.

Table 2: Technical Performance Measures for Lab Activity Grading Criterion

Criterion	Percentage of Grade
Demonstrating the experiment	50
Code Organizing	25
Behavior in Lab	25

Table 3: TPM for Post Lab Activity Grading Criterion

Criterion	Percentage of Grade
Structure and Organization	25
Formatting	25
Clarity and Presentation of Ideas	25
Quality of the Visualization using Figures, Tables	25

5. Course Verification

The assessment criteria for this course are delineated into three distinct components[16] following the IEEE standard on measurements[17,18]. At a macro level, the Measure of Effectiveness (MOE) is gauged, which essentially equates to the overall performance or the final grade attained. Moving more granularly, the Measure of Performance (MOP) encompasses assessments conducted during midterms, final exams, and the evaluation of final projects. For a more targeted examination of students' technical proficiency, Technical Performance Measures (TPMs) are employed, with a focus on each lab activity and the subsequent post-lab assessment. A detailed breakdown of the assessment rubrics for each of these measurement components is provided in the following sections. The measure of effectiveness of educational programs is gauged using metrics such as retention rate, completion rate, and grade point averages.

5.1 TPM and MOP

Table 2 outlines the rubric structure used for evaluating lab activities. It outlines the required criteria to successfully complete the laboratory tasks and present them to the teaching assistant or faculty member, including the ability to write clear C code and maintain appropriate discipline during the experiment or procedure. Additionally, the table presents the weighting of each grading criterion, clarifying how different aspects of the lab work contribute to the overall assessment. Table 3 outlines the rubric structure used for evaluating post-lab activities. This rubric focuses on several criteria, including the structure and format of the lab experiment, document clarity, presentation of configuration management, and the quality of visualizations.

Table 4 presents the rubric structure for evaluating midterm and final exams, which are split into theoretical and practical components. Additionally, this table details the weightings for each grading criterion, providing a clear understanding of how different sections of the exams contribute to the overall assessment.

Table 5 outlines the grading criteria for the final project, categorizing various aspects into two main groups: project content and presentation. The project content includes functional needs,

Table 4: MOP for Exams

Assignment	criterion 1	Percentage of Grade
Mid Term	Practical	60
Final Term	Theory	40

Table 5: MOP for Final Project Grading Criterion

Criterion	Percentage of Grade
Functional Needs (I)	5
Hardware and Software Needs (I)	5
Initial Structure and Behavior (II and III)	10
Requirements (II and III)	10
Engineering Verification(II and III)	20
Validation (IV)	10
Structure and Organization of Report(IV)	5
Formatting in Report (IV)	5
Clarity and Presentation of Ideas in Report(IV)	5
Quality of the Visualization using Figures, Tables in Report (IV)	5
Class Presentation(V)	2
Understanding Problem Statement from Presentation(V)	2
Requirements maturity from Presentation(V)	2
Block Diagram from Presentation(V)	4
Team Members skills and Contribution from Presentation(V)	2
Bill of Material from Presentation(V)	2
Test and Integration from Presentation(V)	6

hardware and software needs, initial structure and behavior, requirements, engineering verification, and validation, each with specific weightings. The report's structure, organization, formatting, clarity, and the quality of visualization are also assessed. Additionally, the presentation component evaluates understanding of the problem statement, requirements maturity, block diagram clarity, team member contributions, bill of materials, and test and integration, each contributing to the final grade. Table 5 also categorizes five assignments related to the final design project, each denoted by Roman numerals I to V. These assignments likely represent different stages or components of the final project, each contributing to the comprehensive evaluation of the student's work. Each assignment focuses on different aspects of the project, from conceptual design, development, verification, validation, and final presentation.

5.2 MOE: Retention Rate

The retention rate is a significant metric in educational settings, indicating the proportion of students who persist in their studies at an institution over a specific timeframe. It's a critical measure of effectiveness for an educational institution's success in maintaining student engagement and interest. Equation 1 provides a quantitative framework for calculating this rate. Figure 8 depicts the lab's measure of effectiveness in student retention, thereby offering a clear representation of how well the Lab course is performing in this key area.

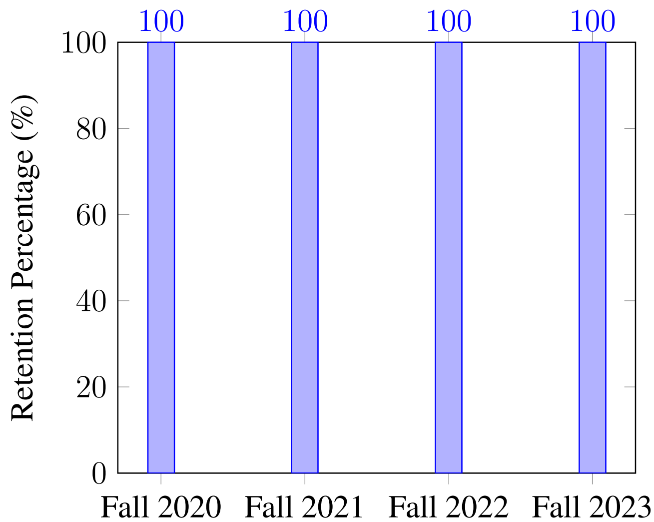


Figure 8: Course Retention Rate Per Semester

$$\text{Retention Rate} = \left(\frac{\text{No. of students that finish the course}}{\text{Total number of enrolled students}}\right) \times 100 \tag{1}$$

5.3 Completion Rate

The completion rate, a crucial measure of effectiveness shown in Equation 2, evaluates the proportion of students who successfully complete the microprocessors lab. This metric is particularly significant in assessing the course's impact and effectiveness. Historically, the completion rate has consistently exceeded 60 percent, a fact highlighted in the Bar chart 2, underscoring the course's efficacy in facilitating student success in this key area.

$$\text{Completion rate} = \left(\frac{\text{No. of students that pass the the course with A,B,C}}{\text{Total number of enrolled students}}\right) \times 100 \tag{2}$$

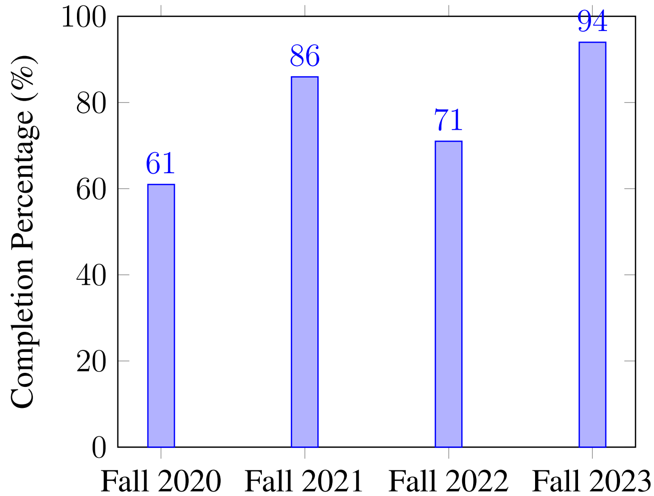


Figure 9: Course Completion Rate Per Semester

5.4 Grade Point Average

The grade point average (GPA) calculation, as detailed in Equation 3, involves the weighted sum of grades A, B, C, D, and F, assigned respective weights of 4, 3, 2, 1, and 1. Over four semesters, the course's GPA summary, depicted in Bar chart 3, consistently exceeded the 2.0 threshold. This demonstrates the effectiveness of the course in maintaining a standard of academic performance, with the weighted grade system providing a clear and quantifiable measure of student achievement.

$$\text{GPA} = \left(\frac{A \times \text{No.of A's} + B \times \text{No.of B's} + C \times \text{No.of C's} + D \times \text{No.of D's} + F \times \text{No. of F's}}{\text{Total number of students}}\right) \times 100 \quad (3)$$

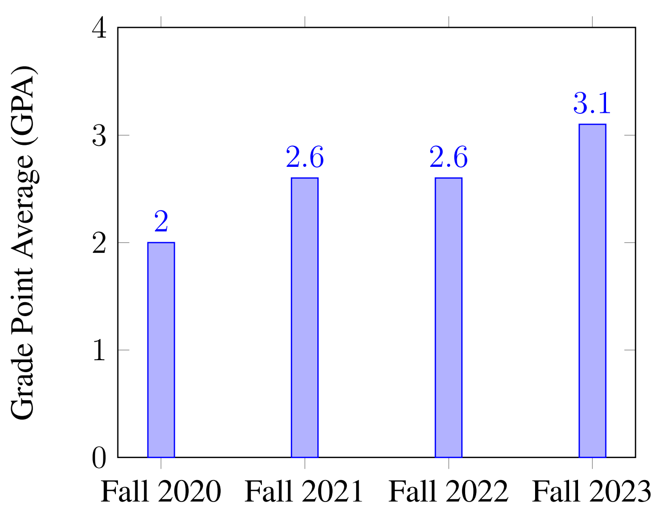


Figure 10: Course Grade Point Average Per Semester

6. Conclusion

In conclusion, the implementation of the microprocessors lab using the presented approach has been highly effective in equipping students with foundational knowledge in microprocessor technology. The hands-on experience with the NXP platform, integrated with the Model-based Systems Engineering approach, has not only allowed for facilitating the learning of fundamental embedded systems concepts but also bridged the gap between academic learning and industry trends. This practical approach fosters a deeper understanding of microprocessors, preparing students for industry-related roles by giving them a realistic perspective of real-world applications and engineering challenges. Active learning engagement via Experiential learning is crucial in developing the skills and ”systems thinking” mindset needed for success in the rapidly evolving field of technology.

References

[1] Ava McCartney. Gartner top 10 strategic technology trends for 2024. https://www.gartner.com/en/articles/gartner-top-10-strategic-technology-trends-for-2024. Accessed: January 29, 2024.

[2] Industry-university partnerships: A review of the literature. *Journal of Technology Transfer*, 45(5):1221–1242, 2020.

[3] Babu George and Ontario Wooden. Managing the strategic transformation of higher education through artificial intelligence. *Administrative Sciences*, 13(9):196, 2023.

[4] S Joshi and PJ Pramod. A collaborative metaverse based a-la-carte framework for tertiary education (co-mate). heliyon 9, e13424 (2023).

[5] Michael Max Bühler, Thorsten Jelinek, and Konrad Nübel. Training and preparing tomorrow's workforce for the fourth industrial revolution. *Education Sciences*, 12(11):782, 2022.

[6] Tiago Amorim, Andreas Vogelsang, Florian Pudlitz, Peter Gersing, and Jan Philipps. Strategies and best practices for model-based systems engineering adoption in embedded systems industry. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 203–212. IEEE, 2019.

[7] NXP Semiconductors. *UM11126 LPC55S6x/LPC55S2x/LPC552x User manual Rev. 2.4*. URL https://www.nxp.com/docs/en/user-guide/UM11126-DUMMY.pdf.

[8] NXP Semiconductors. *LPC55S6x 32-bit Arm Cortex®-M33; M33 coprocessor, TrustZone, PowerQuad, CASPER, 320 KB SRAM; 640 KB flash, USB HS, Flexcomm Interface, SDIO, 32-bit counter/timers, SCTimer/PWM, PLU, 16-bit 1.0 Msamples/sec ADC, Comparator, Temperature Sensor, AES, PUF, SHA, CRC, RNG*, 2022. URL https://www.nxp.com/docs/en/data-sheet/LPC55S6x.pdf. Rev. 2.4 — 8 December 2022.

[9] ISO/IEC 9899:2018 Programming languages - C. International Organization for Standardization, 2018. URL https://www.iso.org/standard/74528.html. Accessed: 02.01.2024.

[10] Prashanth Chetlur Adithya, Shraddha Pandey, Jose A Caballero, Özgür Yürür, and Wilfrido A Moreno. Systems engineering framework to design a laboratory course: A case study. 2018.

[11] NXP Employee Juan-Rodarte. Connecting the lpc55s69 to amazon web services, 2020. URL https://community.nxp.com/t5/NXP-Designs-Knowledge-Base/Connecting-the-LPC55S69-to-Amazon-Web-Services/ta-p/1114386. Last update: 10-27-2020 09:50 AM.

[12] NXP Community. Connecting the lpc55s69 to amazon web services, 2022. URL https://community.nxp.com/t5/NXP-Designs-Knowledge-Base/Connecting-the-LPC55S69-to-Amazon-Web-Services/ta-p/1114386. Accessed: 02.01.2024.

[13] Bruce Powel Douglass and Christian von Holst. *Agile Model-Based Systems Engineering Cookbook: Improve system development by applying proven recipes for effective agile systems engineering*. Packt Publishing Ltd, 2022.

[14] Jin Hee Lee and Dong Hoon Shin. Sustainability — free full-text — continuous improvement and optimization of curriculum system for engineering education accreditation: a questionnaire survey on achievement degrees of graduation requirements. *Sustainability*, 14(21):14554, 2022.

[15] Webster University. Continuous improvement of academic programs. In *2022 ASEE Annual Conference*, 2022.

[16] Garry J. Roedler and Cheryl Jones. Technical measurement: A collaborative project of psm, incose, and industry. Technical Report INCOSE-TP-2003-020-01, International Council on Systems Engineering (INCOSE), December 2005. Prepared by Lockheed Martin and US Army.

[17] IEEE Std. 1220-1998: IEEE Standard for Application and Management of the Systems Engineering Process, 1998. IEEE Std. 1220-1998.

[18] *IEEE Std 982: Standard Dictionary of Measures of the Software Aspects of Dependability*. Institute of Electrical and Electronics Engineers, New York, NY, USA, 1988. Accessed: [Your Access Date Here].