Journal of Engineering Education Transformations ,
Volume 38 , No. 3 , January 2025 , ISSN 2349-2473, eISSN 2394-1707

10.16920/jeet/2024/v38i3/24255

Experimental Based Learning: Microcontroller Simulation Using Virtual Simulation Machine

Mohamed Anees[1], Nagaraja.B.G.[2], Thimmaraja Yadava G[3], Chandrashekar M Patil[4]

[1,2,4]Department of E&CE, Vidyavardhaka College of Engineering, Mysuru, Karnataka, India
[3]Department of E&CE, Nitte Meenakshi Institute of Technology, Bengaluru, Karnataka, India
[1]anees@vvce.ac.in
[2]nagarajbg@gmail.com
[3]thimrajyadav@gmail.com
[4]patilcm@gmail.com

Abstract : Practical learning experiences continue to be essential for students' overall development in engineering education today. However, the construction of physical laboratories is frequently hampered by logistical issues, especially in educational institutions with low funding. This study introduces a novel method of experiential learning through the use of a Virtual Simulation Machine (VSM) to simulate microcontrollers. By utilizing the latest developments in virtualization technology, the suggested approach enables students to imitate microcontroller behavior without the need for actual hardware by establishing an engaging and dynamic platform. Through the simulation of real-world situations and the interaction with virtual elements, students can develop practical abilities in circuit design, system integration, and microcontroller programming. An educational framework is used to perform a series of experimental sessions that evaluate the effectiveness of this strategy in terms of information retention, student engagement, and learning outcomes. Preliminary results indicate significant improvements in student performance and interest in microcontroller projects, confirming the effectiveness of the VSM as a teaching tool. In conclusion, this article promotes the use of virtual simulation technologies in engineering curriculum to improve the usability and effectiveness of experiential learning. By providing students with the tools to experiment, investigate, and create in a virtual setting, teachers can develop a group of skilled and flexible engineers who are prepared to take on real-world problems in embedded systems and related fields.

Keywords : Experimental based learning, VSM Tool, Embedded Systems.

1. Introduction

Embedded System is a combination of hardware and software designed for specific applications and intended to work without human intervention (Zhang & Li, 2023). The major application areas of embedded systems are consumer electronics, aerospace, automotive industry, healthcare, industrial automation, communication systems and many more.

To develop the embedded code Integrated Development Environment (IDE) tools are used (Da Silva & Wagner, 2022a). It provides a comprehensive set of tools and features for the development and testing of code. Many IDEs offer evaluation versions that allow users to evaluate smaller code projects.

Mohamed Anees
Department of E&CE, Vidyavardhaka College of Engineering, Mysuru, Karnataka, India
anees@vvce.ac.in

Some of the features supported by the IDE are(Deepa et al., 2024) (Fezari & Al Dahoud, 2018):

a. Code Editing and Management

b. Compiler and Build Tools

c. Hardware Abstraction

d. Debugging and Testing

e. Emulation and Simulation

f. Deployment and Flashing

After completing the code, it is essential to evaluate it on the actual target hardware. However, obtaining the target hardware for every design iteration is often impractical, especially for educational purposes. Additionally, development boards provided by manufacturers may not encompass all the required peripherals, limiting their utility for learning purposes. In this scenario, VSM software (Roscoe et al., n.d.) provides a platform where designers can develop code, design hardware, and test code functionality seamlessly.

A "Virtual Simulation Machine (VSM) (Roscoe et al., n.d.)" for electronic design refers to a software-based environment that emulates the behavior of electronic circuits, components, and systems (Nagy et al., 2023). It enables engineers and designers to simulate the operation of electronic circuits without physically building them. This technology is widely used in electronics design and development to validate designs, evaluate functionalities, and identify potential issues before moving to the physical implementation phase. VSMs offer several benefits, including cost savings, faster design iterations, and the ability to explore different scenarios without the need for physical prototypes.

In a VSM, you can create and simulate circuits using various electronic components such as resistors, capacitors, transistors, microcontrollers, and more. The software provides tools for connecting components, setting parameters, applying signals, and observing the circuit's behavior through simulation(Metri et al., n.d.).

Some of the key features of VSM software are(Su & Wang, n.d.):

Circuit Simulation: Proteus allows users to simulate electronic circuits containing various components such as resistors, capacitors, inductors, transistors, microcontrollers, sensors, and more. The simulation accurately models the behavior of these components based on their real-world characteristics.

Mixed-Mode Simulation: It supports both analog and digital simulation, making it suitable for mixed-signal circuits that incorporate both analog and digital components.

Real-Time Interaction: Proteus offers interactive simulations where users can control inputs in real-time and observe the circuit's response. This is valuable for debugging and understanding circuit behavior under different conditions.

Virtual Instruments: The software provides a range of virtual instruments like oscilloscopes, function generators, logic analyzers, and more, allowing users to visualize and measure signals within the simulated circuit (Sinha & Dhanush, n.d.; Waller & Foster, 1996).

Microcontroller Simulation: Proteus includes support for simulating microcontrollers and microprocessors. Users can program and simulate the behavior of microcontroller-based systems, including code execution, I/O interactions, and peripherals (Okwu et al., 2013).

Custom Component Creation: Users can create custom components using the Proteus Component Library or by defining their own components using schematic symbols and simulation models.

Printed Circuit Board (PCB) Design Integration: Proteus seamlessly integrates circuit simulation with PCB design. Users can design PCB layouts based on their circuit schematics and verify their functionality through simulation (Okwu et al., 2013).

Interactive Debugging: Proteus provides debugging tools like breakpoints, single-step execution, and variable monitoring. These features help identify and rectify issues within the simulated circuit (Fezari & Al Dahoud, 2018).

Advanced Analysis Tools: Users can perform various analyses, such as transient analysis, frequency-domain analysis, and parameter sweeps to study circuit behavior under different conditions.

Educational Use: The software is used extensively in educational settings to teach electronics concepts and circuit design. Students can experiment with circuit behavior without the need for physical components (Balid et al., 2014).

Component Libraries: Proteus comes with a comprehensive library of pre-designed components, making it easier to construct circuits without starting from nothing (Su & Wang, n.d.).

The rest of the paper is structured as follows: Section II discusses the related work. Designing with virtual simulation machine is described in Section III. Section IV highlights the outcome of activity-based learning(Suchitra et al., 2022) using VSM tool. Finally, the work is concluded in Section V.

2. Related Work

Virtual prototyping platforms like Proteus (Wang et al., 2022) have garnered considerable interest due to their remarkable capacity to emulate both the hardware and software facets of microcontroller-based systems. By granting students the ability to immerse themselves in simulations that closely mirror real-world scenarios, these platforms facilitate a profound comprehension of the intricate interplay between hardware and software elements.

The attention of numerous researchers has been directed towards harnessing the potential of IDE tools tailored for microcontroller-based systems (446-Article Text-1714-1-10-20230718, n.d.; Da Silva & Wagner, 2022b; Lubiana et al., n.d.). These tools play a pivotal role in facilitating code development. However, a significant hurdle emerges when it comes to evaluating the software's compatibility with the hardware. In this context, the emergence of virtual simulation solutions has become invaluable. Scholars have embarked on explorations into the seamless integration of microcontroller-specific IDEs with virtual simulation environments. This amalgamation culminates in a holistic platform, empowering students to not only craft but also verify their embedded software comprehensively. This synergistic approach holds immense promise in bridging the chasm between theoretical learning and its tangible, real-world implementation.

Virtual simulation's scope is not confined to testing microcontroller software alone; it has broader applications. Research underscores its value in hardware design, creating a realm for students to explore component selection, circuit design, and testing. This even extends to providing students with a grasp of the practical details of PCB layout and assembly procedures. This innovative approach involves students in working with virtual components and assemblies, granting educators the ability to offer students a well-rounded journey through the intricacies of embedded system design, all within a supervised and streamlined framework.

Comparative analyses frequently juxtapose experiences with virtual simulations against those with conventional laboratory sessions, elucidating the merits and drawbacks inherent in each approach. The findings indicate that virtual simulations provide not only temporal and access flexibility but also amplify students' capacity to comprehend intricate ideas and navigate various experimental scenarios.

Furthermore, an emerging trend is the increasing inclination towards collaborative and distant learning facilitated by virtual simulation platforms. Research endeavors have delved into the untapped potential of these tools in enabling remote experimentation, offering students the ability to collaboratively engage in projects without geographical boundaries impeding their progress. This shift bears relevance not only in traditional classroom scenarios but also in the realm of distance education and online courses. Scholars are currently devoted to exploring methods that enhance collaboration and interaction dynamics within these virtual realms. Their aim is to ensure that the invaluable advantages of practical learning remain intact during the transition to digital platforms.

A wide range of studies that highlight the educational value, technical integration, efficacy, and collaborative aspects of these innovative tools characterize the landscape of experimental-based learning and microcontroller simulation using virtual simulation machines. In this article, we explore the effectiveness of microcontroller simulation as a tool for experimental-based learning. Furthermore, the study aims to determine how this approach facilitates realistic software testing within simulated environments and enhances students' understanding of microcontroller concepts.

The objectives of this paper include:

- To examine the role of experimental-based learning in enhancing students' comprehension of microcontroller concepts.

- To investigate the potential advantages of using simulation tools in terms of cost-effectiveness, accessibility, and scalability for educational purposes.

- To assess the effect of microcontroller simulation on students' ability to model, evaluate, and troubleshoot software components.

3. Designing With Virtual Simulation Machine

A. Steps involved in simulating microcontroller-based circuit in VSM.

Simulating a microcontroller-based circuit in Proteus involves several steps to set up the microcontroller, program it, design the circuit, and run the simulation. Fig.1 shows the steps involved in simulating microcontroller-based circuits in VSM.

1. Create a New Project: Open VSM and create a new project. You can choose the appropriate project type based on your requirements.

2. Add Microcontroller: In the schematic editor, add the microcontroller component to your workspace. You can find microcontrollers in the component library. Double-click the microcontroller to configure its properties, such as the model and package.

3. Program the Microcontroller: Depending on the microcontroller model, you may need to program it with the appropriate firmware. If the microcontroller supports simulation models, you can skip this step. Otherwise, you need to write the code in the microcontroller's programming language (e.g., C or assembly), compile it, and create a HEX file.

4. Add Power Supply and Clock: Connect power supply components (such as voltage sources and ground) to the microcontroller. Also, add a clock source if the microcontroller requires an external clock signal.

5. Connect Components: Connect to other components, such as LEDs, sensors, buttons, and any external peripherals that your microcontroller interacts with.

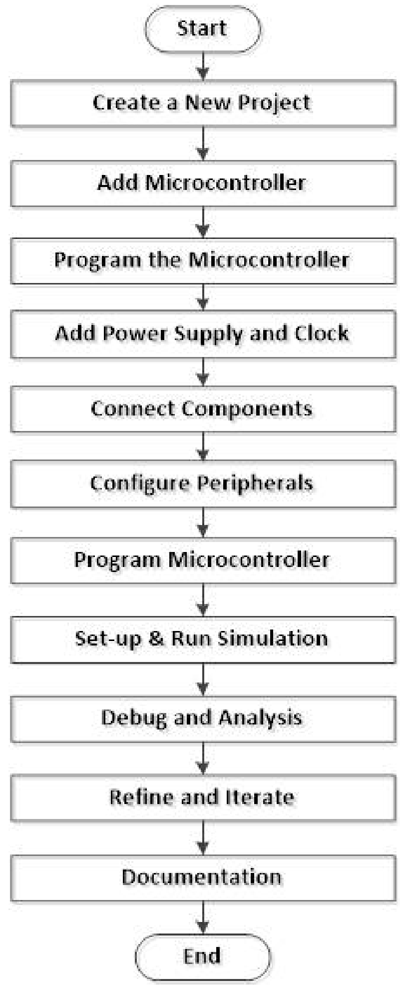

Fig.1 : Steps involved in simulating microcontroller-based circuit in VSM.

6. Configure Peripherals: If your microcontroller has integrated peripherals (like timers, ADCs, UARTs, etc.), configure their settings in the simulation.

7. Program Microcontroller (Simulated): If the microcontroller supports simulation models, you can program it within Proteus. Right-click on the microcontroller, select "Edit Properties," and use the provided editor to write your program code.

8. Set Up Simulation: Go to the "Debug" menu and choose "Run" to set up the simulation. Configure simulation settings, such as the simulation time, step size, and any initial conditions.

9. Run Simulation: Start the simulation to observe how your microcontroller-based circuit behaves. You can monitor signals using virtual instruments like oscilloscopes and logic analyzers.

10. Debug and Analysis: Use debugging tools provided by Proteus to analyze the behavior of your circuit. Set breakpoints, step through code execution, and monitor variables to identify any issues.

11. Review Simulation Results: After the simulation is complete, you can analyze simulation results to ensure that your circuit behaves as expected. Check signal waveforms, values of variables, and overall system behavior.

12. Refine and Iterate: If any issues or unexpected behaviors are identified, refine your circuit design and programming. Make necessary adjustments and iterate through the simulation process again.

13. Documentation: Finally, you can generate a simulation report that documents your circuit design, simulation settings, results, and analysis data. This documentation can be valuable for future references or sharing with collaborators.

B. Advantages of Virtual Simulation Machines (VSM) for Student Learning

VSM software(Deepa M et al., 2021) offers several valuable benefits to students in various fields of study, particularly those related to electronics, electrical engineering, and computer science. Here are some ways in which students can benefit from using VSM:

Learning Electronics Concepts: Proteus provides a hands-on platform for students to learn and experiment with electronic circuit design and simulation. Students can grasp fundamental concepts such as Ohm's law, Kirchhoff's laws, and component behaviors by building and simulating circuits.

Circuit Design and Analysis: Students can design, simulate, and analyze complex electronic circuits without the need for physical components. This allows them to understand how different components interact, how circuits respond to various inputs, and how to troubleshoot issues.(Prabowo & Irwanto, 2023)

Microcontroller Simulation: Proteus supports microcontroller simulation, enabling students to program and simulate the behavior of microcontrollers. This is essential for understanding embedded systems, firmware development, and how software interacts with hardware.

Debugging Skills: Students can practice debugging skills by identifying and rectifying issues within simulated circuits. They learn how to use debugging tools, interpret signal waveforms, and locate errors in code and circuit connections.

Experimentation and Exploration: Students can explore different circuit configurations and test hypotheses without the constraints of physical components. This fosters a creative and experimental approach to learning electronics.

Prototyping without Hardware: Proteus allows students to prototype and evaluate ideas without needing physical components or a physical breadboard. This is particularly useful for quick experimentation and conceptual design.

Digital Logic Design: For computer science and digital systems students, Proteus aids in learning digital logic design, sequential circuits, state machines, and FPGA-based systems.

Project Development: Students can use Proteus to design and simulate components of larger projects before moving to the physical implementation phase. This saves time and resources while ensuring a higher degree of reliability.(Kavitha et al., 2024)

Virtual Labs: Proteus can serve as a virtual lab environment, especially when access to physical laboratories is limited. It provides a safe platform for experimentation without the risk of damaging components.

Documentation and Reports: Students can generate simulation reports, documenting their circuit designs, simulation results, and analysis. This encourages structured thinking and helps with project documentation.

Preparation for Industry: Many industries use software like Proteus for design, testing, and simulation. Familiarity with such tools gives students a head starts when transitioning to professional roles (P. C. Nissimagouar et al., n.d.)(Aruna S. Nayak et al., n.d.).

Educational Projects: Teachers and professors can assign projects that involve designing and simulating circuits in Proteus. This encourages practical application of theoretical concepts and enhances student engagement.

Overall, VSM software offers students a practical and effective platform to bridge the gap between theoretical knowledge and real-world applications in electronics and related disciplines. It is a valuable tool for enhancing conceptual understanding, problem-solving skills, and preparing students for careers in engineering, electronics, and technology.(Deepa M et al., 2021)

C. Case Study – Interfacing LED and a Key to 8051 Microcontroller in VSM.

Problem Statement: Interface an LED and a key with an 8051 Microcontroller (Rakshith et al., 2022; Shanthi, 2021) so that when the key is pressed, the LED should turn ON.

Hardware Design: Table I shows a list of components used in the design along with their VSM part numbers. Pin assignment and the direction of port pins is shown in Table II.

Table 1: List Of Components Used.

Sl. No.	Component Name	VSM Part Number	Quantity
1	8051 Microcontroller	AT89C52	01
2	LEDs	LED-BIBY	01
3	Push Button	BUTTON	01
3	Resistor	RES	01

Table 2 : Port Pin Assignments.

Port Pin	Component Name	Direction
P2.0	LED	Output Pin
P2.1	Push Button	Input Pin

Design Calculations: LEDs are voltage-sensitive devices, meaning their brightness and performance are highly dependent on the current passing through them. If the current is too high, it can damage the LED. Therefore, current-limiting resistors are used to ensure that the LED operates within safe current limits.

The LED selected has forward voltage V_f=2.0 V and requires full drive current I_f=10 mA. Supply voltage is 5V. The value of current limiting resistance is calculated as below.

$$R = \frac{V - V_f}{I_f} = \frac{5 - 2}{10x10^{-3}} = 300\ \Omega$$

Choosing the nearest standard value, R=330 Ω. Power dissipation of the resistor

$$P = I_f^{\,2}\ x\ R = (10x10^{-3})^{\,2}\ x\ 330 = 0.33\ \text{Watts}$$

So, a 330Ω resistor with a power rating of at least 0.33W should be used in series with the LED to limit the current. Fig.2 shows the interfacing of key and switch with 8051 microcontroller.

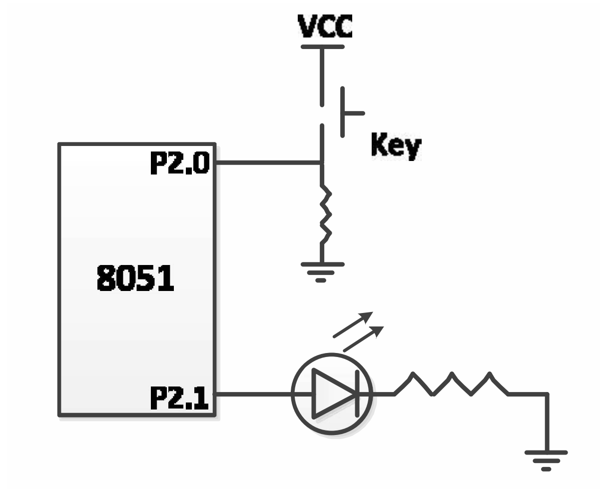

Fig. 2 : LED & Key Interfacing Circuit Diagram.

Fig. 3 shows the software flow of the application in VSM. C code for the application discussed is provided in appendix.

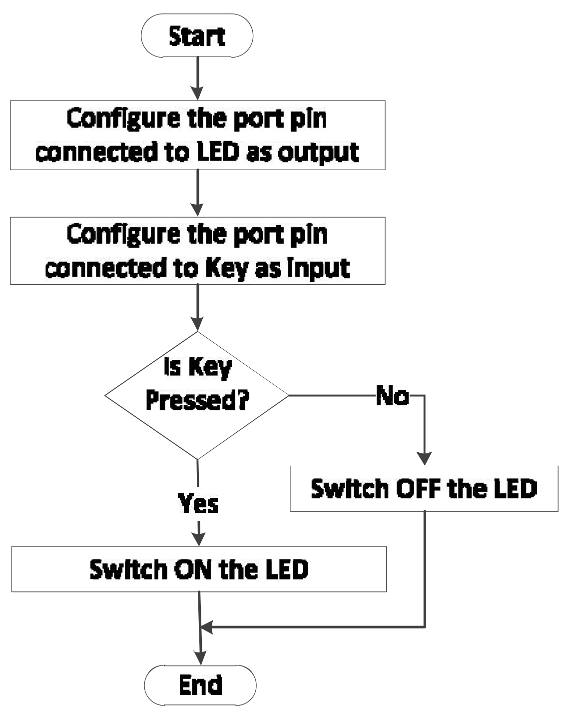

Fig. 3 : Flowchart – Software flow diagram.

Simulation Steps in VSM:

a. Project Creation

- Click on create project and provide a suitable name.
- Choose a desired destination path.
- Choose default template for schematic.
- Enable the option if you are planning to create PCB.
- Programmer has the option to choose from inbuilt compiler or compiler provided by external vendor.

b. Adding and connecting the components.

- Select component mode and add the components related to the design.
- Click on pick devices and search by generic name or part number.
- The component part number in mentioned intable 1.
- Perform the connection as per design.

c. Uploading the hex file

- Double click on the microcontroller component.
- Browse for the hex file and add it to the project.
- Set the clock frequency to 11.0592 MHz.

d. Run the project and analyze the output. Debug in case of any issues and rerun.

4. Outcome of activity based learning using VSM tool

A. Activity assignment

Outcome of the activity: Upon the completion of the activity, students are assessed for the following.

1. Ability to design hardware for 8051 microcontroller.
2. Ability to write firmware for 8051 microcontroller.
3. Ability to integrate hardware and software and show the output on VSM tool.
4. Ability to present their inference/conclusion in the form of report.

Evaluation Rubrics: Each student is evaluated for a total of 10 marks.

- Hardware Design – 3 Marks
- Software Design – 3 Marks
- Working prototype – 4 Marks

Assignment of Questions: A group of 3-4 students were provided with an activity to enhance the non-technical skills like communication, team management, ethics etc.

B. Sample questionnaire.

The activity was structured to encompass a broad range of commonly used peripherals.

Activity 1:

A unipolar stepper motor is interfaced with an 8051 Microcontroller. Two keys are connected to two I/O pins of 8051. Design hardware and software to rotate the stepper motor as below.

- If Key 1 is pressed, the stepper motor should rotate in a clockwise direction.
- If Key 2 is pressed, the stepper motor should rotate in an anti-clockwise direction.

Activity 2:

Eight LEDs are connected to P2. Two keys are connected to two I/O pins of 8051. Design hardware and software to implement the following.

- If Key 1 is pressed, implement an 8-bit Johnson counter on the LED’s connected to P2.
- If Key 2 is pressed, implement an 8-bit ring counter on the LED’s connected to P2.

Activity 3:

An LED is connected to P1.0. A 16x2 LCD is interfaced with an 8051 microcontroller. Design hardware and software to blink the LED continuously and display the status of LED on 16x2 LCD.

a) When LED is ON, call the LCDWRITE function to display “LED ON ”

b) When LED is OFF, call the LCDWRITE function to display “LED OFF”

Activity 4:

Two keys are connected to I/O pins of 8051. Design hardware and software to implement the following.

- If Key 1 is pressed, LCD should display “VVCE MYSURU”.

- If Key 2 is pressed, LCD should display “DEPT. OF E&CE”.

- If no key is pressed, LCD should display “**************”

Activity 5:

Two keys are connected to I/O pins of 8051. Design hardware and software to implement the following.

- If Key 1 is pressed, generate waveform of frequency 5 KHz with 50% duty cycle.

- If Key 2 is pressed, generate a waveform of frequency 10 KHz with 50% duty cycle.

Assume crystal frequency of 11.0592 MHz and Timer 0 in Mode 1 configuration.

Activity 6:

DAC0800 is interfaced with 8051 microcontroller. Design the hardware and software to generate the square waveform of 70% duty cycle.

Activity 7:

ADC0804 is interfaced with 8051 microcontroller. Design hardware and software to display the value of ADC on 16x2 LCD.

C. Course Outcome and Assessment

Outcome of the activity:

Develop firmware and hardware for ARM Cortex M3 and exhibit the operational functionality of the application through VSM.

Table 3 : Direct attainment / Target Levels:

Attainment Level	Attainment Level Value	Target
Zero	0	0% of students scoring >= 50% marks out of relevant maximum marks.
Low	1	50% students scoring >= 50% marks out of relevant maximum marks.
Medium	2	60% students scoring >= 50% marks out of relevant maximum marks.
High	3	70% students scoring >= 50% marks out of relevant maximum marks.

Table 4 : Course Attainment:

Threshold based Attainment	Attainment Level	Average based Attainment %
95.24%	3.0	74.76%

The course attainment was done using IonCudos software.

Threshold based Attainment:

This method calculates course attainment based on a threshold set by the instructor, which for this course was 50%.

For Threshold based Attainment % = $(x/y)\times 100$

x = Count of Students >= to Threshold %

y = Total number of Students Attempted

Average based Attainment:

This method calculates course attainment based on the average score obtained by the class.

For Average based Attainment % = $(x/y)\times 100$

x = Average Secured marks of Attempted Students

y = Maximum Marks.

In both methods, the course attainment level achieved was greater than 70%, resulting in an overall attainment level of 3.

D. Student Feedback on Activity

After the activity feedback was taken from the students. Around 40 students participated in the survey. The summary of the feedback is below.

Q1) How satisfied are you with your experience using EDA tool?

90% of the students were satisfied with the use of the tool.

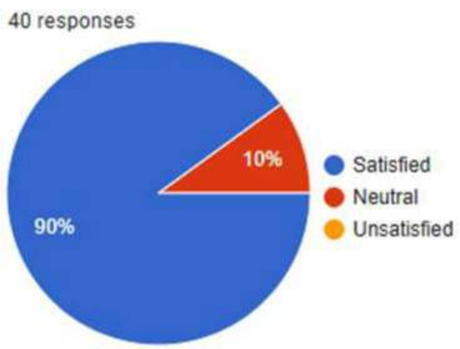

Fig. 4 : Feedback – Experience using EDA Tool

Q2) Has the tool enhanced your design capability?

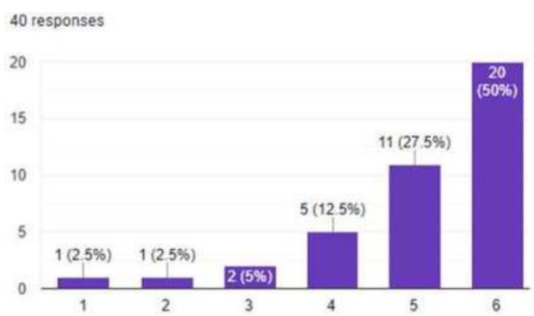

Fig. 5 : Feedback – Design Capability

Q3) How user-friendly do you find the interface of VSM?

90% of the students found the tool to be user friendly.

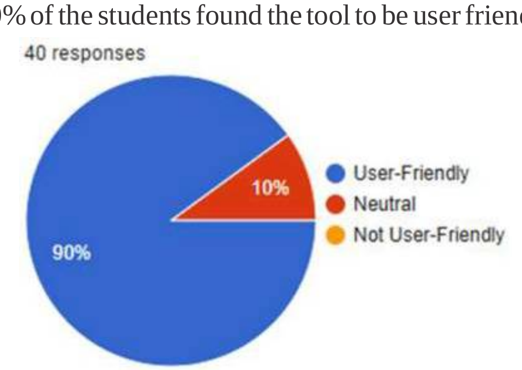

Fig. 6 : Feedback – VSM Interface

Q4) How would you rate the accuracy and realism of the simulations in VSM?

85% of students found the simulations to be accurate.

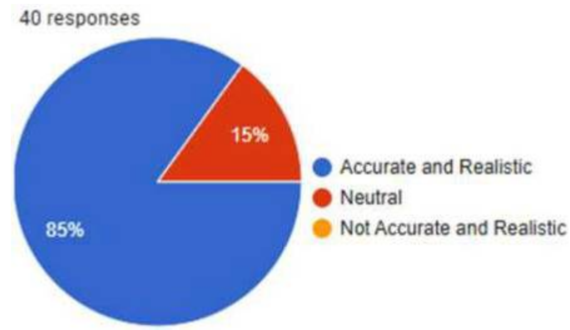

Fig. 8 : Feedback – Continuous Learning

Q5) Are you confident of designing and coding other microcontrollers in future using VSM?

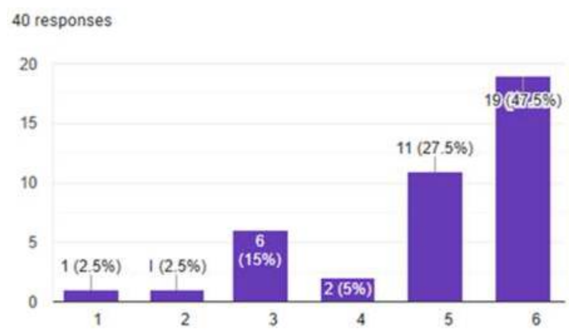

Fig. 7 : Feedback – Accuracy and Realism of Simulations

Conclusion

In conclusion, embedded product design has brought together the realms of embedded hardware and software, transforming the landscape of technology innovation. The intricate interaction of Integrated Development Environment (IDE) tools and hardware has traditionally been required for the development and testing of embedded software, often resulting in considerable time and financial expenditures. This paper explores Activity-Based Learning via Microcontroller Simulation Using a Virtual Simulation Machine, an alternative approach that promises to revolutionize the learning process.

Leveraging VSM for simulation provides a transformative avenue for exploring and comprehending the intricate world of electronics. This simulation platform serves as a virtual laboratory, empowering students to experiment, learn, and innovate without the limitations and risks often associated with traditional physical setups. The dynamic visualizations, real-world accuracy, and diverse components offered by VSM contribute to a holistic learning experience that nurtures practical skills, fosters creativity, and hones problem-solving abilities.

As technology continues to advance, the knowledge gained through VSM simulation equips students with a valuable toolset applicable not only in academia but also in the dynamic landscapes of research, innovation, and professional endeavors. VSM simulation stands as a conduit for bridging theoretical understanding with hands-on proficiency, preparing students for a future rich with possibilities in the realm of electronics and beyond.

Appendix a

```
#include <reg52.h>

sbit        KEY = P2^0;

sbit        LED = P2^1;

void main()

{

P2 = 0x01;

while(1)

{

    if(KEY == 1)

    LED = 1;

    else

    LED = 0;

    }

}
```

Refernces

446446-Article Text-1714-1-10-20230718. (n.d.).

Aruna S. Nayak, Namrata D. Hiremath, Umadevi F.M, Preeti B. Patil, & Satyadhyan Chickerur. (n.d.). Fostering Higher Cognitive Skills Through Industry Collaboration.

Balid, W., Abdulwahed, M., & Alrouh, I. (2014). Development of an educationally oriented open-source embedded systems laboratory kit: A hybrid hands-on and virtual experimentation approach. International Journal of Electrical Engineering and Education, 51(4), 340–353. https://doi.org/10.7227/IJEEE.0006

Da Silva, M. H. A. C. F., & Wagner, S. (2022a). Towards Immediate Feedback for Security Relevant Code in Development Environments. http://arxiv.org/abs/2207.03225

Da Silva, M. H. A. C. F., & Wagner, S. (2022b). Towards Immediate Feedback for Security Relevant Code in Development Environments. http://arxiv.org/abs/2207.03225

Deepa M, Reba P, Santhanamari G, & Susithra N. (2021). Enriched Blended Learning through Virtual Experience in Microprocessors and Microcontrollers Course. In Journal of Engineering Education Transformations (Vol. 34).

Deepa, M., Susithra N, Santhanamari G, & Reba P. (2024). Leveraging Agile Framework for a Project Based Learning Environment in Embedded Systems Design Course. In Journal of Engineering Education Transformations (Issue 37).

Fezari, M., & Al Dahoud, A. (2018). Integrated Development Environment "IDE" For Arduino Integrated Development Environment "IDE" For Arduino Introduction to Arduino IDE. https://www.researchgate.net/publication/328615543

Kavitha, Anitha D, & Manoharan, P. S. (2024). Project based learning and Proctored online assessment in Lab courses to enhance CDIO Skills-A case study of Microcontroller laboratory course. In Journal of Engineering Education Transformations (Issue 37).

Lubiana, T., Lopes, R., Medeiros, P., Silva, J. C., Nicolau, A., Goncalves, A., Maracaja-Coutinho, V., & Nakaya, H. I. (n.d.). Ten Quick

Tips for Harnessing the Power of ChatGPT/GPT-4 in Computational Biology.

Metri, R. A., Bharathpulavarthi, V., Srikanth, I., & Bhattar, C. L. (n.d.). Microcontroller Laboratory practices through project-based learning.

Nagy, M., Lăzăroiu, G., & Valaskova, K. (2023). Machine Intelligence and Autonomous Robotic Technologies in the Corporate Context of SMEs: Deep Learning and Virtual Simulation Algorithms, Cyber-Physical Production Networks, and Industry 4.0-Based Manufacturing Systems. Applied Sciences (Switzerland), 13(3). https://doi.org/10.3390/app13031681

Okwu, P. I., Mbonu, E. S., Ezekwe, C. G., & Ajuzie, C. U. (2013). Quick Prototyping of Real Time Monitoring System Using Proteus Development Tool: A Case for Self Diagnostic Traffic Light System). Quick Prototyping of Real Time Monitoring System Using Proteus Development Tool: A Case for Self Diagnostic Traffic Light System. In African Journal of Computing & ICT. Afr J. of Comp & ICTs (Vol. 6, Issue 5). www.ajocict.net

P. C. Nissimagouar, Gireesha H. M., Basawaraj, R. M. Shet, & Nalini C. Iyer. (n.d.). Enabling Industry skills in Automotive Embedded Systems Courses.

Prabowo, N. K., & Irwanto, I. (2023). The Implementation of Arduino Microcontroller Boards in Science: A Bibliometric Analysis from 2008 to 2022. Journal of Engineering Education Transformations, 37(2), 106–123. https://doi.org/10.16920/jeet/2023/v37i2/23154

Rakshith, P., Shankar, S., Gowtham, N., Savyasachi, G. K., & Avinash, R. (2022). Effective Implementation of Project based learning in Microcontroller Course. Journal of Engineering Education Transformations, 36(Special Issue 2), 308–312. https://doi.org/10.16920/jeet/2023/v36is2/23045

Roscoe, A. J., Yu, M., Dyśko, A., Booth, C., Ierna, R., Zhu, J., & Urdal, H. (n.d.). A VSM (Virtual Synchronous Machine) Convertor Control Model Suitable for RMS Studies for Resolving System Operator / Owner Challenges.

Shanthi, Dr. T. (2021). RFID based Attendance Monitoring System using 8051 Microcontroller. International Journal for Research in Applied Science and Engineering Technology, 9(4), 1042–1046. https://doi.org/10.22214/ijraset.2021.33824

Sinha, A. K., & Dhanush, T. S. (n.d.). Microcontroller Based Design of a Simulation System for Energy Harvesting Application. https://www.researchgate.net/publication/365614888

Su, B., & Wang, L. (n.d.). Application of Proteus Virtual System Modelling (VSM) in Teaching of Microcontroller.

Suchitra, M., Vivek D'Souza, A., Vinay, B. K., & Panchami, S. V. (2022). Activity based teaching learning in Engineering Education, a course on: Embedded Systems. Journal of Engineering Education Transformations, 36(Special Issue 2), 277–282. https://doi.org/10.16920/jeet/2023/v36is2/23040

Waller, J. C., & Foster, N. (1996). Training via the web: a virtual instrument. http://www.scimedia.com/

Wang, X., Wang, J., Weng, Z., Wei, Y., Han, D., & Gong, C. (2022). Reform Exploration of Proteus Virtual Simulation Practice Teaching in Electronic. Advances in Applied Sociology, 12(04), 93–101. https://doi.org/10.4236/aasoci.2022.124009

Zhang, Z., & Li, J. (2023). A Review of Artificial Intelligence in Embedded Systems. In Micromachines (Vol. 14, Issue 5). MDPI. https://doi.org/10.3390/mi14050897