Upy Trainer: IoT Training Tool Innovation Based on Raspberry Pi Pico W For TVET Education

Ku Mohammad Yusri Bin Ku Ibrahim[1], Syahrull Hi-Fi Syam bin Ahmad Jamil[2], Addzrull Hi-Fi Syam Bin Ahmad Jamil[3]
[1]Department of Electrical Engineering, Politeknik Tuanku Syed Sirajuddin.
[2]Department of Mathematics and Science, Politeknik Tuanku Syed Sirajuddin.
[3]Department of Electrical Engineering, Politeknik Seberang Perai.

kmyusri@ptss.edu.my

***Abstract*:** This study presents the development, implementation, and evaluation of the uPy Trainer a modular, MicroPython-based educational platform designed to enhance teaching and learning in the field of Internet of Things (IoT) and embedded systems. Developed using a systematic hardware design workflow with KiCad, the uPy Trainer integrates essential components such as Raspberry Pi Pico W (Pico W), LEDs, push buttons, variable resistors, LDR sensors, OLED displays, and buzzers to provide a comprehensive hands-on learning experience. Following fabrication, the system underwent rigorous functional testing to validate its reliability and operational performance. A usability assessment was conducted with 30 TVET Foundation students enrolled in the FB20094 course at Politeknik Tuanku Syed Sirajuddin, employing the Post-Study System Usability Questionnaire (PSSUQ). Results demonstrated high levels of user satisfaction and system acceptance across all measured dimensions. To evaluate learning effectiveness, a pre- and post-test assessment comprising 10 conceptual and 15 application-based objective questions was administered. The analysis revealed significant improvements in both conceptual understanding and practical skills, with mean scores increasing from 52.5% to 80.2% (conceptual) and from 48.0% to 78.9% (application), respectively. The findings underscore the uPy Trainer's effectiveness as a pedagogically sound, user-friendly and reliable instructional tool, capable of supporting both foundational and advanced IoT education. The integration of physical computing and project-based learning strategies within the platform was found to be instrumental in promoting student engagement and accelerating skill acquisition. This research offers valuable insights for educators seeking to modernize their IoT and embedded systems curriculum and demonstrates the practical benefits of structured, hardware-based educational innovation in technical and vocational education settings.

Keywords: *Raspberry Pi Pico, Raspberry Pi Pico W, Internet of Thing, Micropython, TVET trainer*

1.0 INTRODUCTION

The Fourth Industrial Revolution (Industry 4.0) has placed the Internet of Things (IoT) at the center of technological progress, accelerating changes across sectors such as manufacturing, agriculture, energy, and urban infrastructure (Hassebo & Tealab, 2023; Judijanto et al., 2024; Basavaraj & Patil, 2024; Shreyaskumar Patel, 2024). As a result, industries increasingly require workers who are proficient in automation, data exchange and digital literacy. This shift has intensified the responsibility of Technical and Vocational Education and Training (TVET) institutions to prepare students with relevant, practical IoT competencies.

Despite the urgency to develop these skills, recent statistical data reveals a persistent gap between industry expectations and graduate capabilities. For instance, 99.5% of Malaysian employers consider digital skills highly important, yet most TVET graduates are rated as only "average" or "poor" in key areas such as programming, problem-solving, and IoT literacy (Tee et al., 2024). Furthermore, Malaysia produced just 474,672 TVET graduates between 2016 and 2020, significantly below the

national target of 900,000, raising questions about the effectiveness of current digital skills education (Rodzalan et al., 2022). Many TVET students also lack practical experience in electronics and logical thinking, which are essential for real-world IoT applications (Techanamurthy et al., 2024).

To address these barriers, this project introduces the uPy Trainer—an interactive, modular learning platform based on the Raspberry Pi Pico W (Pico W) microcontroller. The main objectives of this study are as follows: (1) to develop a portable IoT trainer based on the Raspberry Pi Pico W that integrates at least six core modules: LED control, push button, potentiometer, light sensor (LDR), buzzer and OLED display, for hands-on experimentation; (2) to validate the functional reliability and usability of the uPy Trainer through systematic hardware testing and usability assessments; and (3) to evaluate the effectiveness of the trainer in improving students' conceptual understanding and practical skills using pre- and post-test measurements.

Designed specifically for TVET students, the uPy Trainer leverages official MicroPython support and standardized hardware to minimize wiring errors, reduce setup times and facilitate accessible, hands-on IoT experimentation (Loker, 2024; Plauska et al., 2023). Crucially, the uPy Trainer is grounded in evidence-based pedagogical methods, including physical computing, where students write code to interact with real hardware and project-based learning (PBL), a strategy in which educational activities revolve around the design, creation, and evaluation of tangible digital artifacts (Doropoulou et al., 2023). By integrating a range of fundamental electronic components including push buttons, LEDs, OLED displays, buzzers, light dependent resistors (LDR), variable resistors and level shifters, the trainer provides a comprehensive ecosystem that supports practical learning from beginner to advanced levels (Loker, 2024). Its unique advantages over platforms like Arduino and ESP32 include a consistent Python-based environment, reliable hardware-software integration and features tailored specifically for educational settings.

Transitioning to learning MicroPython with the Pico W further opens opportunities for students to explore contemporary applications in IoT, machine learning, robotics (ROS), and web system development. MicroPython's simple and easy-to-understand syntax accelerates the acquisition of foundational programming concepts and hardware integration (Ionescu & Enescu, 2020; Plauska et al., 2023). Moreover, a key advantage of MicroPython especially compared to traditional C/C++ workflows is its ability to upload and execute code dynamically via a serial interface or store scripts on microcontroller flash memory for automatic execution at boot (Plauska et al., 2023). This dynamic workflow enables rapid prototyping, lowers barriers to experimentation and fosters an interactive learning environment making it ideal for both classroom and self-directed study.

Furthermore, the Pico W benefits from official MicroPython support through collaboration between the Raspberry Pi Foundation and the core MicroPython developers. This collaboration ensures firmware stability, consistent updates, and comprehensive documentation accessible via the official Raspberry Pi and MicroPython websites (Loker, 2024). In contrast, for platforms such as ESP32 and Arduino, MicroPython is developed and maintained by third-party communities, without direct

involvement from original hardware manufacturers (Plauska et al., 2023). For example, MicroPython for ESP32 is actively maintained by the open MicroPython community, while support for Arduino is highly limited and mostly unofficial. This official support allows students to begin learning with confidence, without compatibility or technical barriers often faced on other platforms.

Additionally, the experience of learning MicroPython on the Pico W directly assists students in understanding and mastering Python, a leading language for AI, data science, machine learning, and web development (Nimbhorkar, 2024; Patil et al., 2024; Sharma et al., 2024). Early mastery of MicroPython simplifies the transition to using Python on full-featured systems such as the Raspberry Pi SBC, enabling development of more complex and industry-relevant applications.

In summary, the Raspberry Pi Pico W with official MicroPython support, serves as the foundation for the uPy Trainer, a platform designed to bridge the persistent gap between academic preparation and industry demands in the age of Industry 4.0, while meeting measurable objectives in both hardware integration and educational effectiveness.

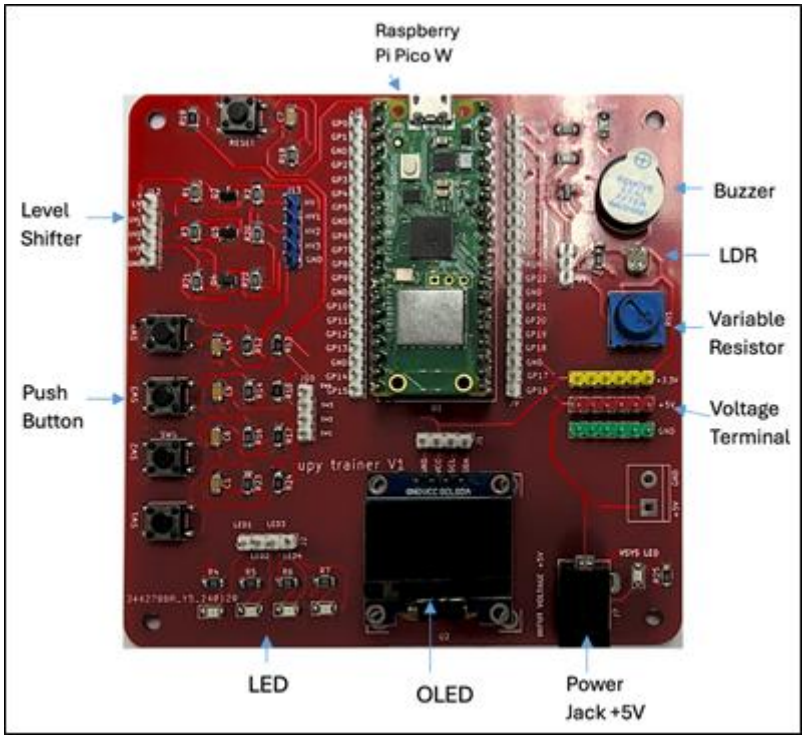

Figure 1: uPy Trainer kits.

2.0 LITERATURE REVIEW

The rapid digital transformation driven by Industry 4.0 and the expanding significance of the Internet of Things (IoT) have created a pressing need for hands-on, modular, and accessible learning platforms within Technical and Vocational Education and Training (TVET). Traditionally, microcontroller training has relied heavily on C/C++ with platforms such as Arduino. While powerful, these platforms can create barriers to rapid experimentation and classroom adoption due to their steeper

learning curves and time-consuming development processes (Ionescu & Enescu, 2020; Plauska et al., 2023).

Recent literature increasingly identifies the Raspberry Pi Pico W (Pico W) paired with MicroPython as an effective solution for both educational and IoT prototyping needs. For example, Loker, (2024) demonstrates that Pico W's Wi-Fi capability, clear pinout, and strong MicroPython compatibility enable a wide range of laboratory activities, from digital and analog interfacing to wireless data acquisition, all within an affordable and scalable package. The accessibility of MicroPython, particularly when used with beginner-friendly tools such as Thonny, is highlighted as a major advantage for introductory and project-based courses (Loker, 2024).

The effectiveness of these platforms is further enhanced when paired with active learning pedagogies. As Doropoulou et al. (2023) found, embedding physical computing and project-based learning (PBL) strategies into IoT and embedded systems courses enables students to directly engage with tangible hardware, collaboratively solve real-world problems, and retain knowledge through authentic project experiences. Specifically, their study implemented laboratory lessons that required students to design, build, and evaluate functional IoT devices, which significantly improved engagement and skill acquisition. Physical computing allows students to observe immediate, real-world effects of their code, while PBL fosters critical thinking and teamwork as students iterate through the process of designing, creating, and evaluating digital objects.

Ionescu & Enescu (2020) and Plauska et al. (2023) further validate the educational strengths of MicroPython. While their performance benchmarks confirm that MicroPython is slower than C/C++, both studies emphasize that MicroPython's high-level syntax, interactive REPL environment, and dynamic code execution make it especially suitable for beginners and for rapid prototyping (Ionescu & Enescu, 2020; Plauska et al., 2023). MicroPython enables immediate code upload and execution, supporting real-time learning and iterative experimentation, capabilities highlighted as crucial for effective TVET and IoT education.

From a broader perspective, the adoption of Python and its derivatives such as MicroPython is closely associated with key trends in modern computing. Multiple reviews underscore Python's position as the language of choice for artificial intelligence, data science, and IoT (Nimbhorkar, 2024; Patil et al., 2024; Sharma et al., 2024). These works consistently point to Python's readability, extensive library ecosystem, and widespread industry adoption as foundational strengths. As Sharma et al., (2024) note, Python's simplicity and cross-domain applicability make it an ideal choice for both beginners and experienced programmers. The literature thus supports the assertion that mastery of Python and MicroPython creates a seamless educational pathway from embedded systems to advanced AI and data applications, directly addressing the digital skills gap in TVET education.

Technical benchmarks also play a role. Longbottom (2021) confirms that, while C delivers higher raw performance on the Raspberry Pi Pico, Python is more accessible for beginners and better

supports educational outcomes through its simplicity and support for rapid development and debugging.

In conclusion, the reviewed literature strongly supports the adoption of MicroPython and the Pico W combined with active pedagogical strategies such as physical computing and PBL for hands-on IoT and AI education, as exemplified by the uPy Trainer platform. These technologies lower barriers to entry, accelerate project-based learning and facilitate the transition to industry-relevant programming skills, thereby directly addressing persistent gaps in TVET curricula.

3.0 METHODOLOGY

This study employed a systematic and multi-phase methodology to design, implement, and evaluate the uPy Trainer as an educational tool for IoT and embedded systems learning. The methodology comprised three key components. First, the development of the uPy Trainer was carried out using a rigorous electronic design workflow including breadboard prototyping, schematic and PCB layout with KiCad, professional fabrication, and comprehensive post-manufacture testing to ensure hardware robustness and educational suitability. This process is illustrated in Figure 2.

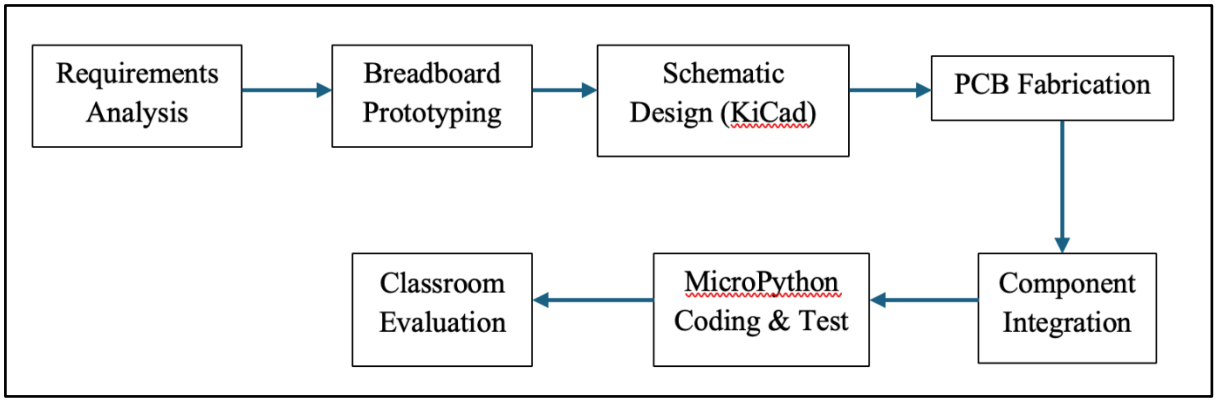

Figure 2: uPy Trainer Design and Implementation Process Flowchart.

The development workflow began with “Requirements Analysis”, establishing educational objectives and technical specifications based on curriculum needs and industry trends. Next, “Breadboard Prototyping” enabled the assembly and preliminary testing of key electronic components such as LEDs, push buttons, potentiometers, LDR sensors, buzzers, and OLED displays to verify functionality and compatibility. Upon successful prototyping, the “Schematic Design” phase used KiCad to formally capture and document the validated circuit configuration. This schematic was then translated into a physical printed circuit board during the “PCB Fabrication” stage through professional manufacturing services.

Following fabrication, “Component Integration” involved mounting all necessary hardware onto the PCB and conducting initial functionality checks. The workflow then proceeded to “MicroPython Coding & Testing”, where firmware was developed and systematically validated to ensure all modules could be effectively controlled and monitored. Finally, the completed uPy Trainer underwent “Classroom Evaluation”, where its usability and educational impact were assessed with real students using both usability questionnaires and pre-/post-tests.

In addition to the development workflow, a usability test was conducted in a real classroom context using a validated Post-Study System Usability Questionnaire (PSSUQ), enabling an in-depth assessment of students’ experiences and perceived system effectiveness. Furthermore, a pre- and post-test assessment strategy was implemented to objectively measure learning gains in both conceptual knowledge and practical application. By integrating structured hardware development, empirical usability testing, and rigorous learning outcome measurement, this methodology provides a reliable foundation for evaluating the pedagogical impact and practical value of the uPy Trainer in a TVET context.

3.1.1 Process of uPy Trainer Developments

The development of the uPy Trainer was systematically conducted using a structured hardware design approach involving initial circuit testing, schematic design and PCB fabrication, all carried out utilizing KiCad software. The project commenced with preliminary circuit testing conducted using a breadboard to validate each electronic component’s functionality and interoperability. The initial phase involved carefully wiring and testing components such as push buttons, LEDs, variable resistors (VR), level shifters, Light Dependent Resistors (LDR), OLED displays and buzzers. These tests ensured each component performed as expected and interacted seamlessly with the Raspberry Pi Pico (Pico) and Raspberry Pi Pico W (Pico W) microcontrollers. Detailed observations and results from this testing phase provided valuable insights and data used as references during the subsequent design stages.

Following successful breadboard testing, the schematic design was carried out using the KiCad Schematic software. KiCad was selected due to its user-friendly interface, extensive community support and capabilities in producing clear and precise electronic schematics. The schematic design involved placing standard symbols for each component, such as Pico W, push buttons, LEDs, VR, level shifter, LDR, OLED and buzzers, systematically onto the schematic sheet as shown in Figure 3. Each component was interconnected accurately according to the validated configurations from the breadboard tests. Clear labeling and annotations were also integrated into the schematic to facilitate easy readability and understanding. This comprehensive schematic served as a foundational reference for PCB layout design.

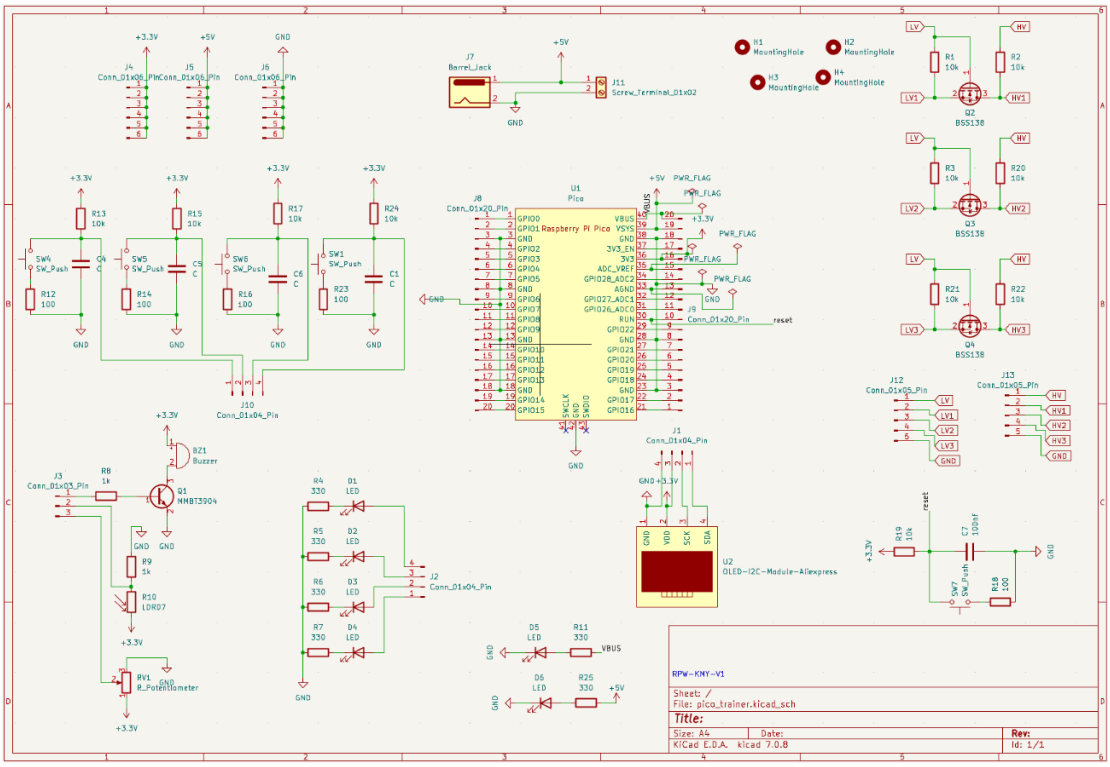

Figure 3: Schematic diagram of the uPy Trainer.

In the subsequent stage, the PCB layout design was accomplished using KiCad PCB (Figure 4). This phase prioritized ergonomic component placement, robust circuit operation, and signal integrity. Components were strategically placed on the PCB to enhance user accessibility and operational convenience. Routing traces were meticulously planned to minimize trace lengths, avoid unnecessary intersections, and reduce electrical noise. Voltage compatibility was ensured and ground planes incorporated to protect against electromagnetic interference. Additionally, built-in simulation tools in KiCad were employed to analyze signal integrity and identify potential design issues before PCB fabrication.

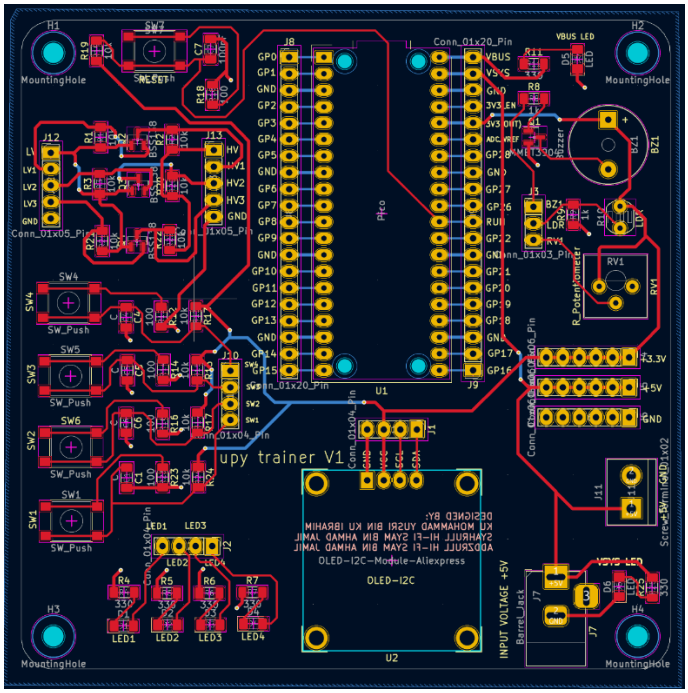

Figure 4: PCB layout of the uPy Trainer.

After finalizing the PCB layout, the design files were sent to professional manufacturers for fabrication. Upon receipt of the fabricated PCBs, comprehensive testing was again conducted to verify the complete functionality of the uPy Trainer. The rigorous testing procedures included validating component functionality, ensuring proper connectivity and assessing overall system reliability. These final tests confirmed that the PCB performed precisely according to design specifications, making it suitable for educational purposes.

Overall, the methodology adopted for developing the uPy Trainer adhered closely to professional electronic design workflows, beginning with rigorous component-level testing, advancing through detailed schematic and PCB design and culminating in thorough testing of the manufactured PCBs. This structured approach ensured that the final product was reliable, user-friendly and robust enough for practical use in Technical and Vocational Education and Training (TVET) institutions. Furthermore, this approach provided students with a valuable, hands-on learning experience, helping them understand the entire IoT system development process, from initial concept validation to final implementation, aligning well with the industry's current emphasis on practical, hands-on education.

3.1.2 Raspberry Pi Pico W Integration

The uPy Trainer was developed to accommodate two primary microcontrollers, the Pico and the Pico W. Although both boards share an identical form factor and pin layout, the Pico W offers integrated wireless functionality that renders it more suitable for IoT applications. The original Pico employs the RP2040 microcontroller, featuring a dual-core Arm Cortex-M0+ processor at 133 MHz,

264 KB of SRAM, 2 MB of flash memory, and up to 26 GPIO pins supporting PWM, UART, I2C and SPI interfaces (Raspberry Pi Ltd, 2020). In contrast, the Pico W retains all of these capabilities while incorporating an Infineon CYW43439 Wi-Fi module compatible with 2.4 GHz (802.11n) networks (Raspberry Pi Ltd, 2022), thereby enabling cloud connectivity, remote control and data logging in educational and prototyping contexts.

3.1.3 MicroPython Firmware Deployment

MicroPython firmware was flashed to the Pico W via the UF2 mechanism to establish a unified programming environment. Activation of the BOOTSEL button during USB connection exposed the board as a mass-storage device, to which the UF2 image was transferred. Following reboot, the Pico W executed under the MicroPython interpreter, facilitating direct execution of Python bytecode on the RP2040 core. Subsequent interfacing with the Thonny integrated development environment established a REPL session over serial, allowing real-time development and testing of scripts that exercised digital I/O, ADC conversion, PWM generation, I2C/SPI peripheral control and Wi-Fi initialization. This workflow supports rapid prototyping of both standalone and network-enabled applications.

3.1.4 Push Button and LED Module

The uPy Trainer incorporates push buttons and LEDs to demonstrate fundamental digital input and output operations. In standalone mode, actuation of a push button triggers a GPIO event on the Pico W, causing an LED to illuminate. In networked mode, button presses generate commands to cloud-based platforms (e.g. ThingSpeak, Blynk, Telegram bot) and the LED state is controlled remotely based on received signals. Through these exercises, students gain insight into signal polling versus event handling, network message formatting and feedback loops in IoT architectures.

3.1.5 Variable Resistor Interface

A variable resistor serves as a core analog input device within the uPy Trainer. Adjustment of its wiper position produces a proportional voltage which is sampled via the Pico’s ADC pin and digitized for microcontroller processing. Students implement remote level monitoring systems in which sampled resistance values are transmitted to cloud services or web dashboards. These analog readings can represent parameters such as motor speed, LED brightness or simulated temperature, illustrating integrated control logic driven by analog inputs.

3.1.6 Level Shifting

Level shifters are employed to ensure voltage compatibility when interfacing the 3.3 V Pico W with 5 V devices such as the Arduino Uno. Proper translation of signal levels protects hardware and maintains reliable communication. Without it, logic thresholds may not be met or components may incur damage.

3.1.7 Light-Dependent Resistor (LDR) Sensor

The uPy Trainer features an onboard LDR sensor to facilitate ambient light experiments without external wiring. LDR resistance changes are converted to voltage readings via the ADC and used to implement smart lighting systems. Sensor data may trigger automatic LED activation under low-light conditions or be streamed over Wi-Fi to real-time dashboards with threshold-based alerts. These projects reinforce concepts in analog signal acquisition, threshold detection and networked data visualization.

3.1.8 OLED Display Interface

An OLED display provides a compact user interface for real-time feedback. In humidity and temperature measurement projects using DHT11/DHT22 sensors, readings are shown on the OLED prior to or during cloud transmission. The display also supports status messages (e.g. “LED ON/OFF”), menu navigation, Wi-Fi connectivity indicators, IP address display and error alerts. Implementation exposes students to I2C programming, text and graphics rendering, and interface layout within constrained pixel dimensions.

3.1.9 Buzzer Module

A programmable buzzer enables audible alerts and user-feedback scenarios. In basic alarm systems, an LDR-based dark-condition detection triggers the buzzer. In user response applications, button presses produce short tones to confirm input reception. Students learn to generate variable-frequency sounds using PWM on digital output pins and to synchronize audio feedback with sensor inputs or network commands.

3.2 Usability Test

This study employed a case study approach to evaluate the effectiveness and usability of the uPy Trainer as a teaching and learning tool for the module FB20094 (TVET PROJECT) at Politeknik

Tuanku Syed Sirajuddin. The sample consisted of 30 semester two students enrolled in the TVET Foundation program, comprising 16 male and 14 female students.

For data collection purposes, a structured questionnaire was used as the main research instrument. The questionnaire was adapted from the Post-Study System Usability Questionnaire (PSSUQ), which is a standard tool for measuring user satisfaction and system usability across websites, software, systems, and products. Originally developed by IBM as the System Usability Metrics (SUMS) project in 1988, PSSUQ has since undergone several refinements, with the version used in this case study being the third iteration. The PSSUQ instrument consists of two sections: a demographic section with two items (gender and age), and a research question section with sixteen items. These sixteen items assess four key aspects: overall satisfaction (items 1 to 16), system usefulness (SYSUSE; items 1 to 6), information quality (INFOQUAL; items 7 to 12), and interface quality (INTERQUAL; items 13 to 15).

Upon completion of the IoT training sessions using the uPy Trainer, students completed the questionnaire to provide feedback on their learning experience.

3.3 Pre- and Post-Test Assessment

To rigorously evaluate the effectiveness of the uPy Trainer in facilitating student learning outcomes, a pre- and post-test methodology was implemented as part of this study's research design. This approach aimed to objectively measure both theoretical understanding and practical skills before and after the instructional intervention.

Prior to the commencement of the uPy Trainer module, all participating students were administered a pre-test designed to assess their baseline knowledge and competencies in key areas, including IoT fundamentals, MicroPython programming, embedded hardware concepts and system integration. The assessment consisted of 10 multiple-choice conceptual objective questions, which evaluated foundational theory and 15 application-based objective questions that required students to interpret code snippets, troubleshoot simulated circuit scenarios and solve practical IoT-related problems.

The results of the pre-test established a benchmark for each student's initial competency and informed subsequent instructional planning, enabling targeted emphasis on identified areas of weakness. Upon completion of the four-week instructional module which integrated hands-on exercises, project-based learning and physical computing activities, all students completed a post-test. The post-test was carefully constructed to mirror the pre-test in structure, coverage, and difficulty level, while using alternate questions and scenarios to minimize test-retest bias.

Both pre- and post-test results were collected, anonymized, and analyzed using descriptive statistics. The primary indicators included mean scores and percentage improvements for both conceptual and application domains. This comparative analysis enabled the quantification of

individual and group learning gains, providing robust evidence of the uPy Trainer's pedagogical impact. Additionally, item-level analysis was conducted to identify specific topics or skills with the greatest improvement.

All assessment procedures were conducted in accordance with institutional ethical guidelines. Student participation was voluntary and all responses were treated with strict confidentiality.

4.0 RESULTS

This section presents the outcomes of the uPy Trainer's implementation and evaluation, focusing on three key areas: hardware validation, usability, and student learning performance. First, the results of comprehensive functional testing are detailed to demonstrate the reliability and capability of the uPy Trainer's hardware components. Second, usability test findings are reported, reflecting student satisfaction and system effectiveness as measured by the PSSUQ instrument. Finally, the impact of the uPy Trainer on student learning is analyzed through pre- and post-test comparisons, providing quantitative evidence of conceptual and practical skill improvements following the instructional module. The results collectively highlight the uPy Trainer's potential as an effective educational platform for IoT and embedded systems learning in a TVET context.

4.1 Functional Testing and Hardware Validation

The uPy Trainer was designed in KiCad using its schematic capture and PCB layout modules. After fabrication, the main electronic components including LEDs, push buttons, buzzers, LDRs, and variable resistors were mounted and subjected to a systematic series of functional tests. The outcomes of these hardware validation procedures are summarized in Table 1.

The LED Blink test achieved a stable 1 Hz toggle rate, confirming precise timing control of the GPIO output and demonstrating reliable management of digital signals for visual feedback applications. For the push button, polling a GPIO pin and linking it to console output or LED control produced accurate and immediate state changes, validating the integrity of digital input capture.

The ADC readings for the variable resistor changed smoothly across the full input scale (0 to 65,535), indicating dependable analog sampling and precise mapping of voltage variations to digital values. Similarly, the LDR sensor showed a high reading of 42,000 in sunlight and a low value of 1,200 in darkness, confirming the sensor's effective detection range and sensitivity to ambient light.

The buzzer generated an audible 2 kHz tone for one second, verifying correct PWM signal generation and the system's ability to provide audio feedback. The OLED display successfully rendered the "Hello, uPy!" string and a basic graphic via I2C communication, confirming proper driver integration and reliable data transmission.

Further functional checks included an I2C bus scan that accurately identified device addresses, demonstrating bus integrity and connectivity. The UART echo test established full-duplex serial communication, with immediate echoing of input data. PWM servo control generated precise pulses, with the servo holding neutral and adjusting correctly in response to pulse width changes. Wi-Fi connectivity was also validated, as the board successfully connected to a known SSID and returned the correct network configuration.

Collectively, these results demonstrate that all core hardware modules of the uPy Trainer operate reliably and as intended. This comprehensive validation establishes the platform's readiness for educational deployment, supporting a wide range of IoT and embedded systems experiments.

Table 1

Functional tests and output results for the uPy Trainer board.

No	Test	Procedure	Output Results
1	LED Blink	Run a short script that toggles the LED on and off in 500 ms intervals.	LED visibly blinks at a 1 Hz rate.
2	Push Button	Write a loop that polls a GPIO pin tied to a button and prints its state or toggles an LED.	Console prints "Pressed" / "Released" or the LED changes state each time the button is actuated.
3	Variable Resistor (ADC)	Read from the ADC channel connected to the onboard potentiometer and print the value.	Numeric reading (0 - 65535) changes smoothly as the knob is turned.
4	LDR Sensor (ADC)	Sample the ADC pin tied to the onboard LDR under different lighting conditions.	ADC reading increases in bright light and decreases in darkness (42000 in sunlight , 1200 in dark).
5	Buzzer (PWM)	Output a 2 kHz PWM signal to the buzzer pin for 1s.	Audible 2 kHz tone for one second.
6	OLED Display	Send a "Hello, uPy!" string via I2C and draw a basic graphic.	"Hello, uPy!" appears on the screen and a simple shape is rendered.

7	I2C Bus Scan	Run an I2C scanner script over SDA/SCL.	List of detected device addresses (0x3C for the OLED).
8	UART Echo	Connect TX to RX on a UART port, send characters from a terminal.	Characters typed in the terminal are echoed back immediately.
9	PWM Servo Test	Generate a 1.5 ms pulse every 20 ms on a PWM pin connected to a servo.	Servo holds neutral (90 degree) adjusting pulse width moves the servo accordingly.
10	Wi-Fi Connectivity	Use network.WLAN to connect to a known SSID and print wlan.ifconfig()	Tuple with board's IP, subnet mask, gateway, and DNS Network configuration: IP address : 172.20.10.2 Subnet mask : 255.255.255.240 Gateway : 172.20.10.1 DNS server : 172.20.10.1

4.2 Usability Test Results

Figure 5 presents the findings from the survey responses regarding the acceptance of the uPy Trainer as a teaching aid for the FB20094 (TVET PROJECT) course. The Y-axis represents the average scores, while the X-axis corresponds to the four elements measured for product effectiveness. The results indicate that the OVERALL dimension achieved an average score of 1.10, while INTERQUAL (interface quality) recorded an average of 1.15. This was followed by INFOQUAL (information quality) with an average score of 1.20, and finally SYSUSE (system usefulness) at 1.16. These findings demonstrate that the uPy Trainer is highly suitable as an instructional tool for the FB20094 course. It is important to note that in the PSSUQ evaluation, lower average scores (closer to 1) reflect better performance and higher user satisfaction. Therefore, the consistently low scores across all dimensions provide strong evidence of the uPy Trainer's effectiveness and its positive reception among students.

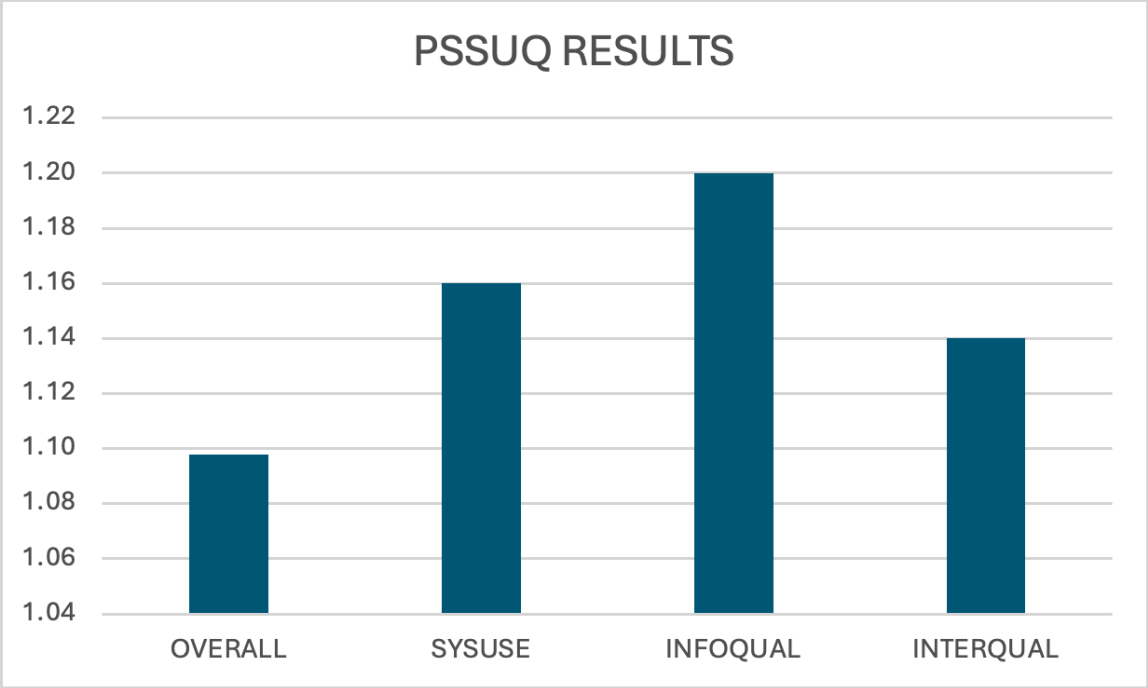

Figure 5: Survey findings using the questionnaire. The Y-axis represents the average score, and the X-axis represents the four elements used to measure product effectiveness.

4.3 Pre- and Post-Test Analysis of Student Performance

To assess the impact of the uPy Trainer on student learning, a pre- and post-test assessment was administered to all participants. The test consisted of 10 conceptual objective questions and 15 application-based objective questions, administered before and after the four-week instructional module. Results are summarized in Table 2.

Table 2

Comparison of Pre- and Post-Test Mean Scores for Conceptual and Application Objective Questions.

Question Categories	Pre-Test Mean (%)	Post-Test Mean (%)	Improvement
Conceptual	52.5	80.2	27.7
Application	48	78.9	30.9

Analysis revealed a significant improvement in both areas. The mean score for conceptual questions increased from 52.5% in the pre-test to 80.2% in the post-test, indicating a 27.7% gain in students' theoretical understanding of IoT, embedded systems and related concepts. Application-based questions showed an even larger improvement, with the mean score rising from 48.0% to 78.9%, representing a 30% increase. These findings suggest that the hands-on, project-based, and physical

computing strategies embedded in the uPy Trainer not only improved students' conceptual grasp but also substantially enhanced their practical problem-solving and technical application skills. Table 2 summarizes the marked gains in both domains, underscoring the effectiveness of the uPy Trainer as a pedagogically sound educational platform.

5.0 CONCLUSIONS AND RECOMMENDATIONS

This study demonstrates the effectiveness of the uPy Trainer as a modular, reliable, and user-friendly educational platform for IoT and embedded systems learning. Through rigorous hardware development, usability testing, and pre- and post-assessment, the uPy Trainer has been shown to significantly improve both conceptual understanding and practical application skills among TVET students. The integration of core modules such as LED control, push button, potentiometer, LDR, buzzer, and OLED display facilitates hands-on experimentation and supports diverse learning activities. The use of MicroPython, combined with structured, project-based learning in the classroom, has further enhanced the accessibility and depth of IoT education.

The uPy Trainer was successfully implemented as the primary platform for project-based coursework, where students applied their knowledge in real IoT projects. This experience demonstrated that the trainer is not only suitable for regular lessons, but also highly effective in guiding students through collaborative projects and practical assignments. Based on these findings, it is recommended that the uPy Trainer continue to be used in regular coursework, student-driven projects, and extracurricular competitions to further strengthen digital skills and foster creativity. Educators are encouraged to adapt and expand the provided modules to suit various skill levels and curricular needs. Moreover, engaging students in group and problem-based activities using the uPy Trainer can further help connect classroom theory to real-world technological applications.

While the uPy Trainer has demonstrated substantial educational value, several limitations should be acknowledged. First, the current study focused on semester two students in the Asasi TVET program at a single institution, which may limit the generalizability of the findings. Second, although the platform is designed for accessibility, its suitability for students with no prior experience in electronics or programming may require additional scaffolding and instructional support. Third, the evaluation period was limited to four weeks, which means that long-term retention of skills and the platform's impact on student performance in more advanced IoT or engineering courses were not assessed.

Future research should address these limitations by conducting longitudinal studies to evaluate the sustained impact of the uPy Trainer on student skills and career readiness. Expanding the sample size and including students from diverse backgrounds and educational levels will help determine the trainer's broader applicability. Furthermore, investigating the use of the uPy Trainer in conjunction

with advanced IoT modules, integration with cloud platforms, or interdisciplinary projects could further enhance its value and adaptability.

REFERENCES

Doropoulou, D., Angelopoulos, V., Amanatidis, P., Lagkas, T., & Karampatzakis, D. (2023). Teaching Embedded Systems and IoT at the University using MicroPython and Raspeberry Pi Pico. *ACM International Conference Proceeding Series*, 130–135. https://doi.org/10.1145/3635059.3635079

Hassebo, A., & Tealab, M. (2023). Global Models of Smart Cities and Potential IoT Applications: A Review. In *Internet of Things* (Vol. 4, Issue 3, pp. 366–411). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/iot4030017

Ionescu, V. M., & Enescu, F. M. (2020). Investigating the performance of MicroPython and C on ESP32 and STM32 microcontrollers. *2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging, SIITME 2020 - Conference Proceedings*, 234–237. https://doi.org/10.1109/SIITME50350.2020.9292199

Judijanto, L., Yulistiyono, A., & Febriyanto, W. (2024). Bibliometric Analysis on the Application of IoT in Smart Manufacturing System in Industry 4.0. In *West Science Interdisciplinary Studies* (Vol. 02, Issue 08).

Loker, D. R. (2024). *Data Acquisition Using the Raspberry Pi Pico W*.

Nimbhorkar, A. (2024). A STUDY ON THE APPLICATION OF AI IN PYTHON PROGRAMMING IN 21ST CENTURY. *Journal of Trends and Challenges in Artificial Intelligence*, *1*(4), 123–126. https://doi.org/10.61552/jai.2024.04.001

Patil, H., Mahandule, V., & Gunjal, A. (2024). *Python in the Evolution of AI: A Comparative Study of Emerging Technologies 1**. https://ssrn.com/abstract=5075929

Plauska, I., Liutkevičius, A., & Janavičiūtė, A. (2023). Performance Evaluation of C/C++, MicroPython, Rust and TinyGo Programming Languages on ESP32 Microcontroller. *Electronics (Switzerland)*, *12*(1). https://doi.org/10.3390/electronics12010143

Publication Basavaraj, S., & Patil, S. S. (2024). A Review of Agricultural IoT: Challenges, Applications & Future Research Direction. *Technology, and Social Sciences (IJMTS) A Refereed International Journal of Srinivas University*, *9*(4), 2581–6012. https://doi.org/10.5281/zenodo.13922569

Raspberry Pi Ltd. (2020). *Raspberry Pi Pico Datasheet*. https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf

Raspberry Pi Ltd. (2022). *Raspberry Pi Pico W Datasheet*. https://datasheets.raspberrypi.com/picow/pico-w-datasheet.pdf

Rodzalan, S. A., Noor, N. N. M., Abdullah, N. H., & Saat, M. M. (2022). TVET Skills Gap Analysis in Electrical and Electronic Industry: Perspectives from Academicians and Industry Players. *Journal of Technical Education and Training*, *14*(1), 158–177. https://doi.org/10.30880/jtet.2022.14.01.014

Sharma, Mrs. A., Choudhary, A., Himanshu, H., & Chaudhary, A. (2024). Exploring Python: A Comprehensive Guide for Data Science, Machine Learning, and IoT. *INTERANTIONAL JOURNAL OF SCIENTIFIC*

RESEARCH IN ENGINEERING AND MANAGEMENT, *08*(10), 1–3. https://doi.org/10.55041/IJSREM37816

Shreyaskumar Patel. (2024). IoT Applications in Healthcare and Industry: Current State, Challenges, and Future Perspectives. *International Journal of Advanced Research in Science, Communication and Technology*, 89–96. https://doi.org/10.48175/ijarsct-22614

Techanamurthy, U., Ahmad, F., Kahar, N., Salahuddin, A. R. P., & Ahmad, Z. (2024). Developing a Micro-Credential Curriculum in IoT for Malaysian Community Colleges: Key Topics and Projects. *Journal of Technical Education and Training*, *16*(2), 304–316. https://doi.org/10.30880/JTET.2024.16.02.027

Tee, P. K., Wong, L. C., Dada, M., Song, B. L., & Ng, C. P. (2024). Demand for digital skills, skill gaps and graduate employability: Evidence from employers in Malaysia. *F1000Research*, *13*. https://doi.org/10.12688/f1000research.148514.1