Available online at www.HighTechJournal.org

HighTech and Innovation Journal

ISSN: 2723-9535

Vol. 6, No. 4, December, 2025

Design of a Cost-Effective Educational Unmanned Ground Vehicle Platform with an Auxiliary Computer

Regillkent Esquivias [1], Isaac Anthony Ferrolino [1], Wrenn Cyruss Castro [1], Joshua Erick Lao [1], Nickolai Lim [1], Alvin Chua [1*]

[1] *Department of Mechanical Engineering, De La Salle University Manila, 2401 Taft Ave, Malate, Manila, 1004 Metro Manila, Philippines.*

Received 11 August 2025; Revised 22 November 2025; Accepted 26 November 2025; Published 01 December 2025

Abstract

This paper presents the development of a cost-effective, modular, and easy-to-assemble educational unmanned ground vehicle (UGV) system designed for hands-on robotics instruction for high school students. Its methodology incorporates frame redesign using CAD and 3D printing, software integration with DroneKit and Ardupilot, as well as the design of activity-based learning modules. Various performance evaluations, including incline testing, Aruco marker performance tests, and focus testing with students, highlighted successful system operation, system engagement, and learning improvements. The UGV could handle slopes of up to 25 degrees, and vision-guided marker tracking worked with precision. Student feedback was positive, with average Likert scale results of 4.63 for excitement and 4.42 for ease of use. Comparative surveys showed increased user satisfaction with the improved design, though wiring organization, GPS accuracy, and occasional snap-fit difficulties were noted for refinement. A two-tailed t-test showed no change in student interest after testing, but many indicated increased confidence if robotics were further offered in senior high school. The novelty and contribution of this study lie in the integration of a snap-fit 3D-printed modular frame, accessible hardware, autonomous capabilities, and curriculum-oriented learning modules, making robotics education more affordable, engaging, and practical for schools with limited resources.

Keywords: 3D-Printed; Camera Integration; Modular Design; Raspberry Pi; Snap-Fit Assembly; STEM Education.

1. Introduction

Robotics education has been an integral part of STEM development, as it introduces the basics of engineering through teaching simple mechanics and electronics. Through implementations of lectures, hands-on activities, and educational robots, gaps due to a lack of interest are minimized. This is already seen in the present with the increase in the implementation of robotics courses across learning institutions during the K-12 implementation. Palconit et al. [1] found that social interest has minimal influence on robotics development in the Philippines, whereas policy plays a decisive role by directly shaping educational implementation. Key suggestions include integrating fundamental concepts in the early stages of learning, investment in education, and research and development [2]. There are several schools that aim to incorporate robotics education using well-known robotics kits such as LEGO Mindstorm and VEX. However, such kits are commonly `too expensive to cover several batches of students as they are purchased abroad. Darmawansah et al. Mentions that other diverse robotics tools other than LEGO should be considered, and broader, more practical educational contexts should be pursued [3].

* Corresponding author: alvin.chua@dlsu.edu.ph

http://dx.doi.org/10.28991/HIJ-2025-06-04-03

UGVs are autonomous systems designed to navigate across land without onboard human operators. UGVs can come equipped with a variety of sensors and are able to navigate accurately in both indoors and outdoors environments. They are typically used in a wide range of applications, including disaster response, logistics, and monitoring, making them an effective medium for learning about robotics, automation, and STEM [4]. To support the potential for learning from the platform, UGVs must also provide educational modules that introduce students to programming, integrating sensors, and managing actuators. There is a clear need to develop an affordable, easy-to-build, and easy-to-use UGV to make robotics more accessible in common secondary and tertiary learning contexts. The platform should minimize uncertainties during the build phase so that students are less distracted by mechanical issues and bugs to allow students to focus on the robotics and programming aspects.

The paper discusses a new education UGV system aimed at making robotics easier and more affordable. In comparison to past iterations developed by Corpuz et al. [5], this new system employs a snap-fit modular frame that is 3D printed to allow easy assembly. A Raspberry Pi and camera are used to process data and allow for computer vision-based activities, respectively. DroneKit, a high-level Python library, is used to control the robot and allow students to develop autonomous behavior with scripts. While the combination of the Raspberry Pi and DroneKit itself is not novel, it enables students to easily program the UGV with straightforward Python syntax and relatively accessible hardware. Together with educational modules, these attributes improve the price, accessibility, and enjoyment of the platform for the student, while retaining useful benefits like outdoor navigation, sensors, and interactive programming. The contribution of this research lies primarily in its packaging and accessibility: bringing together the low-cost modular nature of the design, autonomous outdoor capabilities, and curriculum-oriented learning materials to make robotics education a practical option for schools with limited resources.

This introduction focuses on the growing interest in robotics, robotics kits, and delves into UGVs. The following parts of this article consist of Section 2, which presents the review of related literature, covering educational robotics globally and locally, robotics kits in the Philippines, and unmanned ground vehicles. Section 3 tackles the methodology, including the UGV frame redesign, component selection, software implementation, educational module development, and testing procedures for both hardware and software. Section 4 discusses the results and analysis of the final UGV design, incline testing, camera calibration, and software performance. Section 5 provides the conclusion. Section 6 presents the declarations, and finally, Section 7 lists the references.

2. Review of Related Literature

2.1. Educational Robotics

With advancements in engineering, computing, and automation, robotics has been a rapidly growing field since the 20th century. As such, industries have used automation and robotics to maintain or increase performance efficiency in various aspects (i.e., manufacturing, labor management, agriculture, and healthcare), and to reduce costs by removing labor expenses. The education industry adjusted to this trend by also integrating or expanding robotics, automation, artificial intelligence, and software development in curricula, especially in engineering and computer science tertiary programs. Robotics learning in STEM started in the early 1970s, when the LOGO Turtle was used and programmed to move and draw geometric shapes [6]. Eventually, educational robots were applied in universities and schools, teaching programming and engineering. A notable innovation in this concept is the integration of robotics with toys, which makes it more appealing, especially for children. Mitchel Resnick, who has been exploring toys with computers and machine learning since the 1980s, was able to develop the first "programmable brick" made possible with the sponsorship and collaboration with the LEGO company in 1985 [7]. Today, there are several educational robot platforms that are catered to different groups. VEX Robotics focuses on basic engineering principles and collaboration. Other robots such as MINDS-i Robotics emphasizes real-world problems and tasks and uses Arduino C++ to execute programming [8, 9]. Competitions are also conducted to challenge users of the platform to solve problems more analytically and think outside the box through collaboration. These competitions are held in various locations and promote robotics interest and inclusion in different areas. While research identifies robots as suitable for early educational exposure, their actual effectiveness remains unclear [10].

In the meta-analysis by Ouyang & Xu (2024) [11], comprising 21 robot-assisted STEM education studies from 2010 to 2022 with 2,433 participants, results showed that educational robotics generally had significantly positive and moderate effects on STEM education compared to instruction without robotics. Specifically, educational robotics had a moderate-to-large effect on improving students' learning performances and a moderate effect on influencing students' learning attitudes. The analysis also indicated that the effects of educational robotics showed no differences across educational levels (mostly primary school, followed by middle school, higher education, high school, and kindergarten), intervention durations (1 day to 1 month), types of robots (programming robots were mostly used, while social robots were rarely used), and interactive types (one-on-one or group activities). Supporting these findings, a study at Gumaca National High School in Quezon province, Philippines, reported that Grade 9 students used robotics to learn physics, and findings suggest that hands-on robotics activities enhanced understanding of abstract concepts, such as motion, speed, velocity, momentum, and collisions, by allowing students to visualize these concepts [12]. This approach was found to have improved comprehension and retention and enabled students to relate physics principles to real-life experiences. Robotics projects also promoted teamwork, problem-solving, creativity, and innovation as students designed, built, and programmed their robots. Combining these findings, it can be inferred that integrating robotics into STEM education, including physics, enhances learning experiences and fosters interest in STEM subjects.

Despite these stated benefits, robotics also introduces societal challenges such as the digital divide. One prominent issue is that the quality of devices and internet connections varies significantly per user, country, or area. The accessibility of resources is also crucial, as the quality of devices affects knowledge acquisition. In the meta-analysis of ten articles which were filtered per continent [13], several patterns were observed regarding how robotics classes were conducted in early grades. In Asia, a consistent three-phase methodology was used. The first step involves a theoretical introduction to robotics, computational thinking, and programming. Followed by firsthand familiarization with tools and robotics to ensure understanding and minimize mistakes. Finally, an evaluation of computational thinking was conducted through problem-solving tasks involving robotics programming. These interventions were done primarily in medium socioeconomic contexts. In America and Europe, due to various constraints, information on implementation is limited, but evidence indicates feasibility in medium- to high-socioeconomic contexts. Overall, meta-analysis suggests that educational robotics is easier to implement in wealthier schools or communities [13]. While it can significantly boost computational thinking in students, the meta-analysis of Ouyang & Xu (2024) found a minor and insignificant effect on computational thinking, highlighting minor inconsistencies in reported outcomes [11].

In the Philippines, robotics programs are often integrated into its educational system, particularly across secondary, selected programs in tertiary levels, and as an elective in primary levels. Depending on the availability of trained teachers to teach the subject and the availability of kits, robotics is taught as a lecture or a laboratory class. Often, educational institutions can only afford a limited selection of robotics kits and provide compensation for a small number of accredited teachers. As such, the quality of education among institutions in the country is not guaranteed to be consistent, with the bare minimum of teaching to ensure students understand the general application of programming and robotics. There are some notable programs which have been implemented to assist with the challenges in the field, including the RoboTeach Extension Project, which served as a week-long workshop for teachers to understand basic concepts and applications of robotics, and the school-to-school programs of the Philippine Robotics Academy that offer LEGO NXT Base sets, computer laptops, teacher's manual, student textbooks, quarterly teacher's training, and certificates and prizes for outstanding projects [9, 14]. Others may be in the form of sponsorships from external organizations, such as the intervention of the Teacher Training Program by the Japanese Government for international teachers, which taught robotics by letting trained teachers use available materials for robot learning. In Camarin High School, which was composed of 347 staff members and 9923 students divided into 224 sections, seminars for local teachers were conducted, discussing software programming, electronics, hardware, mechanisms, and teaching techniques.

The robotics subject was employed through the introduction of Arduino, an open-source electronics platform, for understanding the microcontroller, and the use of Tinkercad simulations before physically implementing designs [15]. In the same direction, Ianthe Christian Academy in Olongapo City, Zambales, has invested in integrating robotics into its curriculum to equip students with essential skills while keeping learning enjoyable and engaging. A study by Ahillon Jr. et al. (2025) [16] evaluating the robotics class highlighted several key findings. First, by equipping teaching staff with a strong understanding of robotics fundamentals, they were able to communicate concepts effectively and create engaging lessons that sustained student interest. The study also reported notable improvements in students’ problem-solving skills, logical reasoning, and confidence in tackling technological challenges due to early exposure to STEM through robotics. Students demonstrated increased enthusiasm for learning, asked thoughtful questions, and actively participated in discussions. Their firsthand experience with robotics also provided technical advantages and enhanced curiosity. Student responses indicated satisfaction with the overall learning environment, tools, and support provided. However, the study also noted that some parents did not fully recognize the value of robotics compared to traditional subjects. Reasons behind this could be unfamiliarity with its benefits or the lack of visible impact on other academic areas. Most importantly, the study highlights robotics as an important part of modern education.

2.2. Robotics Kits in the Philippines

The Philippines is slowly coming to terms with the technological developments that characterize Industry 4.0. As Pangandaman et al. [17] noted, education is integral to how modern technology will be embraced, with robotics being one of the tools necessary for youth empowerment. Despite reforms like the K-12 program, Philippine robotics education has not become mainstream due to a lack of access to properly equipped laboratories and the availability of funding for robotics education. To overcome these barriers and limitations, many robotics kits have been implemented in schools in the Philippines.

Note. Image from LEGO® MINDSTORMS® EV3 31313, by LEGO, n.d., LEGO. (https://www.lego.com/en-us/product/lego-mindstorms-ev3-31313) [18].

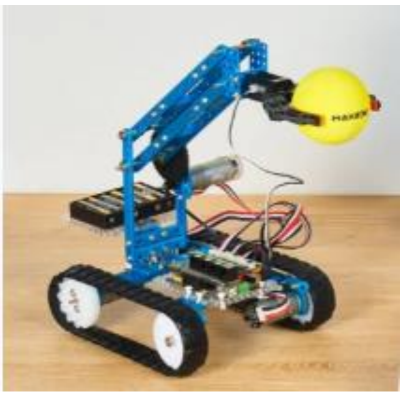
Figure 1. LEGO Mindstorms EV3 Inventor Robotics Kit

The LEGO Mindstorms EV3 kit, shown in Figure 1, is currently one of the most popular platforms worldwide and in the Philippines. The kit encourages modularity and a block-based programming environment for beginners while providing intermediate and advanced coding opportunities. Being friendly for novices is the reason the EV3 kit is utilized in school and university-based robotics clubs.

Note. Image from UGOT Robot-01, by PHR Robotics, n.d., PHR Robotics. (https://www.phr-robotics.com/en/robot-ugot-phr-robotics) [19].

Figure 2. UGOT Robot-01

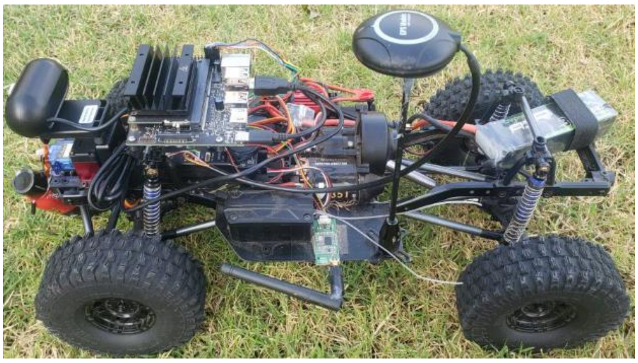
Figure 2 shows the UGOT Robot-01, an advanced kit available through PHR Robotics. Compared to LEGO, UGOT seeks to move forward into using real-world projects as it provides sensors and is also Arduino and Python programmable, making it one of the most flexible kits for secondary and tertiary learners.



Note. Image from mBot Ultimate 10-in-1 Robot Kit, by Makeblock, n.d., Makeblock. (https://www.makeblock.com/pages/mbot-ultimate-robotics-kit) [20].

Figure 3. Makeblock mBot Ultimate Robotics Kit

Another option, the Makeblock mBot Ultimate Robotics Kit, as shown in Figure 3, offers a 10-in-1 building experience for kids, promoting creativity and problem-solving. The use of metal components and Arduino-based programming is geared towards students, exploring the physical interaction with hardware in a more comprehensive way than the plastic nature of kits based on LEGO.



Note. Image from Smart Rover Kit, by Drone Dojo, n.d., Drone Dojo. (https://dojofordrones.com/product/rover-kit/) [21].

Figure 4. Drone Dojo Smart Rover Kit

Finally, the Drone Dojo Smart Rover Kit shown in Figure 4 introduces rover-based robotics, linking the physical nature of ground mobility outdoors with the programming exercises. The rover makes use of DroneKit and a companion computer, either a Raspberry Pi or a Jetson Nano. Open-source design means one can apply it to whatever flexibility is desired in the classroom. Because of the complexity, this is a particularly costly product, which would limit the number of kits purchased by schools and pay for public education.

Looking forward, one promising development is Shape Robotics, a Danish educational technology company. As of 2025, Radenta Technologies, a prominent solutions integrator in the Philippines, expanded its partnership with Shape Robotics, intending to support and boost robotics-based learning in the country [22]. The Fable is a modular robotics kit suited for classroom use and enables students to build and program their own robots. It is equipped with beginner and advanced programming and uses visual coding tools like Blockly and text-based languages such as Python. It's hardware includes servo motors, sensors, and an AI-supported camera, making it compatible with platforms like Raspberry Pi and Arduino. This kit is known for its capabilities, with over 12,000 schools globally using it to teach STEM through technology, signaling growth in the Philippines as it is becoming more prominent in academic institutions through Radenta's outreach.

Aside from these robotic kits, robotics workshops are also gaining momentum in the Philippines. One example is Nullspace Robotics PH, the country's first learning center focused on coding and robotics. Its first branch opened in 2023, and a second branch followed in 2024. Nullspace Robotics PH aims to introduce children to engineering and programming concepts through hands-on learning. Students are to build and program robots, create games, and work on electronic projects, providing a fun and interactive learning environment. Nullspace Robotics PH also aims to strengthen STEM education in primary schools by conducting pilot programs and teacher training [23]. Some of the kits they use include the LEGO Education SPIKE Prime and SPIKE Essentials, which help develop robotics proficiency using sensors and automation. In addition to robotics, they offer coding workshops such as the Minecraft Series: Coding Workshops, where children explore Minecraft Education through block-based coding, text commands, and other fundamental programming concepts. These activities help build creativity, problem-solving, and computational thinking skills. Their programs are fee-based, with 5- or 10-session packages and depending on age level. Other workshops are also emerging. For example, Bytelift offers structured robotics programs for students from Grades 1 to 10, covering introductory robotics, hands-on assembly, coding basics, advanced robotics concepts, AI integration, and real-world applications. Bytelift hosts robotics workshops across the Philippines and uses educational kits such as the mBot2 and the mBot2 Smart World 3-in-1 Add-on Pack.

Low-cost options and open-source platforms are crucial and integral in robotics education, and finally in STEM development, as access to commercially available practical and learning materials are generally costly. This is apart from other factors that affect STEM interest such as its perceived complexity. Providing more accessible educational technology would invite more people into the discipline, which can create opportunities for further innovation in the future. While these platforms are powerful, their cost, proprietary ecosystems, and limited local availability restrict the number of units schools can purchase, which limits hands-on time spent per student. In contrast, open and low-cost systems based on 3D-printed frames, off-the-shelf electronics, and open-source software can be replicated at a fraction of the price of kits like LEGO Mindstorms or VEX, while still supporting core STEM outcomes such as sensor integration, autonomous navigation, and structured problem-solving. This study positions the X-Lakbay UGV within this space by evaluating whether a modular, snap-fit, 3D-printed rover backed by Python-based activities can deliver learning experiences comparable to those of proprietary kits, but at significantly lower cost and with greater hardware flexibility.

2.3. Unmanned Ground Vehicles (UGVs)

As parts and components of a system become more complex, brought by restrictions and standards, various manufacturing processes can be considered. 3D printing has emerged as a practical option for both prototyping and production, offering flexibility, reduced material waste, and lower costs. It allows for easy customization of parts through CAD design, making it ideal for developing modular components. In the field of robotics, including UGVs, the use of 3D printing is growing. Several robotics systems have successfully integrated 3D printing. For example, Aurora Flight Sciences developed an Unmanned Aerial Vehicle (UAV) using Fused Deposition Modeling (FDM) with ULTEM 9085 resin, producing a jet-powered aircraft composed of 80% 3D-printed parts. The use of lightweight, high-performance thermoplastics reduced build time by 50% while maintaining aerospace-grade durability [24]. Another example is the Yonder Deep student organization, which used 3D printing to build a low-cost, fully autonomous underwater vehicle (AUV) for ocean data collection. An entry-level AUV used for arctic environments typically costs somewhere in the range of $300,000. However, by using 3D printing technology, Yonder Deep was able to create a full-scale AUV prototype at a cost of around $3,000, consuming about 2 weeks of printing time [25]. Another notable example is the AUV developed by Yang et al. [26], known as the Advanced Robotic Marine Systems (ARMs) 1.0. It is a modular AUV designed for educational use, made from PLA filament. Each of its four 3D-printed sections can be easily removed or replaced, allowing students to reconfigure the system and experiment with different sensor setups. With the mentioned robotics systems, it can be gathered that 3D printing is not yet widely adopted in UGV development. However, it is steadily gaining traction due to its lightweight, durable, and customizable nature, offering advantages including reduced energy consumption, lower costs, and improved modularity.

The integration of companion computers into unmanned systems significantly enhances their functionality, particularly in applications that require advanced data processing and sensor integration. One of the most widely used companion computers is the Raspberry Pi (RPi), a low-cost, relatively powerful single-board computer initially developed to promote ICT education [27]. Its popularity in robotics stems from its flexibility, affordability, and compatibility with various hardware interfaces and open-source software platforms like Linux. The Raspberry Pi can support multiple peripherals and sensors, including USB cameras, making it suitable for applications such as computer vision, real-time video streaming, and object detection in UGVs.

Sensor integration plays a crucial role in educational robotics by allowing students to interact with their robots through data acquisition and control tasks. The EUROPA platform, an educational robotics system based on the Raspberry Pi 3 B+, illustrates this well. It incorporates a variety of sensors such as ultrasonic modules and cameras, enabling students to explore tele-operation, line-following, and robotic arm manipulation through Python scripts. Karalekas et al. stated that these activities could help students with STEM concepts, including programming, geometry, and algebra [28].

In terms of software, DroneKit stands out as an effective open-source framework that enables autonomous operation in unmanned vehicles. It uses Python as its primary language and allows developers to create high-level applications that can control vehicle movement and sensor integration. Pulungan et al. [29] demonstrated the effectiveness of DroneKit in enabling autonomous flight in a quadcopter, highlighting its utility in research and development contexts.

Shown in Figure 5 below is the UGV design by Corpuz et al. [5]. It is a fully 3D-printed working model developed to be low-cost, modular, and educational. The design incorporates the open-source Pixhawk controller with ArduPilot Mission Planner software, while the open-source CAD files allow users to create additional modules suited to their experiment/objective.

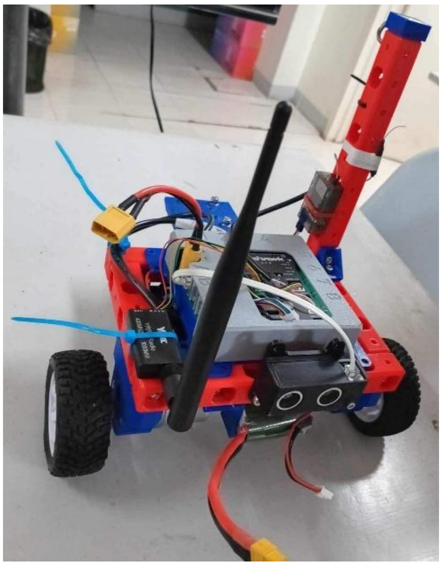


Figure 5. UGV design by Corpuz et al. [5]

Based on the recommendations of Corpuz et al. [5], it was stated that the UGV design could be further improved by exploring alternative 3D printing materials, such as ASA, to enhance durability and environmental resistance. In terms of assembly, the development of snap-fit joints was recommended to make the frame easier to assemble without compromising connection strength. Furthermore, Corpuz et al. [5] recommended exploring alternative drive configurations, including tracked systems, to improve mobility across terrains. Finally, for the companion computer, they proposed the use of more powerful units, such as the RPi, to enable more features and functionality.

3. Research Methodology

This section outlines the methods and procedures used in designing, manufacturing, and testing the improved UGV, with emphasis on modularity, cost-effectiveness, and integration of auxiliary components. The manufacturing process employed 3D printing for the frame, while the testing procedures evaluated both hardware performance and educational value.

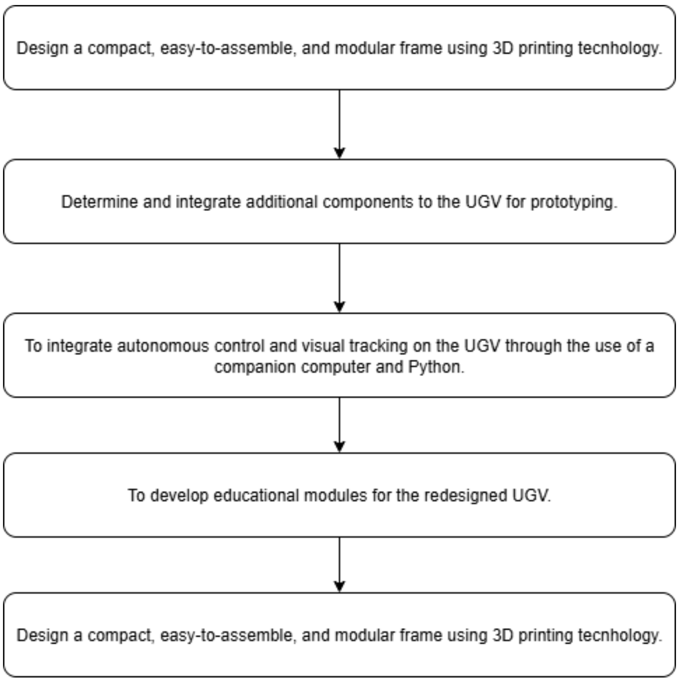


Figure 6. Methodological framework

Figure 6 illustrates the methodological framework, beginning with the redesign and fabrication of a compact, modular, and easy-to-assemble frame using 3D printing. This is followed by the selection and integration of additional components such as the Raspberry Pi and camera, prototype testing, development of educational modules, and finally, user testing and evaluation of the complete UGV platform.

3.1. UGV Design Considerations

To uphold the goals of creating a low-cost, modular, and educational mobile robot, the UGV was reassessed across multiple design criteria, including power, sensor operation, and battery life. The redesign utilizes 3D-printed components for affordability and incorporates a Raspberry Pi to retain open-source programmability while supporting more complex tasks. Cost-effectiveness was also achieved by reusing functional components from the previous design and purchasing only necessary additions such as the camera. The redesign builds upon the strengths of the earlier model, addresses assembly difficulties, and introduces new features that expand learning opportunities.

3.1.1. Power

Power requirements for the UGV were assessed based on power consumption and reliable task execution. Energy consumption is commonly determined using experimental data; however, this often excludes factors relating to robot dynamics and terrain conditions. The required power Pt of a UGV is calculated using Equation 1 [30]:

$$P_t = \frac{v}{\eta}(F_t) \quad (1)$$

where, v is linear velocity (m/s), Ft is traction force (N), and η is the overall efficiency (approximately 0.9 for direct drive systems).

3.1.2. Torque Conditions

Locomotion requires adequate motor torque. Although wheel count, diameter, and motor output were predetermined, Equation 2 was used to confirm the suitability of the previous specifications [30]:

$$\tau = \frac{1}{N_W} * \frac{D_W}{2} * F_t \tag{2}$$

where, NW is the number of wheels, DW is wheel diameter, and Ft is traction force determined from Equation 1.

3.1.3. Sensor Operation

A digital camera was integrated into the UGV to support computer vision tasks. Cameras use photosensors that detect light intensity but not color; thus, a Bayer filter mosaic allows red, green, and blue color detection. Afterward, demosaicing algorithms reconstruct full-color images [31]. These capabilities enable computer-vision-based tasks—including classification, tracking, identification, and optical character recognition—by pairing the camera with processing on the Raspberry Pi. Ultrasonic sensors, GPS, and the Pixhawk controller further allow indoor and outdoor deployment and broaden the range of educational activities.

3.1.4. Power Requirements

Power calculations were performed for the Raspberry Pi 4 Model B and the USB webcam. The companion computer requires 5 V DC at up to 3 A, while the webcam draws power through USB depending on model specifications. Individual component power consumption was calculated using:

$$P = I \cdot V \tag{3}$$

After obtaining the power consumption of each component, the total power consumption of the UGV can be obtained by adding up the power consumption of each component. The equation for this is shown below.

$$P_{total} = P_{companion\ computer} + P_{camera\ module} \tag{4}$$

3.1.5. Battery Life

The battery serves as the UGV's primary power source; lithium-ion batteries are preferred due to their high energy density and long lifespan. An important specification is the C-rate, calculated as [32]:

$$C - rate = \frac{Current\ (A)}{Battery\ Capacity\ (Ah)} \tag{5}$$

High C-rates shorten battery lifespan, while low C-rates underutilize capacity; thus, an optimal balance is required. The minimum battery capacity needed was determined using [33]:

$$Battery\ Capacity\ (Ah) = \frac{Total\ Power\ Consumption\ (W) \cdot Operating\ Time\ (h)}{Battery\ Voltage\ (V)} \tag{6}$$

This equation considers voltage and load to ensure adequate operating time without excessive battery weight.

3.2. UGV Frame Redesign

The frame was redesigned to improve modularity, functionality, and ease of assembly while maintaining low cost. The research team reverse-engineered the previous UGV using manual measurements to guide modifications. Snap-fit connections were introduced as the primary structural improvement, enabling tool-free assembly without compromising strength. Additional connecting mechanisms—peg-and-hole and sliding mechanisms—were implemented for component flexibility. Custom housing for the Raspberry Pi and camera was designed in Onshape and 3D-printed. Components were positioned to allow flexibility in layout without affecting performance. Figures 7 to 10 illustrate the final design and the three groups of connection mechanisms (snap-fit, peg-and-hole, and sliding).

Shown in Figure 7 is the final UGV design developed by researchers. The design can be classified into different categories based on component connectivity. The first group is the snap-fit components, shown in Figure 8, which include the main frame, wheel mounts, battery holder, caster wheel mount, and GPS pole. The second group is the peg-and-hole components, shown in Figure 9, consisting of the ESC holder board, Raspberry Pi holder board, and Pixhawk holder board. Lastly, the third group is the sliding mechanism components, shown in Figure 10, which includes the RC receiver housing, camera mount, and antenna housing.

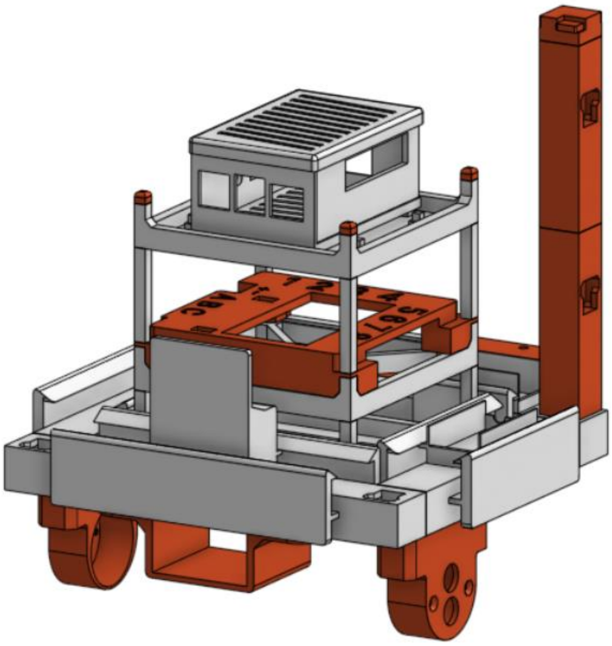


Figure 7. Final UGV design

The snap-fit components shown in Figure 8 consist of the main frame, wheel mounts, battery holder, caster wheel mount, and GPS pole. Each of these components features a custom-designed hexagonal male connector that mates with a corresponding female slot on the main frame. The mechanism functions by first aligning the male and female parts, then the male part is inserted and slid downward until it locks into place. This provides a secure attachment and also allows for quick assembly and disassembly.

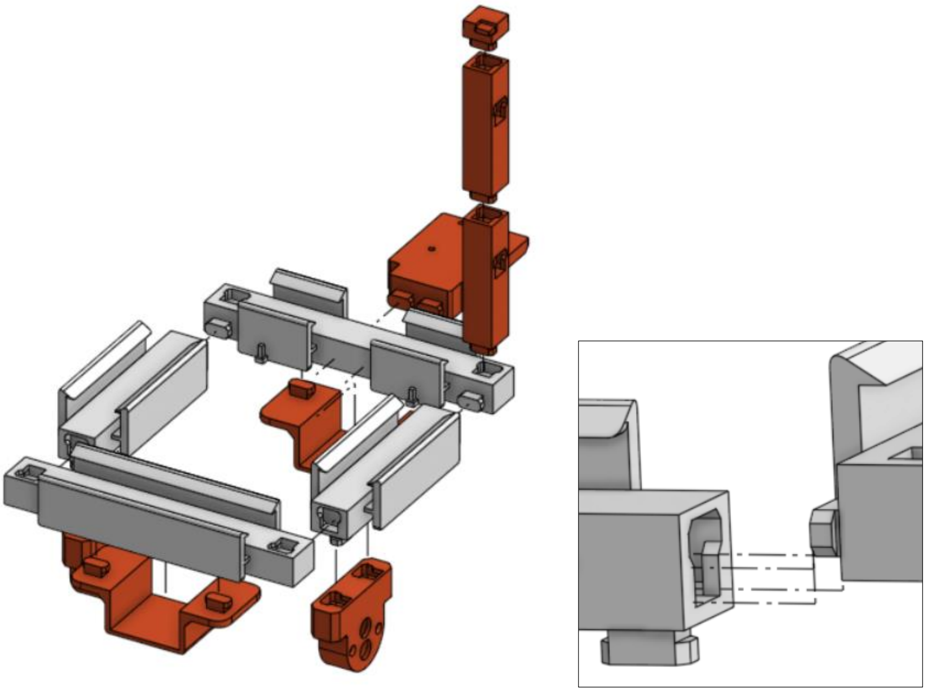


Figure 8. Snap-fit components of the UGV design

The peg-and-hole components, also shown in Figure 9, include the ESC holder board, Raspberry Pi holder board, and Pixhawk holder board. Each board is designed with a set of holes at either face into which pegs are inserted. This feature enables the boards to be mounted onto the frame or stacked onto one another. It also contributes to the ease of assembly and disassembly by allowing the boards to be easily removed along with their pegs, providing better access to components during troubleshooting or maintenance of the UGV.

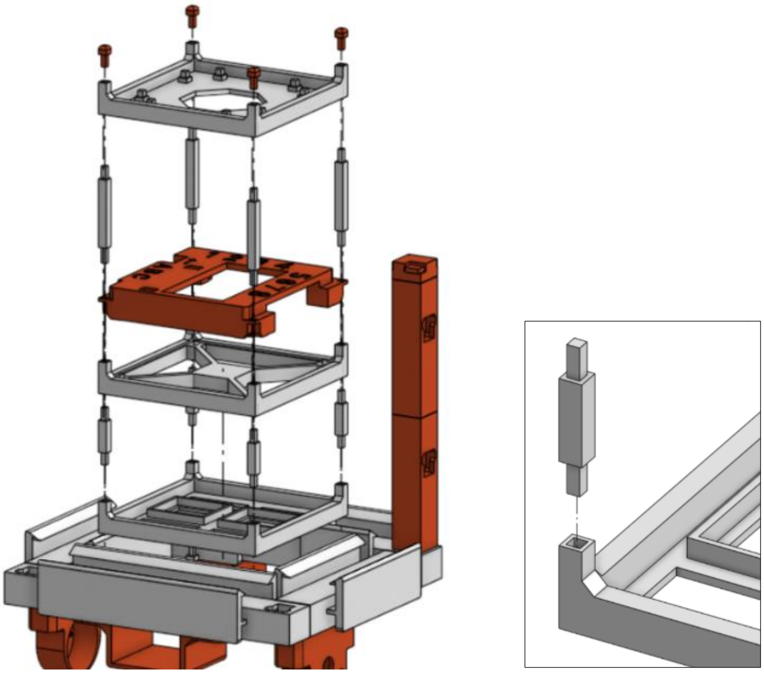


Figure 9. Peg-and-hole components of the UGV design

The sliding mechanism components, shown in Figure 10, are composed of the RC receiver housing, camera mount, and antenna housing. Each component is designed with extruded features that fit into corresponding slots on the main frame. The mechanism operates by sliding the component along the slot until it reaches the desired position, where it is secured. This sliding capability allows for a stable connection and provides adjustable positioning of sensors and modules for convenience and functionality.

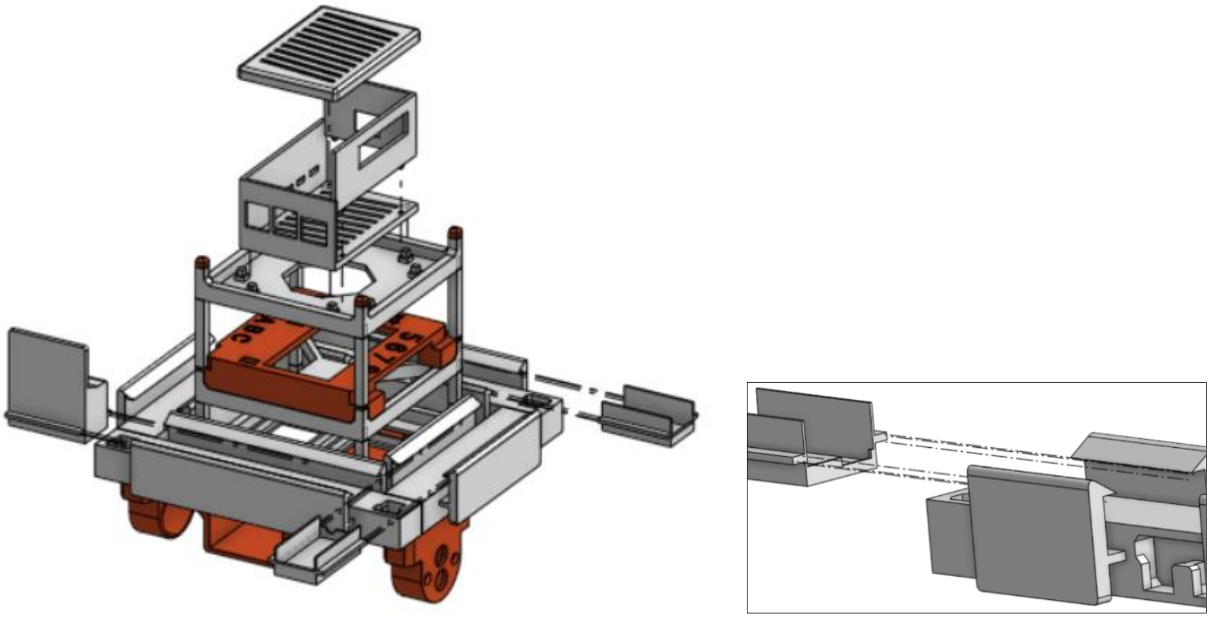


Figure 10. Sliding mechanism components of the UGV design

3.3. Component Selection

A Raspberry Pi 4 Model B was selected as the companion computer, using its GPIO pins to interface with the Pixhawk. A UBEC was installed to step down the LiPo battery voltage to a stable 5 V, 4 A power supply for the Raspberry Pi. Custom cabling—including a soldered 6-pin JST to jumper-wire interface—enabled communication between the Raspberry Pi and Pixhawk. A USB webcam was chosen for marker detection, computer vision tasks, and teleoperation (see Figures 11 and 12).

Figure 11. Power cable for the mainboard and RPi

Figure 12. 6-pin JST connected to jumper wire for Pixhawk-RPi connection

3.4. Software Implementation

The Raspberry Pi was set up using Raspberry Pi OS Bullseye Lite (64-bit, Legacy), chosen for its compatibility with DroneKit and OpenCV. The software stack included DroneKit-Python for MAVLink velocity control, OpenCV for ArUco marker detection, and MAVProxy for Pixhawk interfacing. A software-in-the-loop (SITL) environment was configured using ArduPilot on Ubuntu (via WSL) to validate scripts before deploying them to the physical UGV (Figure 13).

The system also incorporated ArUco marker detection and tracking using calibrated camera parameters to determine the marker's lateral offset and forward distance. These values enabled real-time control—adjusting turning behavior based on the marker's position and automatically reducing speed upon approach. A safety feature was implemented wherein the UGV stops and disarms if the marker is lost for more than one second, then rearms once the marker is re-detected within one meter. This tracking system demonstrates a lightweight navigation method that does not rely on GPS or machine learning, using only a USB webcam and printed markers.

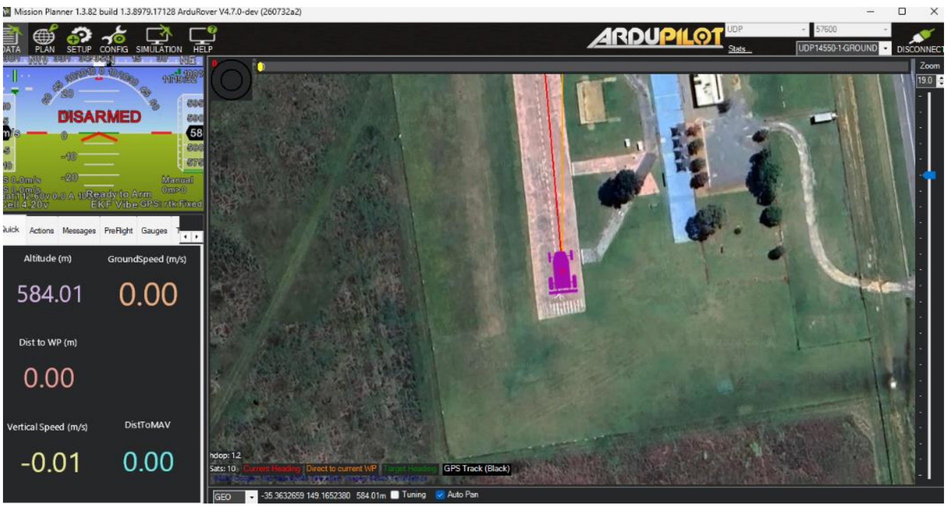


Figure 13. Software-in-the-loop (SITL) environment used for testing scripts

3.5. Educational Module Development

The educational modules were developed using established learning theories—constructivism, experiential learning, and outcome-based education. Constructivism emphasizes active engagement, while experiential learning highlights reflection and hands-on participation. Modules incorporate activities such as assembly, programming, experimentation, and collaboration, aligned with Bloom's taxonomy to promote higher-order thinking skills [34-36].

The lessons were structured into an online learning platform containing chapter-based content, procedural steps, media resources, and sample code. Visual consistency, clear text hierarchy, and "*Manlalakbay* Notes" were added to enhance clarity, engagement, and safety. The platform supports both adolescents and adults by providing opportunities for problem-solving, simulation-based practice, and mechanical assembly (Figure 14).

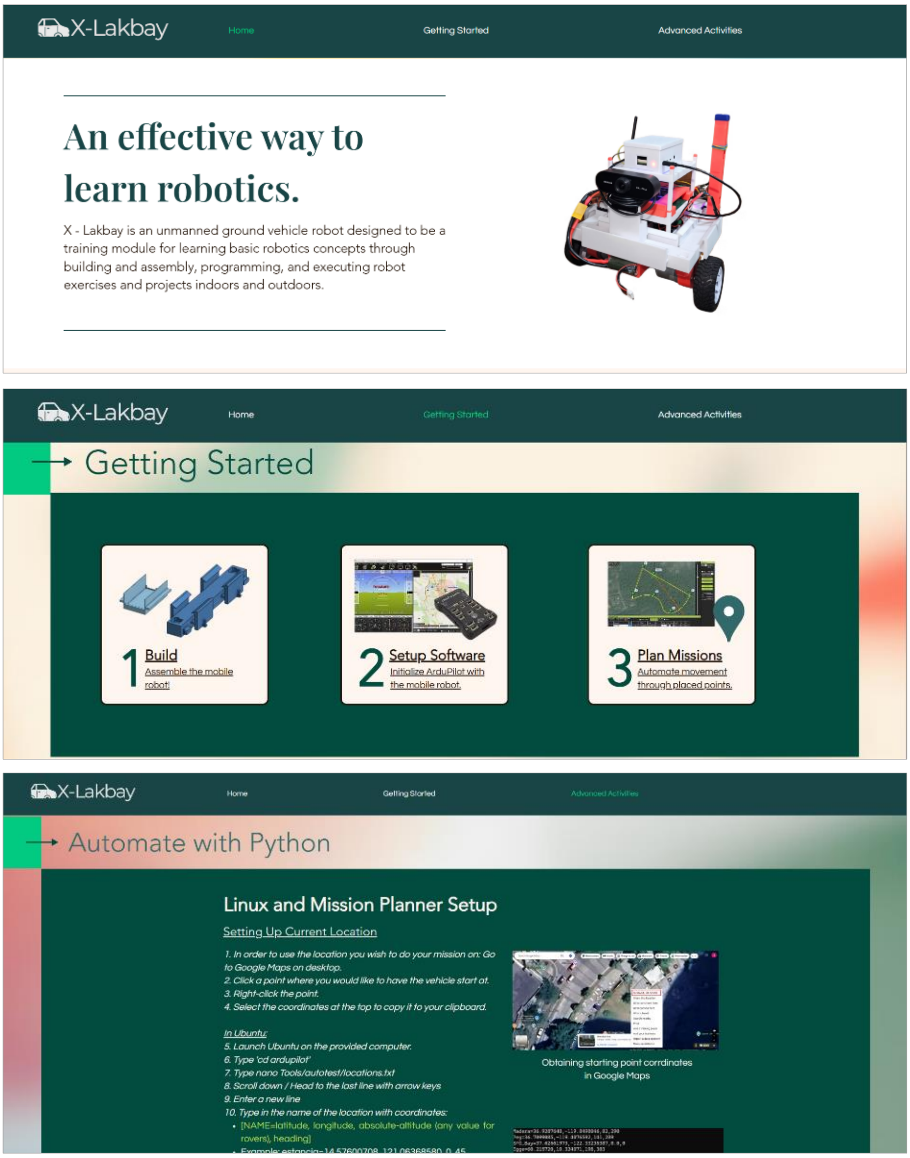


Figure 14. X-Lakbay website / Online educational modules (https://regillkentesquivia.wixsite.com/x-lakbay)

3.6. Testing and Evaluation

3.6.1. Incline Testing

For the UGV incline testing, a wooden ramp measuring 88.6 cm × 29.7 cm × 10 cm was used. This wooden ramp served as the primary surface on which the UGV traversed during testing due to its rigidity. The ramp was tilted using various household items, such as empty filament rolls, a suitcase, boxes, and books, to achieve incline angles in 5° increments. For example, a filament roll created a 5° slope, while a small box gave 15°, and a suitcase produced about 20°. Books were added for fine 1° adjustments. The Phyphox phone app was used to measure the inclination angle. A cardboard paper bag taped to the base served as a transition ramp, and a plastic lid at the top acted as a landing platform. This setup simulated real-world terrain and was used to evaluate the UGV's ability to climb and maneuver inclines. For each angle, five uphill trials were conducted. The incline angle was measured before and after each trial to ensure consistency. The target incline was around 20°, aligning with the minimum slope capability referenced for UGVs in the National Institute of Standards and Technology (NIST) Standard Test Methods for Response Robots [37, 38].

The complete setup of the incline test can be seen in Figure 15, where (a) illustrates the height and angle variation along with the plastic container lid, (b) displays the wooden ramp, and (c) shows the cardboard paper bag used as the transition ramp.

(a) (b) (c)

Figure 15. Overall incline testing setup

3.6.2. Software Testing

3.6.2.1. GPS Testing

Dronekit was designed primarily for outdoor use and requires a GPS lock to function properly. A GPS lock requires at least 6 satellites to connect to the GPS of the UGV. To verify what types of environments the UGV could function in, the researchers noted the satellites acquired by the UGV in the different locations around De La Salle University - Manila Campus. The number of satellites the UGV is connected to can be seen in the Mission Planner application. The test was conducted first with a continuous fix to the satellites, starting in an area with open access to the sky, and then brought to the different test sites. The second set of tests was done with a cold start, meaning the UGV was fully powered off, then powered on, and allowed to try and detect satellites. This was done to assess the UGV's satellite acquisition capabilities in both ideal and obstructed conditions.

Figure 16. Pictures of Test Sites (a) LS Facade (top left), (b) CADS (edge) (top right), (c) Henry Sy-Yuchengco (bottom left), (d) Henry Sy Lawn (bottom right)

Figure 16 shows the test sites used. LS Facade and the Henry Sy lawn (Figures 16a and 16d) both give the robot open access to the sky, while CADS and Henry Sy-Yuchengco (Figures 16b and 16c) give the robot only partial access to the sky. All areas are located in the same general dense urban area, with multiple buildings surrounding the different locations.

3.6.2.2. Velocity Testing

A controlled test was conducted to verify the UGV's velocity performance while utilizing DroneKit. Initially, the starting position of the UGV was marked with masking tape. The vehicle was then programmed to proceed north relative to its orientation at a velocity of 0.5 m/s using local velocity movement with DroneKit. Then, the actual distance travelled by the UGV was measured using a tape measure. This procedure was repeated five times to ensure data consistency and reliability.

3.6.2.3. Straight-Line Testing

To evaluate the straight-line performance of the UGV, it was programmed to move at a constant speed of 0.5 m/s for 5 seconds. A compass app was used to record the yaw angle both before and after each run. The test was repeated five times. Additionally, before measurement, the compass was calibrated by moving the phone in a figure-eight motion until the app confirmed successful calibration.

3.6.2.4. Turning Testing

To assess the turning performance of the UGV, it was first positioned at a specific heading, then programmed to move east (rotate clockwise) at a speed of 0.5 m/s. The yaw angle was recorded before and after each turn using the compass app. This procedure was repeated five times to ensure consistent results. The results were compared to simulation results using software-in-the-loop (SITL) in Mission Planner.

3.6.2.5. Multiple Aruco Marker Performance Test

The functionality of assigning distinct commands to specific markers was also evaluated. Using procedures similar to the velocity, straight-line, and turning tests, the response of the UGV to three designated markers shown in Figure 17 was measured. Each marker corresponded to a unique action, namely moving forward, turning left, or turning right, executed at a speed of 0.25 m/s. IDs 3, 272, and 836 from the original Aruco marker dictionary at 19x19cm were printed out and used to represent the left command, forward command, and right command, respectively. To ensure consistency, five trials were conducted for each marker. Additionally, a sequential detection test was performed wherein the UGV was shown all three markers one after the other. For each trial, the researchers recorded which marker the UGV detected and the action it performed in response. This sequence was repeated five times to assess detection reliability and command execution.

Figure 17. Multiple Markers (a) Left Marker - ID #3 (left), (b) Forward Marker - ID #272 (middle), (c) Right Marker - ID #836 (right)

3.6.2.6. Aruco Marker Range Testing

A 19cm x 19cm ID 72 Aruco marker was printed out and pasted on a foam board. To assess the range and accuracy at different resolutions, the webcam was first calibrated at both 480p and 720p. The Aruco marker board was then held at various distances, with each actual distance measured using a tape measure. At each point, the distance reported by the program was also recorded. Additionally, the minimum and maximum distances at which the marker could be detected were measured.

3.6.2.7. Aruco Marker Parking Testing

To evaluate the Aruco marker parking performance, the UGV was positioned approximately 2.5 meters from the marker. It was then programmed to approach the marker and stop at a distance of 1.5 meters. A piece of tape was placed on the floor to indicate where the marker should be held for consistency. Once the UGV came to a stop, the distance between the camera lens and the center of the Aruco marker was measured using a tape measure.

3.6.2.8. Aruco Marker Follow-Me Performance

To evaluate the ability of the UGV to follow an Aruco marker, a test was conducted in which the robot was initially placed 1.5 meters in front of the marker. The marker was then moved backwards over a walking distance of 5 meters, with the UGV programmed to follow. This procedure was repeated ten times to assess consistency and tracking performance.

3.6.3. Assessing the Ease of Use and Educational Value of the Platform

To evaluate the user-friendliness and educational impact of the X-Lakbay UGV, a focus test was conducted with 22 high school students, mostly from branches of Philippine Science High School in different regions in the country, with varying levels of robotics experience. Students were given educational modules to follow, wherein the researchers guided them throughout the process. After which, they were asked to answer a survey to collect feedback.

The survey utilized a Likert scale where participants rated aspects like Mechanical Assembly, Electronics, Software, Overall UGV Performance, and Educational Value on a scale from 1 (Highly Dissatisfied) to 5 (Highly Satisfied). Open-ended questions were also included to gather detailed feedback. The results from both numerical and written feedback can be used to guide future improvements.

4. Results and Analysis

4.1. Final Design of the UGV

This study builds on the 3D-printed PETG UGV chassis of the previous group by further improving its lightweight, low-cost, and modular design. Main improvements focused on ease of assembly and interchangeable components for various applications. As a micro UGV, maintaining a compact yet lightweight form is essential. The unified wiring system from the earlier design was retained for improved ease of assembly. PETG remained the material of choice due to its affordability and strength, as demonstrated by previous researchers, who found that PETG frames exhibited structural strength comparable to aluminum extrusions. The final UGV design, shown in Figure 18, measures 31 cm in length, 27.3 cm in width, and 35.7 cm in height. Its parts are labelled accordingly.

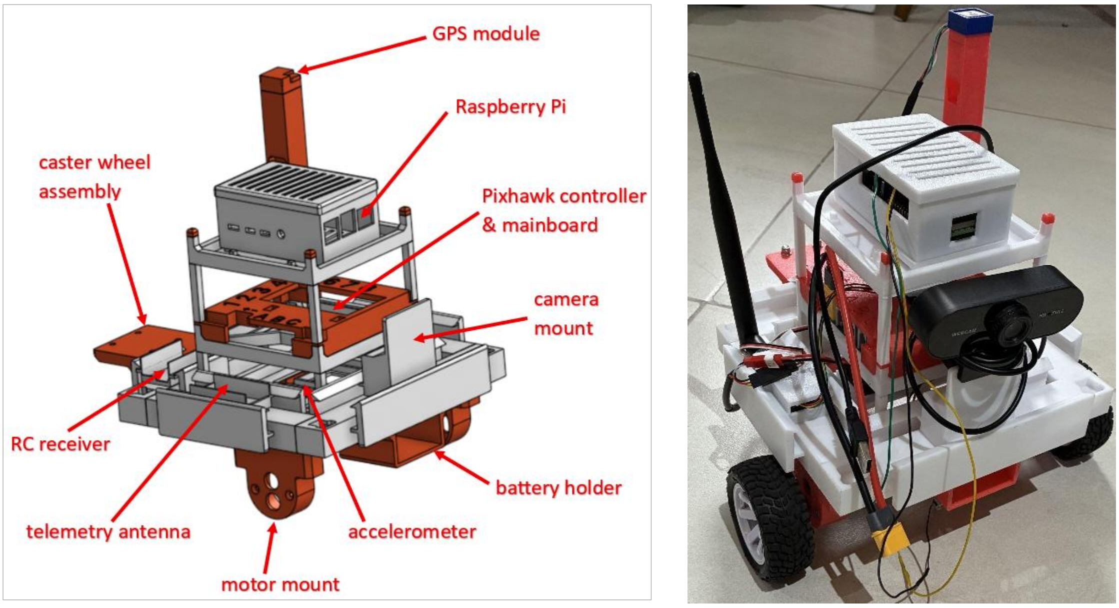


Figure 18. Final UGV design

Compared to its predecessor, the redesigned UGV is heavier by approximately 650 grams, primarily due to the addition of new components. The predecessor had a total weight of approximately 1.5 kilograms, with the frame itself weighing around 365 grams, whereas the redesigned UGV design has a total weight of approximately 2.15 kilograms, with the frame weighing about 720 grams. Even so, the current weight category of the UGV falls within the micro UGV class, in line with the project's objectives.

As a possible approach to improve the design, exploration on the use of different materials or multi-material print outputs, if feasible, can be considered. For example, Thermoplastic Polyurethane (TPU), such as TPU98A, can be used for the motor mounts. When printed purely in TPU, the component can function as a suspension system for the UGV, potentially absorbing a portion of the impact it experiences as it travels. Although not as UV- and temperature-resistant as PETG, a multi-material approach, combining TPU with PETG or ASA, can also be explored. ABS (Acrylonitrile Butadiene Styrene) and ASA (Acrylonitrile Styrene Acrylate) are excellent alternatives to PETG. They offer higher stress resistance, stiffness, impact resistance, heat resistance, and UV resistance than PETG, making them suitable for parts such as the main frame and snap-fit components, especially in the context of the Philippines, a tropical country with relatively high temperatures and a high UV index. ASA would be able to withstand these conditions better than PETG, potentially extending the component's service life. In addition, they are also less dense than PETG. Theoretically, when the same infill density and settings are used, ABS would yield the lightest UGV weight, followed closely by ASA. However, the downsides of these materials are that they offer less fatigue resistance and require much more attention during printing, as they are more sensitive. For example, ABS is more prone to warping and would require adjustments and fine-tuning of necessary parameters to achieve the optimal print settings. Additionally, ABS and ASA are more expensive than PETG, which could be a limiting factor for a low-cost UGV.

4.2. Incline Testing Results

As shown in Table 1, the UGV completed all trials without issues at 5°, 10°, and 15° inclines. At 20°, it climbed successfully but showed minor forward tilting when stopping during descent, likely due to its center of gravity being located towards the front of the unit. At 25°, the same tilting occurred, and the battery began to shift forward slightly due to limited friction and momentum. Additional tests were done in 1° increments beyond 25° to determine the UGV's maximum incline capability. At 26°, the UGV climbed all trials but continued to show tilting during descent. At 27°, it successfully climbed the ramp only in some trials, with minimal wheel slip and the same descent issues. At 28°, it also climbed in some trials, but at low speed, with more severe issues such as wheel slipping during ascent and greater forward tilt during descent, increasing the risk of rollover. Overall, the UGV attained the objective of matching or exceeding its predecessor's slope-climbing ability, achieving a higher maximum incline of approximately 26°. It also attained the minimum standard for UGV incline performance. However, similar to its predecessor, operation at these higher inclines requires a steady and controlled descent; otherwise, the risk of failure due to extreme tilt would occur. The findings suggest that improvements in center-of-gravity placement through chassis redesign or rearranging of placement of components may be needed for safer operation on steep slopes.

Table 1. Summary of incline test results

Incline (°)	Trial no.	Ascent time (s)	Ascent speed (m/s)	Ascent Result	Descent Result	Observed Issues
5	1	16.39	0.088	passed	passed	none
	2	14.12	0.102	passed	passed	none
	3	14.67	0.099	passed	passed	none
	4	15.72	0.092	passed	passed	none
	5	15.57	0.093	passed	passed	none
10	1	16.74	0.086	passed	passed	none
	2	15.53	0.093	passed	passed	none
	3	16.14	0.090	passed	passed	none
	4	15.94	0.091	passed	passed	none
	5	14.24	0.102	passed	passed	none
15	1	17.12	0.084	passed	passed	none
	2	16.65	0.087	passed	passed	none
	3	17.09	0.085	passed	passed	none
	4	14.69	0.098	passed	passed	none
	5	15.11	0.096	passed	passed	none
20	1	17.61	0.082	passed	passed	minimal tilt during descent
	2	19.18	0.075	passed	passed	minimal tilt during descent
	3	16.05	0.090	passed	passed	minimal tilt during descent
	4	14.52	0.100	passed	passed	minimal tilt during descent
	5	16.58	0.087	passed	passed	minimal tilt during descent
25	1	16.67	0.087	passed	passed	minimal tilt during descent
	2	14.33	0.101	passed	passed	minimal tilt during descent
	3	14.08	0.103	passed	passed	minimal tilt during descent
	4	16.65	0.087	passed	passed	minimal tilt during descent
	5	16.52	0.088	passed	passed	minimal tilt during descent
26	1	16.85	0.086	passed	acceptable	minor tilt during descent
	2	14.76	0.098	passed	acceptable	minor tilt during descent
	3	15.87	0.091	passed	acceptable	minor tilt during descent
	4	15.13	0.096	passed	acceptable	minor tilt during descent
	5	14.41	0.100	passed	acceptable	minor tilt during descent

27	1	20.44	0.071	passed	failed	moderate tilt during descent
	2	18.41	0.079	passed	failed	moderate tilt during descent
	3	18.00	0.080	passed	failed	moderate tilt during descent
	4	DNF	0.102	failed	failed	minor wheel slip during ascent; moderate tilt during descent
	5	17.53	0.083	passed	failed	moderate tilt during descent
28	1	DNF	0.071	failed	failed	wheels slip during ascent; moderate tilt during descent
	2	DNF	0.069	failed	failed	wheels slip during ascent; moderate tilt during descent
	3	DNF	0.105	failed	failed	wheels slip during ascent; moderate tilt during descent
	4	DNF	0.067	failed	failed	wheels slip during ascent; moderate tilt during descent
	5	DNF	0.072	failed	failed	wheels slip during ascent; moderate tilt during descent

Note: DNF = Did Not Finish (UGV failed to complete the incline climb during the trial).

4.3. Camera Calibration Results

The camera may introduce radial distortion to the images taken, hence requiring calibration using OpenCV. Here, thirty images of a checkerboard pattern are taken at different angles and fed into the calibration program. Patterns were then drawn by the program onto the images to find the corners of the checkerboard, generating a resulting camera matrix and distortion coefficients as shown below.

The resulting camera matrix is shown below in Equation 7:

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 997.71 & 0 & 405.57 \\ 0 & 1004.77 & 399.57 \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

where, f_x and f_y are the focal lengths and c_x and c_y are the optical centers. The distortion coefficients are shown below in Equation 8.

$$(k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3) = (0.369 \quad 0.1789 \quad 0.00214 \quad -0.00303 \quad 0.29332) \tag{8}$$

where, k_1 is the first-order radial distortion, k_2 is the second-order radial distortion, k_3 is the third-order radial distortion, p_1 is the horizontal distortion, and p_2 is the vertical distortion.

The camera matrix describes the internal parameters of the camera and how it portrays points from its three-dimensional coordinate space into two-dimensional image coordinates. These are used by OpenCV in calculating angles and distances with greater accuracy during image processing. Meanwhile, the distortion coefficients are used for image correction in the case of lens-induced warping. These values will be different for every camera, even if two cameras are the same model. Thus, calibration will be needed for each camera used to ensure accuracy. After applying these calibration values, the resulting image will be undistorted and will allow for more reliable results in later activities utilizing computer vision.

4.4. Software Performance

4.4.1. GPS Test Results

To evaluate the GPS performance of the UGV under varying environmental conditions, satellite acquisition tests were conducted using two methods: a cold start and a continuous fix. In the cold start scenario, the UGV was powered on at each test site and attempted to acquire satellites from scratch. In contrast, the continuous fix method involved first acquiring a GPS lock in an open-sky location before moving the UGV to other sites.

As shown in Figure 19, it was found that when the UGV attempted to connect to satellites from a cold start, areas providing clear access to the sky, specifically the LS Facade and the Henry Sy lawn, demonstrated good GPS signal acquisition, connecting with 15 satellites. This number more than meets the minimum requirement of 6 satellites necessary for a reliable GPS lock and proper functionality of DroneKit scripts. On the other hand, locations with limited sky visibility, such as CADS (edge) and Henry Sy - Yuchengco (ground level), detected 0 satellites and were unable to establish a GPS fix, thus making DroneKit non-functional. However, when the UGV was allowed to first acquire a GPS fix in an open-sky environment before being moved to outdoor, yet roofed, areas like CADS (edge) or the Henry Sy - Yuchengco (ground level), the UGV successfully maintained a good GPS signal, acquiring 12 and 13 satellites, respectively. This suggests that a continuous GPS fix, maintained while transitioning from open to partially obstructed environments, enables the GPS unit to retain satellite positions more effectively despite the weaker signal. This means that the robot must be allowed to connect to satellites in outdoor environments to properly function. However, in case of inclement weather (i.e. rain, snow, etc.), the robot can be moved to a roofed location after obtaining a GPS lock and continue to function properly.

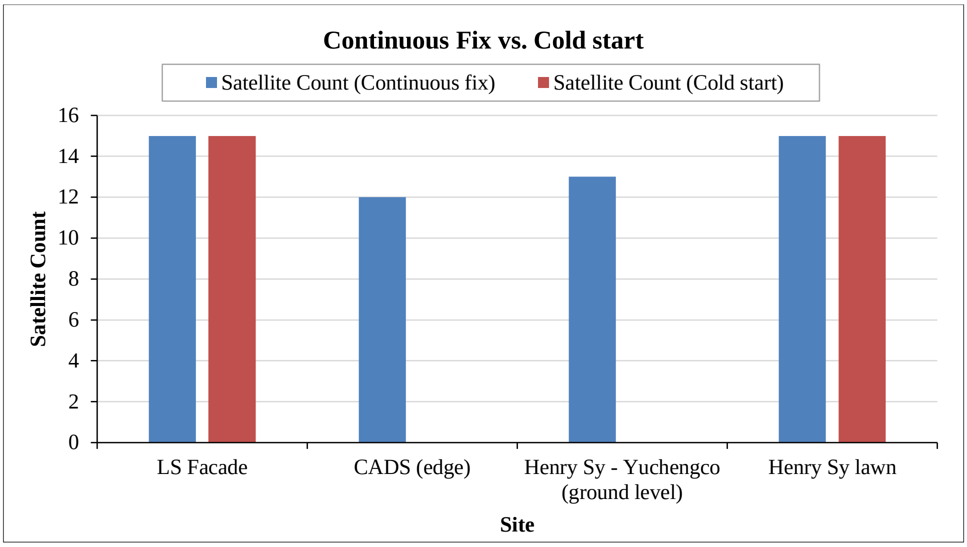


Figure 19. Satellite count: continuous fix vs. cold start

This examination is not to be confused with the GPS accuracy mentioned in Corpuz et al.'s [5] study. GPS accuracy pertained to the precision of the mobile robot's real-time location as it moves or operates in a given area, while the GPS testing conducted for this study emphasized the ability of the robot to connect to at least 6 satellites to enable ArduPilot control on roofed and un-roofed areas. Generally, the inaccuracy of the ArduPilot GPS signal in both circumstances are due to the isolated structural interferences surrounding the campus of De La Salle University - Manila. Nonetheless, testing the robot on a widely open location such as Rizal Park, to which was accomplished by Corpuz et al. [5], has a significant improvement towards such.

4.4.2. Velocity Performance

In Figure 20, the data indicates that the UGV's actual velocity was consistently lower than the expected speed of 0.5 m/s, averaging 0.416 m/s. This resulted in an average reduction of approximately 16.8% from the target velocity. A potential cause for this discrepancy observed during testing was a slight sagging of the wheels, which may have introduced additional friction, thereby affecting the UGV's overall speed. This issue is attributed to the increased weight of the current UGV at 2.15 kgs compared to the 1.5 kg UGV in Corpuz et al.'s [5] study. The increased weight improved traction on steeper surfaces but also increased wheel deflection and rolling resistance. Future designs could look at a revised layout, moving the motors and the wheels further into the body to reduce the moment generated by the frame on the wheels. Besides this, a lighter chassis or tighter tolerances with the snap-fit components could also reduce the amount of sagging experienced by the wheels. Despite the reduced average speed, the actual velocities recorded were generally consistent, with a standard deviation of 0.024 m/s. This low standard deviation shows a high degree of consistency in the UGV's speed during forward movement. The UGV's reduced speed primarily impacts the distance covered, but this can be offset by proportionally extending its travel time. Because the velocity drop is consistent, reliable module completion remains achievable, as autonomous scripts can readily adapt to a slightly lower yet stable speed profile.

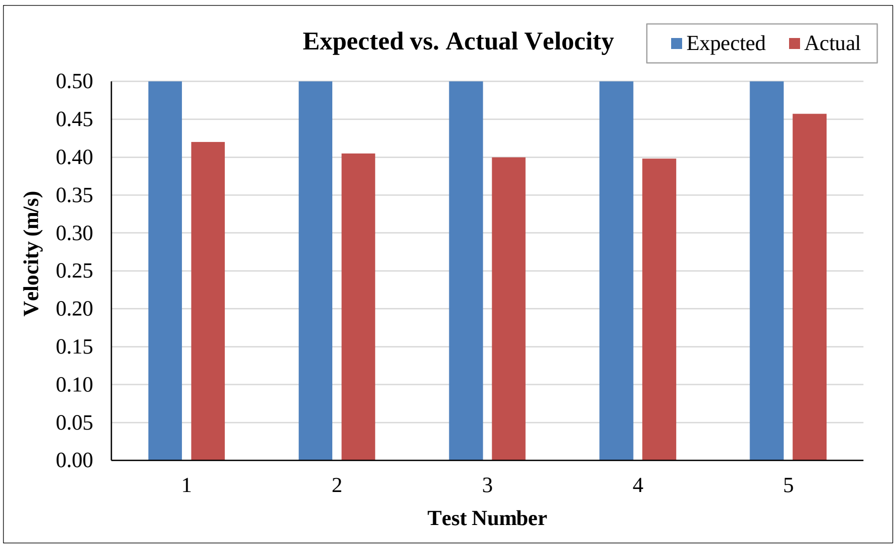


Figure 20. Summary of expected vs. actual velocity results

4.4.3. Straight-Line Performance

Observing the graph in Figure 21, the UGV had a mean deviation of 2.0 degrees and a mean absolute deviation of 2.8 degrees. This indicates that the UGV is generally quite accurate at going in a straight line. However, the standard deviation was 3.74 degrees, mainly due to an 8-degree error in Trial 3. This singular large deviation highlights that the system may be vulnerable to disturbances such as uneven terrain or tire irregularities. While large errors are infrequent, their impact should not be underestimated, as deviations can add up over long distances. For missions requiring high precision, such as in navigating confined spaces, these deviations could compromise effectiveness unless corrective mechanisms can be put into place. Possible solutions could include using adaptive control algorithms to ensure that the robot maintains its initial heading after receiving a command, thereby reducing the accumulation of directional errors over long distances.

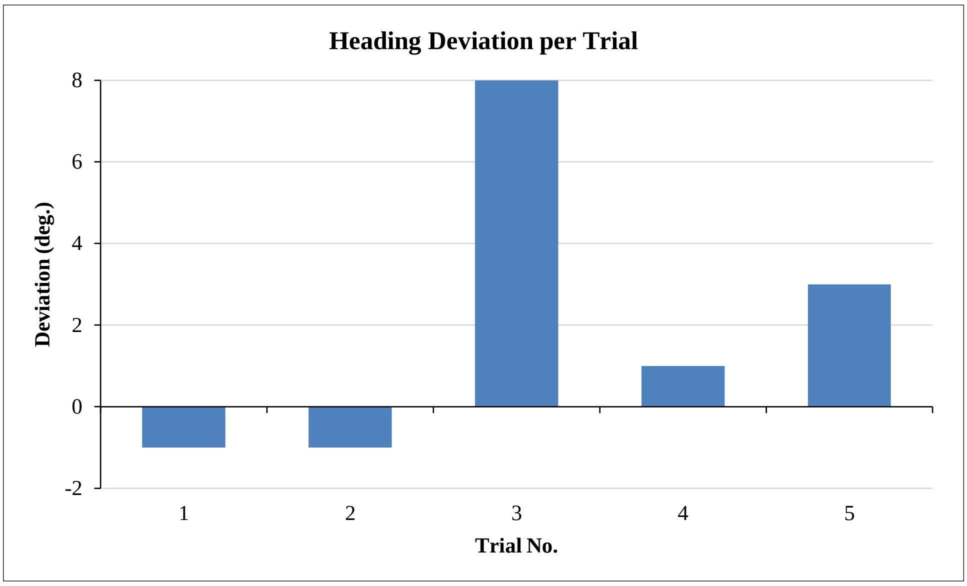


Figure 21. Heading deviation per trial

4.4.4. Turning Testing

As shown in Table 2, both the simulation and actual results were close to the expected 90° turn, with the simulated turning difference being 89.6 degrees. and the actual average being 90.4°. The simulation had a low standard deviation of 0.55°, while the actual data showed a high standard deviation of 33°. This was due to Trial 3 turning 129 degrees and Trial 4 only 37°, which together still totaled close to the expected 180°.

Table 2. Summary of turning test results

Trial No.	Simulated Yaw (°)	Compass app (°)	Expected Difference (°)	Actual Difference (°)	Absolute Percent Difference (%)
0 (Initial Position)	157	158	N/A	N/A	N/A
1	247	256	90	98	8.89%
2	336	353	89	97	8.99%
3	66	122	90	129	43.33%
4	155	149	89	37	58.43%
5	245	250	90	91	1.11%

Based on Figure 22, most trials get values close to the simulation results, meaning the UGV can get close to executing a 90° turn, although there was a large deviation for Trial 3. However, it self-corrected in Trial No. 4. More trials are needed to confirm if the large error was just a one-time issue, since the overall average was still within an acceptable range. Nonetheless, this can prove to be unreliable during operation if large errors regularly happen, as the user will not know if the robot will self-correct or proceed to turn 90 degrees accurately.

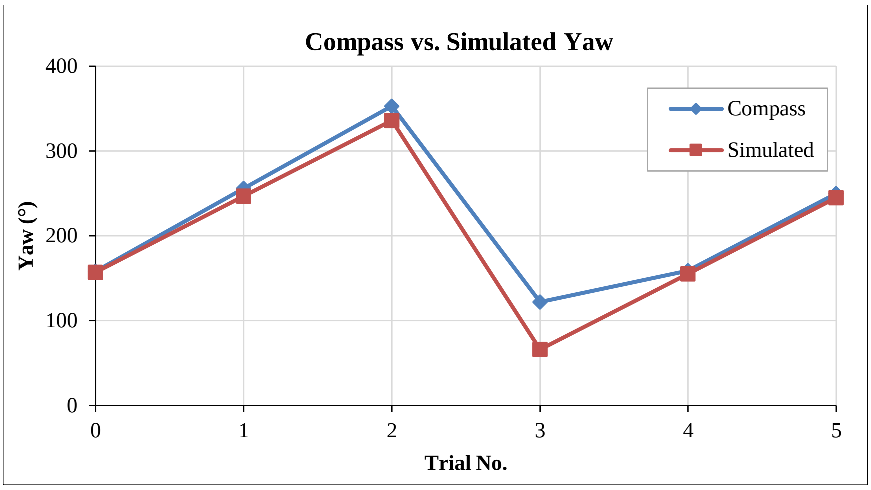


Figure 22. Compass vs. simulated yaw

4.4.5. Multiple Markers Performance

Upon placing the forward marker in front of the camera, the UGV was able to consistently execute forward motion. Figure 23 showed the difference in heading of the UGV while Figure 24 shows the difference in the expected and actual velocity. The average percent difference between its expected and actual heading was calculated to be -5.56%, indicating minor deviation from its intended trajectory. Its velocity was also recorded at an average of 0.22 m/s, which is approximately 10.76% lower than the target velocity of 0.25 m/s. This shows that the robot is able to accurately follow Aruco markers that are programmed to move the robot forward.

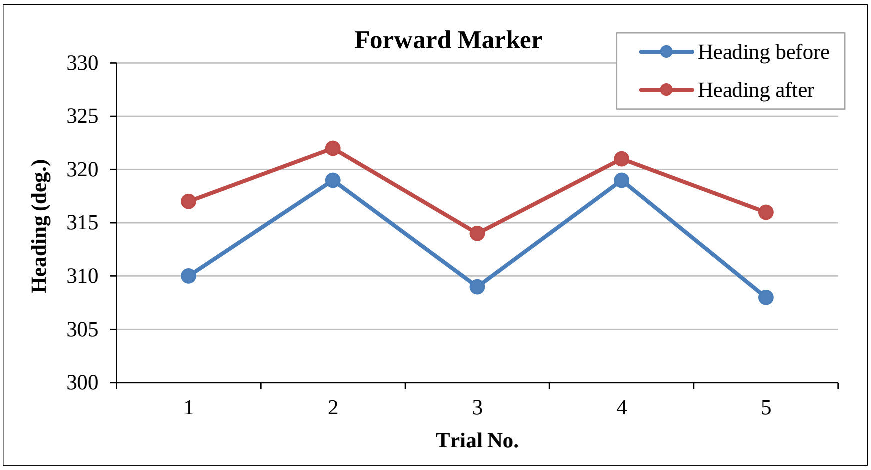


Figure 23. Heading of UGV - Forward Marker

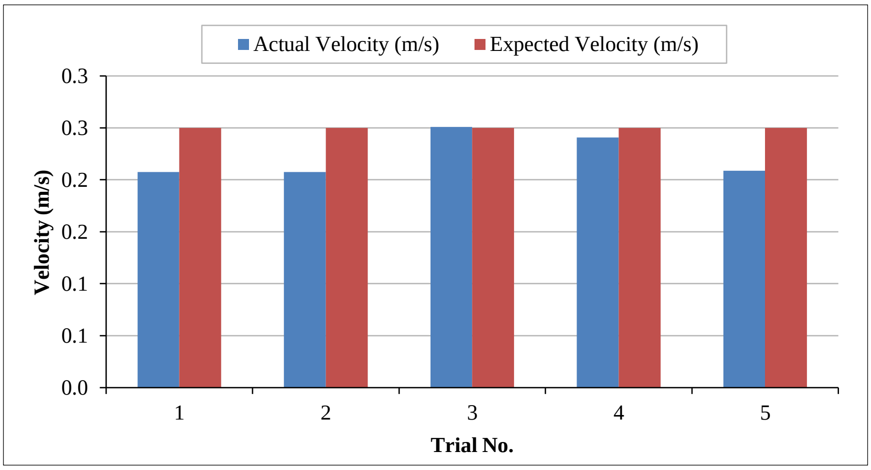


Figure 24. Velocity of UGV- Forward Marker

When responding to the right marker, the UGV exhibited a -5.56% deviation on average from the expected 90-degree turn, and a -6.22% deviation when responding to the left marker. These results indicate that the UGV was able to execute both right and left turns with reasonable accuracy upon detecting the corresponding markers. However, minor deviations may affect longer missions over time, which means using markers to navigate is ideal only for shorter distances such as a few meters. This can be seen in Figure 25, which illustrates the difference in the expected and absolute difference in identifying the right and left markers.

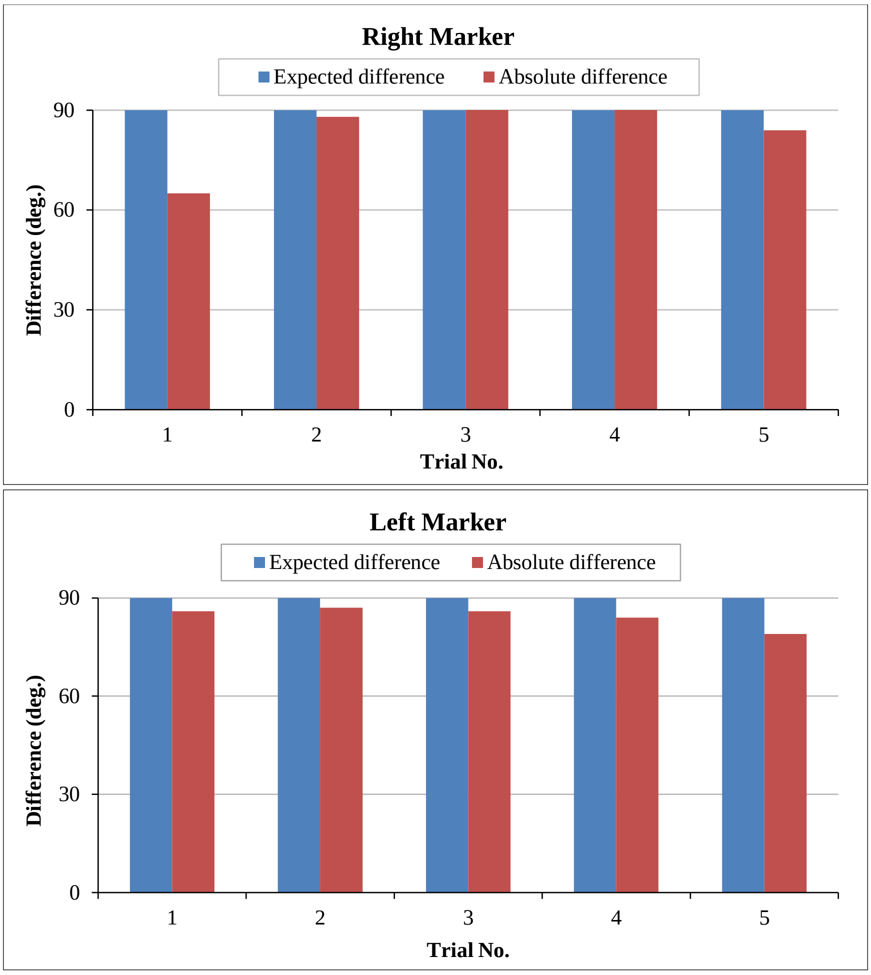


Figure 25. Right and Left Marker Deviation from 90 degrees.

Upon showing the UGV multiple markers in a row, it can be seen that the rover was able to correctly and consistently identify the marker shown to it, allowing it to perform basic functions such as turning left, turning right, and moving forward upon detection of a specific marker. As such, different markers can be programmed for different actions, and can be shown successively to the robot, which can be useful for creating games or missions for students. The complete results of the test can be seen in Table 3

Table 3. Multiple Markers Consecutive Run

Trial No.	Marker Shown (ID Number)	Marker Detected (ID Number)	Action Expected	Action Taken
1	3 836 272	3 836 272	Left Right Forward	Left Right Forward
2	3 836 272	3 836 272	Left Right Forward	Left Right Forward
3	3 836 272	3 836 272	Left Right Forward	Left Right Forward
4	3 836 272	3 836 272	Left Right Forward	Left Right Forward
5	3 836 272	3 836 272	Left Right Forward	Left Right Forward

4.4.6. Aruco Marker Range Performance

The marker was placed at varying distances from the camera of the UGV, with the camera set at either 480p or 720p, and the distance of the marker was measured.

Looking at the absolute percent difference between the detected and actual distances in Table 4, both resolutions showed accurate results, with 480p being slightly more accurate than 720p by a margin of 0.18%. Since the differences are small, they are likely due to human error during measurement. The minimum and maximum detection ranges of both resolutions were found to be the same, with a minimum distance of 40 cm and a maximum distance of 1220 cm. Based on this, we can conclude that resolution does not make a big difference in accuracy. Instead, factors like lighting conditions, marker size, and camera calibration play a bigger role in determining accuracy and reliability. As such, lower-resolution cameras and programs set to process lower-resolution markers can be used to save on both financial and computational resources.

Table 4. 480p and 720p Distance Testing

640 × 480p			1280 × 720p		
Aruco Distance (cm)	**Actual Distance (cm)**	**Absolute Percent Difference (%)**	**Aruco Distance (cm)**	**Actual Distance (cm)**	**Absolute Percent Difference (%)**
100	97	3.09%	100	91	9.89%
150	137	9.49%	150	150	0.00%
200	203	1.48%	200	198.5	0.76%
250	250.5	0.20%	250	246.3	1.50%
300	299.7	0.10%	300	293	2.39%
Total	-	**14.36%**	-	-	**14.54 %**

4.4.7. Aruco Marker Parking Performance

The expected stopping distance is set to 92.5 cm because the UGV only stops after detecting the marker within 150 cm five times, with a 0.05-second loop delay. Since it slows to 0.3 m/s below 200 cm, it travels about 7.5 cm during this delay. Therefore, 7.5 cm is subtracted from 150 cm to get the expected distance of 142.5 cm.

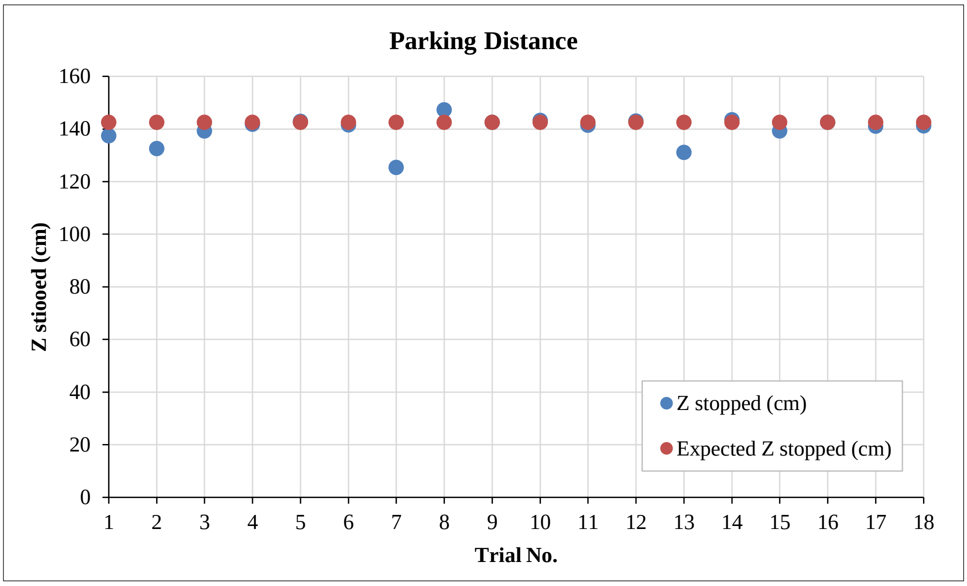


Figure 26. Actual vs. expected parking distance

Figure 26 above illustrates the comparison between the actual and expected stopping distances of the UGV relative to the Aruco marker. On average, the UGV stopped at a distance of 139.4 cm, which is 2.16% below the expected value of 142.5 cm. This minimal deviation suggests that the program is generally reliable in achieving the intended stopping distance. It also indicates that the UGV is well-suited for short-distance maneuvers, docking, or alignment tasks where the UGV precise stopping near a marker or object is required.

4.4.8. Aruco Marker Follow-Me Performance

The UGV was further evaluated using its “Follow-Me” functionality, which relied on Aruco marker tracking. The objective was for the robot to follow the marker along a 5-meter path while maintaining a target distance of 1.5 meters. To achieve this, the UGV was programmed to dynamically adjust its speed based on the marker’s position. During the test, positional data – X, Y, and Z, corresponding to the horizontal offset of the marker from the center of the frame in cm and the distance of the marker from the camera, respectively – along with the UGV’s velocity and turning rate were recorded and exported as CSV files for analysis. The horizontal deviation (X) from the center of the camera frame was averaged across trials and plotted against the mean traversal time. On average, the UGV completed the 5-meter path in approximately 12.9 seconds.

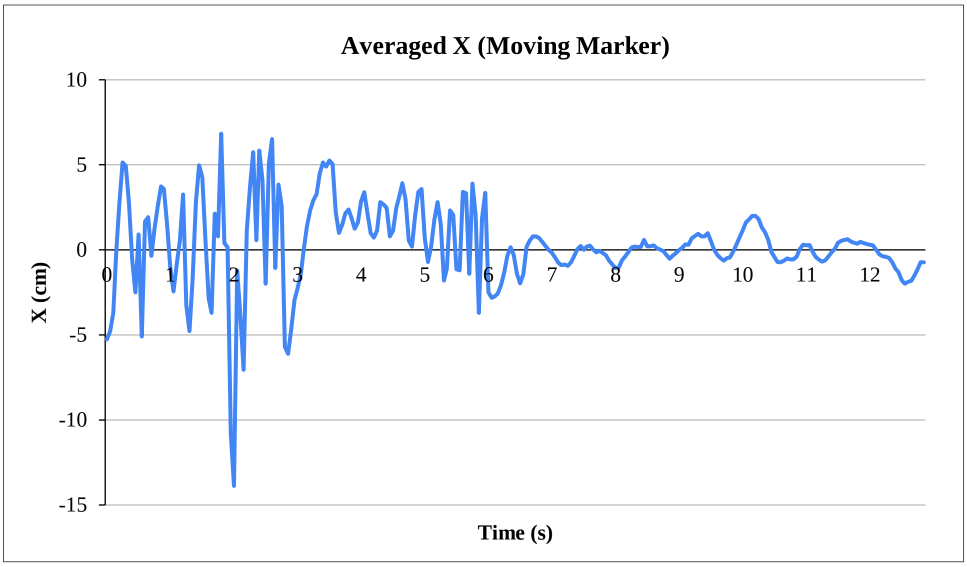


Figure 27. Averaged X (Moving Marker)

Figure 27 presents the horizontal deviation or x-value, of the marker from the center of the camera frame of the UGV as a function of time. The plot initially exhibits irregular fluctuations, which can be attributed to minor lateral swaying due to the marker manually moving backward. These fluctuations indicate transient instability caused by rapid changes in the marker’s position relative to the camera’s field of view. The deviation, however, progressively decreases, with the trajectory converging towards zero. This demonstrates the rover’s ability to continuously correct its heading and realign itself with the marker through its feedback and control mechanisms.

The mean absolute horizontal deviation across all the experimental trials was measured at 1.67 cm, indicating a relatively high degree of accuracy, considering that the marker’s motion was inconsistent as it was manually controlled. The minimal deviation suggests that the system is capable of maintaining stable lateral positioning even under non-ideal operating conditions, validating the effectiveness of the vision-based tracking algorithm and control response.

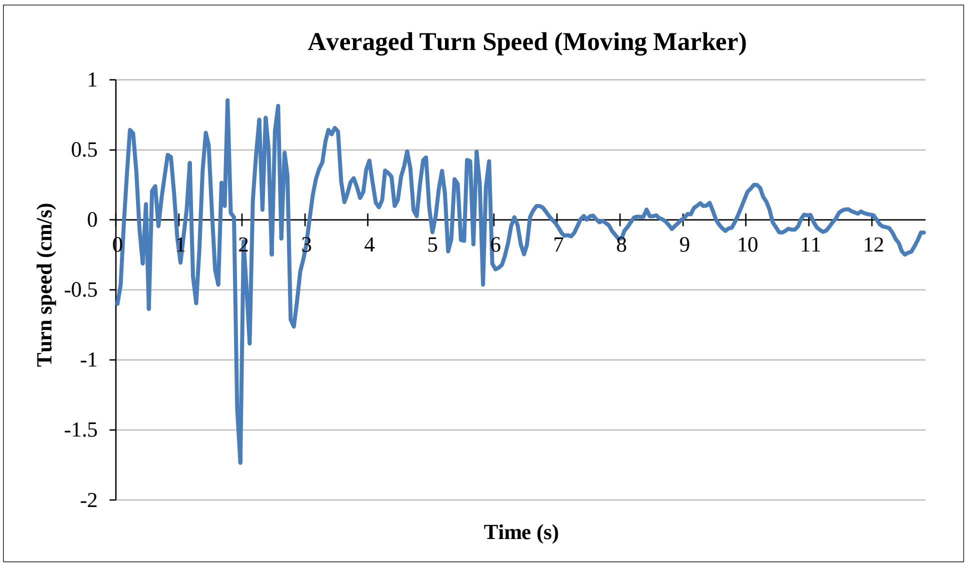


Figure 28. Averaged Turn Speed (Moving Marker)

Figure 28 illustrates the turn speed of the UGV as a function of time. The plot exhibits behavior consistent with that shown in Figure 25, which is expected given that the turn speed is directly derived from the horizontal deviation, or x-value, detected by the camera. As the rover adjusts its orientation to align with the marker, the turn speed responds proportionally to changes in lateral displacement, resulting in observable fluctuations corresponding to periods of manual movement and the rover's subsequent corrective actions.

Across all trials, the average turn speed was measured at 0.21 cm/s. This indicates that the UGV performs small, incremental adjustments rather than abrupt steering changes, reflecting a degree of stability in its tracking behavior. This suggests that the feedback mechanism moderates rotational response to minimize overshoot and maintain smooth alignment with the marker, despite the non-uniform motion of the manually controlled marker and natural disturbances.

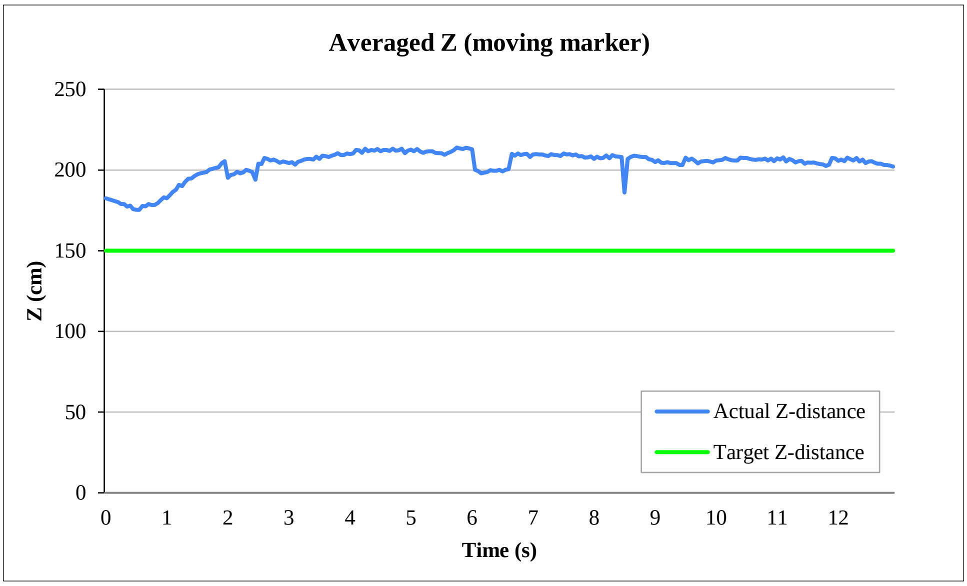


Figure 29. Averaged Z (Moving Marker)

This chart presents the z-axis distance of the UGV from the Aruco marker over time, plotted alongside the target following distance of 1.5 meters, during which the robot tracked a moving marker for an average duration of 12.9 seconds. At the start of the trial, the UGV is positioned approximately 180 cm from the marker and subsequently adjusts its position to achieve a tracking state. Rather than converging to the exact target value, the robot ultimately stabilizes slightly above 200 cm, maintaining a relatively consistent separation once the tracking behavior is established. This indicates that the system is capable of regulating longitudinal distance during motion and sustaining a steady offset within the camera frame.

Additionally, although the nominal target distance was set to 1.5 meters, the robot maintained an average tracking distance of 2.03 meters, which is approximately 36.8% higher than the intended setpoint. This deviation suggests a systematic steady-state error rather than random noise, and the behavior implies that the controller stabilizes prematurely due to insufficient corrective force, causing it to maintain a larger gap to avoid overshooting rather than fully converging toward the desired distance.

Figure 30. Setup of Follow-Me test (Moving Marker)

Figure 30 shows the setup of the moving marker test, in which the robot must maintain a specified distance z from the marker throughout the duration of its movement, over a distance of 1.5 m. To maintain this distance, the robot must adjust its speed in proportion to the distance z. In the developed code for this test, there are two equations used. The first equation determines the error, as shown in Equation 9:

$$error = z - target_z \tag{9}$$

where, z is the actual distance of the robot from the marker and $target_z$ is the required distance from the marker. The next equation determines the speed of the robot, as shown in Equation 10:

$$speed = Kp \times error \tag{10}$$

where, *Kp* is the proportional gain, which was set to 0.005 in the program, and *error*, is the distance error calculated from the first equation.

Figure 31 clearly depicts UGV behavior in real-world following conditions. With an increase in the Z-distance between the robot and the ArUco marker, the robot's speed also increases almost linearly, especially past the approximate 150 cm point. This indicates that the UGV is attempting to "catch up" to the marker as it detects that the object of interest is moving away. This behavior is favorable in real-world operation as it depicts the UGV behaving as intended and can adjust the speed of its approach based on the marker's relative movement.

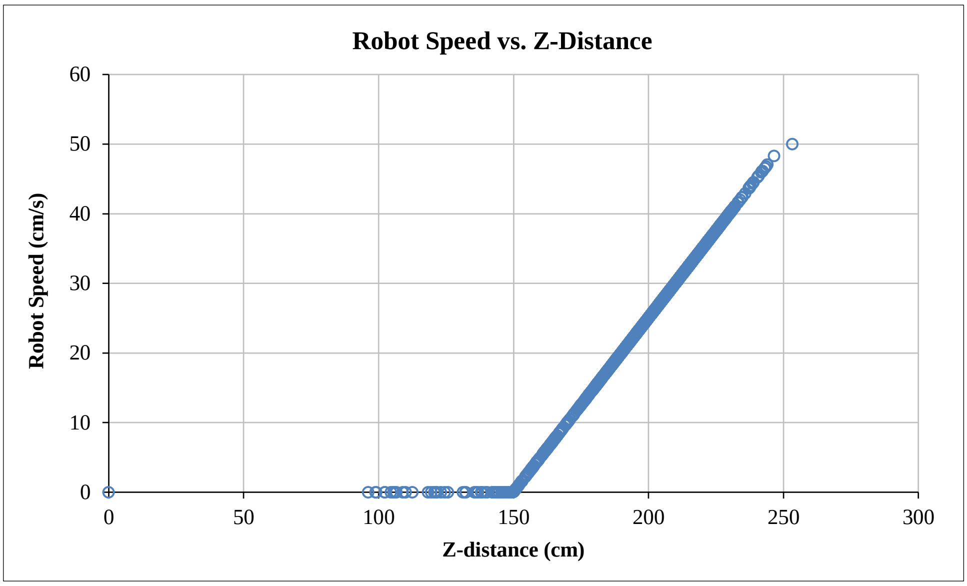


Figure 31. Robot Speed vs. Z-Distance (Moving Marker)

Nonetheless, the data also suggests further limitations. While the intended stopping distance was 150 cm, the robot was able to stabilize at an average of 203 cm, corresponding to a deviation of 35.8%. This means the control parameters (specifically proportional gain, speed scaling, and/or Aruco depth estimation) had not been tuned for high-performance stopping. In practice, inadequate tuning of these parameters could result in overshooting or maintaining unnecessarily large buffers when following a person or object which leads to reduced accuracy of tracking.

Another point to note is that, for distances less than ~140–150 cm, the robot remains nearly stationary. It appears that there is a "deadband" or threshold of sensitivity that leads the robot to not react to small changes in its distance to the target. In practical use, this has the result that the robot will not try to make fine turns when the target is near may lead to minor drift or delayed stopping.

Despite these constraints, the robot still shows that it can successfully detect increases in Z-distance and adjust with increasing speeds. This shows that the perception and decision-making pipeline is working. With additional tuning—potentially in terms of speed scaling curves, PID gains, or Z-distance readings, the robot can achieve more efficient and reliable real-world following behavior.

4.5. Focus Testing

For the evaluation of the effectiveness of the revised and added features of the mobile robot, a 2.5-hour workshop was conducted with 22 high school students from different regions and with varying experience in robotics and programming. With three separate day sessions depending on the availability of the respondents, the assembly of the snap-fit designed frame and electronic parts, familiarization with ArduPilot Mission Planner, and the integration of Python-programmed tasks with Ubuntu and PuTTY, which is used to remotely access the RPi, were accomplished in groups of approximately 5 people. The workshop flow can be seen in Figure 32. This, together with the constructed online website / educational module that provided the students with a video guide for mechanical assembly construction, and *Try-Out* activities that exercise and execute the commands and techniques in the sample codes on the discussion portion of the module. Aspects of the mobile robot: (1) Mechanical Assembly, (2) Electronics Assembly, (3) Software, (4) Overall UGV Performance, (5) Robot Educational Value were assessed in this regard, including the respondents' change in interest before and after the workshop.

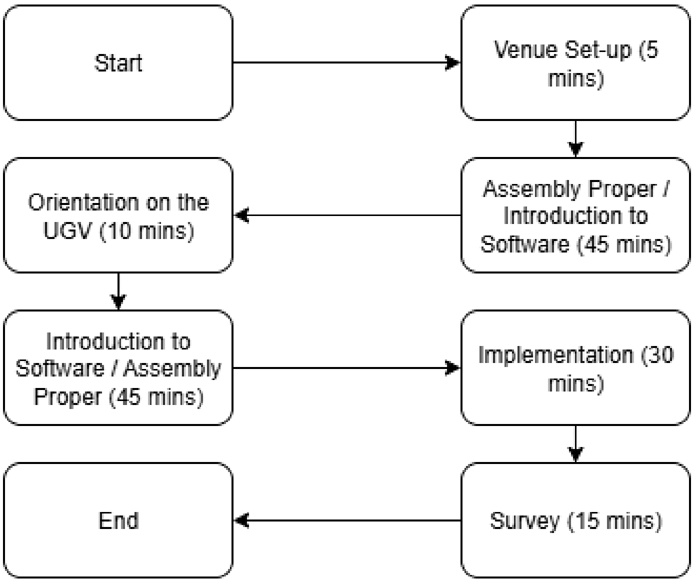


Figure 32. Educational workshop program flow

As shown in Figure 33, the new chassis design had a weighted average of responses of 4.59 out of 5, with the snap-fit fastener at 4.64. The 3D printing finish, 3D printing material used, color coding design, wheel performance, and hardware tolerance received ratings ranging from 4.64 to 4.72, all of which exceed satisfactory levels. There were minimal comments and suggestions about some pieces being difficult to attach due to some rough edges and finishes, the need for notches for electronic covers, and the need for more detailed descriptions of each piece.

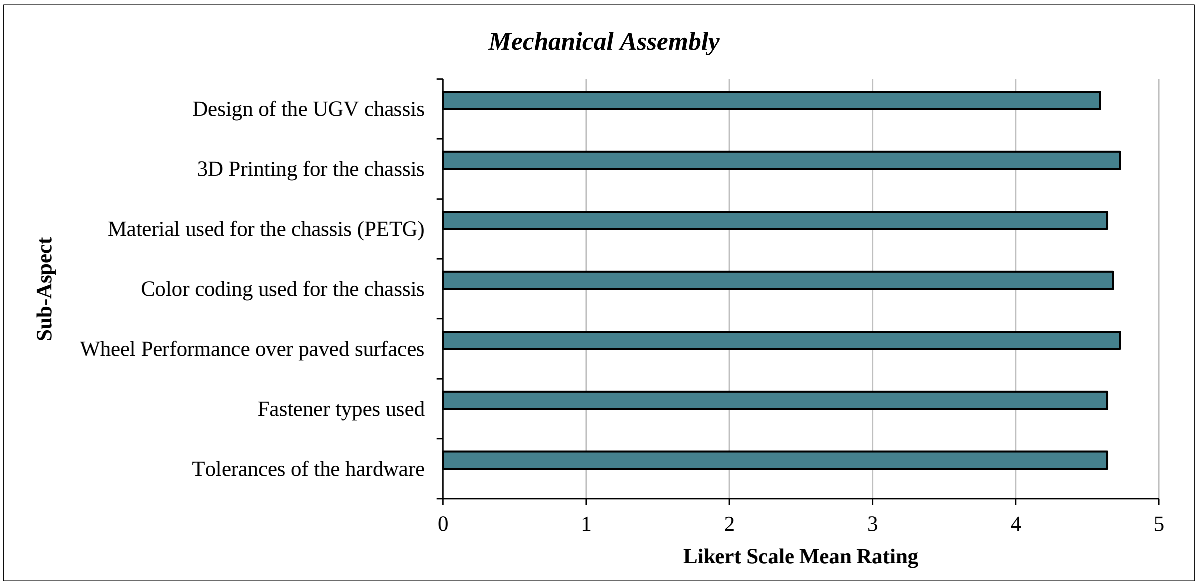


Figure 33. Mechanical Assembly Assessment

In the electronics assembly, as shown in Figure 34, the accuracy of the camera module was 4.59 out of 5, 4.64 for the incorporated sensors, such as the ultrasonic sensor and computer vision-capable camera, 4.59 for electrical connection effectiveness, 4.82 for battery capacity, and performance of the Pixhawk at 4.68. The accuracy of the GPS module was rated at 4.59; however, some outdoor testing activities could not be conducted due to inclement weather on some days of the workshop, which may have contributed to further inaccuracy in this value. This is the testing area—an open space at the St. La Salle Building in De La Salle University - Manila, being surrounded by multiple construction sites and buildings that may significantly affect the GPS signal, despite being the most viable and practical area for outdoor testing. The wiring management was rated 4.41, which is higher than satisfactory but lower than the other categories. This could be because of the absence of wire clips to prevent wires from spreading out, which caused a disorganized appearance, as noted by some respondents. A comparison of all ratings is shown in Figure 34.

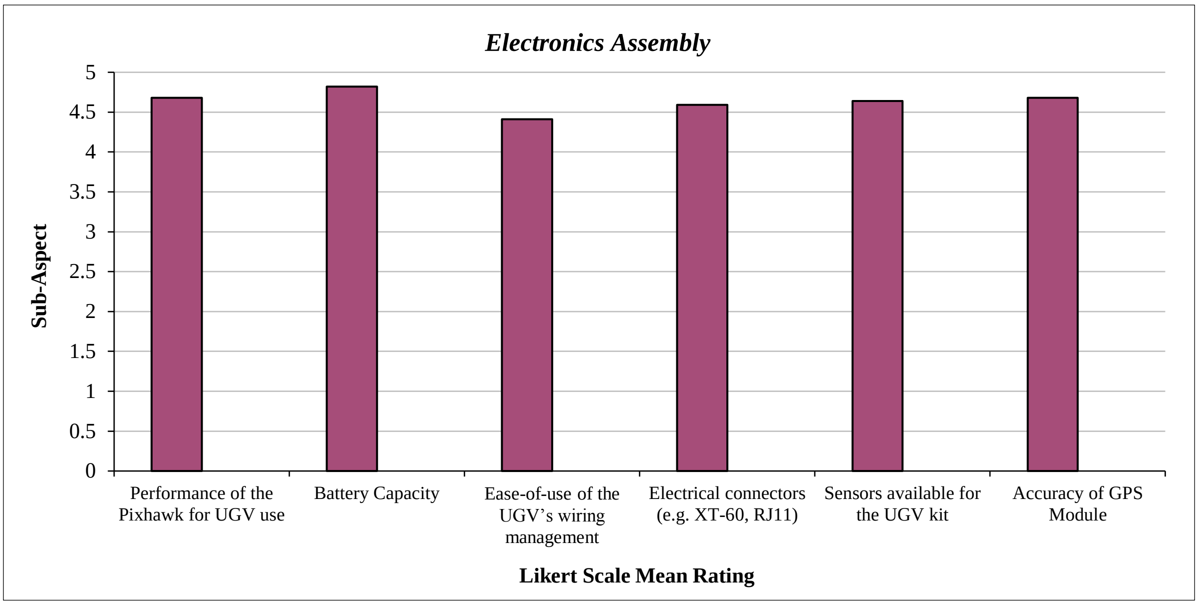


Figure 34. Electronics Assembly Assessment

The software assessment also received positive feedback for its user engagement in coding and creating missions using Mission Planner, especially when guided by the educational module. Primarily testing the Ubuntu software, the weighted average score for its user interface was 4.73, 4.33 for its capabilities, 4.77 for compatibility with Mission Planner, and 4.68 for its open-source nature. It also received 4.73 for its learning curve, highlighting that the provided code samples and educational module allowed respondents to grasp the concepts easily. These ratings are presented in Figure 35. Respondents' insights further affirmed that using Ubuntu together with the modules was both engaging and enjoyable.

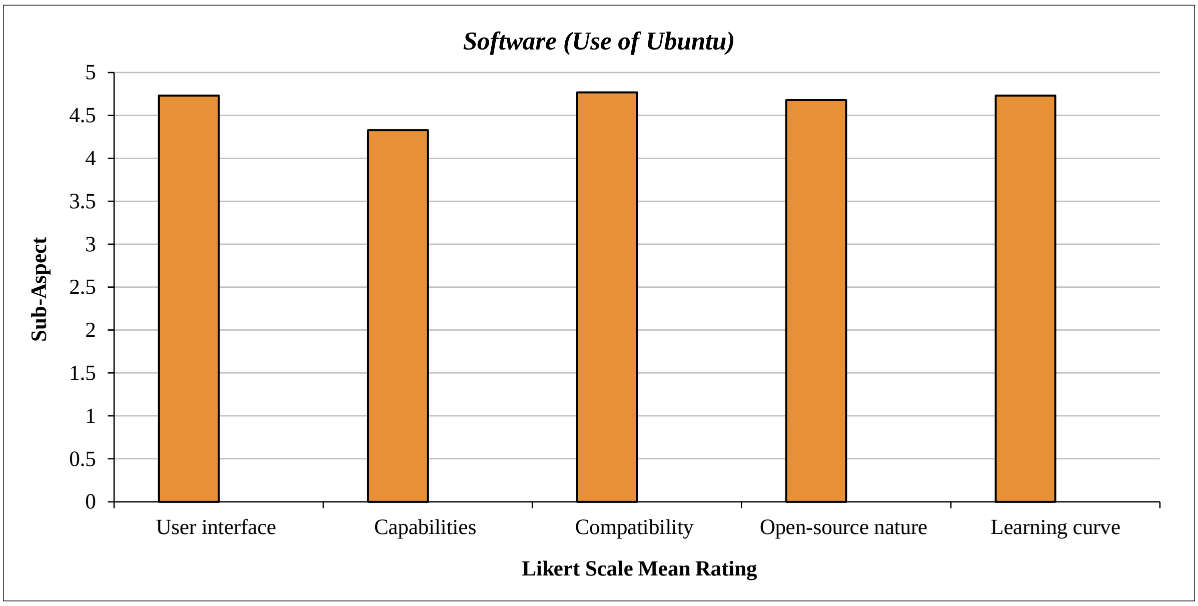


Figure 35. Software Assessment

Looking at overall UGV performance, Figure 36 shows the weighted averages: 4.45 for speed, acceleration 4.59, tuning capability 4.59, maximum incline 4.55, and navigation performance 4.50. Respondents noted that navigation performance did not always align with the Mission Planner simulation, which may have been due to GPS drift. Additionally, respondents suggested reducing the robot's sensitivity to manually input via the remote control and adjusting its speed to improve overall control.

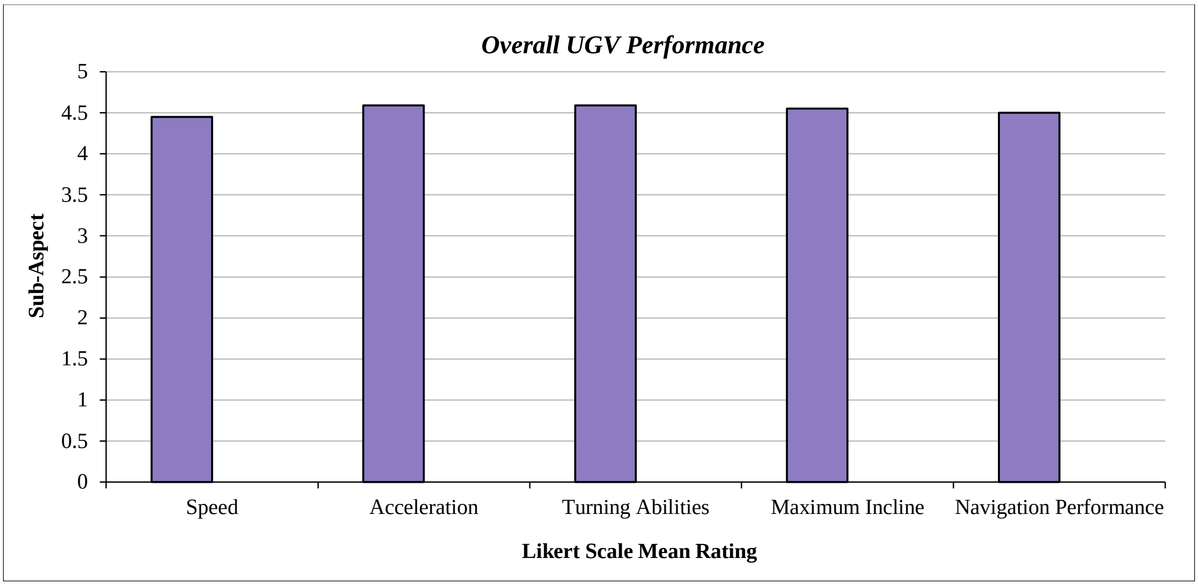


Figure 36. Overall UGV Assessment

For the educational value of the robot and the assessment of the educational module, student discussion data regarding ArduPilot Mission Planner and the Linux operating system yielded weighted averages between 4.73 and 4.82, as shown in Figure 37. Troubleshooting program errors and visual summary presentation aids were both rated 4.73, while the learning curve received a weighted average of 4.59. Respondents noted that they wanted more detailed descriptions of the mechanical assembly parts and a more explicit troubleshooting procedure. Nonetheless, students reported a high level of satisfaction with the online learning module and found it to be a useful learning tool. Although there was no substantial change in robotics interest between pre- and post-workshop surveys, most students (68.2%) indicated that an in-depth learning experience with X-Lakbay, integrated into the senior high school robotics course, would enhance their robotics skills. Additionally, 9.1% of students noted that greater confidence or experience in robotics could further improve learning gains.

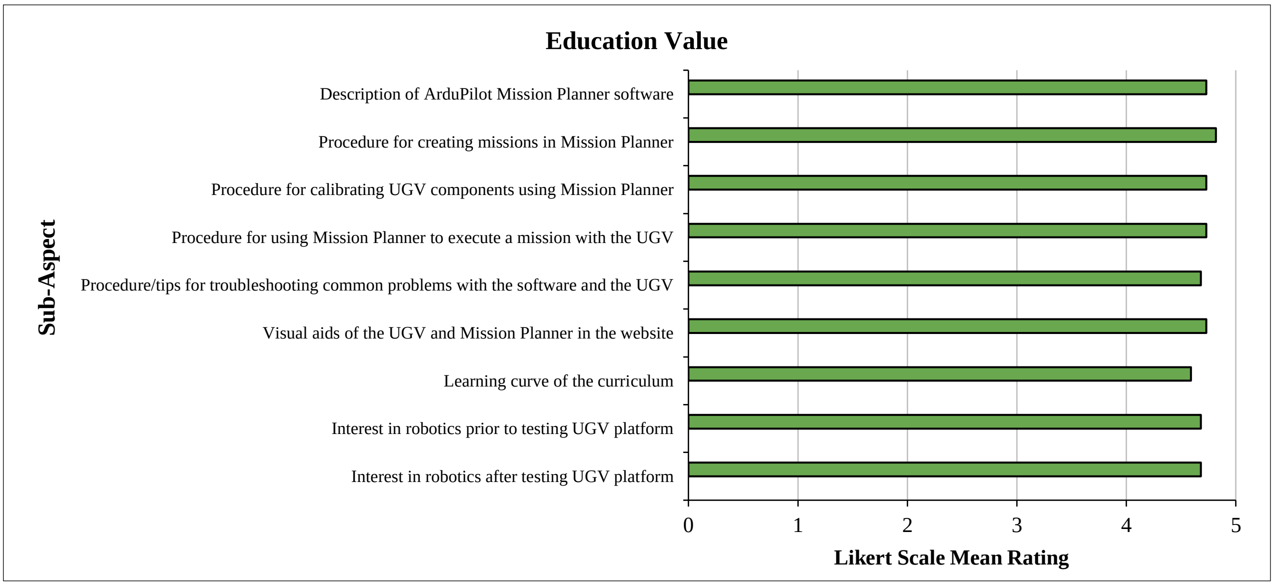


Figure 37. Educational Value Assessment

The focus testing conducted in this study, despite sharing similar criteria and parameters with Corpuz et al. [5], cannot be directly correlated with that study, as the participants in this examination were not as diverse. While the previous study included students from tertiary and secondary levels, as well as an instructor, this study focused exclusively on secondary-level students from different schools and regions across the country. Therefore, it cannot be concluded whether the new design of X-Lakbay is more effective in educating students than previous iterations.

Overall, these findings indicate that the X-Lakbay UGV could serve as a viable alternative to proprietary educational kits, engaging participants effectively, and achieving comparable learning outcomes. Participants who had prior experience with commercial robotics platforms reported that building the snap-fit frame, wiring the electronics, and programming missions in Ubuntu and ArduPilot was just as engaging and intuitive as working with LEGO- or VEX-based activities. Average ratings for Mechanical Assembly (4.59), Software (4.63–4.77), and Educational Value (4.59–4.82) suggest that students successfully translated building the robot into programming it to move autonomously within a single 2.5-hour workshop, providing an experience similar to proprietary kits without being restricted to a closed ecosystem.

Furthermore, the use of standard components (Raspberry Pi, Pixhawk, USB camera) and open-source tools (DroneKit, ArduPilot, Ubuntu) allows schools to avoid vendor lock-in for future repairs or replacements. Instead of purchasing additional licensed bricks or motors, teachers can print additional frames, reuse electronics from existing robotic designs, or reconfigure the robot for different learning activities. This modularity and reusability position X-Lakbay not merely as a low-cost alternative, but as a suitable pedagogical platform comparable to LEGO- or VEX-style kits in resource-constrained educational environments.

4.6. Cost-Effectiveness

The redesigned UGV must be priced reasonably and prioritize affordability for educational purposes compared to other products selling on the market to be competitive in the Philippine marketplace. Current competitors for educational kits and regional contests are VEX Robotics and LEGO Mindstorms. Both companies produce robot kits with modular parts. However, these are often only modular in that the software and hardware modifications are only possible within their existing proprietary product lines.

Table 5 presents a comparison of various robotics kits available in the Philippines, adapted from previous research. Prices for these kits have increased significantly, now ranging from approximately Php 28,540 to Php 65,000 due to fluctuations in the exchange rate. The previous UGV version cost Php 15,600.00 to build. With the redesign, including added components such as the Raspberry Pi companion computer, the estimated cost increased by Php 3,353.84, bringing the total to Php 18,953.84. Applying a 50% markup results in a suggested retail price (SRP) of Php 28,430.76. This pricing remains competitive, particularly when compared to similar kits such as the Drone Dojo, which costs nearly twice as much. The updated UGV offers enhanced features, supports outdoor use, and remains more affordable than most local options, making it a strong alternative in terms of both functionality and cost-effectiveness.

Table 5. A comparison of the features of complete robotics kits available in the Philippines. Adapted from Corpuz et al. (2024) [5]

Model	Open-Source	Camera	Distance Sensor	Line Sensor	Bumper Sensor	GPS	Telemetry Module	Optional Co-processor	Cost (in PHP)
VEX IQ Education Kit (2nd gen.)			Yes	Yes	Yes				28,540.56
VEX Robotics V5 Classroom Starter Kit			Yes	Yes	Yes				47,596.37
VEX Robotics EXP Classroom Starter Kit			Yes	Yes	Yes				47,596.37
LEGO Mindstorms EV3 Inventor Robotics Kit			Yes	Yes	Yes				29,344.12
Drone Dojo Smart Rover Kit	Yes	Yes				Yes	Yes	Yes	50,731.47
UGOT Robot-01	Yes	Yes	Yes				Yes	Yes	65,000.00
Makeblock MBot Ultimate	Yes		Yes	Yes			Yes	Yes	33,365.00
Our UGV Design	Yes	Yes	Yes			Yes	Yes	Yes	28, 430.76

5. Conclusion

This study successfully developed a cost-effective and modular unmanned ground vehicle (UGV) designed as an educational platform supported by structured learning modules. The system addresses the need for accessible robotics education by introducing an assembly methodology that reduces both setup complexity and duration, enabling faster deployment in instructional environments. Furthermore, the addition of a camera and a companion computer facilitates configuration, autonomous control, and practical experimentation, enhancing student engagement in robotics programming and application-based learning. The results of the development process demonstrate that modularity, affordability, and educational applicability can be achieved simultaneously without compromising functional performance. Consequently, the system contributes to ongoing efforts to expand robotics education across the country, including in academic institutions with limited resources. These findings highlight the feasibility of integrating advanced robotics technologies into educational settings while maintaining cost-effectiveness and ease of use.

Despite its demonstrated capabilities, several areas require further improvement. These include the utilization of spare parts, tolerance testing of manufactured components, improvements in wire management and power delivery, enhancement of software features, and further development of the educational modules. Addressing these limitations is expected to improve system robustness, operational reliability, and effectiveness as a learning instrument. Implementing these enhancements may also strengthen the platform’s potential for future commercialization, expanding its role in supporting STEM learning and enhancing technology-based education across Philippine institutions. The findings and recommendations presented in this study may additionally serve as a practical reference for those seeking to replicate, validate, or extend this work, as well as for those aiming to develop their own scalable, accessible, and effective educational robotics systems.

6. Declarations

6.1. Author Contributions

Conceptualization, A.Y.C.; methodology, I.A.C.F. and R.P.E.; software, J.E.C.L.; validation, W.C.P.C. and I.A.C.F.; formal analysis, I.A.C.F., J.E.C.L., R.P.E., and W.C.P.C.; investigation, R.P.E.; resources, N.G.L.; data curation, W.C.P.C.; writing—original draft preparation, I.A.C.F.; writing—review and editing, J.E.C.L., N.G.L., and W.C.P.C.; visualization, N.G.L.; supervision, A.Y.C.; project administration, J.E.C.L.; funding acquisition, N.G.L. and J.E.C.L. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding and Acknowledgments

The authors acknowledge the support provided by the Unmanned Aerial Vehicle Laboratory of De La Salle University Manila, particularly in supplying the major components used in the research.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

7. References

[1] Palconit, M. G. B., Concepcion, R. S., Culaba, A. B., Dadios, E. P., Alejandrino, J. D., Rosales, M. A., & Magsumbol, J. A. V. (2020). Diffusion of Robotics in the Philippines: Impact of Policies, Laws, R D, and Word of Mouth. In R. laws & D. (Eds.), 2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management, HNICEM 2020, 1–6. doi:10.1109/HNICEM51456.2020.9400094.

[2] Montemayor, T. (2018). Why include robotics in PH school curriculum. Philippine News Agency, Quezon City, Philippines. Available online: https://www.pna.gov.ph/articles/1040784 (accessed on November 2025).

[3] Darmawansah, D., Hwang, G. J., Chen, M. R. A., & Liang, J. C. (2023). Trends and research foci of robotics-based STEM education: a systematic review from diverse angles based on the technology-based learning model. International Journal of STEM Education, 10(1), 12. doi:10.1186/s40594-023-00400-3.

[4] Unmanned Systems Technology. (2025). Unmanned ground vehicles (UGV) manufacturers. Unmanned Systems Technology, Dorset, United Kingdom Available online: https://www.unmannedsystemstechnology.com/expo/unmanned-ground-vehicle-ugv-manufacturers/ (accessed on November 2025).

[5] Corpuz, C. S. D. D., Cuaycong, G. A., Diño, R. G. D., Estacio, R. C., Señar, G. A. C., & Chua, A. Y. (2025). A Novel Cost-Effective Unmanned Ground Vehicle Platform for Robotics Education. HighTech and Innovation Journal, 6(1), 303–327. doi:10.28991/HIJ-2025-06-01-020.

[6] Proven Robotics. (2025). Robotics in STEM: Transforming classrooms and inspiring creativity. Proven Robotics, Riyadh, Saudi Arabia. Available online: https://provenrobotics.ai/robotics-in-stem-education-2/ (accessed on November 2025).

[7] Evans, M. K. (2016). Member collaboration: LEGO Mindstorms. MIT Media Lab, Massachusetts, United States. Available online: https://www.media.mit.edu/posts/member-collaboration-lego-s-mindstorms/ (accessed on November 2025).

[8] Tech-Labs. (2025). STEM Robotics Drones Lab – UAV (45-Hour). Tech-Labs, Texas, United States. Available online: https://tech-labs.com/products/stem-robotics-drones-lab-uav-45-hour (accessed on November 2025).

[9] VEX Robotics. (2025). The STEM Solution for Your Classroom. VEX Robotics, Greenville, Texas, United States. Available online: https://www.vexrobotics.com/education/overview (accessed on November 2025).

[10] Felta Multi-Media Inc. (2025). Philippines Robotics Academy launched in several schools. Felta Multi-Media Inc., Quezon City, Philippines. Available online: https://www.felta.ph/2012/philippines-robotics-academy-launched-in-several-schools/ (accessed on November 2025).

[11] Ouyang, F., & Xu, W. (2024). The effects of educational robotics in STEM education: a multilevel meta-analysis. International Journal of STEM Education, 11(1), 7. doi:10.1186/s40594-024-00469-4.

[12] Mecantina, J. (2025). The Lived Experiences of Students Using Robotics as a Tool in Learning Physics. Aloysian Interdisciplinary Journal of Social Sciences, Education, and Allied Fields, 1(10), 78-95. doi:10.5281/zenodo.17451729.

[13] Alonso-García, S., Rodríguez Fuentes, A. V., Ramos Navas-Parejo, M., & Victoria-Maldonado, J. J. (2024). Enhancing computational thinking in early childhood education with educational robotics: A meta-analysis. Heliyon, 10(13), 33249. doi:10.1016/j.heliyon.2024.e33249.

[14] Gonzales, M., Andal, E., Ching, D., Gaffud, M., & Tabo, E. (2021). Assessing the Efficacy of RoboTeach Extension Project on Public School Teachers. International Journal of Educational Management and Development Studies, 2(3), 78–100. doi:10.53378/348742.

[15] Galino, J. R., & Tanaka, H. (2021). Initiatives and development of secondary school robotics education in camarin high school. Journal of Robotics, Networking and Artificial Life, 8(1), 60–65. doi:10.2991/jrnal.k.210521.014.

[16] Ahillon Jr, R. C., Cadiong, S. R. C., Tomeldan, A. B., Belgira, A. C. B., Ayaya, A. R. A., Paredes, A. D. M., ... & Quindra, K. C. (2025). Evaluation of Robotics Class in a Private School in the Philippines. ASEAN Journal of Educational Research and Technology, 4(1), 23-32.

[17] Pangandaman, H. K. (2019). Philippine Higher Education Vis-À-Vis Education 4.0: A Scoping Review. International Journal of Advanced Research and Publications, 3(3), 65–69.

[18] The LEGO Group. (2025). LEGO MINDSTORMS EV3 (31313). The LEGO Group, Billund, Denmark. Available online: https://www.lego.com/en-us/product/lego-mindstorms-ev3-31313 (accessed on November 2025).

[19] PHR Robotics. (2025). UGOT robot. PHR Robotics, Seoul, South Korea. Available online: https://www.phr-robotics.com/en/robot-ugot-phr-robotics (accessed on November 2025).

[20] Ubuy. (2025). Makeblock DIY Ultimate Robot Kit – Premium Quality 10-in-1 Robot. Ubuy, Kuwait City, Kuwait. Available online: https://www.ubuy.com.ph/product/1GM2BS4OW-makeblock-diy-ultimate-robot-kit-premium-quality-10-in-1-robot-st (accessed on November 2025).

[21] Dojo for Drones. (2025). Rover Kit. Dojo for Drones, California, United States. Available online: https://dojofordrones.com/product/rover-kit/ (accessed on November 2025).

[22] TechSabado. (2025). Education: Radenta, Shape Robotics bring modular tech education to PH schools. TechSabado, Quezon City, Philippines. Available online: https://techsabado.com/2025/06/11/education-radenta-shape-robotics-bring-modular-tech-education-to-ph-schools/ (accessed on November 2025).

[23] The Asianparent. (2025). First learning center for coding and robotics opens in Manila – Nullspace. theAsianparent, Singapore. Available online: https://ph.theasianparent.com/first-learning-center-for-coding-and-robotics-opens-in-manila-nullspace (accessed on November 2025).

[24] Pearson, A. (2020). World's first jet-powered, 3D printed UAV tops 150 mph with lightweight Stratasys materials. Minnesota, United States. Available online: https://www.stratasys.com/en/resources/blog/aurora-uav-3d-printing/ (accessed on July 2025).

[25] Corselli, A. (2023). 3D-printed AUV aims to thwart climate change. Tech Briefs Media Group, New York, United States. Available online: https://www.techbriefs.com/component/content/article/47729-3d-printed-auv-aims-to-thwart-climate-change (accessed on November 2025).

[26] Yang, L., Xiang, X., Kong, D., & Yang, S. (2024). Small Modular AUV Based on 3D Printing Technology: Design, Implementation and Experimental Validation. Brodogradnja, 75(1), 75104. doi:10.21278/brod75104.

[27] Molloy, D. (2016). Exploring Raspberry Pi: interfacing to the real world with embedded Linux. John Wiley & Sons, New Jersey, United States. doi:10.1002/9781119211051.

[28] Karalekas, G., Vologiannidis, S., & Kalomiros, J. (2020). Europa: A case study for teaching sensors, data acquisition and robotics via a ROS-based educational robot. Sensors (Switzerland), 20(9), 2469. doi:10.3390/s20092469.

[29] Pulungan, A. B., Putra, Z. Y., Sidiqi, A. R., Hamdani, & Parigalan, K. E. (2024). Drone Kit-Python for Autonomous Quadcopter Navigation. International Journal on Informatics Visualization, 8(3), 1287–1294. doi:10.62527/joiv.8.3.2301.

[30] Oyekola, P., Lambrache, N., Mohamed, A., Pumwa, J., Olaru, L., N'Drelan, B., & Ebere, C. (2019). Design and construction of an unmanned ground vehicle. Proceedings of the International Conference on Industrial Engineering and Operations Management, 487–496.

[31] Arrow Electronics, Inc. (2019). Introduction to Bayer filters. Arrow Electronics, Inc., Colorado, United States. Available online: https://www.arrow.com/en/research-and-events/articles/introduction-to-bayer-filters (accessed on November 2025).

[32] Qu, J. G., Jiang, Z. Y., & Zhang, J. F. (2022). Investigation on lithium-ion battery degradation induced by combined effect of current rate and operating temperature during fast charging. Journal of Energy Storage, 52, 104811. doi:10.1016/j.est.2022.104811.

[33] Willmon, W. G. (2021). Design, Manufacture, and Test of a Hybrid Aerial-Ground Robotic Platform. Master's Thesis, University of Denver, Colorado, United States.

[34] Piaget, J., & Wedgwood, S. (2015). The grasp of consciousness: Action and concept in the young child. In The Grasp of Consciousness: Action and Concept in the Young Child. Psychology Press, London, United States. doi:10.4324/9781315722382.

[35] Kolb, D. A. (1984). Experiential Learning: Experience as the Source of Learning and Development. Prentice Hall, New Jersey, United States. doi:10.1016/B978-0-7506-7223-8.50017-4.

[36] Spady, W. G. (1994). Outcome-Based Education: Critical Issues and Answers. American Association of School Administrators, Arlington, Virginia. American Association of School Administrators.

[37] Spady, W. G. (1994). Outcome-Based Education: Critical Issues and Answers. American Association of School Administrators, Arlington, United States.

[38] Guston, D. (2012). National Institute of Standards and Technology (U.S.). Encyclopedia of Nanoscience and Society, SAGE Publications, New York, United States. doi:10.4135/9781412972093.n332.