robotics

MDPI

Systematic Review

ROS-Enabled DIY and Open-Source Wheeled Robots for Higher Education Learning and Competitions: A Systematic Review

Rúben Pereira [1], Benedita Malheiro [1,2] and Manuel F. Silva [1,2,*]

[1] ISEP, Polytechnic of Porto, Rua Dr. António Bernardino de Almeida, 431, 4249-015 Porto, Portugal; 1181915@isep.ipp.pt (R.P.); mbm@isep.ipp.pt (B.M.)
[2] INESC TEC, Campus da FEUP, Rua Roberto Frias, 4200-465 Porto, Portugal
* Correspondence: mss@isep.ipp.pt

Abstract

This study systematically characterizes Do It Yourself (DIY) and open-source wheeled robotic platforms used in higher education and academic competitions. It also analyzes Robot Operating System (ROS)-based designs with respect to real-time performance and multi-sensor integration, following Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2020 guidelines. A total of 20 high-quality studies were identified across five major digital libraries (Dimensions, Web of Science, SpringerLink, ScienceDirect, and IEEE Xplore), which were searched on 12 January 2026. Eligibility was restricted to peer-reviewed English-language studies published between 2005 and 2026 that explicitly implement ROS-based wheeled platforms in higher education contexts. Results were synthesized through qualitative analysis using a structured data extraction form implemented in the Parsifal systematic review platform. Methodological quality and risk of bias were assessed using a structured appraisal checklist. The results show a dominant trend toward distributed dual-processor architectures, which separate low-level real-time control from high-level processing. Most platforms target an accessible price range of 50 € to 500 € for open-source and DIY platforms. ROS has emerged as the standard middleware, enabling multi-sensor integration and supporting digital twin workflows. There is also a clear shift toward open-source hardware and Three-Dimensional (3D)-printed modular designs, which reduce production costs. However, challenges remain, including software obsolescence and the lack of maintenance plans. The findings highlight the need for interoperable reference architectures and automated deployment workflows to ensure long-term sustainability. Evidence is limited by heterogeneity, inconsistent reporting, and small sample sizes, which introduce risks of bias and imprecision. This review was formally registered with protocols.io.

Keywords: DIY robotics; open-source robotics; wheeled mobile robots; educational robotics; academic robotics competitions; robot operating system (ROS/ROS 2); systematic review; PRISMA

Check for updates

Academic Editor: Giulio Reina

Received: 14 April 2026
Revised: 22 June 2026
Accepted: 26 June 2026
Published: 30 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

This systematic review assesses and summarizes the current state of the art in wheeled robots equipped with ROS for use in engineering programs in higher education and academic competitions. Historically, the integration of mobile robotics into Science, Technology, Engineering and Maths (STEM) curricula has been saturated with toy-like kits primarily designed for primary education (e.g., Bee-bot, Lego Mindstorms) [1,2]. However, there

 https://doi.org/10.3390/robotics15070123

is a high demand for university-level platforms capable of supporting Industry 4.0/5.0 skills, specifically ROS, computer vision, and embedded co-design. In these advanced contexts, DIY and open-source hardware are heavily preferred over proprietary "black-box" systems. Open architectures expose students to real-world complexity, hardware assembly, and hardware–software integration.

Furthermore, it is important to recognize that not all robotic platforms are general-purpose. Many are designed for specific objectives, whether focused on teaching fundamental robotics concepts (e.g., locomotion kinematics, human–robot interaction) or tailored for specific competition domains (e.g., social robotics, robotic soccer, autonomous racing). This review fills the current gap in the literature by synthesizing the technical architectures of affordable platforms, ultimately helping educators avoid software obsolescence and logistical burdens. It aims to guide the design of such dual-purpose platforms, which implement *de facto* standard middleware and promote teamwork and learning in the field of robotics [3–5].

Educational robotics uses physical robots to support learning by doing. In higher education, it is often used to learn programming, control, electronics, and perception (e.g., using platforms like EUROPA or TurtleBot 3) [2,6]. In competitions, it is used to test systems under temporal constraints and to benchmark solutions (e.g., in the RobotCraft or Robot@Factory 4.0 intensive programs) [4,5]. In both cases, the platform should be safe, easy to maintain, and robust in repeated use.

A wheeled robot platform usually includes a mobile base, a power system, sensors, motors, wheels, and computing units. A common pattern is to split the system into two levels: a microcontroller (e.g., Arduino or STM32) for low-level motor control and time-critical tasks, and a single-board computer (e.g., Raspberry Pi or Intel NUC) for high-level functions such as mapping, planning, and user interfaces [4,7,8]. This separation maintains control stability and allows for complex tasks.

Software is now a major part of educational robotics. Many recent platforms rely on ROS to support modular code development, sensor integration, and reuse of existing packages. Modular hardware platforms such as TurtleBot (https://www.turtlebot.com/turtlebot3/, (accessed on 12 January 2026)) and open designs such as FOSSBot [9] with ROS can shorten the setup time and promote reuse between cohorts. In practice, this can reduce the effort of teaching staff and give students a clean development workflow [6,9]. This review focuses on wheeled robots that can support both learning and competition needs, identifies common design choices, and reports gaps that affect long-term use and reproducibility. Specifically, this study investigates the electronic and hardware architectures that characterize current wheeled robotic platforms used in educational and competitive settings, with particular attention to how these architectures are implemented in ROS-based designs. Emphasis is placed on real-time performance constraints and the integration of heterogeneous sensor modalities.

The remainder of this article is organized as follows. Section 2 details the methodology applied in this systematic review, including the PRISMA guidelines, search strategy, and eligibility criteria. Section 3 presents the results, categorizing the findings by hardware architectures, software middleware, and manufacturing approaches. Section 4 discusses the results in depth, answering the research questions while identifying persistent challenges and pedagogical gaps. Finally, Section 5 concludes the study and outlines its methodological limitations.

https://doi.org/10.3390/robotics15070123

2. Method

This section describes the literature review of educational- and competition-oriented wheeled robotic platforms, gathering design decisions, identifying common technical choices, and recurring gaps.

2.1. Review Scope and Research Questions

The scope was limited to wheeled platforms used in higher education courses and academic competitions. The review questions (Table 1) focus on (*i*) hardware architecture and (*ii*) the adoption of ROS-based middleware for integration and performance.

Table 1. Research questions.

ID	Question
RQ1	What are the prevailing electronic and hardware architectures in wheeled robotic platforms used for both student-centric learning and competitions?
RQ2	What are the most relevant ROS-based educational robotic platform designs, particularly regarding real-time performance and multi-sensor integration?

2.2. Protocol, Tools, and Reporting

The review follows the PRISMA 2020 reporting guidelines, which provide a 27-item checklist and a standard flow diagram to describe how records are identified, screened, assessed for eligibility, and included in the final synthesis [10–12]. In this section, PRISMA is used to guide the review process in a clear and traceable way. The corresponding flow diagram is shown in Figure 1.

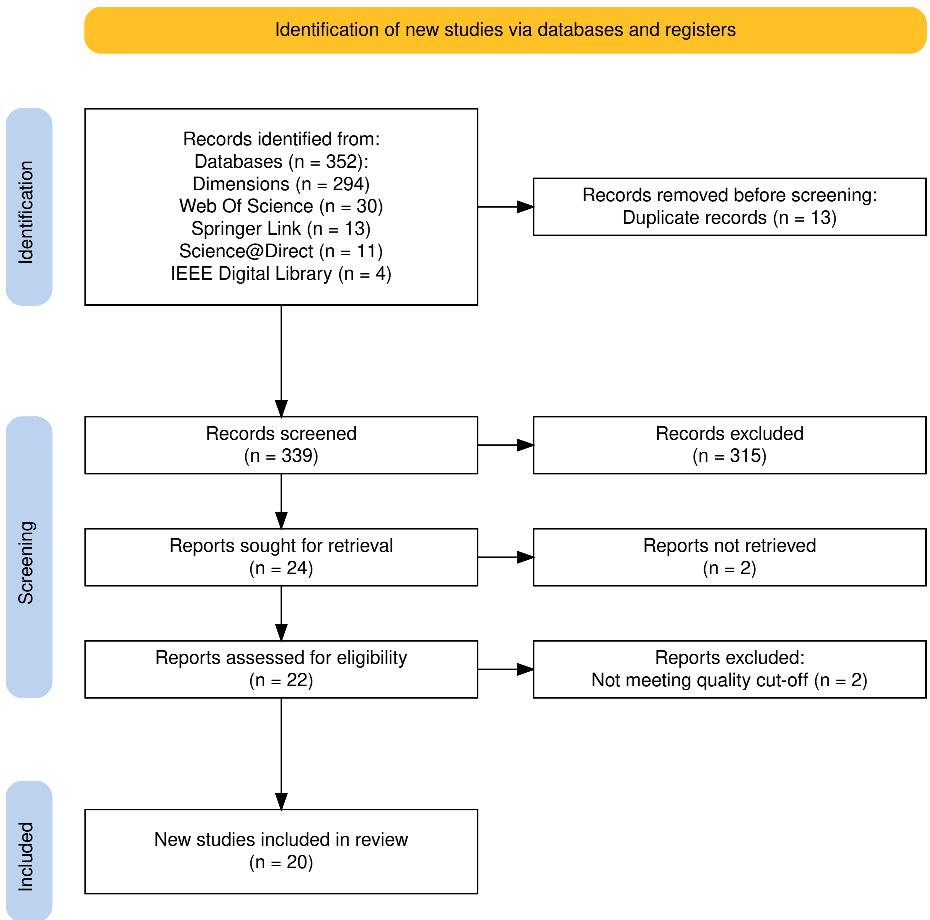


Figure 1. PRISMA flow diagram [13].

The protocol defines (*i*) the scope and research questions (Table 1) of the review, (*ii*) the digital libraries to be searched (Table 2), (*iii*) inclusion and exclusion criteria, and (*iv*) the data extraction and quality assessment plan. The workflow relied on Parsifal (https://parsif.al/ (accessed on 12 February 2026)), an online tool for planning, conducting, and reporting systematic literature reviews. Parsifal was used to store the protocol, import records, record screening decisions, and manage quality assessment and data extraction forms. This review was conducted and managed using Parsifal, with the protocol registered on protocols.io [14]. No amendments were made to the protocol during the review process.

Table 2. Databases searched and records identified.

Database	Records (*n*)
Dimensions [1]	294
Web of Science [2]	30
SpringerLink [3]	13
ScienceDirect [4]	11
IEEE Xplore [5]	4
Total	**352**

[1] https://www.dimensions.ai/, (accessed on 12 January 2026), [2] https://www.webofscience.com/, (accessed on 12 January 2026), [3] https://link.springer.com/, (accessed on 12 January 2026), [4] https://www.sciencedirect.com/, (accessed on 12 January 2026), [5] https://ieeexplore.ieee.org, (accessed on 12 January 2026).

The PRISMA diagram in Figure 1 was produced using the PRISMA 2020 tool [13]. All values reported in the diagram and in the supporting tables and figures were obtained and exported from Parsifal to reduce manual transcription errors.

2.3. Search, Screening, and Selection

2.3.1. Search Strategy

The searches were executed on 12 January 2026 in five digital libraries: Dimensions (https://www.dimensions.ai/, (accessed on 12 January 2026)), Web of Science (https://www.webofscience.com/, (accessed on 12 January 2026)), SpringerLink (https://link.springer.com/, (accessed on 12 January 2026)), ScienceDirect (https://www.sciencedirect.com/, (accessed on 12 January 2026)), and IEEE Xplore (https://ieeexplore.ieee.org, (accessed on 12 January 2026)). The number of records retrieved from each source is reported in Table 2. Records were manually downloaded from each interface (including bibliographic metadata and abstracts where available) and then imported into Parsifal for screening and tracking. ACM Digital Library (https://dl.acm.org/, (accessed on 12 January 2026)) was also considered during protocol refinement; however, the final query returned no eligible records. Scopus (https://www.scopus.com/, (accessed on 12 January 2026)) was not searched because institutional access was not available. IEEE Robotics and Automation Society conference papers were considered through IEEE Xplore, and no additional IEEE-specific filters were applied.

2.3.2. Search Strings

The search terms were derived from the Population, Intervention, Comparison, Outcome, and Context (PICOC) elements (Table 3) and organized into three concept blocks: (*i*) higher education, (*ii*) robotic platforms and ROS, and (*iii*) educational robotics competitions. This level of reporting is also used in recent robotics systematic reviews [12,15]. The Population element consists of higher education settings or students in higher education; the Intervention element corresponds to robotic platforms and ROS-based software (including ROS, ROS 2, and related operating system terms); the Outcome element is not explicitly defined in the search string; and the Context element is that of educational robotics and academic competitions. The Comparison element was omitted.

Table 3. PICOC keywords (matched to search string).

	Keywords
Population	"Higher education"
Intervention	platform, ROS, ROS2, "ROS 2"
Comparison	-
Outcome	-
Context	"Educational robotics", "Competition"

These blocks were combined using Boolean operators. Table 4 presents the base Boolean expression. Table 5 lists the corresponding expressions used in each database interface, together with the additional filters applied per database. The baseline logic was adapted to the syntax of each digital library and, where possible, applied to the title, abstract, and keyword fields.

Table 4. Base Boolean search string used to retrieve candidate studies.

Concept Block	Terms (Combined with OR)
Combined query	''Educational robotics'' AND platform AND (ROS OR ROS2 OR ''ROS 2'') AND (''Higher education'' OR ''Competition'')

Table 5. Database-specific implementation of the base search query.

Database	Search String	Additional Filters
Dimensions	(''Educational robotics'' AND platform AND (ROS OR ROS2 OR ''ROS 2'') AND (''Higher education'' OR ''Competition''))	Publication Type (Preprint, Proceeding, and Article); Publication Year (2005 to 2026); Open Access
Web of Science	(''Educational robotics'' AND platform AND (ROS OR ROS2 OR ''ROS 2'') AND (''Higher education'' OR ''Competition''))	—
SpringerLink	(''Educational robotics'' AND platform AND (ROS OR ROS2 OR ''ROS 2'') AND (''Higher education'' OR ''Competition''))	Content type (Conference paper); Data Published 2005 to 2026); Languages (English)
ScienceDirect	(''Educational robotics'' AND platform AND (ROS OR ROS2 OR ''ROS 2'') AND (''Higher education'' OR ''Competition''))	Article Type (Review articles, and Research Articles); Languages (English)
IEEE Xplore	(''Educational robotics'' AND platform AND (ROS OR ROS2 OR ''ROS 2'') AND (''Higher education'' OR ''Competition''))	—

The application of additional filters varied by database according to their respective search functionalities and content coverage. For Dimensions, filters for publication type (Preprint, Proceedings, and Article) and Open Access status were applied to ensure a focus on peer-reviewed and openly accessible literature. The publication year range of 2005 to 2026 was selected to capture the earliest development and initial release of ROS, which originated in 2007 with the first code commit made on 7 November 2007 [16]. This range provides a buffer to include foundational research leading up to and following the first code release, ensuring comprehensive coverage of the formative period of ROS. For SpringerLink, filters for content type (Conference paper) and language (English) were necessary due to the platform's broad, multidisciplinary scope, which includes a significant number of non-English and non-peer-reviewed items. Similarly, ScienceDirect required filters for article type (Review articles and Research Articles) and language (English) to isolate substantive research contributions from its extensive database. No additional filters were applied to Web of Science and IEEE Xplore, as their core collections are already highly curated and predominantly consist of peer-reviewed, English-language publications relevant to engineering and computer science, thus aligning well with the scope of this review without further restriction. Two reviewers independently screened titles and abstracts, followed by full-text screening. Disagreements were resolved through discussion.

2.3.3. Duplicate Removal and Screening

After importing the records into Parsifal, duplicate detection was performed using the available identifiers (title, authors, year, and Digital Object Identifier (DOI)). After removing duplicates (n =13), 339 records were screened by title and abstract (Figure 1). At this stage, 315 records were excluded according to the predefined criteria. The exclusion reasons are reported in Table 6 and Figure 2.

Table 6. Exclusion codes used during screening (n = 315).

Code	Reason	*n*	%
E1	Irrelevant platform	103	32.70
E2	School only (primary/secondary)	56	17.78
E3	Aerial/marine/industrial only	39	12.38
E4	Simulation only (no physical robot)	37	11.75
E5	No hardware	34	10.79
E6	Not education/competition context	28	8.89
E7	Not English	18	5.71

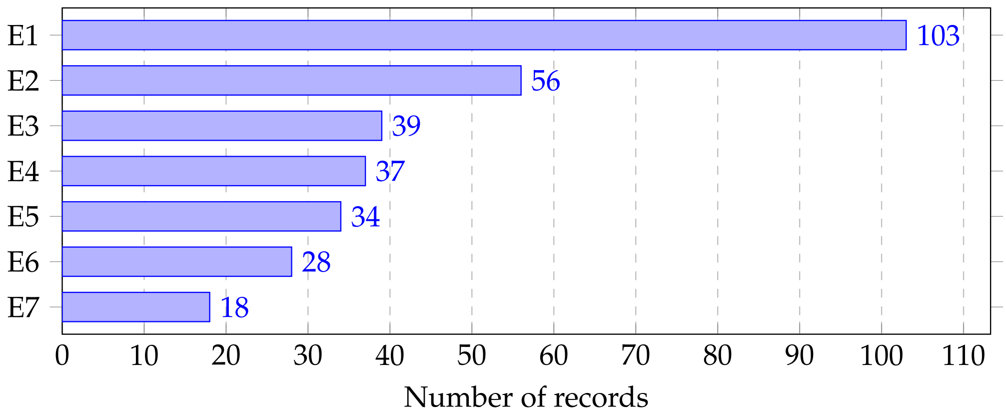


Figure 2. Reasons for exclusion during screening (n = 315)—see Table 6 for codes.

Inclusion and exclusion criteria were established a priori based on the PICOC framework (Table 3) following the PRISMA 2020 guidelines to specify eligibility criteria [10]. The selection process was designed to identify studies that directly address research questions concerning robotic platforms and ROS within the context of higher education and educational robotics.

Studies were considered for inclusion if they focused on higher education settings or student populations within higher education, corresponding to the Population element. The Intervention element required the use of robotic platforms that explicitly mention ROS 1, ROS 2, or related operating system terminology, aligning with the core technical focus of this review. Studies that mentioned ROS only tangentially were excluded unless they described the role of ROS in the platform implementation, hardware–software integration, middleware architecture, or experimental workflow. Only peer-reviewed journal articles, conference proceedings, and full conference papers were included to ensure a baseline level of academic rigor. The search was restricted to works published in English and within the publication year range of 2005 to 2026.

In contrast, studies were excluded if they did not involve higher education contexts, such as those focused solely on primary or secondary education without a clear link to the defined Population. Publications lacking a specific focus on ROS-based platforms, or those discussing general robotics kits (e.g., LEGO Mindstorms (https://www.lego.com/en-gb/themes/mindstorms/about, (accessed on 12 January 2026))) without ROS integration, were also excluded as they did not meet the Intervention criteria. Simulation-only studies were excluded because this review focuses on physical wheeled platforms and their hardware architecture for hands-on learning. Review and survey papers were not included in the primary synthesis. Non-peer-reviewed sources, including editorials, opinion pieces, gray literature such as theses and dissertations, and publications not in English, were excluded to maintain focus on formally published research. Furthermore, studies published before 2005 were excluded since they predate the development of ROS. The application of these criteria ensured a focused and methodologically sound selection of the literature for synthesis.

Table 6 details the exclusion reasons and associated codes. "Irrelevant platform" (E1) refers to any system that does not meet the core technical criteria, such as non-wheeled or closed proprietary solutions without middleware details. "Higher education only" (E9) applies to studies that describe curricula but lack a technical description of the robotic hardware.

Full-text assessment was attempted with the remaining 24 reports; 2 reports could not be retrieved. The remaining 22 reports were assessed for eligibility and quality. Table 7 displays the outcome of the screening stage.

Table 7. Record status in Parsifal (after import).

Status	Records (*n*)	Percentage (%)
Rejected	317	90.1
Accepted	22	6.3
Duplicated	13	3.7

As detailed in Table 8, two reports were excluded after quality assessment, resulting in a total of 20 studies for synthesis (Figure 1). In the Parsifal export, "not retrieved" and screened-out items are both registered as rejected records. Specifically, 315 records were excluded during the title and abstract screening phase, and 2 records were excluded because they could not be retrieved, resulting in a total of 317 rejected records in the Parsifal system.

Table 8. Reports excluded after full-text assessment and reasons for exclusion ($n = 2$).

Study Title	Score	Primary Reason for Exclusion
Criteria for the design of an educational robotics platform	5.0	E5/E6:Focuses on general design criteria without providing a specific physical hardware implementation for synthesis.
Building and understanding robotics—a practical course for different levels of education	5.0	E1/E9: Focuses on a broad educational course structure rather than a detailed technical description of a ROS-based wheeled platform.

High-end commercial platforms (e.g., Clearpath (https://clearpathrobotics.com/, (accessed on 12 January 2026)), PAL Robotics (https://pal-robotics.com/, (accessed on 12 January 2026))) and mobile manipulators were explicitly excluded from this review. These exclusions were applied to maintain a strict pedagogical focus on the fundamental mechatronic challenges of building and controlling a mobile wheeled base, which remains the primary learning objective in the targeted higher education courses. By participating in the full hardware–software co-design cycle, students avoid the limitations of proprietary 'black-box' systems, which have been shown to significantly boost student confidence, problem-solving skills, and engagement [3,17].

2.4. Quality Assessment and Data Extraction

A short checklist was used to determine whether the documents provide enough technical detail to support reuse (Table 9). Studies above the quality threshold were included. Figure 3 summarizes their scores. Data extraction was performed independently by two reviewers using a standardized form in Parsifal. The structured appraisal checklist evaluates methodological quality and risk of bias, while the structured data extraction form supports qualitative synthesis.

Table 9. Quality appraisal checklist (Parsifal "Quality Assessment").

QA Item	Question
QA1	Are the objectives or aims of the study clearly stated?
QA2	Is the wheeled robotic platform described in sufficient technical detail (hardware and software)?
QA3	Is the educational or competitive application context clearly defined (e.g., higher education course, competition type)?
QA4	Is the use of ROS or ROS 2 explicitly stated and technically explained?
QA5	Are the methods used to develop, deploy, or evaluate the platform clearly described?
QA6	Does the study report any form of evaluation, validation, or testing (educational outcomes, performance metrics, competition results)?
QA7	Does the study acknowledge limitations or challenges related to the platform or its application?
QA8	Is the study directly relevant to wheeled robotic platforms for educational or competitive applications in higher education?

The QA checklist evaluates reporting quality and methodological completeness rather than risk of bias in the strict PRISMA sense. Although not a formal risk-of-bias tool, it evaluates methodological clarity and reporting completeness. Table 10 presents the quality score distribution of the twenty included studies.

Table 10. Quality score distribution for included studies ($n = 20$).

Score	Studies (n)	Percentage (%)
5.5	1	5.0
6.5	3	15.0
7.0	4	20.0
7.5	6	30.0
8.0	6	30.0

https://doi.org/10.3390/robotics15070123

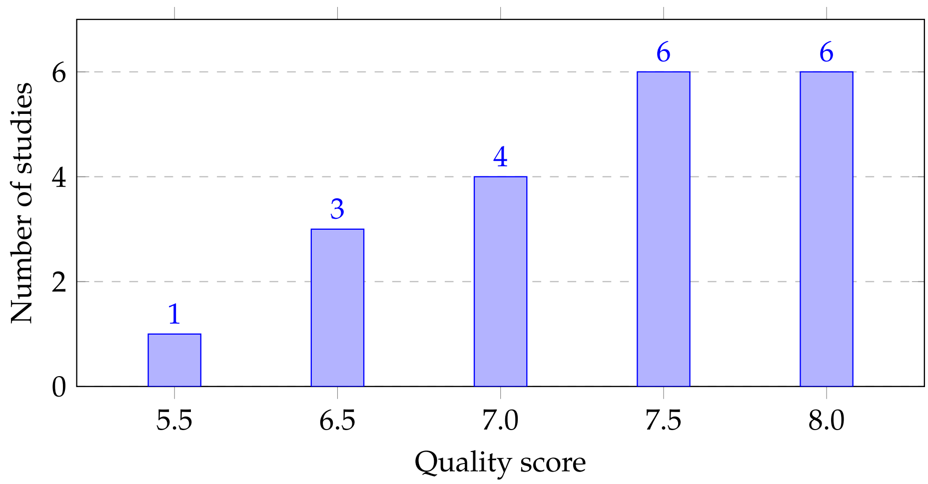


Figure 3. Quality score distribution histogram for included studies.

2.4.1. Data Completeness and Harmonization

The data extraction form contains fields not consistently reported in all included studies. Table 11 details the missing information, namely software configuration (e.g., ROS version), course or competition name, and the target educational level. For this reason, the synthesis focuses on well-documented fields, namely middleware, platform type, and application context, and reports when a relevant field is missing. Missing data were recorded as 'not reported,' and no imputation was performed.

Table 11. Missing values in the data extraction fields (included studies, $n = 20$).

Field	Missing (n)	Missing (%)
ROS version	16	80.0
Course or competition name	14	70.0
Educational level	7	35.0
Platform name or model	4	20.0
Role of ROS/ROS 2	4	20.0
Application context	2	10.0
Hardware description	2	10.0
Platform name	2	10.0
Research questions addressed	2	10.0
Source type	2	10.0
Wheeled platform	1	5.0
Title	1	5.0

Findings are supported by studies with quality scores ≥ 5.5, indicating moderate to high reporting quality.

2.4.2. Reporting Bias and Certainty Assessment

No formal assessment of reporting bias was conducted due to the qualitative nature of the synthesis. The certainty of evidence was not formally assessed due to the heterogeneity of study designs and outcomes. This systematic review consistently applied the same search string across all databases, using only repository-specific filters dictated by each interface. During the screening process, the reviewers strictly adhered to the predefined QA checklist. A quantitative synthesis (meta-analysis) was not appropriate due to heterogeneity in study designs, outcomes, and reporting formats. Methodological quality and risk of bias were assessed using a structured appraisal checklist, and results were synthesized qualitatively based on a data extraction form.

3. Results

This section analyzes the selected works by applying a consistent review protocol and by summarizing evidence across studies. Table 12 lists for the obtained studies the platform name, middleware ROS usage, hardware description, and application context.

Most educational robotics papers present a specific platform and describe its use within a course or a competition. Examples include TurtleBot-based teaching setups, PiBot, EdUcational Ros rObot PlAtform (EUROPA), and the Robotont line of platforms. These papers are useful because they provide concrete bills of materials, software stacks, and lessons learned from deployment [1,2,6,8,18,19].

The integration of mobile robotics into STEM curricula has shifted from the use of proprietary, black-box platforms (e.g., KUKA (https://www.kuka.com/, (accessed on 12 January 2026)), Pioneer 3-DX (https://robots.ros.org/pioneer-3-dx/, (accessed on 12 January 2026)), TurtleBot (https://www.turtlebot.com/, (accessed on 12 January 2026))) toward open-source hardware (OSH) and DIY solutions. Prior educational-robotics studies show extensive use of low-cost kits in pre-university contexts; however, higher-engineering courses require platforms that expose students to ROS, perception, embedded control, and hardware–software integration. The reviewed university-level platforms support Industry 4.0/5.0 skills—specifically ROS, computer vision, and embedded co-design—at an accessible price range of 50 € to 500 € [3,6,8,17,19].

Other works focus on reusable design ideas and platform requirements. For instance, Rosillo et al. [17] discussed general criteria for educational robots and their underlying architectures. Such contributions prove particularly valuable when designing a new platform that must balance cost, ease of assembly, sensor coverage, and long-term maintainability. Another example is the 5dpo robotic platform (Figure 4), a system initially developed for competitive use and later adopted for laboratory classes in an Autonomous Systems Master's course. This work aims to bridge the gap between simulation and physical deployment by providing hands-on experience with real robotic hardware [5].

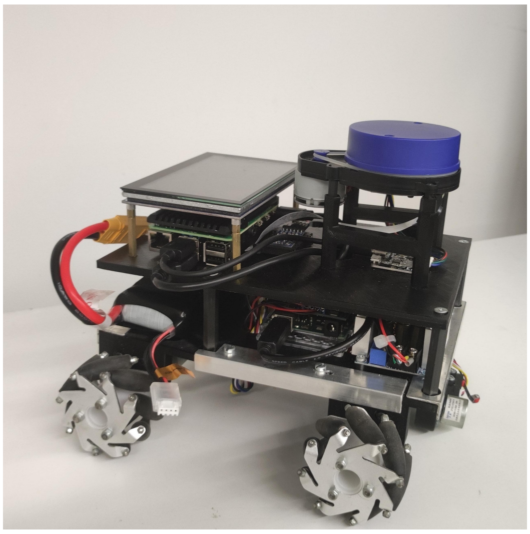


Figure 4. The 5dpo platform adapted from competition to educational use. Image kindly provided by the A. P. Moreira.

As summarized in Table 12, the platforms from these 20 studies can be grouped by their primary application domain to enhance analytical depth. In the domain of Industrial

https://doi.org/10.3390/robotics15070123

Logistics and Competitions, platforms such as the 5dpo platform and RobotCraft are highlighted for their use in simulated factory navigation, holonomic movement, and intensive deadline-driven learning [4,5,20,21]. Conversely, in the domain of Foundational STEM, platforms like FOSSBot, CrowBot BOLT, and X-Lakbay focus heavily on rapid prototyping, snap-fit modularity, and accessibility for foundational programming and structural learning [3,9,22]. Meanwhile, in the domain of Higher Education and Advanced Perception, platforms such as the TurtleBot 3, Robotont, and SROS are utilized for teaching autonomous navigation, 3D mapping, and Simultaneous Localization And Mapping (SLAM) without requiring industry-priced hardware [6–8,19]. The PiBot and Andruino-R2 focus heavily on computer vision and machine learning tasks using onboard or smartphone cameras, while the EUROPA platform links data acquisition and physics with an advanced sensor suite [1,2,18,23]. For specialized engineering curricula, the PSoC 4 and Balancing Robot platforms target Embedded Systems Co-design and Control Systems, and platforms like Mona are specifically engineered for Swarm Robotics research [24–26].

Table 12. Overview of included studies ($n = 20$).

Reference	Year	Platform	Middleware	Context	Application
[27]	2015	TraxBot (v1, v2) and StingBot (Arduino-based)	ROS 1	Higher education	Autonomous Navigation/Multi-Robot
[4]	2019	RobotCraft	ROS 1	Competition, higher education	Competition/Industrial Logistics
[18]	2018	PiBot	Other	–	Computer Vision/Foundational STEM
[20]	2018	RobotCraft	ROS 1	Competition, higher education	Competition/Industrial Logistics
[21]	2018	RobotCraft	ROS 1	Higher education	Competition/Industrial Logistics
[1]	2019	EUROPA	ROS 1	Higher education	Data Acquisition/Physics
[25]	2019	Mona	ROS 1	Higher education	Swarm Robotics
[24]	2019	ROS-controlled Balancing Robot	ROS 1	Higher education	Control Systems
[6]	2019	Turtlebot 3 Burger	ROS 1	Higher education	Autonomous Navigation/SLAM
[17]	2020	MATLAB/ROS framework	ROS 1	Higher education	Autonomous Navigation/Control
[2]	2020	EUROPA	ROS 1	–	Data Acquisition/Physics
[7]	2020	SROS	ROS 1	Higher education	Autonomous Navigation/SLAM
[23]	2021	Andruino-R2	ROS 1	Higher education	Intelligent Control/Machine Learning
[9]	2022	FOSSBot	Other	Higher education	Foundational STEM
[26]	2023	PSoC 4 evaluation kit	Other	Higher education	Embedded Systems/Hardware Co-design
[19]	2023	Robotont	ROS 1	Higher education	Autonomous Navigation/3D Mapping
[8]	2024	Robotont 3	ROS 2	Higher education	Autonomous Navigation/3D Mapping
[3]	2024	CrowBot BOLT	ROS 1	Higher education	Foundational STEM/Intro to ROS 1
[5]	2025	5dpo Robotics Team platform	ROS 1/ROS 2	Competition, Higher education	Competition/Industrial Logistics
[22]	2025	X-Lakbay UGV	Other	Higher education	Foundational STEM

3.1. Hardware Architectures

Current state-of-the-art platforms predominantly adopt a distributed dual-processor architecture. This design separates real-time kinematic control (low-level) from computationally intensive perception and planning (high-level). Figure 5 illustrates the concept.

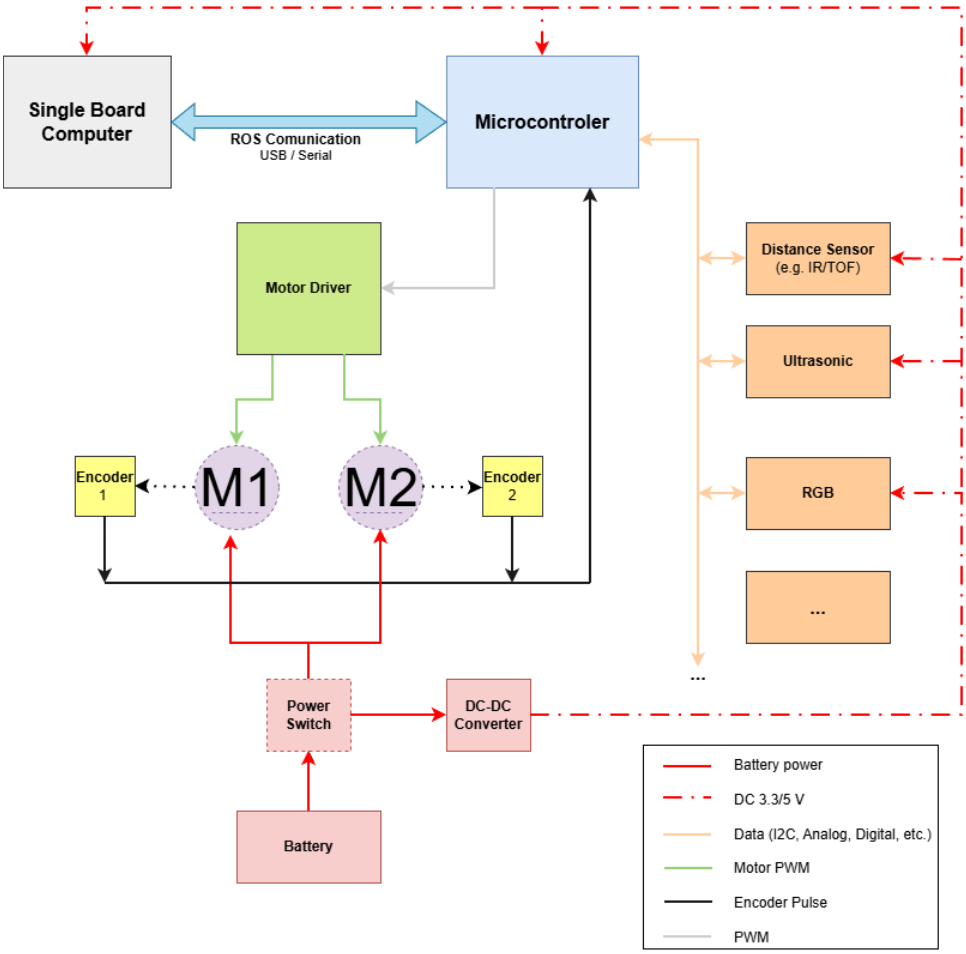


Figure 5. Generic hardware–software architecture (adapted from RobotCraft [4]).

3.1.1. Low-Level Control Units

The low-level controller manages motor Pulse Width Modulation (PWM), quadrature encoder acquisition, and safety interrupts. The Arduino Mega 2560 is widely adopted for this role due to its high Input/Output (I/O) count, notably in the RobotCraft and SROS platforms. RobotCraft employs it to interface ultrasonic and infrared (IR) sensors via a custom shield [4,20], while SROS uses it to handle differential-drive kinematics [7].

STM32 and Programmable System-On-Chip (PSoC) are used in newer platforms, migrating to more powerful 32-bit Micro-Controller (μC). Robotont 3 utilizes an ARM Cortex-M4 (STM32F407VGT6) to handle holonomic kinematics and power management on a single consolidated Printed Circuit Board (PCB) [8]. Similarly, the PSoC-based robot uses a Cypress PSoC 4 Bluetooth Low Energy (BLE), allowing students to design custom hardware peripherals inside the chip (hardware/firmware co-design) [26]. The CrowBot BOLT leverages the ESP32 System-on-Chip (SOC), integrating Wireless Fidelity (Wi-Fi) and Bluetooth directly into the low-level controller, reducing costs to approx. 50 $ while maintaining ROS connectivity via Transport Control Protocol (TCP)/Internet Protocol (IP) [3].

3.1.2. High-Level Computing Units

High-level units run the operating system (typically Ubuntu Linux) and the ROS Master node (ROS 1). The Raspberry Pi (RPi) 3 Model B+ is used in EUROPA, SROS, and TurtleBot 3 (Figure 6a) to run SLAM algorithms and image processing [2,6,7]. The RPi Zero is used in FOSSBot (Figure 6b) and PiBot (Figure 6d) to minimize power consumption and footprint [9,18]. Intel Next Unit of Computing (NUC) is used in Robotont (Figure 6c) to provide x86 architecture power to process depth data from RealSense cameras, enabling 3D mapping tasks that overwhelm ARM-based Single-Board Computers (SBCs) [19]. Alternatively, the Andruino-R2 platform integrates the student's smartphone via the Universal Serial BUS (USB) On-The-Go (OTG) communication mechanism. This low-cost approach eliminates the need for a dedicated SBC by exploiting the phone's Inertial Measurement Unit (IMU), Global Positioning System (GPS), camera, and high-resolution display [23].

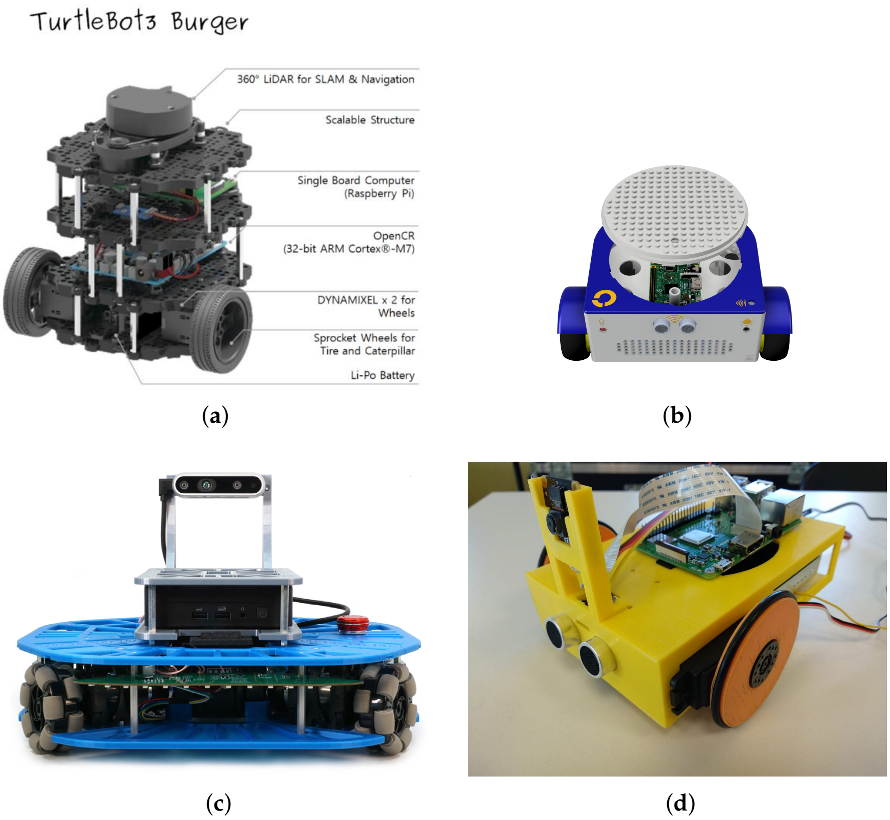


Figure 6. Illustrative educational/competition platforms. (**a**) TurtleBot3 Burger. Reproduced from the TurtleBot3 e-Manual [28] (CC BY 4.0). (**b**) FOSSBot. Cropped from [9], Figure 3, under the Creative Commons Attribution License (CC BY). (**c**) Robotont 3. Cropped from [8] Figure 1A, under the Creative Commons Attribution License (CC BY). (**d**) PiBot. Cropped from [18], Figure 5, under the Creative Commons Attribution License (CC BY).

3.1.3. Actuators

Standard educational platforms include Direct Current (DC) motors with magnetic quadrature encoders. RobotCraft employs the Hercules Dual 15 A 6 V to 20 V motor controller to drive high-torque DC motors. The SROS uses a Pololu Dual VNH5019 shield to drive 12 V motors with a 1:19 gear ratio. The Robotont 3 integrates an MC33886 H-bridge directly onto its main PCB to drive three motors in a Kiwi (120° offset) holonomic configuration [4,5,7,8,19–21].

3.1.4. Sensors

Light Detection and Ranging (LiDAR) sensing is essential for Two-Dimensional (2D) SLAM. TurtleBot 3 uses the LDS-01 360° scanner, while the Robot@Factory 4.0 platform uses the YDLIDAR X4PRO for beacon detection. Considering depth cameras, Robotont employs the Intel RealSense D435i Red-Green-Blue–Depth (RGB-D) camera for 3D point cloud generation. SROS uses the older Microsoft Kinect v1 (Xbox 360), keeping costs under 200 €. For standard perception, all platforms include the HC-SR04 ultrasonic sensor and Sharp GP2Y0A21 IR sensors for basic obstacle avoidance and wall following [5–7,9,18,19]. Camera-based sensing complements LiDAR, ultrasonic, and infrared sensors by enabling visual perception, object detection, RGB-D mapping, and vision-based navigation. In the reviewed platforms, RGB-D cameras such as Intel RealSense and Microsoft Kinect are used for depth perception and point-cloud generation, while smartphone-based platforms exploit integrated cameras to reduce hardware cost. These sensors increase pedagogical value but also introduce calibration, driver-maintenance, and computational-load constraints.

3.1.5. Power Systems

The reviewed platforms indicate a design shift from raw Lithium Polymer (LiPo) batteries, reported in platforms such as TraxBot and RobotCraft, toward consumer-grade power-tool batteries with integrated protection circuits. Robotont 3, for example, uses 18 V Makita Lithium-ion (Li-ion) batteries, which reduce handling complexity in student laboratories [8,19,27]. No included study reported quantified battery-safety incident rates; therefore, this trend should be interpreted as a design-safety rationale rather than as a statistically validated safety comparison.

3.2. Middleware and Software Architectures

The transition from raw sensor data to intelligent autonomous behavior in educational robotics is fundamentally enabled by the software architecture that governs the robot's operation. Central to this architecture is the concept of middleware—a software layer that abstracts the complexities of heterogeneous hardware, manages inter-process communication, and provides reusable services such as device drivers and control algorithms. In modern robotics education, this middleware layer determines both the accessibility of the platform for beginners and its capability to demonstrate advanced concepts. This section reviews the main software frameworks used in educational platforms, focusing on the industry-standard ROS middleware, simulation, and digital twinning that support scalable and flexible learning.

3.2.1. Robot Operating System

ROS 1 and, increasingly ROS 2, form the software backbone of ROS-compliant platforms. Bridges are used for communication between the low-level µC and the ROS network, e.g., the `rosserial` protocol adopted by SROS, RobotCraft, and CrowBot. This protocol serializes ROS messages over the Universal Asynchronous Receiver-Transmitter (UART), allowing the Arduino to publish topics like `odom` and subscribe to `cmd_vel`. Some platforms, like Andruino, use custom serial protocols over USB OTG to reduce the overhead associated with `rosserial` on limited microcontrollers [7,23,27].

3.2.2. Simulation and Digital Twins

The state of the art proposes simulation and digital twinning to allow scalable testing and teaching. PiBot, Robotont, and SROS provide full Unified Robot Description Format (URDF) models for Gazebo. This allows students to run the exact same navigation code in simulation as they would on the physical robot. At the Katholieke Universiteit Leuven, digital twinning is explored at master's level using TurtleBot 3 and MATLAB Robotics System Toolbox (https://www.mathworks.com/products/robotics.html, (accessed on 12 January 2026)), enabling engineering students to interact with ROS topics using MATLAB code. While Gazebo remains prevalent, other industry-standard simulation environments, such as Webots (https://cyberbotics.com/, (accessed on 12 January 2026)) CoppeliaSim (https://www.coppeliarobotics.com/, (accessed on 12 January 2026)), and NVIDIA Isaac Sim (https://developer.nvidia.com/isaac/sim, (accessed on 12 January 2026)), are also highly relevant for robotics education, though they are less frequently detailed in the reviewed literature [6,7,18,19].

3.3. Manufacturing and Modularity

The transition from Computer Numerical Control (CNC)-milled parts to fused deposition modeling (FDM) is ubiquitous. Robotont 3 reduced chassis production costs from 200 € with CNC polycarbonate to 25 € with Polylactic Acid (PLA)/Polyethylene Terephthalate Glycol (PETG) filament by redesigning the chassis with a lattice structure for rigidity. The X-Lakbay rover (Figure 7) features a snap-fit modular design printed in PETG, allowing assembly without screws, lowering the barrier for high school students. Modern designs rely on generative techniques to optimize strength-to-weight ratios, ensuring that 3D-printed chassis can support payloads (e.g., laptops, manipulators) without warping [8,22].

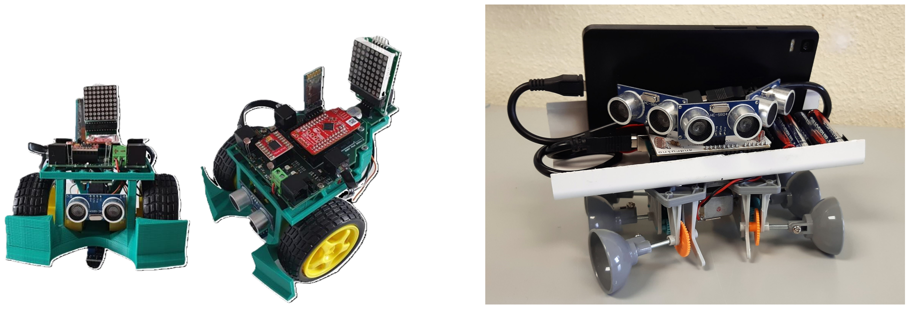
Figure 7. X-Lakbay snap-fit 3D-printed robotic platform. Cropped from [22], Figure 1A, under the Creative Commons Attribution License.

4. Discussion

The included studies show repeated patterns: modular hardware, separation between low-level and high-level control, and strong use of ROS for integration and reuse. The following discussion addresses the research questions and identifies the current challenges and gaps in educational and competition-oriented wheeled robotic platforms. Table 12 lists the source, year, middleware, and application context of the identified robotic platforms.

4.1. Hardware Architectures for Wheeled Robotic Platforms (RQ1)

One dominant configuration identified across the reviewed literature is the adoption of a distributed dual-processor architecture to balance cost, performance, and educational flexibility. This architecture typically separates low-level, real-time kinematic control from high-level cognitive processing. µCs, such as the Arduino Mega, Arduino Uno, STM32, or PSoC (Figure 8a), are universally employed to handle time-critical tasks like managing motor drivers, reading encoders, and processing basic sensor data. Meanwhile, high-level processing is typically transferred to a companion SBC, most commonly a Raspberry Pi or an Intel NUC, which provides the necessary computational power to run Linux-based operating systems and complex algorithms. In highly resource-constrained scenarios, platforms like Andruino-R2 (Figure 8b) innovate by replacing the SBC with individual student Android smartphones, leveraging the built-in camera, IMU, and processing power to dramatically reduce costs [2,5,7,19–21,23,26,27].



(a) (b)

Figure 8. Robotic platforms with high-level processing. (**a**) Mobile robotic platform integrating a PSoC. Reproduced from [26], Figure 7, under the Creative Commons Attribution License (CC BY). (**b**) Andruino-R2 robotic platform integrating an Android phone. Cropped from [23], Figure 1, under the Creative Commons Attribution License (CC BY).

To foster student-centric learning and accessibility, citizen manufacturing via 3D printing has become a staple in modern platform design. Platforms such as PiBot, FOSSBot, and X-Lakbay utilize 3D-printed chassis for easy replication, rapid prototyping, and modularity. The X-Lakbay Unmanned Ground Vehicle (UGV) specifically features a snap-fit modular frame that requires no screws, minimizing mechanical frustration and allowing students to focus on programming. For competition-oriented environments, hardware robustness is prioritized without sacrificing open-source principles. The 5dpo Robot@Factory 4.0 platform utilizes Mecanum wheels for omnidirectional movement in simulated industrial logistics. Similarly, the RobotCraft training program requires students to fully assemble a differential drive robot from components, exposing them to the practical mechatronic challenges of real-world robotics [5,9,18,20,22].

4.2. ROS-Based Designs for Multi-Sensor Integration and Real-Time Operation (RQ2)

ROS has emerged as standard middleware, providing essential hardware abstraction, message-passing, and software modularity for educational platforms. To bridge the high-level ROS network with the real-time physical hardware, most platforms rely on serial communication protocols. The `rosserial` package is widely used (e.g., in SROS, RobotCraft, and the MATLAB/ROS framework) to wrap ROS messages over the UART or USB interfaces, allowing the low-level µC to directly publish sensor topics (like `/odom`) and subscribe to velocity commands (like `/cmd_vel`). To reduce communication latency and overhead, some platforms like Andruino-R2 deploy custom serial protocols over bidirectional USB OTG connections [4,7,20,23,27].

This ROS-centric design greatly facilitates multi-sensor integration for advanced perception tasks. Educational platforms seamlessly integrate 360° LiDARs, Red-Green-Blue (RGB) and RGB-D cameras (such as Microsoft Kinect or Intel RealSense), and standard ultrasonic/IR sensors to perform SLAM and autonomous navigation. Real-time performance is preserved by isolating non-deterministic, computationally heavy ROS nodes on the companion computer, while the µC executes strict real-time Proportional–Integral–Derivative (PID) control loops for motor actuation. Furthermore, ROS allows the creation of digital twins in simulation environments such as Gazebo and Stage, enabling students to safely test sensor fusion and control algorithms before deploying them on physical hardware [5–8,19,20,23,27].

4.3. Challenges and Gaps

Despite the advantages of open-source robotic platforms, several significant challenges and gaps remain.

First, robot management and software maintenance in a classroom environment pose a major logistical burden. As noted in the Turtlebot 3 study, manually updating individual robots—via Secure Digital (SD) card cloning or Secure SHell (SSH)—is inefficient for teaching staff, highlighting the absence of automated mechanisms for batch wireless updates [6].

Second, the physical robustness and wiring complexity of DIY robots often hinder the learning experience. Platforms built with prototyping boards and loose jumper wires frequently suffer from connection failures, leading students to incorrectly blame the hardware for software bugs. To mitigate this, evolutionary designs like Robotont 3 have transitioned from loose wiring harnesses to a single consolidated PCB. However, this configuration reduces the scope for students to gain hands-on experience in component-level electronic assembly.

Third, the steep learning curve associated with ROS and Linux remains a barrier for novice students. Courses often require extensive preliminary training or must rely on abstraction layers, such as the MATLAB Robotics System Toolbox, to shield students from the underlying operating system complexities. Students repeatedly identify ROS integration as the most challenging component of their robotics coursework in multiple surveys [4,6,8,20,26].

Finally, there is a technological gap regarding software obsolescence. Much of the current educational literature and platform infrastructure relies on ROS 1 (e.g., Noetic), which has reached its end of life. Migrating entire software stacks and curricula to ROS 2 to maintain industry relevance requires significant development effort, a transition that only a few platforms, like Robotont, have actively addressed [8]. Although studies rarely provide long-term maintenance and upgrade plans, they often include open-source code and build details for reproducibility.

Beyond software and hardware maintenance, there is a notable pedagogical gap regarding advanced manipulation tasks. While modern, robust chassis like the X-Lakbay can mechanically support the integration of manipulators [22], and the EUROPA platform [1] successfully incorporates a 3-DOF robotic arm to teach basic inverse kinematics, the overwhelming majority of the reviewed curricula remain strictly dedicated to mobile wheeled base kinematics, sensor fusion, and perception. Complex manipulation in unstructured environments remains outside the scope of these low-cost DIY platforms.

The negative findings reported across the included studies indicate that open-source educational platforms remain difficult to scale beyond a small age group. Manual SD-card cloning, SSH-based updates, loose jumper wiring, connector failures, and inconsistent ROS deployments increase maintenance effort and reduce reproducibility. These issues are particularly relevant in competition-oriented scenarios, where robots are repeatedly assembled, repaired, and redeployed under time pressure. Future platforms should therefore include automated deployment, version-controlled software images, connectorized wiring, and documented maintenance procedures. The absence of Scopus access and the lack of eligible ACM Digital Library results further constrain the breadth of the search and should be considered when interpreting the generalizability of the synthesis.

5. Conclusions

This systematic literature review on open-source DIY wheeled robotic platforms designed for higher education in engineering and competitions has identified a trend toward 3D-printed, modular, ROS-compatible, open-source designs. However, the results reveal persistent limitations in terms of hardware maintenance and scalability, software lifecycle management, and long-term sustainability. Key open challenges include the definition of interoperable reference architectures, explicit design for maintenance, scalability, and sustainability, automated deployment of multi-robot systems, and reproducible integration workflows. Addressing these challenges is essential to support sustainable teaching practices while effectively bridging educational, competitive, and research-driven goals.

Limitations

This review acknowledges distinct limitations within both the synthesized primary studies and the review methodology itself. Regarding the primary literature, there is a high prevalence of missing technical data—specifically, 80 % of studies lack ROS version information and 70 % omit course or competition contexts. This significantly hinders reproducibility and longitudinal comparisons. Methodologically, this review restricts its scope to physical, open-source, and DIY wheeled platforms. Consequently, the findings do not generalize to commercial, legged, or research-grade robotic systems. Furthermore, although screening and data extraction were performed independently, they were not double-blinded, and the restriction to English-language publications across the selected databases may have introduced language and selection bias, potentially excluding relevant platforms developed in other languages.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/robotics15070123/s1.

Author Contributions: Conceptualization, B.M. and M.F.S.; methodology, R.P.; software, R.P.; validation, R.P., B.M. and M.F.S.; formal analysis, R.P.; investigation, R.P.; resources, R.P., B.M. and M.F.S.; data curation, R.P.; writing—original draft preparation, R.P.; writing—review and editing, B.M. and M.F.S.; visualization, R.P.; supervision, B.M. and M.F.S.; project administration, B.M. and M.F.S.; funding acquisition, M.F.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by Portuguese national funds through FCT—Fundação para a Ciência e a Tecnologia, I.P., under UID/50014/2025 (https://doi.org/10.54499/UID/50014/2025).

Data Availability Statement: The data extraction forms and extracted data supporting the findings of this study are available from the corresponding author upon reasonable request. This protocol was also registered with protocols.io [14]. The checklist can be downloaded at Supplementary Materials.

Acknowledgments: The authors used Copilot and Grammarly for editing and grammar enhancement, and NotebookLM for article analysis. All content was reviewed and edited by the authors, who take full responsibility for the final work.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

µC	Micro-Controller
2D	Two-Dimensional
3D	Three-Dimensional
BLE	Bluetooth Low Energy
CNC	Computer Numerical Control

DC	Direct Current
DIY	Do It Yourself
DOI	Digital Object Identifier
EUROPA	EdUcational Ros rObot PlAtform
FDM	fused deposition modeling
GPS	Global Positioning System
I/O	Input/Output
IMU	Inertial Measurement Unit
IP	Internet Protocol
IR	infrared
Li-ion	Lithium-ion
LiDAR	Light Detection and Ranging
LiPo	Lithium Polymer
NUC	Next Unit of Computing
OSH	open-source hardware
OTG	On-The-Go
PCB	Printed Circuit Board
PETG	Polyethylene Terephthalate Glycol
PICOC	Population, Intervention, Comparison, Outcome, and Context
PID	Proportional–Integral–Derivative
PLA	Polylactic Acid
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
PSoC	Programmable System-On-Chip
PWM	Pulse Width Modulation
RGB	Red-Green-Blue
RGB-D	Red-Green-Blue–Depth
ROS	Robot Operating System
RPi	Raspberry Pi
SBC	Single-Board Computer
SD	Secure Digital
SLAM	Simultaneous Localization And Mapping
SOC	System-on-Chip
SSH	Secure SHell
STEM	Science, Technology, Engineering and Maths
TCP	Transport Control Protocol
UART	Universal Asynchronous Receiver-Transmitter
UGV	Unmanned Ground Vehicle
URDF	Unified Robot Description Format
USB	Universal Serial BUS
Wi-Fi	Wireless Fidelity

References

1. Karalekas, G.; Vologiannidis, S.; Kalomiros, J. EUROPA—A ROS-based Open Platform for Educational Robotics. In *Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*; IEEE: New York, NY, USA, 2019; Volume 1, pp. 452–457. [CrossRef]
2. Karalekas, G.; Vologiannidis, S.; Kalomiros, J. EUROPA: A Case Study for Teaching Sensors, Data Acquisition and Robotics via a ROS-Based Educational Robot. *Sensors* **2020**, *20*, 2469. [CrossRef] [PubMed]
3. Varela-Aldás, J.; Palacios-Navarro, G. A ROS-Based Open Tool for Controlling an Educational Mobile Robot. *Int. J. Online Biomed. Eng. (iJOE)* **2024**, *20*, 23–39. [CrossRef]
4. Couceiro, M.S.; Araújo, A.G.; Tatarian, K.; Ferreira, N.M.F. RobotCraft: The First International Collective Internship for Advanced Robotics Training. In *Proceedings of the Robotics in Education*; Lepuschitz, W., Merdan, M., Koppensteiner, G., Balogh, R., Obdržálek, D., Eds.; Springer: Cham, Switzerland, 2019; pp. 31–43. [CrossRef]

5. Lopes, M.S.; Ribeiro, J.D.; Moreira, A.P.; Rocha, C.D.; Martins, J.G.; Sarmento, J.M.; Carvalho, J.P.; Costa, P.G.; Sousa, R.B. From Competition to Classroom: A Hands-on Approach to Robotics Learning. In *Proceedings of the 2025 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*; IEEE: New York, NY, USA, 2025; pp. 170–176. [CrossRef]
6. Amsters, R.; Slaets, P. Turtlebot 3 as a Robotics Education Platform. In *Proceedings of the Robotics in Education*; Merdan, M., Lepuschitz, W., Koppensteiner, G., Balogh, R., Obdržálek, D., Eds.; Springer: Cham, Switzerland, 2020; pp. 170–181. [CrossRef]
7. Kuzin, S.; Sziebig, G. SROS: Educational, Low-cost Autonomous Mobile Robot Design Based on ROS. In *Proceedings of the 2020 IEEE/SICE International Symposium on System Integration (SII)*; IEEE: New York, NY, USA, 2020; pp. 1052–1057. [CrossRef]
8. Mõtshärg, E.; Vunder, V.; Raudmäe, R.; Muro, M.; Drikkit, I.; Tšigrinski, L.; Köidam, R.; Aabloo, A.; Kruusamäe, K. Robotont 3—An accessible 3D-printable ROS-supported open-source mobile robot for education and research. *Front. Robot. AI* **2024**, *11*, 1406645. [CrossRef] [PubMed]
9. Chronis, C.; Varlamis, I. FOSSBot: An Open Source and Open Design Educational Robot. *Electronics* **2022**, *11*, 2606. [CrossRef]
10. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, n71. [CrossRef] [PubMed]
11. Page, M.J.; Moher, D.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. PRISMA 2020 explanation and elaboration: Updated guidance and exemplars for reporting systematic reviews. *BMJ* **2021**, *372*, n160. [CrossRef] [PubMed]
12. Sousa, R.B.; Sobreira, H.M.; Moreira, A.P. A Systematic Literature Review on Long-Term Localization and Mapping for Mobile Robots. *J. Field Robot.* **2023**, *40*, 1245–1322. [CrossRef]
13. Haddaway, N.R.; Page, M.J.; Pritchard, C.C.; McGuinness, L.A. PRISMA 2020: An R Package and Shiny App for Producing PRISMA 2020-Compliant Flow Diagrams, with Interactivity for Optimised Digital Transparency and Open Synthesis. *Campbell Syst. Rev.* **2022**, *18*, e1230. [CrossRef] [PubMed]
14. Pereira, R.; Malheiro, B.; Silva, M.F. Protocol for: ROS-Enabled DIY and Open-Source Wheeled Robots for Higher Education Learning and Competitions: A Systematic Review. 2026. Available online: https://www.protocols.io/view/protocol-for-ros-enabled-diy-and-open-source-wheel-14egnpbbpv5d/v1 (accessed on 25 June 2026).
15. Costa, G.d.M.; Petry, M.R.; Moreira, A.P. Augmented Reality for Human–Robot Collaboration and Cooperation in Industrial Applications: A Systematic Literature Review. *Sensors* **2022**, *22*, 2725. [CrossRef] [PubMed]
16. Starkman, J. The Robot Operating System in Transition: Experiments and Tutorials. Master's Thesis, Case Western Reserve University, Cleveland, OH, USA, 2018. Available online: http://rave.ohiolink.edu/etdc/view?acc_num=case1517240947102584 (accessed on 12 January 2026).
17. Rosillo, N.; Montés, N.; Alves, J.P.; Ferreira, N.M.F. A Generalized Matlab/ROS/Robotic Platform Framework for Teaching Robotics. In *Proceedings of the Robotics in Education*; Merdan, M., Lepuschitz, W., Koppensteiner, G., Balogh, R., Obdržálek, D., Eds.; Springer: Cham, Switzerland, 2020; pp. 159–169. [CrossRef]
18. Vega, J.; Cañas, J.M. PiBot: An Open Low-Cost Robotic Platform with Camera for STEM Education. *Electronics* **2018**, *7*, 430. [CrossRef]
19. Raudmäe, R.; Schumann, S.; Vunder, V.; Oidekivi, M.; Nigol, M.K.; Valner, R.; Masnavi, H.; Singh, A.K.; Aabloo, A.; Kruusamäe, K. ROBOTONT—Open-source and ROS-supported omnidirectional mobile robot for education and research. *HardwareX* **2023**, *14*, e00436. [CrossRef] [PubMed]
20. Fonseca Ferreira, N.M.; Araujo, A.; Couceiro, M.S.; Portugal, D. Intensive summer course in robotics—Robotcraft. *Appl. Comput. Inform.* **2020**, *16*, 155–179. [CrossRef]
21. Fonseca Ferreira, N.M.; Freitas, E.D.C. Robotics as multi-disciplinary learning: A summer course perspective. In *Proceedings of the 2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*; IEEE: New York, NY, USA, 2018; pp. 536–541. [CrossRef]
22. Esquivias, R.; Ferrolino, I.A.; Castro, W.C.; Lao, J.E.; Lim, N.; Chua, A. Design of a Cost-Effective Educational Unmanned Ground Vehicle Platform with an Auxiliary Computer. *HighTech Innov. J.* **2025**, *6*, 1138–1158. [CrossRef]
23. Lopez-Rodriguez, F.M.; Cuesta, F. An Android and Arduino Based Low-Cost Educational Robot with Applied Intelligent Control and Machine Learning. *Appl. Sci.* **2021**, *11*, 48. [CrossRef]
24. Chebotareva, E.; Gavrilova, L. Educational Mobile Robotics Project "ROS-Controlled Balancing Robot" Based on Arduino and Raspberry Pi. In *Proceedings of the 2019 12th International Conference on Developments in eSystems Engineering (DeSE)*; IEEE: New York, NY, USA, 2019; pp. 209–214. [CrossRef]
25. Arvin, F.; Espinosa, J.; Bird, B.; West, A.; Watson, S.; Lennox, B. Mona: An Affordable Open-Source Mobile Robot for Education and Research. *J. Intell. Robot. Syst.* **2019**, *94*, 761–775. [CrossRef]
26. Suarez, A.; García-Costa, D.; Perez, J.; López-Iñesta, E.; Grimaldo, F.; Torres, J. Hands-on Learning: Assessing the Impact of a Mobile Robot Platform in Engineering Learning Environments. *Sustainability* **2023**, *15*, 13717. [CrossRef]

27. Araújo, A.; Portugal, D.; Couceiro, M.S.; Rocha, R.P. Integrating Arduino-Based Educational Mobile Robots in ROS. *J. Intell. Robot. Syst.* **2015**, *77*, 281–298. [CrossRef]
28. ROBOTIS. TurtleBot3 OpenSource and Licences. ROBOTIS e-Manual (Online). Documents Released Under CC BY 4.0. Available online: https://emanual.robotis.com/docs/en/platform/turtlebot3/features/#components (accessed on 6 March 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.