International Journal Of Health, Engineering And Technology (IJHET) *E-ISSN 2829 - 8683*
Volume 5, Number 2, July 2026, *Page. 1710 - 1719*
Email : editorijhess@gmail.com

Design And Development Of An ESP32-Based Transporter Robot For Lightweight Goods Using The Arduino IDE

Achmad Hosen Zaini[1)*], Wincoko[2)], Yanuar Sinatra[3)]
[1,2,3)] Sekolah Tinggi Teknik Multimedia Internasional Malang

*Corresponding Author
Email : achmadhosenzaini@gmail.com

Abstract

This research develops a transporter robot controlled by a wireless joystick for light cargo transportation applications. The system uses the ESP32 microcontroller to process commands from the joystick via a wireless connection, which then controls the DC motor to drive the robot wheels using PWM signals, and a servo motor to operate the gripper that lifts light items up to 300 grams. The robot is equipped with BTS7960 drivers for the DC motors and L293D drivers for the servo motor. The robot's power source is a 18650 battery, which supports operation for 50-65 minutes. The method used is Research and Development (R&D), focusing on the design, implementation, and testing of the transporter robot. Testing was conducted to evaluate the robot's ability to transport items, movement stability, and response to joystick commands. The test results show the robot can move at a speed of 0.6 m/s and turn with precision. The gripper can lift items up to 280 grams stably, and the robot has a fast joystick response time of less than one second. The contribution of this research is the development of an efficient wireless-controlled transporter robot, with applications in light cargo transportation in confined environments, such as industries and warehouses. The simple and efficient system makes this robot a practical solution for improving operational efficiency in various applications.

Keywords: ***Robot transporter; joystick wireless; ESP32***

INTRODUCTION

Nowadays, technological growth is developing rapidly. In the near future, society will enter the era of automation, where many human tasks and needs will be supported or replaced by robots integrated into various systems. Basically, the purpose of using robots is to simplify human work, not to completely replace human labor Kaptein (Kaptein et al., 2022).

In industry, the use of automation processes has become very common. Almost all production processes in large industries today have implemented automated systems. In a production process, high precision, consistent results, and fast operation are often required. On the other hand, humans have limitations in terms of speed and precision, especially when physical abilities decline, which can reduce work quality Jabbar and Yasdar (Jabbar and Yasdar, 2022).

Rapid technological advancement has impacted various aspects of human life. One field of technology that plays an important role is robotics. Robotics not only serves as the foundation for future technological development, but also provides opportunities to improve critical thinking, creativity, and collaborative skills Didik (Didik, 2024).

A transporter robot is a type of robot used to pick up or move objects that can be controlled either automatically or manually. Manual control on currently developed transporter robots generally still uses conventional remote controls such as joysticks. These transporter robots play an important role in moving goods in various industrial sectors Ahsani, Rahmawati, and Wibisono (Ahsani, Rahmawati and Wibisono, 2020).

A gripper is one of the devices that plays a role in automation processes. Its function is to pick up and move objects to predetermined positions. The gripper plays a major role in grasping materials to be lifted. Since the materials used have different shapes, materials, and sizes, the gripper needs to be adjusted according to the specifications of the material Yudha et al. (2024).

A transporter robot becomes more useful when equipped with a wireless joystick, allowing easier and more flexible control because it does not require cables. G-Code is a language used to provide commands to computers in machine manufacturing or control processes. G-Code functions as instructions regarding where the robot should move, how it should move, and which path should be followed Mardiyyah and Wincoko (Mardiyyah and Wincoko, 2021).

Several previous studies have developed transporter robots for carrying lightweight goods. For example, research conducted by Nizam, Haris Yuana, and Zunita Wulansari (2022) used the Arduino UNO microcontroller to control DC motors in a transporter robot system. The results showed that the robot could move automatically, although motor control was still limited to simple movements and load response was less optimal. Another study by Eka Maulana and Nurpulaela (2024) utilized ESP8266 for wireless communication, but unstable sensor data processing caused a decrease in robot navigation accuracy.

Currently, most developed transporter robot systems face significant limitations, particularly in motor control accuracy, sensor integration, and program flexibility. Many previous systems used conventional microcontrollers with limited processing capacity, making them unable to execute commands in real-time with high precision, especially when combined with additional sensors. Furthermore, wireless connectivity in existing systems is often unstable, potentially causing high latency or signal interference, which affects the robot's responsiveness to user commands. As a result, the robot becomes less efficient in transporting lightweight goods, particularly under operational conditions requiring fast coordination and high precision REDDY G et al. (2025).

In this context, ESP32 as a microcontroller offers significant potential that has not been optimally utilized in previous systems. ESP32 has higher processing capability, numerous GPIO pins, Bluetooth and Wi-Fi wireless communication support, and PWM control capabilities for DC motors and servos with high precision. By utilizing these capabilities, transporter robots can be designed to overcome existing limitations, especially in terms of more accurate motor control, better sensor integration, and program flexibility that allows adaptation to various operational needs. In addition, ESP32 enables modular programming using the Arduino IDE, simplifying the development of control logic, input-output management, and integration of additional sensors without changing the overall code structure Ade Chairany, Relita Buaton, and Ratih Puspadini (2025).

Based on the explanation above, this research focuses on designing an ESP32-based transporter robot controlled through a wireless joystick. The selection of a joystick as the input device provides intuitive and real-time control, enabling users to move the robot and operate the gripper precisely. This system is designed so that the robot can transport and move lightweight goods with high stability, optimize the performance of DC motors and servos, and maintain balanced load distribution during operation. The integration of a wireless joystick with ESP32 enables responsive wireless control, supports operational flexibility, and opens opportunities to implement additional control algorithms, such as obstacle avoidance or multi-robot coordination.

In robotics, the microcontroller is the central component that functions as the brain of the robot. The primary role of the microcontroller is to process data received from various sensors and convert it into commands sent to actuators, allowing the robot to perform certain tasks automatically. Without a microcontroller, a robot would not have the ability to respond to its environment, regulate movement, or adjust actions according to changing conditions. In other words, the microcontroller acts as a bridge between sensor inputs and outputs in the form of motor, servo, or other actuator movements, enabling the entire robotic system to operate in a coordinated and precise manner Rutuja S. Patil et al. (2025).

The selection of a microcontroller is one of the key factors in robot development. In previous robot systems, many used basic microcontrollers such as Arduino UNO to control DC motors or simple actuators. Arduino UNO has advantages in programming simplicity and the availability of many libraries supporting various hardware devices, making it easier for developers to create robot prototypes. However, the limitation of Arduino UNO lies in its low processing capacity and limited number of pins. This makes it less suitable for robot systems requiring complex sensor integration, high-precision motor control, and simultaneous data processing. As a result, robots controlled using Arduino UNO tend to have slower and less stable responses when executing multiple commands simultaneously Wang et al. (2025).

Meanwhile, some robot systems use ESP8266 for wireless communication, especially for remote control through Wi-Fi. ESP8266 enables robots to be controlled remotely with fairly good wireless connectivity, allowing robots to transport or manipulate objects without direct control. Nevertheless, ESP8266 has limited processing capacity. When the robot must process sensor data simultaneously while controlling motors, the system often experiences delays or reduced movement accuracy. This indicates that although ESP8266 excels in wireless communication, its capability for real-time and precise robot control is still limited Cern and Ze (2023).

ESP32 emerges as a more advanced microcontroller solution for modern robotics. ESP32 is a low-power microcontroller classified as a System on a Chip (SoC), with higher processing capability compared to Arduino UNO and ESP8266. The ESP32 board provides a dual-core 32-bit CPU operating up to 240 MHz, 448 KB ROM, 520 KB RAM, and more than 30 input-output pins that can be used for various functions. In addition, ESP32 has integrated Wi-Fi and Bluetooth communication modules, including Bluetooth Low Energy (BLE), supporting stable and fast wireless communication. With these capabilities, ESP32 can control motors and servos with high precision, process sensor data simultaneously, and maintain wireless communication stability while the robot moves or carries loads Nuanmeesri and Poomhiran (2021).

Besides processing and connectivity capabilities, ESP32 has advantages in simultaneously controlling multiple actuators, which is very important for transporter robots. Modern robotic systems often require coordination among several DC motors, servo motors, and additional sensors, so the ESP32's ability to execute commands in parallel allows the robot to move with high precision. This means the robot can regulate wheel movement, open and close the gripper, and adjust speed simultaneously without delays or command conflicts, making robot operation more efficient and accurate G S et al. (2026).

The integration of ESP32 with a wireless joystick also increases robot control flexibility. The use of a joystick enables intuitive and real-time control, where users can move the robot forward, backward, turn, and operate the gripper simultaneously. The combination of ESP32 and a wireless joystick minimizes signal latency, ensures fast and accurate robot response, and allows direct movement adjustment according to user needs Leblanc et al. (2025).

Furthermore, the use of ESP32 allows transporter robots to have broad development capabilities. The modular programming structure and comprehensive library support facilitate the development of additional algorithms, such as obstacle avoidance, automatic navigation, or gripper position adjustment based on object weight and size. Thus, the robot not only functions as a lightweight goods transporter, but can also be developed into an adaptive intelligent robotic platform. This flexibility is highly beneficial for further research and industrial applications, as it allows developers to adapt the system to increasingly complex operational needs without changing the main hardware design.

ESP32 development boards, such as the ESP32 DevKit V1 DOIT, enable the design of more complex yet efficient transporter robots. This board can be used to connect wireless joysticks, distance sensors, load sensors, or additional actuators, allowing the robot to execute various tasks automatically. For example, the robot can adjust motor speed based on the transported load, open and close the gripper precisely, and move stably on different surfaces. The combination of fast processing, wireless connectivity, and sufficient pin numbers makes ESP32 highly flexible for various robotic configurations Martínez and Rendón (2023).

In addition, the use of ESP32 enables modular programming using the Arduino IDE. This simplifies the development of complex control logic, integration of additional sensors, and management of various robot operational scenarios without changing the overall program structure. This flexibility is crucial for transporter robots that must adapt to different operational needs, such as transporting lightweight goods accurately, avoiding obstacles, or adjusting gripper movements according to the size and weight of objects Varela-Aldás and Palacios-Navarro (2024).

Furthermore, this research is expected to contribute significantly to the development of robotic control systems. By combining advanced microcontroller technology, precise motor control, and stable wireless communication, this transporter robot not only improves operational efficiency but also becomes a model adaptable to various lightweight transportation applications, whether in manufacturing industries, warehousing, or laboratory research. This approach also opens opportunities for other developers to explore wireless joystick-based control systems, thereby expanding understanding of hardware and software integration and providing a foundation for innovation in intelligent robotics.

RESEARCH METHODS

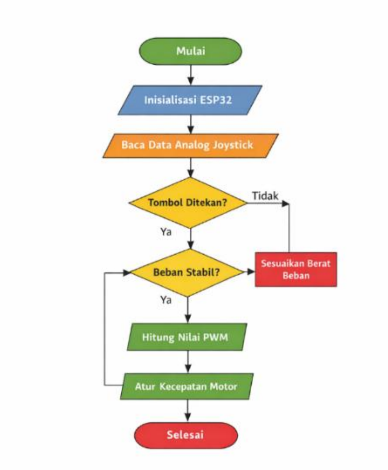

Figure 1. Research Flowchart

This research uses a Research and Development (R&D) approach and aims not only to understand phenomena, but also to develop, implement, and test an operational transporter robot system. The research process begins with identifying the requirements and specifications of the device, followed by designing the system and hardware according to the robot's objectives. Furthermore, the components are assembled and integrated, then tested to ensure that the functionality is in accordance with the planned design. If any discrepancies are found, improvements and retesting are carried out so that the robot can operate optimally Wincoko, Mahmud, and Kurniasari (2022).

The research data sources consist of primary and secondary data. Primary data were obtained through direct observation of the robot's hardware and software, including technical specifications, performance, and system functionality, enabling researchers to obtain accurate and relevant data. Secondary data were obtained through literature studies related to robotics technology, robot drive systems, and previous transporter robot developments, which provide theoretical foundations and additional insights for developing a better system. Data collection techniques include direct observation, documentation studies, and literature reviews, allowing researchers to comprehensively understand the interaction between the robot and its environment and identify aspects that need improvement.

The research instruments include both hardware and software. The hardware consists of an ESP32 microcontroller, DC motors and servo motors, batteries, motor drivers (BTS7960 and L293D), a wireless joystick, gripper, laptop, and supporting cables. The software used includes Arduino IDE version 2.3.3 and the Windows 11 operating system. The robot was developed with dimensions of 30 × 20 × 15 cm, capable of carrying lightweight loads of less than 300 grams, and has an average energy consumption of 13.32 Wh per hour.

Table 1. Hardware Components

No	Hardware / Component	Specifications / Description
1	Laptop	Axioo, Intel Celeron N4020 @1.10 GHz
2	ESP32 Microcontroller	Dual-core 32-bit, 3–5 V, Wi-Fi & BLE, 48 GPIO pins
3	DC Motor	Voltage 3–6 V, current 100 mA–10 A
4	Servo Motor	Voltage 4–6 V, current 1–2 A
5	Battery	18650, 3.7 V / 1200 mAh
6	BTS7960 Motor Driver	Voltage 5.5–27 V, 28 A continuous current
7	L293D Motor Driver	Voltage 4.5–36 V, 600 mA per channel
8	Wireless Joystick	2.4 GHz, 10–12 buttons, control range up to 10 m
9	Gripper	Load capacity <300 grams, size 8 × 8 cm, width up to 16 cm
10	Ultrasonic Distance Sensor	HC-SR04, used for obstacle detection

Table 2. Software Components

No	Software	Specifications / Description
1	Arduino IDE	Version 2.3.3
2	Windows 11	Operating System

Table 3. Robot Parameters

Parameter	Value
Robot Voltage (V)	11.1 V
Average Motor Current (I)	1.2 A
Power (P)	13.32 W
Energy Consumption per Hour (E)	13.32 Wh

The power calculation in this study is based on the electrical power equation:

$$P = V \times I$$

Based on the measured voltage of 11.1 V and average current of 1.2 A, the resulting power consumption is 13.32 W. The hourly energy consumption is calculated using the following equation:

$$E = P \times t$$

where $t = 1$hour, resulting in an energy consumption value of 13.32 Wh.

RESULTS AND DISCUSSION

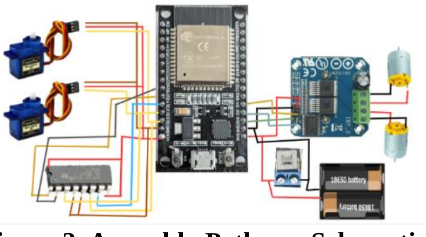

Figure 2. Assembly Pathway Schematic

The robot system utilizes a wireless joystick as the primary input device, which sends control signals wirelessly to the ESP32 microcontroller. The ESP32 acts as the central data processing unit, translating joystick signals into movement commands for the DC motors controlling the wheels and the servo motor controlling the gripper. The BTS7960 driver regulates the current and voltage supplied to the DC motors, while the L293D driver controls the gripper with precision. The robot is capable of moving forward, backward, and turning accurately, as well as lifting objects weighing up to 300 grams. The joystick response remains stable within a range of 2–10 meters, with a response time of less than one second, enabling effective real-time control.

The robot development process began with planning and selecting components, including the ESP32, DC and servo motors, BTS7960 and L293D drivers, 18650 battery, and wireless joystick. Afterward, the hardware was assembled according to the design, followed by ESP32 programming to connect the joystick with the motors and gripper, as well as system integration through gradual testing. Program testing showed that the robot was able to respond properly to joystick commands, achieving an average speed of 0.6 m/s on a 2-meter track, completing a 90° turn in less than 2 seconds, and enabling the gripper to stably grasp loads up to 300 grams. The 18650 battery was able to support robot operation for approximately 50–65 minutes, consistent with the calculated energy consumption of 13.32 Wh per hour.

The robot assembly pathway schematic demonstrates a systematic integration: wireless joystick → ESP32 → drivers (BTS7960/L293D) → DC motor & servo motor → robot gripper/wheels, powered by an 18650 battery supply. The ESP32 processes joystick input and sends PWM signals to regulate motor speed and direction. The program utilizes the Bluepad32 library for Bluetooth communication, EEPROM for data storage, and Servo_ESP32 for servo motor control. The feasibility test results, including material, design, and hardware testing, indicate that the components, documentation, and system function according to the research objectives, while remaining safe and effective.

Table 4. Robot Device Specifications

No	Component	Specifications / Description
1	ESP32 Microcontroller	Dual-core 32-bit, 3–5 V, Wi-Fi & BLE, 48 GPIO, 520 kB SRAM, 4 MB flash
2	DC Motor	Voltage 3–6 V, current 100 mA–10 A
3	Servo Motor	Voltage 4–6 V, current 1–2 A
4	BTS7960 Driver	Voltage 5.5–27 V, 28 A continuous current, overcurrent and overtemperature protection
5	L293D Driver	Voltage 4.5–36 V, 600 mA per channel, thermal shutdown protection
6	Gripper	Load capacity <300 grams, size 8 × 8 cm, width up to 16 cm
7	Battery	18650, 3.7 V / 1200 mAh
8	Wireless Joystick	2.4 GHz, 10–12 buttons, 10-meter control range
9	Laptop	Axioo, Intel Celeron N4020 @1.10 GHz

Table 5. Load Testing Results

No	Load Weight	Lifting Status	Robot Stability	Servo Condition	Description
1	100 grams	Successful	Stable	Normal	Smooth movement
2	350 grams	Unsuccessful	Unstable	Unable to operate	Gripper failed to lift

Table 6. System Block Testing Results

No	Component	Voltage (V)	Current (A)	Description
1	18650 Battery	7.85	2.2	Normal
2	ESP32 (standby/joystick processing)	5.38	0.18	Normal
3	DC Motor BTS7960 (without load)	3.0	1.3	Normal
4	DC Motor BTS7960 (with load)	3.0	1.3	Normal
5	Servo Motor L293D (without load)	3.2	1.0	Normal
6	Servo Motor L293D (gripping load)	3.2	1.0	Normal

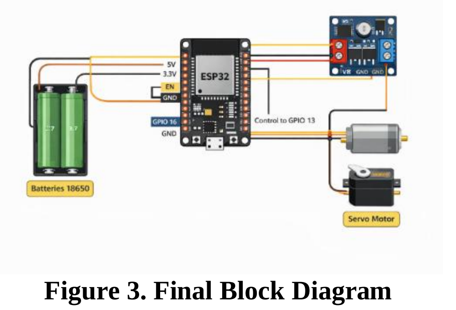

Figure 3. Final Block Diagram

The wiring schematic illustrates the integration of the ESP32-based transporter robot system with its main components, namely the 18650 battery as the power source, DC motors, servo motor, and motor drivers. The 18650 battery supplies the main voltage for the entire system, with 5V, 3.3V, and GND lines ensuring stable power delivery to the ESP32, motor drivers, and servo motor. This power distribution is designed so that all components receive sufficient current, enabling the robot to operate stably for approximately 50–65 minutes according to the power consumption tests.

The ESP32 functions as the control center, receiving input signals from the wireless joystick and translating them into PWM commands for the DC motors and servo motor. The control line connected to GPIO 13 regulates the motor driver, allowing the DC motors to move the robot wheels forward, backward, and turn with high precision. The servo motor, used for the gripper mechanism, receives PWM signals directly from the ESP32, enabling accurate gripping and releasing of lightweight objects up to 300 grams.

This schematic demonstrates good modular integration between input, processing, and actuation systems. The ESP32 is capable of processing commands in real-time, coordinating wheel and gripper movements so that the robot operates responsively and stably. Clearly separated power and signal pathways also help maintain voltage stability, prevent performance degradation, and simplify maintenance or future system development, including the addition of sensors or extra actuators.

Table 7. ESP32 Pin Configuration

No	ESP32 Pin	Pin Function	Description	Notes
1	GPIO 13	PWM Output	DC motor & servo control	Generates PWM signals for smooth movement
2	GPIO 16	Digital Input	DC motor control	Sends signals from ESP32 to motor driver
3	V5	Power Supply	5V power from battery	Ensures stable supply for ESP32 & components
4	GND	Ground	Ground connection	Balances voltage throughout the system

The research results indicate that the ESP32-based transporter robot programmed using the Arduino IDE was successfully implemented effectively. The system uses a wireless joystick as the primary input device, capable of transmitting wireless control signals to the ESP32 with a response time of less than one second within a range of 2–10 meters. This latency is consistent with Bluetooth wireless communication theory for real-time control applications, which states that the ideal latency ranges between 50–100 ms. The ESP32 acts as the data processing center, interpreting joystick signals into movement commands for the DC motors and servo motor. Through this mechanism, the joystick serves as the interface between the user and the robot, while the ESP32 functions as the brain of the system that regulates all robot movements in real-time. This integration ensures that the robot can respond to commands with high precision while enabling simultaneous wheel and gripper movements.

The DC motors are controlled using the BTS7960 driver with PWM signals generated by the ESP32, allowing the robot to move forward, backward, and turn precisely. A two-meter track test showed that the robot achieved an average speed of 0.6 m/s, while the 90° turning time was recorded at less than two seconds. These results confirm the effectiveness of the motor driver and PWM combination in supporting precise maneuverability, consistent with PWM control theory, which states that duty cycle variations directly affect motor speed and rotation direction. The servo motor operating the gripper is controlled through the L293D driver, enabling stable gripping and releasing of lightweight objects up to 300 grams. Testing showed that the gripper operated optimally with loads between 100–300 grams, while loads exceeding 300 grams caused performance degradation, confirming that the robot's mechanical capacity matches the original design.

The robot development stages included component planning, hardware assembly, ESP32 programming using the Arduino IDE, as well as system integration and preliminary testing. Components such as the ESP32, DC motors, servo motor, BTS7960 driver, L293D driver, 18650 battery, and wireless joystick were selected based on suitability, availability, and research objectives. Hardware assembly focused on component stability and load distribution to ensure balanced robot movement. ESP32 programming utilized the Bluepad32 library for joystick communication, EEPROM for data storage, and Servo_ESP32 for servo motor control, allowing the system to operate responsively and safely. The modular program structure also enables future development without modifying the core control functions, while simplifying maintenance and troubleshooting Nizam, Haris Yuana, and Zunita Wulansari (2022).

The system power testing demonstrated that the 18650 battery could support robot operation for approximately 50–65 minutes, in accordance with the calculated energy consumption of 13.32 Wh per hour. Efficient current distribution through the 5V, 3.3V, and GND lines helped maintain voltage stability, prevent performance degradation, and ensure that the DC motors and servo motor operated simultaneously without interference. System block analysis showed that all components operated within normal voltage and current ranges, confirming that the integration between hardware and software was optimal. Feasibility testing, including material, design, and hardware evaluations, showed that the robot met the planned technical, safety, and aesthetic standards.

The assembly pathway schematic and system block diagram describe the logical relationship between joystick input, ESP32 processing, DC motor and servo actuators, and the resulting robot movement and object transportation output. Proper power distribution and compatible drivers enable the robot to operate stably while carrying lightweight loads up to 300 grams. The modular design also provides flexibility for future development, including the addition of distance sensors or ultrasonic sensors so that the robot can operate on uneven or obstacle-filled terrain. This design also simplifies component replacement, maintenance, and modification without dismantling the entire system.

Although the robot functions optimally for transporting lightweight goods, several limitations were identified. The robot can only lift loads up to 300 grams, making it unsuitable for heavier objects. The effective joystick control range is limited to 10 meters, and the signal quality decreases beyond this range. The ESP32 processing capacity is also limited, meaning that the integration of complex control algorithms or additional sensors requires code optimization to maintain system stability. Furthermore, the system is not yet equipped with overcurrent protection or advanced obstacle avoidance sensors.

Mechanical testing demonstrated the robot's stability while carrying lightweight loads across a distance of approximately 2 meters. The average speed of 0.6 m/s and the 90° turning time of less than 2 seconds indicate that the DC motors and BTS7960 driver operated according to PWM control theory. The gripper was capable of gripping 100-gram objects with high stability but failed to lift 350-gram loads due to reduced servo motor torque.

CONCLUSIONS

The ESP32-based transporter robot programmed using the Arduino IDE was successfully developed and operated according to the objectives of the research. The wireless joystick system enabled responsive wireless control, allowing the robot to move forward, backward, turn, and operate the gripper with precision. The integration of mechanical, electronic, and software systems proved effective, with the gripper capable of lifting lightweight objects up to 300 grams and load distribution maintaining movement stability. The 18650 battery supported approximately 50–65 minutes of operation, while the modular design simplified maintenance and future development. Limitations were identified in the maximum load capacity, battery duration when carrying heavier loads, and joystick control range; however, overall, the robot achieved the research objectives, demonstrating stable performance, precision, and readiness for lightweight goods transportation applications.

REFERENCES

Ade Chairany, Relita Buaton and Ratih Puspadini (2025) 'Penerapan Internet of Things pada Mekanik Prototipe Robot Pengaduk Gabah', *Repeater : Publikasi Teknik Informatika dan Jaringan*, 3(3). Available at: https://doi.org/10.62951/repeater.v3i3.617.

Ahsani, I.A., Rahmawati, D. and Wibisono, K.A. (2020) 'Kendali Robot Transporter Berdasarkan Pergerakan Pergelangan Tangan Menggunakan Leap MotionDengan Metode Decision Tree', *Jurnal Riset Rekayasa Elektro*, 2(2). Available at: https://doi.org/10.30595/jrre.v2i2.9046.

Cern, Y.S. and Ze, Y.S. (2023) 'The Design of a New 3D Print-in-place Soft Four-Legged Robots with Artificial Intelligence', *Jurnal Kejuruteraan*, 35(3). Available at: https://doi.org/10.17576/jkukm-2023-35(3)-20.

Didik, W. (2024) 'Robot avoider berbasis arduino uno untuk edukasi tik di smp annur lembah kemuning', 2(3).

Eka Maulana, F. and Nurpulaela, L. (2024) 'Konfigurasi Mikrokontroler Stm32 Untuk Membaca Push Button Dengan Arduino Ide Pada Prototipe Smart Charger Di Pt. Pasifik Satelit Nusantara', *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(4), pp. 7278–7284. Available at: https://doi.org/10.36040/jati.v8i4.10184.

G S, G. *et al.* (2026) 'Autonomous Navigation Robot for Military Applications', *International Journal of Scientific Research in Engineering and Management*, 10(01). Available at: https://doi.org/10.55041/ijsrem55911.

Jabbar, A.A. and Yasdar, Y. (2022) 'Sistem Kendali Gripper Adaptif Pada Robot Transporter', *Jurnal Mosfet*, 2(2). Available at: https://doi.org/10.31850/jmosfet.v2i2.1974.

Kaptein, F. *et al.* (2022) 'A Cloud-based Robot System for Long-term Interaction: Principles, Implementation, Lessons Learned', *ACM Transactions on Human-Robot Interaction*, 11(1), pp. 1–27. Available at: https://doi.org/10.1145/3481585.

Leblanc, A. *et al.* (2025) 'Commercial Hoverboard Reverse Engineering and Repurposing for a Stabilized Platform: A Recyclable Solution for Modular Robotic Bases', *Sensors*, 25(12). Available at: https://doi.org/10.3390/s25123833.

Mardiyyah1, N.K. and Wincoko (2021) 'Jurnal ilmiah sinteks issn : 1907-2007 e-issn : 2579-7115', 10(2), pp. 10–16.

Martínez, F. and Rendón, A. (2023) 'Autonomous Motion Planning for a Differential Robot using Particle Swarm Optimization', *International Journal of Advanced Computer Science and Applications*, 14(4). Available at: https://doi.org/10.14569/IJACSA.2023.0140490.

Nizam, M.N., Haris Yuana and Zunita Wulansari (2022) 'Mikrokontroler Esp 32 Sebagai Alat Monitoring Pintu Berbasis Web', *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(2), pp. 767–772. Available at: https://doi.org/10.36040/jati.v6i2.5713.

Nuanmeesri, S. and Poomhiran, L. (2021) 'Optimization Shortest One-Way Path for Energy Saving Auto Robot Collecting Floating Garbage using Fast Approximate Nearest Neighbor Search', *International Journal on Advanced Science, Engineering and Information Technology*, 11(2). Available at: https://doi.org/10.18517/ijaseit.11.2.13191.

REDDY G, V.K. *et al.* (2025) 'Orientacart: Assistive Navigation Cart for the Visually Impaired Using RFID and Line Following', *International Journal of Scientific Research in Engineering and Management*, 09(11). Available at: https://doi.org/10.55041/ijsrem54682.

Rutuja. S. Patil *et al.* (2025) 'Hazardous Gas Detection and Environmental Monitoring in Coal Mines Using IoT and GSM-Based Autonomous Robot', *International Research Journal on Advanced Engineering Hub (IRJAEH)*, 3(07). Available at: https://doi.org/10.47392/irjaeh.2025.0478.

Varela-Aldás, J. and Palacios-Navarro, G. (2024) 'A ROS-Based Open Tool for Controlling an Educational Mobile Robot', *International journal of online and biomedical engineering*, 20(1). Available at: https://doi.org/10.3991/ijoe.v20i01.46573.

Wang, A. *et al.* (2025) 'A Hierarchical Distributed Control System Design for Lower Limb Rehabilitation Robot', *Technologies*, 13(10). Available at: https://doi.org/10.3390/technologies13100462.

Wincoko, W., Mahmud, M. and Kurniasari, I. (2022) 'Sistem Aplikasi Coding Dan Indexing Pasien Rawat Jalan Di Puskesmas Bringkoning Madura', *POSITIF : Jurnal Sistem dan Teknologi Informasi*, 8(1), pp. 1–7. Available at: https://doi.org/10.31961/positif.v8i1.1213.

Yudha, Y.F. *et al.* (2024) 'Perancangan Sistem Deteksi Objek Pada Robot Transporter Menggunakan Metode Darknet YOLOv8', *Electrician : Jurnal Rekayasa dan Teknologi Elektro*, 18(2), pp. 161–170. Available at: https://doi.org/10.23960/elc.v18n2.2595.