https://doi.org/10.22306/atec.v11i2.283 *Received: 23 Apr. 2025; Revised: 13 May 2025; Accepted: 05 June 2025*

Prototype design of an inverted pendulum two-wheel vehicle

Dang Anh Viet
VNU University of Engineering and Technology, 144 Xuan Thuy Street, Cau Giay District, Ha Noi, Vietnam,
vietda@vnu.edu.vn

Keywords: two-wheeled, self-balancing vehicle, prototype, wireless.
Abstract: This study presents the design and implementation of a two-wheeled self-balancing vehicle based on the inverted pendulum principle, controlled wirelessly via a PC interface. The system integrates an ATmega-based embedded platform with MRF24J40MA wireless transceiver modules for real-time bidirectional communication between the vehicle and a host computer. The vehicle's structure includes a custom mechanical frame, sensor modules (including accelerometers and encoders), motor drivers, and an LCD interface for status display. Control algorithms, including PID tuning, are executed through a PC-based graphical interface, allowing precise adjustments and live monitoring. A dual-microcontroller configuration—ATmega8 for PC interface and ATmega32 for onboard control—facilitates modularity and reliable data handling. The results confirm the system's effectiveness in maintaining balance and responding to commands under various test conditions. This prototype serves as a foundational model for further exploration in autonomous robotics, wireless control systems, and real-time embedded applications.

1 Introduction

Wireless data transmission is a large area in information electronics, the transmitted data can be analogue or digital [1-5]. In wireless data transmission, the most effective is still transmission by electromagnetic waves or radio waves, because of the advantages of long-distance transmission, multi-directional, high operating frequency. The article aims to build a simple system consisting of an intermediate board that performs data forwarding between the computer and the RF transceiver module, another RF transceiver module mounted directly on the self-balancing two-wheeled vehicle with the task of sending control commands to the robot and collecting data that needs to be processed and sending it back to the computer. To solve this problem, the research takes advantage of the microcontroller's ability [6-10] to transmit and receive serial data thanks to the UART in the chip. The microcontroller has the ability to perform multi-processing communication, which is very suitable for data transmission in a wireless network system consisting of many processors. Wireless technology is applied to create compact, smart, flexible robot models on all terrains. By adding a remote observation camera, people can control them as desired, creating the desired work efficiency.

A self-balancing two-wheeled vehicle is a robot that can balance itself in a vertical position with only two coaxial wheels [11-15]. The balancing process is performed by the controller [16-19]. The two wheels will move forward or backward to always maintain the vertical state of the vehicle so that it does not fall. The two-wheeled vehicle is also capable of moving on different terrains like other means of transport. The content of the article is divided into 3 parts. The first part is responsible for studying the theoretical basis. The second part is to study the hardware features, and the last part is to design the control software, hardware circuits and the results achieved. This research not only demonstrates the feasibility of wireless control in dynamic robotic systems but also highlights the synergy between embedded microcontroller platforms and RF communication modules. The proposed design offers a practical and scalable solution for developing intelligent mobile robots. Ultimately, the study provides a comprehensive approach to designing and implementing a self-balancing two-wheeled vehicle, from theoretical modelling to hardware integration and control software development.

2 Hardware and software design

The MRF-AVR-PC communication circuit schematic diagram is shown in Figure 1. The circuit is a complete embedded wireless communication system built around the ATmega8L microcontroller and the MRF24J40MA wireless transceiver module. At the heart of the system, the ATmega8L handles core processing and communication tasks. It operates using a stable clock signal provided by an external crystal oscillator, connected via two capacitors to form a classic crystal-based timing circuit. To ensure proper operation and maintain clean reset conditions, a reset circuit is included, consisting of a push-button and passive component that allow manual system reinitialization.

Power is supplied to both the microcontroller and the wireless module through dedicated voltage regulators. The ATmega8L typically operates at 5V, while the MRF24J40MA module requires a regulated 3.3V supply. Each power line is stabilized using filtering capacitors to reduce voltage ripples and electrical noise. The ATmega8L communicates with the MRF24J40MA using the SPI protocol, employing pins for data in (MOSI), data out (MISO), clock (SCK), and chip select (CS). Additional control lines such as RESET, INT, and WAKE are used to manage the wireless module's operational states and event

Acta Tecnología - International Scientific Journal about Technologies
Volume: 11 2025 Issue: 2 Pages: 93-100 ISSN 2453-675X

handling. To enable communication with a personal computer, the system integrates a MAX232 IC, which converts TTL logic levels from the microcontroller to RS-232 voltage levels compatible with a standard DB9 serial port. This allows for seamless UART communication between the microcontroller and external software on the PC. The MAX232 utilizes external capacitors as part of its internal charge pump to generate necessary voltage swings for RS-232 signalling.

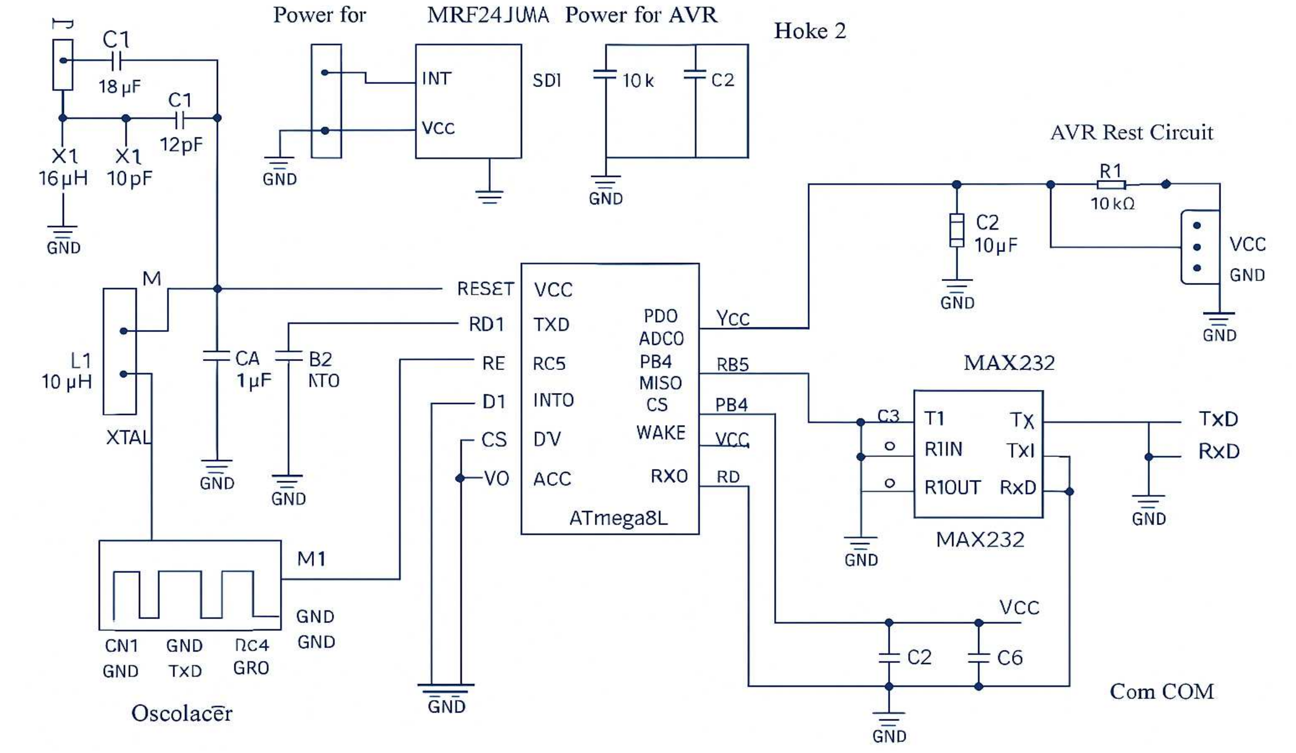

Figure 1 MRF – AVR - PC communication circuit schematic diagram

For development and firmware updates, an in-circuit serial programming (ICSP) header is provided. This interface allows programmers to upload or modify the ATmega8L firmware without removing the chip from the board. Overall, the design supports wireless data exchange between the PC and remote devices using ZigBee protocol, while ensuring stable operation, programmable control, and robust communication through well-integrated analog and digital subsystems.

Figure 2 represents a complete embedded system for a robot control unit, designed around the ATmega32 microcontroller. The system integrates multiple functional modules to support sensor input, motor control, wireless communication, user interface, and system monitoring.

At the core of the circuit is the ATmega32, a versatile 8-bit AVR microcontroller that interfaces with several peripheral devices. A 16 MHz crystal oscillator and two capacitors ensure stable clock operation for precise timing and execution. The microcontroller receives regulated power through two voltage regulators: the LM1117 (3.3V output) and the 7805 (5V output), each stabilized by decoupling capacitors to reduce voltage noise and fluctuations. A manual reset circuit with a tactile switch and pull-up resistor is included for system reinitialization.

Sensor input is handled by an MMA accelerometer module, connected to the ATmega32 through I2C or SPI pins, depending on configuration. A motor encoder module is connected to digital I/O pins to track motor position and speed, feeding essential feedback to the control algorithm. The MOTOR block indicates output connections to a motor driver circuit, allowing the microcontroller to control motor rotation based on commands or sensor feedback.

For wireless control, the system includes an MRF24J40MA ZigBee module, which communicates with the ATmega32 via SPI and is powered through the 3.3V regulator. The control signals from this RF module enable the robot to receive commands remotely. An ISP (In-System Programming) header is provided to upload firmware directly to the microcontroller, making it easier to update the software without removing the chip.

User interaction and system status are facilitated through a 16x2 LCD module connected to the microcontroller's PORTC. This display outputs relevant data like sensor readings or mode indicators. Additionally, an LED with a current-limiting resistor (R3) serves as a simple status indicator. A push button (SW1) allows user input, such as mode switching or command triggering.

Overall, the system combines multiple subsystems—including motion sensing, wireless communication, motor control, display, and user input—into a single cohesive unit. It is ideal for mobile robot applications, remote sensing platforms, or any embedded control system requiring real-time feedback and user interaction.

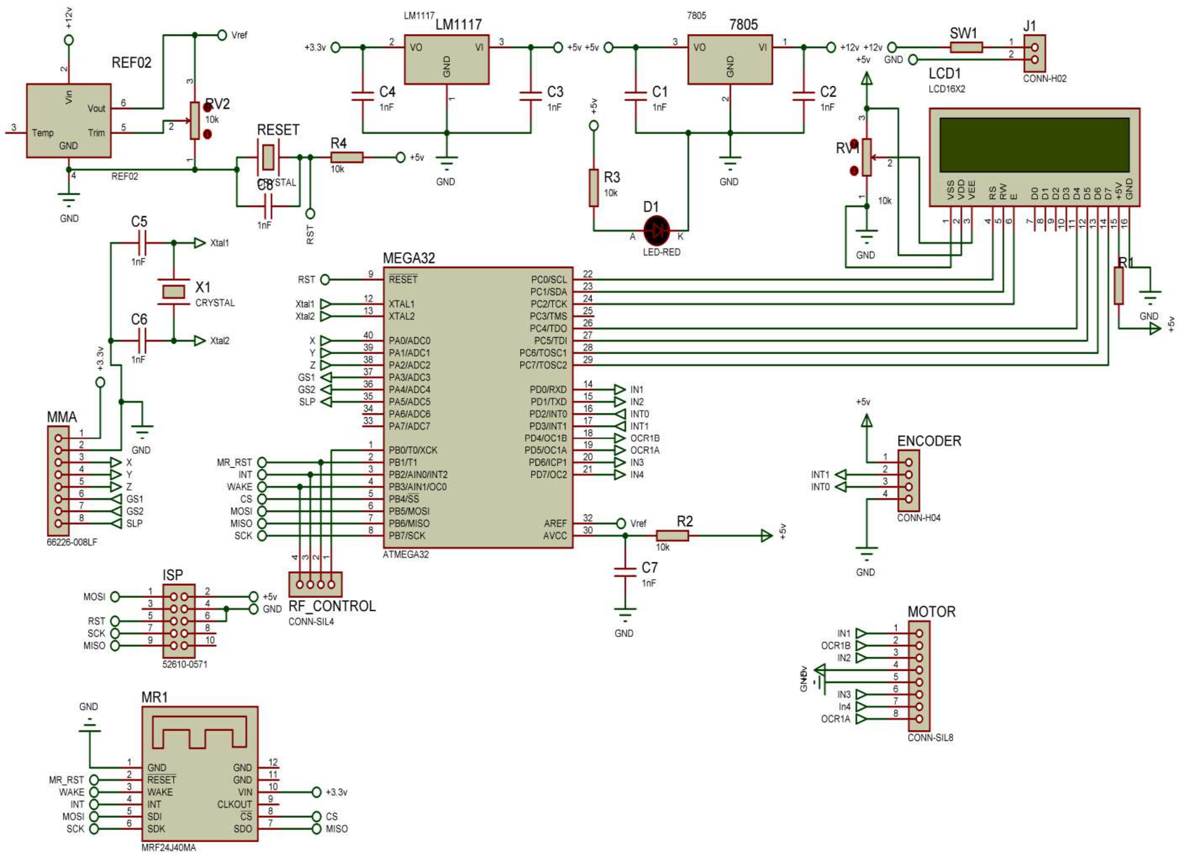

Figure 2 Robot control circuit schematic diagram

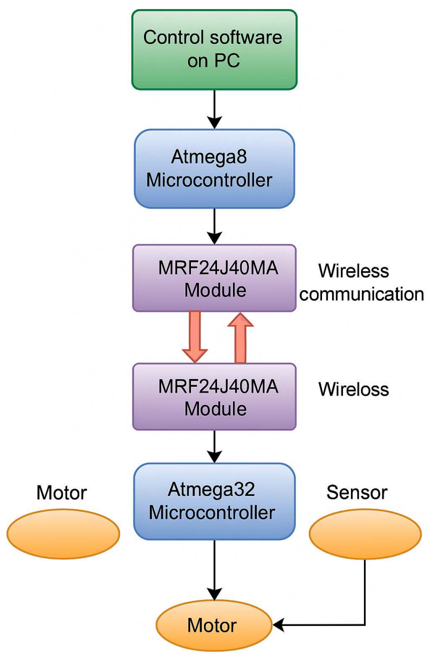

Figure 3 Block diagram of system

The block diagram of system as shown in the Figure 3. The system begins with the PC-based control software, which sends commands to the Atmega8 microcontroller. This microcontroller communicates wirelessly via a MRF24J40MA module. The signal is received by another MRF24J40MA module, which forwards the data to the Atmega32 microcontroller. The Atmega32 processes the data, reads information from the sensors, and drives the motors accordingly. This forms a wireless closed-loop control system between the PC and the robot.

The Figure 4 shows a graphical user interface (GUI) for a program titled “Program for Controlling a Self-Balancing Two-Wheeled Vehicle.” The layout is visually divided into several functional sections designed to facilitate communication, control, and monitoring of the vehicle. On the left side, the "Communication Control" section allows the user to select a COM port and baud rate (e.g., COM1 and 2400) to establish a connection with the vehicle via a wireless module like the MRF24J40. Below this is the "Motor Control (PID)" section, where the user can input PID (Proportional, Integral, Derivative) values for tuning the vehicle's balancing response. Buttons for confirming or cancelling the settings are also provided.

The control software on the computer is designed as follows:

Figure 4 Control program interface

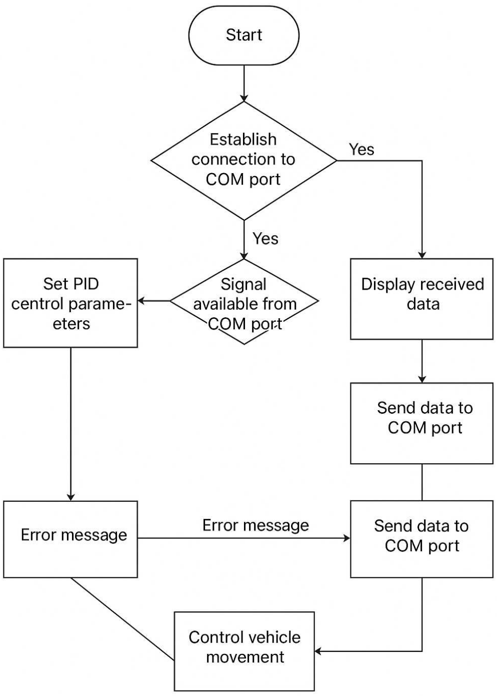

Figure 5 Algorithm flow chart of the control program on PC

Figure 5 presents a flowchart that outlines the operational logic of a program used to control a system—likely a self-balancing vehicle—through a COM port communication interface. The process begins with establishing a connection to the COM port. Once connected, the program checks for signal availability from the port. If a signal is detected, it allows two parallel operations: displaying the received data and plotting the vehicle's status on a graph, indicating real-time feedback from the vehicle's sensors.

Concurrently, the program also proceeds to set PID (Proportional, Integral, Derivative) parameters for the motor. If the parameters are set correctly, data is transmitted to the vehicle through the COM port. If an error occurs during the PID configuration, the system generates an error message. Additionally, if there is no issue, the program continues to control the vehicle's movement based on the received instructions. Overall, the flowchart illustrates a well-organized structure for managing serial communication, motor tuning, real-time monitoring, and error handling, essential for ensuring responsive and stable control of a self-balancing robotic system.

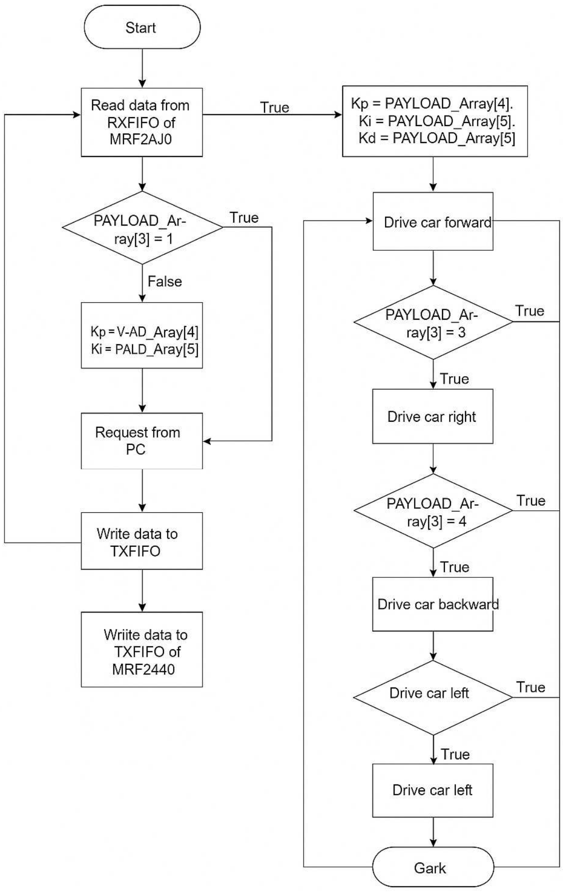

Figure 6 Algorithm flow chart of remote control of two-wheeled vehicle

Figure 6 displays a structured flowchart representing a control algorithm for a car system that communicates with a PC via a wireless transceiver module (MRF24J40). The process begins at the "Start" block and immediately proceeds to read incoming data from the RXFIFO register of the MRF24J40. If data is successfully read, the system evaluates specific control conditions based on the content of PAYLOAD_Array [3].

If PAYLOAD_Array [3] equals 1, the program assigns values to the PID parameters Kp, Ki, and Kd using elements from the payload array. Subsequent decisions branch based on whether the array's index equals 2, 3, 4, or 5, which trigger respective motion commands: driving the car forward, right, backward, and finally, left. Each condition leads to an action box indicating a directional command to the vehicle.

If the data read condition is false or no relevant control condition is met, the flow proceeds to check if there is a data request from the PC. If true, the necessary data is written into the TXFIFO buffer and transmitted back via the MRF24J40 module. The flowchart uses standard symbols with decision diamonds and process rectangles and includes corrected English for logic terms like “True” and “False,” ensuring clarity. However, a minor typo is present in the terminal block labelled "Gark" instead of "End" or "Finish," and that could be revised for improved accuracy.

3 Result and discussion

The Figure 7 shows a custom-built communication circuit board, likely used for serial data transmission in embedded systems. At the center of the board is a dual in-line package (DIP) microcontroller, which serves as the main processing unit. To its left, a red module labeled “MRF24J40MA” is mounted, which is a 2.4GHz IEEE 802.15.4 wireless transceiver, commonly used for ZigBee-based communication. On the right side, there's a DB9 serial connector, indicating that the board supports RS232 communication, allowing it to interface with PCs or other serial devices.

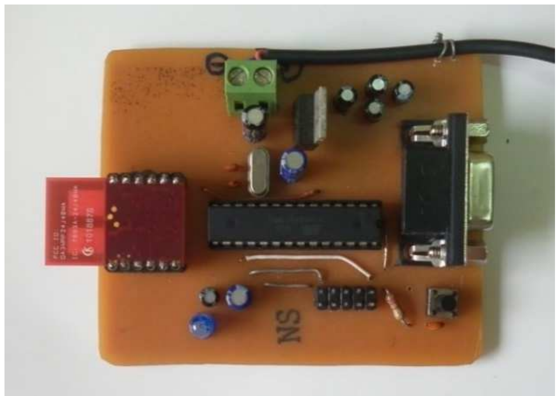

Figure 7 MRF24J40MA communication module to connect to computer

The top part features a green terminal block for power supply or other external connections, along with various supporting electronic components such as capacitors, resistors, a crystal oscillator (for clock signal generation), and voltage regulators.

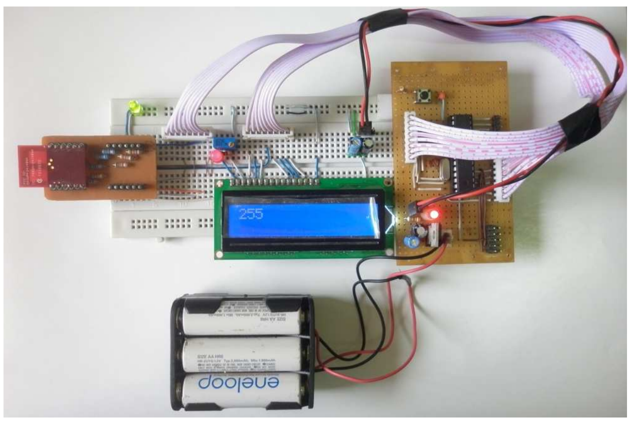

Figure 8 MRF24J40 Board to test wireless transmission between two MRF24J40 modules

The Figure 8 shows a test setup for a microcontroller-based wireless communication system. On the left, the red wireless module (MRF24J40MA) is connected to a small custom PCB, which is mounted on a white breadboard. This module handles IEEE 802.15.4 communication, often used in ZigBee applications. The breadboard also includes supporting electronic components such as LEDs, resistors, and a 16x2 character LCD display that is currently showing the value “255,” suggesting it is displaying data received wirelessly or processed by the microcontroller. A custom-designed main control board is visible to the right, connected via a ribbon cable. This board likely contains a microcontroller and associated circuitry for managing communication and data processing. Power is supplied from a battery pack at the bottom, containing two AA Eneloop rechargeable batteries.

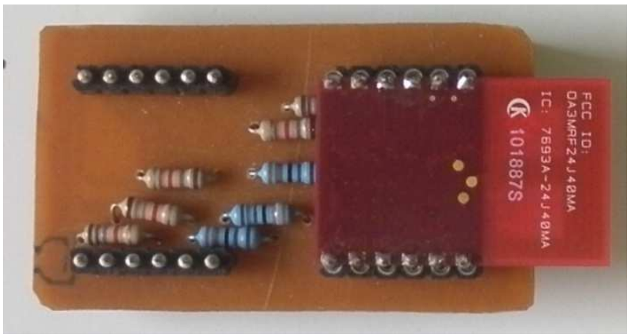

Figure 9 MRF24J40MA module mounted on self-balancing two-wheeled vehicle

The Figure 9 shows a custom-made adapter board for a wireless communication module, specifically the MRF24J40MA, which is a 2.4GHz IEEE 802.15.4/ZigBee transceiver manufactured by Microchip. The red module is securely soldered onto a brown single-layer PCB, and multiple through-hole resistors are mounted to condition or protect the data lines connected to the module. Two rows of pin headers are used for interfacing this board with other microcontroller systems, such as development boards or test circuits, via jumper wires or ribbon cables. The resistors likely function as pull-up, pull-down, or current-limiting resistors to ensure reliable data transmission and electrical compatibility.

The Figure 10 displays a prototype of a two-wheeled self-balancing robot, designed to operate based on the principles of an inverted pendulum. The structure is primarily composed of a metal frame with two wheels at the base, allowing mobility and dynamic stabilization. Mounted vertically on the frame is a tall support beam, which likely serves as the main axis for balance control. Near the middle of the frame is an electronics section, consisting of several custom-built circuit boards and an LCD screen, indicating real-time feedback or sensor readings such as angle, speed, or control output. At the top of the robot, a battery pack consisting of multiple cylindrical cells is securely attached to power the system. Numerous wires connect the sensors, actuators, and controllers, showing a test or development phase. This setup is commonly used in research or educational projects to study real-time control systems, especially PID or fuzzy logic controllers, applied to robotics and mechatronics. The robot is likely intended to demonstrate wireless communication, autonomous balancing, and mobility in embedded systems.

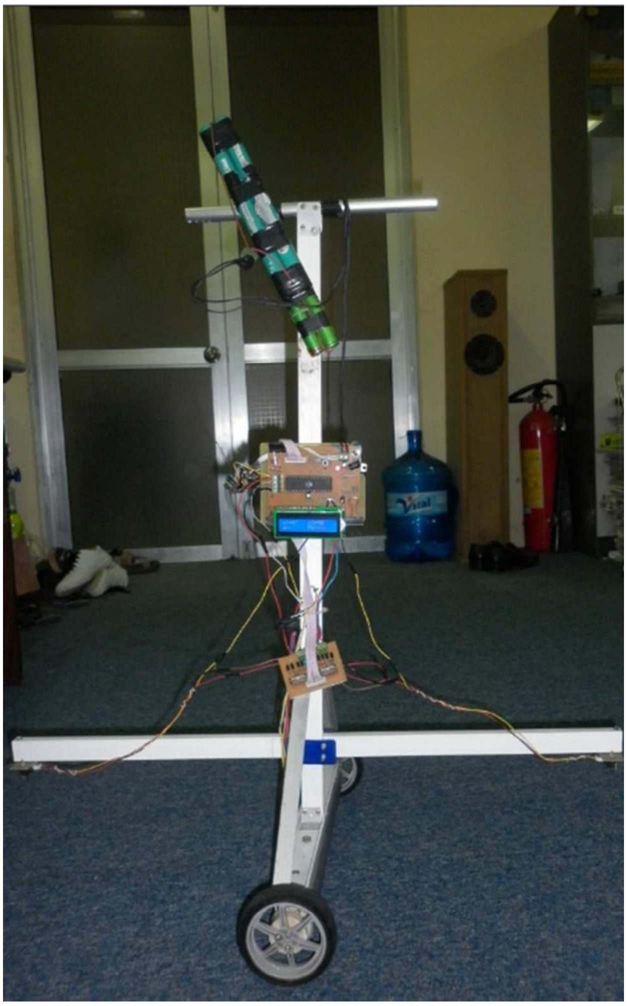

Figure 10 Self-balancing two-wheeler prototype

The experimental evaluation of the self-balancing two-wheeled vehicle prototype yielded positive results, validating both the hardware integration and the wireless control strategy. The wireless communication system, using the MRF24J40MA transceiver modules, demonstrated reliable and stable data transmission between the PC and the vehicle in real-time. The communication board (Figure 7) and wireless test setup (Figure 8) confirmed that the modules could maintain continuous two-way communication with minimal data loss within the operational range tested.

In terms of system performance, the embedded control unit, composed of the ATmega8 and ATmega32 microcontrollers, effectively processed sensor inputs and executed the PID control algorithms, enabling the robot to maintain balance even under slight external disturbances. The practical deployment of the communication system onto the vehicle (Figure 9) illustrated that the compact integration of the wireless module did not interfere with the dynamic stabilization of the system.

The full vehicle prototype (Figure 10) successfully demonstrated the inverted pendulum control principle. Throughout testing, the robot maintained an upright position and responded accurately to directional commands (forward, backward, left, right) issued via the PC interface. The LCD feedback system allowed real-time monitoring of system parameters, improving user interaction and facilitating fine-tuning of the PID control parameters.

However, minor limitations were observed during testing. The vehicle's response time showed slight latency under longer-range wireless transmission, primarily due to the lower baud rate (2400 bps) used to enhance stability. Moreover, while the current control algorithm handled moderate disturbances well, performance degradation was noticeable under aggressive tilting or rough terrain, suggesting a need for enhanced control strategies, such as adaptive PID control or implementation of a Kalman filter for sensor fusion.

Overall, the results affirm that the designed system meets the intended objectives: demonstrating wireless real-time control of a self-balancing two-wheeled vehicle, validating modular embedded design, and offering a platform for further research in advanced control algorithms and wireless robotic systems.

4 Conclusion

The research successfully demonstrates the development of a self-balancing two-wheeled vehicle based on the inverted pendulum principle, integrating both hardware and software components with wireless communication capabilities. By employing the MRF24J40MA module in conjunction with AVR microcontrollers, the system enables real-time data transmission and remote control via a PC interface. The control software allows users to input PID parameters and monitor the vehicle's state dynamically, creating a responsive and adaptable platform for balance and motion control. Hardware subsystems, including motion sensors, motor drivers, and an LCD interface, were effectively integrated into a compact and functional design. Experimental validation through various test setups confirmed the system's ability to maintain balance and respond accurately to remote commands. This prototype offers a valuable foundation for further development in wireless robotic control systems, educational tools for control theory, and advanced mobility research. Future enhancements may focus on refining the algorithm, improving energy efficiency, and expanding autonomy for more complex real-world applications.

References

[1] SURENTHER, I., SRIDHAR, K.P., ROBERTS, M.K.: Maximizing energy efficiency in wireless sensor networks for data transmission: A Deep Learning-Based Grouping Model approach, *Alexandria Engineering Journal*, Vol. 83, pp. 53-65, 2023. https://doi.org/10.1016/j.aej.2023.10.016

[2] PANAHI, U., BAYILMIŞ, C.: Enabling secure data transmission for wireless sensor networks based IoT applications, *Ain Shams Engineering Journal*, Vol. 14, No. 2, 101866, pp. 1-11, 2023. https://doi.org/10.1016/j.asej.2022.101866

[3] ALUVALU, R., KUMARAN, S.V.N., THIRUMALAISAMY, M., BASHEER, S., ALDHAHRI, E.A., SELVARAJAN, S.: Efficient data transmission on wireless communication through a privacy-enhanced blockchain process, *PeerJ Computer Science*, Vol. 9, e1308, pp. 1-24, 2023. https://doi.org/10.7717/peerj-cs.1308

[4] LIAZID, H., LEHSAINI, M., LIAZID, A.: Data transmission reduction using prediction and aggregation techniques in IoT-based wireless sensor networks, *Journal of Network and Computer Applications*, Vol. 211, 103556, pp. 1-10, 2023. https://doi.org/10.1016/j.jnca.2022.103556

[5] KHUSHBOO, J., KUMAR, A., SINGH, A.: Data transmission reduction techniques for improving network lifetime in wireless sensor networks: An up-to-date survey from 2017 to 2022, *Transactions on Emerging Telecommunications Technologies*, Vol. 34, No. 1, e4674, 2023. https://doi.org/10.1002/ett.4674

[6] JACKO, P., DURANKA, P., VARGA, R.: Advantages of Bistable Microwires in Digital Signal Processing, *Sensors*, Vol. 24, No. 8, 2423, pp. 1-24, 2024. https://doi.org/10.3390/s24082423

[7] PAWASE, C., D'SOUZA, S.: *Revolutionizing Biomedical Engineering with Microcontroller Applications: A Comprehensive Review*, In: Tripathi, A., Soni, A., Tiwari, M., Swarnkar, A., Sahariya, J. (eds) Intelligent Computing Techniques for Smart Energy Systems. ICTSES 2023. Lecture Notes in Electrical Engineering, Vol. 1277, Springer, Singapore, 2023. https://doi.org/10.1007/978-981-97-8429-5_19

[8] TAMBA, V.K., BIAMOU, A.L.M., PHAM, V.-T., GRASSI, G.: Multistable memristor synapse-based coupled bi-hopfield neuron model: Dynamic analysis, microcontroller implementation and image encryption, *Electronics*, Vol. 13, No. 12, 2414, pp. 1-19, 2024. https://doi.org/10.3390/electronics13122414

[9] BROCKMANN, S., SCHLIPPE, T.: Optimizing convolutional neural networks for image classification on resource-constrained microcontroller units, *Computers*, Vol. 13, No. 7, 173, pp. 1-18, 2024. https://doi.org/10.3390/computers13070173

[10] LEE, D., KIM, J.-S., HONG, S.: Dual-Core based Microcontrollers Inference Design and Performance

Analysis, *IEEE Access*, Vol. 12, pp. 120326-120336, 2024. https://doi.org/10.1109/ACCESS.2024.3443406

[11] YAN, C., LI, X.: Research on Stability Control System of Two-Wheel Heavy-Load Self-Balancing Vehicles in Complex Terrain, *Applied Sciences*, Vol. 14, No. 17, 7682, pp. 1-15, 2024. https://doi.org/10.3390/app14177682

[12] DEULGAONKAR, V.R., NAIKWADI, V.A., ROTE, P.V., KAWADE, S.A., MODAKWAR, S.P., MANE, R.S.: Design and Analysis of Self-Balancing Two Wheeler Vehicle, *International Journal of Vehicle Structures & Systems*, Vol. 15, No. 5, pp. 716-720, 2023. https://doi.org/10.4273/ijvss.15.5.23

[13] MA, Y., MENG, F., XIONG, S.: Design and Implementation of a Two-Wheeled Self-Balancing Car Using a Fuzzy Kalman Filter, *Applied Sciences*, Vol. 14, No. 12, 5296, pp. 1-14, 2024. https://doi.org/10.3390/app14125296

[14] GHAHREMANI, A., RIGHETTINI, P., STRADA, R.: Simultaneous adjustment of balance maintenance and velocity tracking for a two-wheeled self-balancing vehicle, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, Vol. 238, No. 10, pp. 4608-4627, 2024. https://doi.org/10.1177/09544062231209144

[15] SUJITHA, S., DIXEN, D.A., DEVADIGA, D.V., DONY SNEHIT, P., HARSHITA, K.: *Design of Electric Vehicle with Self-Balance Approach using Gyroscopic Effect and Autonomous Systems*, 2024 8th International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, pp. 540-543 2024. https://doi.org/10.1109/ICISC62624.2024.00095

[16] DHARMASIRI, R.A.D.D., JAYANANDA, M.K.: A Study of Two-Wheel Self-Balancing Robot Dynamics for Control Systems Learning, *Vidyodaya Journal of Science*, Vol. 27, No. 2, pp. 25-36, 2024. https://doi.org/10.31357/vjs.v27i02.7834

[17] LI, S., HUANG, X., YOU, S., MAO, D., NIAN, H.: *Two Wheeled Self Balancing Vehicle Based on Wireless Remote Control*, 2024 IEEE 7th International Conference on Electronic Information and Communication Technology (ICEICT), Xi'an, China, 2024, pp. 1236-1239, 2024. https://doi.org/10.1109/ICEICT61637.2024.10670785

[18] YAN, J., YANG, Q., ZHU, C., CHEN, Y., WANG, Y.: Control system of two-wheeled self-balancing handling robot based on vision, *Journal of Physics: Conference Series*, Vol. 2975, No. 1, pp. 1-6, 2025. https://doi.org/10.1088/1742-6596/2975/1/012023

[19] DA ROSA, P.H.R., STEFFANELLO, V.R., CALDEIRA, A., RECH, C., MAIDANA, C.F., VENTURINI, S.F.: Two-wheeled self-balancing car, modeling, nonlinear system simulation and linearization, *Ciência e Natura*, Vol. 46, No. 3, pp. 1-15, 2024. https://doi.org/10.5902/2179460X87047

Review process
Single-blind peer review process.