



ORIGINAL CONTRIBUTION

DESIGN OF A BLUETOOTH-CONTROLLED ESP32-BASED SMART RC CAR WITH ULTRASONIC OBSTACLE DETECTION

¹Subhendu Hait, ²Subhadip Maity, ³Tushar Kanti Shit, ⁴Asim Kuilya

^{1,2}UG student, Department of Electronics and Communication Engineering, Haldia Institute of Technology, Haldia, Purba Medinipur, West Bengal

³School of Electronics Engineering, KIIT Deemed to be University, Bhubaneswar

⁴Department of Electronics and Communication Engineering, Haldia Institute of Technology, Haldia, Purba Medinipur, West Bengal

ABSTRACT

This paper presents the design and development of an RC car that can be operated wirelessly using a remote controller. The system consists of a chassis, DC motors, motor driver, battery, microcontroller, and a wireless communication module. The controller transmits commands to the receiver, which controls the movement of the car in different directions, including forward, reverse, left, and right. The main objective of this project is to demonstrate the integration of electronics, embedded systems, and wireless communication in a practical application. The RC car offers reliable performance, ease of operation, and low power consumption, making it suitable for educational, research, and hobby purposes. This project also provides a foundation for future enhancements such as obstacle avoidance, autonomous navigation, GPS tracking, and IoT-based remote monitoring.

KEYWORDS: RC Car, Remote Control, Obstacle Avoidance, Autonomous Navigation, IoT, GPS Tracking.

1. INTRODUCTION

Nowadays the radio controlled cars have developed to a high level and are used in many cases. Depending on their use, they vary in properties, design and special features. There are two major types of radio controlled cars, one that uses as energy source a conventional battery and the other type of radio controlled car utilizes a small internal combustion engine as an energy converter, which uses fossil fuel as energy source. Depending on its use a radio controlled car can be: (i) an army robot to detonate a buried bomb, (ii) an android robot to do a human job, (iii) a mobile robot to move around the environment like an automatic guided vehicle (AGV), (iv) a children's toy car or an RC car that can be operated with a radio wave, and (v) a radio controlled car with the purpose to detect fault on equipment with restricted access. These are some examples where a radio controlled car can be used. Radio control (regularly shortened to R/C or basically RC) is the utilization of radio signals to remotely control a gadget. Mechanical, military,

and logical research associations make utilization of radio-controlled vehicles also. This research work incorporates a robotic car consisting of an RF receiver, decoder, motor driver and DC motor [1], [2].

- ESP32 → main controller
- Bluetooth → wireless control
- Smart RC car → manual + autonomous operation
- Ultrasonic → obstacle detection
- Obstacle avoidance → automatic decision making

2. METHODS AND MATERIALS

2.1 Hardware Specification

Wheel: Wheels are of many types:

1. Standard / Fixed wheel
2. Orientable wheel
3. Ball wheel
4. Off-centred wheel

5. Omni wheel

6. Mecanum wheel

Here, we use a fixed wheel that can rotate forward or reverse. The center of the wheel is fixed to the robot chassis. The angle between the robot chassis and wheel plane is constant.

Geared DC Motor: A geared DC motor has a gear assembly attached to the motor. The speed of the motor is counted in terms of rotations of the shaft per minute and is termed as RPM. The gear assembly helps in increasing the torque and reducing the speed.

Servo Motor: A servo motor is a rotary or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration in a mechanical system.

Ultrasonic Sensor: An ultrasonic sensor is a device that uses ultrasonic sound waves to measure distance or detect objects. It consists of a transmitter that emits high-frequency sound pulses and a receiver that detects the reflected waves. By calculating the time taken for the sound to bounce back, the sensor determines the distance to the object [3].

Microcontroller:

- A microcontroller is a small integrated circuit used to control electronic devices and embedded systems [2], [4].
- It contains a processor (CPU), memory (RAM & ROM), and input/output ports on a single chip [2].
- It reads input from sensors or buttons, processes the data, and controls outputs like motors, LEDs, or displays [2].
- Microcontrollers are widely used in robots, home appliances, cars, and IoT devices [2].

ESP32: The ESP32 is a powerful and cost-effective microcontroller designed by Espressif Systems. It is a low-power System on Chip microcontroller with integrated Wi-Fi and dual-mode Bluetooth [3].

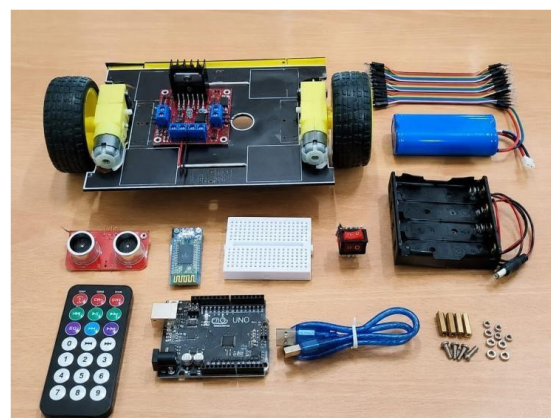


Figure 1: Components

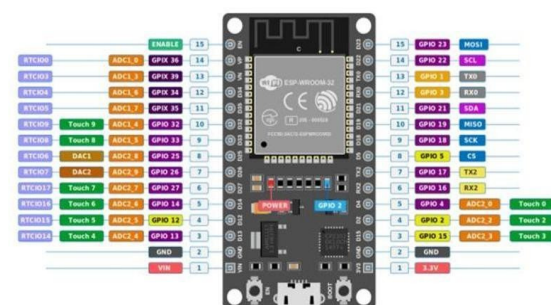


Figure 2: ESP32 pin configuration

2.2 Software

Arduino IDE: The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards, ESP boards etc..

Pulse Width Modulation (PWM): Pulse width modulation or PWM is a commonly used control technique that generates analog signals from digital devices such as microcontrollers. The signal thus produced will have a train of pulses, and these pulses will be in the form of square waves [5].

3. WORKING PRINCIPLE

The proposed RC car is built using an ESP32 microcontroller, an L298N motor driver, DC motors, an SG90/MG996R servo motor, and a rechargeable battery. The ESP32 serves as the main controller and communicates wirelessly with a smartphone or remote controller through Bluetooth or Wi-Fi. When the user sends a command such as Forward, Backward, Left, Right, or Stop, the ESP32 receives the command and processes it. For forward and backward movement, the ESP32 generates digital control signals that are sent to the L298N motor driver. The motor driver amplifies these signals and supplies the required current to the DC motors,

causing the wheels to rotate in the desired direction.

For steering, the ESP32 generates a Pulse Width Modulation (PWM) signal to control the SG90 or MG996R servo motor. The servo motor rotates to a specific angle, turning the front wheels left or right according to the received command. This provides accurate and smooth steering of the vehicle. A rechargeable battery powers the ESP32, L298N motor driver, servo motor, and DC motors. The coordinated operation of these components enables reliable wireless control, efficient movement, and precise steering [8].

This system provides real-time wireless control, low power consumption, and stable performance, making it suitable for educational, robotics, and research applications.

4. CIRCUIT DIAGRAM

The circuit diagram showing the interconnections among the ESP32, L298N motor driver, geared DC motors, servo motor, and ultrasonic sensor is illustrated in Fig. 3.

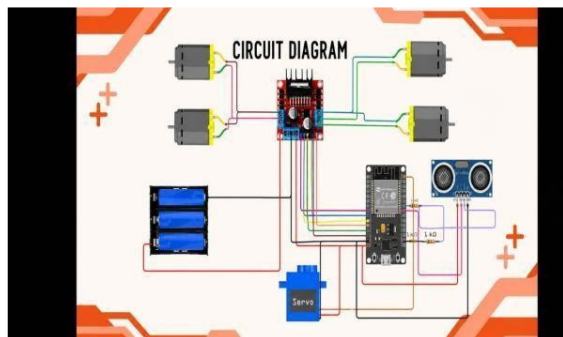


Figure 3: Circuit diagram

5. DISCUSSION

Before presenting the experimental results, the connections between the Arduino IDE and ESP32, computer and ESP32 through Bluetooth, and the controller and motor are tested.

5.1 Arduino–ESP32 Connection

The Arduino and ESP32 connection was first tested using the Arduino IDE. The ESP32 was connected to the computer using a USB cable, and the required Arduino C++ program was uploaded to the ESP32. After uploading the program, the Serial Monitor was opened to verify the operation of the ESP32. Test messages were displayed on the Serial Monitor to confirm that the program was running correctly [2].

During the initial testing, an error occurred and no proper response was observed. The problem was solved by checking the USB connection, power supply, program settings, and communication pins. After correcting the connections and uploading the program again, the ESP32 operated successfully.

5.2 Computer–ESP32 Bluetooth Connection

After successfully testing the ESP32, the next step was to establish a Bluetooth connection between the computer and ESP32. The ESP32 Bluetooth function was programmed using Arduino C++. The computer was paired with the ESP32 through Bluetooth. After successful pairing, the computer could send control commands wirelessly to the ESP32. The received Bluetooth commands were processed by the C++ program running on the ESP32.

Different commands were used to control the movement of the RC car, such as forward, reverse, left, right, and stop. The Bluetooth communication was tested at different distances to determine the effective operating range. The connection was considered successful when the ESP32 received the commands correctly and the car responded according to the transmitted command.

5.3 Controller and Motor Connection

After establishing the Bluetooth connection between the computer and ESP32, the controller and motor connections were tested. The Arduino C++ program was used to process the Bluetooth commands received by the ESP32. Four basic movement directions were tested: forward, reverse, left, and right. A stop command was also used to stop the motors.

When the forward command was received, both DC motors rotated in the forward direction. When the reverse command was received, both motors rotated in the opposite direction. For turning right, the motors were controlled to produce a right turn, while the left command produced a left turn.

During the initial testing, some commands were not received correctly and the motors did not always respond as expected. The problem was solved by checking the Bluetooth communication, motor-driver connections, power supply, and Arduino C++ program. The final test demonstrated that the ESP32 successfully

received Bluetooth commands from the computer and controlled the DC motors through the driver.

5.4 Range-Testing Results

Range testing was conducted in two environments: with obstacles and without obstacles. The tests were performed indoors. The obstacle environment contained walls, doors, furniture, and other objects, while the unobstructed environment provided a more open path.

The computer was connected to the ESP32 through Bluetooth, and commands were sent to control the RC car. The distance between the computer and the RC car was gradually increased to determine the effective Bluetooth communication range.

From Table 1, it can be observed that the Bluetooth communication performance depends on the distance and surrounding obstacles. At shorter distances, the ESP32 provides reliable Bluetooth communication and the RC car responds properly to the commands. At approximately 10 meters, the system operates effectively without obstacles, while obstacles reduce the signal quality. At approximately 15–20 meters, the Bluetooth signal becomes weaker

and the car may respond intermittently. At 25 meters, the Bluetooth signal was not detected during the test. Therefore, the practical operating range depends on factors such as distance, obstacles, interference, and the surrounding environment [3].

Table 1. Bluetooth Range-Testing Results

Distance	With Obstacles	Without Obstacles	Comments
1 meter	Excellent	Excellent	Bluetooth signal is strong.
5 meters	Excellent	Excellent	Bluetooth signal is strong.
10 meters	Good	Excellent	Signal strength decreases with obstacles.
15 meters	Poor (car still moves)	Good	Signal becomes weaker with obstacles.
20 meters	Poor (car still moves)	Poor (car still moves)	Bluetooth signal is weak and intermittent.
25 meters	Negative	Negative	Bluetooth signal is not detected.

6. RESULT

These authors control the motion of the car through the radio frequency application [7]. The movement of the car using the transmitter is shown in Fig. 4.

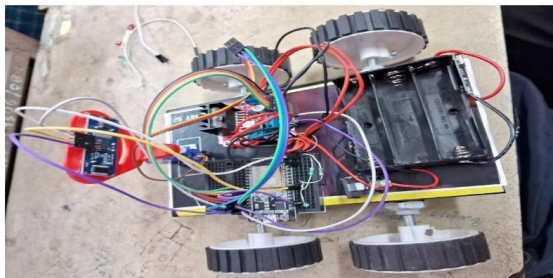


Figure 4: RC car

Overall, the experiment demonstrates that ESP32 Bluetooth communication combined with an Arduino C++ control program can provide reliable wireless control of the RC car within the tested range.

7. CONCLUSION

In this paper, it has been shown that it is possible to perform communication between the transmitter and receiver using radio frequency. The car can be moved forward and backward and can be turned left and right. The circuit can be modified and transformed into an amphibious mechanism which can run both on land and water efficiently. This technique may also be implemented for other purposes. The feasibility of the proposed technique has been shown.

REFERENCES

- [1] K. V. Alexandra, "Prototype model tests of a high speed wheeled amphibious vehicles," in *Proceedings of the Eighth Australasian Fluid Mechanics Conference*, University of Newcastle, N.S.W., Australia, 1983.
- [2] J.-D. Warren, J. Adams, and H. Molle, *Arduino Robotics*, 1st ed. Berkeley, CA, USA: Apress, 2011, 628 pp., ISBN 978-1-4302-3183-7.
- [3] T. Bräunl, *Embedded Robotics: From Mobile Robots to Autonomous Vehicles with Raspberry Pi and Arduino*, 4th ed. Singapore: Springer, 2022, 519 pp., ISBN 978-981-16-0803-2.
- [4] R. Kamal, *Embedded Systems: Architecture, Programming & Design*. New Delhi, India: McGraw-Hill Education, 2014.
- [5] R. Grimmer, *Arduino Robotic Projects: Build Awesome and Complex Robots with the Power of Arduino*. Birmingham, U.K.: Packt Publishing, 2014, ISBN 978-1-78398-982-9.
- [6] L. Vachhani, P. Vyas, and A. G. K. Arunkumar, *Embedded Control for Mobile Robotic Applications*. Hoboken, NJ, USA: Wiley, 2022, ISBN 978-1-119-81238-8.
- [7] K. Dubey and N. K. Khemka, "Line follower robot using radio frequency (RF) technology," *IOSR Journal of Mechanical and Civil Engineering*, vol. 3, pp. 1–5, 2013.
- [8] M. A. Othman, M. M. Ismail, H. A. Sulaiman, M. H. Misran, and Y. Z. A. Mohd, "Wireless controlled omni directional monitoring robot with video support," *International Journal of Engineering Research and Applications*, vol. 2, no. 4, pp. 2228–2232, 2012.
- [9] A. Canamar and L. Smrcek, "2050 visionary concept: Advance amphibious preliminary design," *International Journal of Engineering Research and Development*, vol. 3, no. 6, pp. 1–12, 2012.
- [10] S. Banerji, "Design and implementation of an unmanned vehicle using a GSM network with microcontrollers," *International Journal of Science, Engineering & Technology Research*, vol. 2, no. 2, 2013.
- [11] W. M. Khaing and K. Thiha, "Design and implementation of remote operated spy robot control system," *International Journal of Science, Engineering and Technology Research*, vol. 3, no. 7, 2014.
- [12] T. Bräunl, *Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems*, 3rd ed. Berlin/Heidelberg, Germany: Springer, 2008.
- [13] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. Cambridge, MA, USA: MIT Press, 2011, 472 pp., ISBN 978-0-262-01535-6.
- [14] R. Kamal, *Microcontrollers: Architecture, Programming, Interfacing and System Design*, 2nd ed. Pearson Education India.