



# DESIGN AND DEVELOPMENT OF AN ESP32-BASED SMART MOBILE MANIPULATOR FOR REMOTE OBJECT HANDLING

Kuncham Preetham<sup>1</sup>, Dr. K. Raja Kumar<sup>2</sup>

Department of Computer Science Engineering (AI/ML), Andhra University<sup>1</sup>

Guide, Associate Professor, Dept. of CS&SE, AUCE (A)<sup>2</sup>

**Abstract:** The integration of robotic mobility and object manipulation provides a practical solution for remote robotic operations. This work presents a manually controlled mobile robotic platform integrated with a servo-based robotic arm for basic object handling. An ESP32 microcontroller coordinates the DC motors, servo motors, and ultrasonic sensing according to user commands and sensed distance. The vehicle supports forward, backward, left, right, and stop operations, while the robotic arm provides base rotation, shoulder and elbow movement, and gripper open/close functions. A wireless control interface provides remote operation, and an ultrasonic sensor is used for obstacle detection. Experimental demonstration shows successful vehicle movement, obstacle stopping, arm actuation, and pick-and-place operation. The resulting platform is compact, portable, economical, and suitable for educational, remote-handling, and basic mobile robotic applications.

**Keywords:** ESP32, mobile manipulator, robotic arm, ultrasonic sensor, wireless control, servo motor, obstacle detection, pick-and-place.

## 1. INTRODUCTION

Robotic systems combine mobility, control, sensing, and mechanical operation to perform tasks remotely and efficiently. A mobile robotic platform equipped with a robotic arm can move from one location to another and interact with lightweight objects, combining the advantages of a mobile vehicle and a manipulator in one system. The developed platform uses an ESP32 microcontroller as the central control unit. Wireless commands are received by the controller and translated into vehicle and arm actions, while an ultrasonic sensor provides distance information for obstacle detection.

The work is intended as a compact and practical demonstration platform for embedded control, wireless communication, motor control, servo actuation, sensing, and robotic manipulation. The project material describes the system as a low-cost and flexible platform for educational, industrial, and remote-handling applications. The implemented prototype extends a conventional remote-controlled vehicle by adding a multi-joint arm and gripper, allowing basic pick-and-place tasks.

## 2. PROBLEM STATEMENT

Conventional remote-controlled cars primarily provide mobility and do not provide a mechanism for handling objects. Robotic arms can manipulate objects but require suitable positioning. Therefore, a compact system is required that combines vehicle movement and robotic arm operation while maintaining simple wireless manual control.

- Provide mobile movement and object-handling capability on a single robotic platform.
- Provide simple wireless manual control for both the vehicle and robotic arm.
- Use an ESP32-based controller to coordinate motors, servos, and sensor information.
- Provide obstacle detection using an ultrasonic sensor.

## 3. OBJECTIVES

- Develop a mobile robotic vehicle using DC geared motors and a suitable chassis.
- Integrate a servo-operated robotic arm with a gripper for lightweight object handling.
- Implement wireless manual control through the ESP32.
- Implement forward, backward, left, right, and stop vehicle commands.
- Implement base, shoulder, elbow, and gripper control.



- Detect nearby obstacles using an ultrasonic sensor and stop the vehicle at the configured threshold.
- Demonstrate a complete pick-and-place operation.

#### 4. LITERATURE SURVEY AND TECHNICAL BACKGROUND

The supplied project material identifies technical areas relevant to the proposed platform: ESP32-based robotic vehicle control, wireless-controlled robotic arm movement, ultrasonic-sensor-based obstacle avoidance, manual wireless robot control, and IoT-oriented robotic communication. These areas form the technical foundation for combining mobility, manipulation, sensing, and wireless control in one embedded robotic platform.

The ESP32 is used as the central embedded controller because it coordinates multiple input and output functions and provides integrated wireless communication capabilities. Motor-driver control interfaces the controller with DC motors, servo motors provide controlled angular movement of the robotic arm joints and gripper, and ultrasonic distance measurement supplies environmental information for obstacle detection.

The present implementation focuses on a practical prototype with manual wireless control and sensor-assisted stopping. AI/ML-based object recognition, advanced autonomous navigation, improved arm control, and IoT/cloud connectivity are identified in the supplied material as future extensions rather than implemented features.

#### 5. PROPOSED SYSTEM

The proposed system consists of a manually controlled robotic car integrated with a robotic arm. The ESP32 acts as the central control unit and receives commands through a wireless interface. DC motors provide vehicle movement, servo motors control the robotic arm and gripper, and an ultrasonic sensor measures the distance to an obstacle. The controller coordinates these functions according to user commands and sensor information.

- ESP32 microcontroller: central processing and control unit.
- DC geared motors: vehicle propulsion and directional movement.
- Motor driver: electrical interface for driving DC motors.
- Servo motors: base, shoulder, elbow, and gripper actuation.
- Ultrasonic sensor: obstacle-distance measurement.
- Battery: power source for the controller and actuators.
- Wireless control interface: remote command input.

#### 6. SYSTEM ARCHITECTURE

The system architecture is organized around the ESP32 controller. The wireless controller provides user commands to the ESP32. For vehicle operation, the ESP32 sends control signals to the motor driver, which drives the DC motors and wheels. For manipulation, the ESP32 drives the servo motors associated with the robotic arm. The power supply supports the controller, sensors, and actuators, while the ultrasonic sensor provides distance information for obstacle detection.

#### 7. HARDWARE COMPONENTS

| Component                 | Function                                                                                |
|---------------------------|-----------------------------------------------------------------------------------------|
| ESP32                     | Main controller; processes sensor information and controls the vehicle and robotic arm. |
| Ultrasonic Sensor         | Detects nearby objects by measuring distance.                                           |
| Motor Driver              | Drives and controls the DC motors.                                                      |
| DC Geared Motors          | Provide movement to the robotic vehicle.                                                |
| Robot Chassis with Wheels | Mechanical frame and support for the vehicle.                                           |
| Servo Motors              | Control robotic arm joints and gripper.                                                 |
| Battery                   | Supplies power to the ESP32, sensors, motors, and servos.                               |
| Connecting Wires          | Provide electrical connections between components.                                      |

#### 8. ROLE OF ESP32

The ESP32 acts as the main controller of the Smart RC Car. It receives distance information from the ultrasonic sensor, processes the sensor data, and manages the control decisions required by the prototype. It controls the motor driver for vehicle movement and the servo motors for robotic-arm operation. Its integrated wireless communication capability supports remote control through the control interface. The controller therefore provides a single coordination point for mobility, sensing, and manipulation.



## 9. ROBOTIC CAR MECHANISM

The robotic car uses an ultrasonic sensor to measure the distance to nearby objects. The sensor provides distance information to the ESP32. Based on the configured obstacle condition, the ESP32 controls the motor driver and DC motors. When the path is clear, the vehicle can move forward; when an obstacle is detected within the specified threshold, the vehicle stops automatically. The control chain is: Ultrasonic Sensor → ESP32 → Motor Driver → DC Motors → Car Movement.

## 10. ROBOTIC ARM INTEGRATION

The robotic arm adds object-handling capability to the mobile platform. The arm is mounted on the vehicle chassis and is operated using servo motors for controlled movement of its joints and gripper. The demonstrated functions include base rotation in both directions, shoulder movement up and down, elbow movement, and gripper opening and closing. The ESP32 coordinates servo commands with vehicle functions, allowing the platform to approach an object and perform a basic pick-and-place task.

## 11. MANUAL WIRELESS CONTROL SYSTEM

The user provides movement and arm-control commands through a wireless control interface. The ESP32 receives and interprets the commands and generates the required motor or servo actions. The vehicle supports Forward, Backward, Left, Right, and Stop commands. The arm interface provides Base +/−, Shoulder +/−, Elbow +/−, Gripper Open, and Gripper Close controls. The demonstrated interface is a browser-based Wi-Fi interface hosted by the ESP32. For the vehicle: User Command → ESP32 → Motor Driver → DC Motors → Vehicle Movement.

For the robotic arm: User Command → ESP32 → Servo Motors → Robotic Arm Movement.

## 12. COMMUNICATION AND COORDINATION

The ESP32 acts as the central communication and coordination unit. User commands are transmitted wirelessly to the controller, which coordinates vehicle and arm operations. The motor driver controls the DC motors, servo motors control the manipulator, and the ultrasonic sensor supplies environmental distance information. This coordination enables the integrated platform to perform navigation and manipulation using a common controller.

## 13. SOFTWARE REQUIREMENTS

- Arduino IDE: programming and uploading firmware to the ESP32.
- Embedded C/C++: development of the control program.
- ESP32 board package: libraries and board support.
- Motor and servo libraries: control of vehicle motors and arm servos.
- Wireless communication support: communication between the control interface and ESP32.
- Serial Monitor: testing, debugging, and monitoring commands.

## 14. CONTROL LOGIC

The control logic manages the movement of the robotic car and robotic arm through the ESP32. The user first sends a movement or arm-control command. The ESP32 receives and interprets the command. For vehicle movement, it controls the motor driver and DC motors; for arm movement, it controls the servo motors. The resulting actions include Forward, Backward, Left, Right, Stop, arm-joint movement, and gripper operation.

User Command → ESP32 → Command Processing → Motor/Servo Actuation → Car and Robotic Arm Action.

## 15. OVERALL WORKING

During operation, the user communicates with the ESP32 through the wireless interface. The controller processes the received command and activates the corresponding motor or servo output. The ultrasonic sensor provides distance information for obstacle detection. The vehicle therefore combines manual mobility with sensor-assisted stopping, while the robotic arm performs manipulation commands. The complete workflow is: user/manual control → wireless communication → ESP32 controller → motor driver and servo motors → vehicle movement and robotic-arm operation.

## 16. PROTOTYPE IMPLEMENTATION

The final prototype integrates the vehicle chassis, DC motor drive system, ESP32 controller, ultrasonic sensor, battery supply, servo-operated robotic arm, and gripper. The prototype was powered on and tested using the wireless control interface. The demonstration confirms that the integrated system can perform vehicle movement, obstacle detection, arm movement, and object pick-and-place operations.



## 17. RESULTS AND DEMONSTRATION

The experimental results supplied with the project show successful operation of the major functions. The final demonstration includes the powered-on prototype, Wi-Fi control interface, vehicle movement in different directions, obstacle detection, robotic-arm movements, functional tests, and a pick-and-place sequence. In the obstacle-detection demonstration, the vehicle stops automatically when an obstacle is within the configured threshold shown at approximately 12 cm, while it continues forward when the obstacle is beyond 20 cm. The robotic-arm demonstration shows base rotation, shoulder movement, gripper opening and closing, and a complete object pick-and-place sequence.

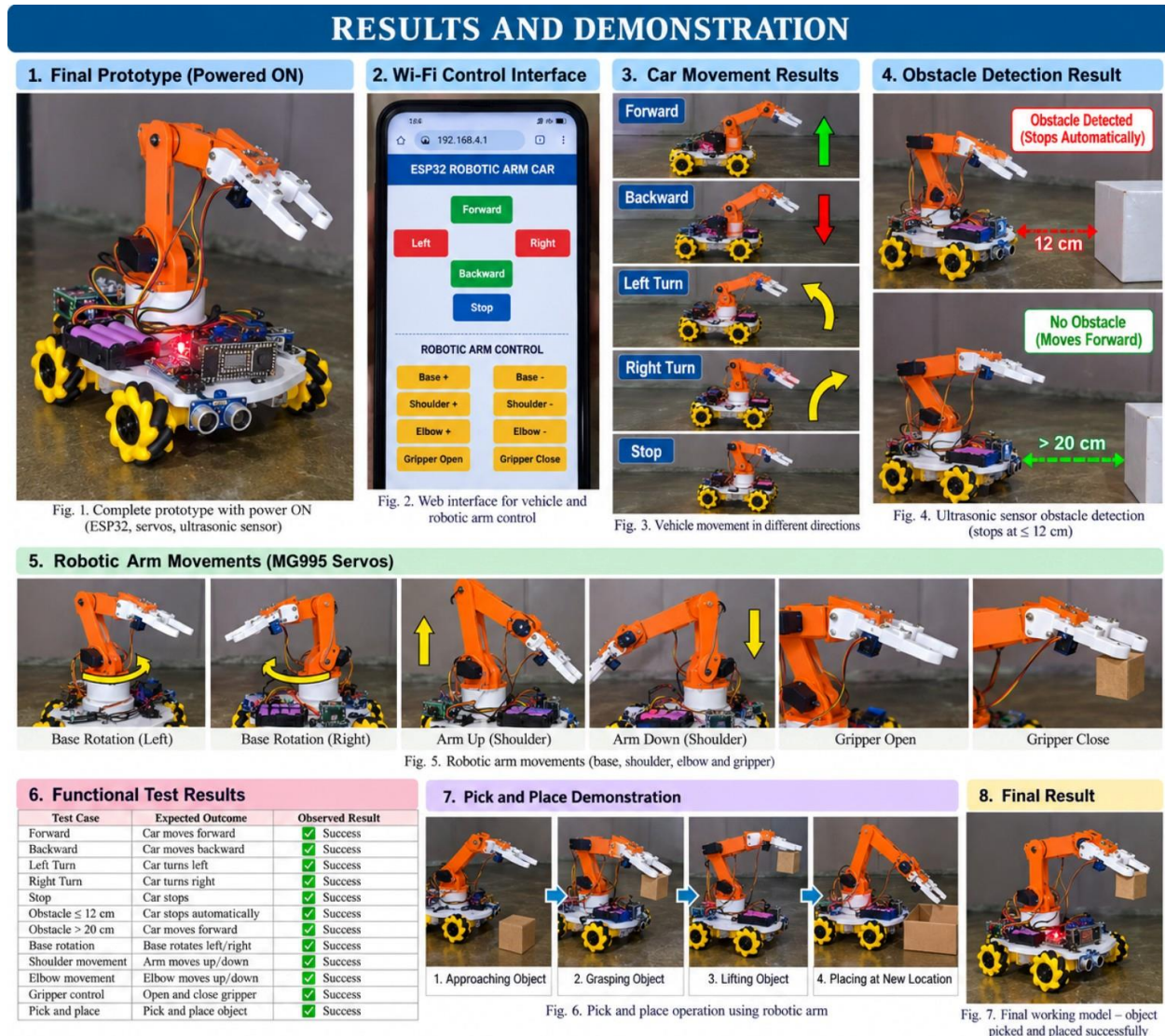


Fig. 1. Results and demonstration of the final prototype, wireless control, vehicle movement, obstacle detection, robotic-arm actuation, functional tests, and pick-and-place operation.

## 18. FUNCTIONAL TEST RESULTS

| Test Case             | Expected Outcome        | Observed Result |
|-----------------------|-------------------------|-----------------|
| Forward               | Car moves forward       | Success         |
| Backward              | Car moves backward      | Success         |
| Left Turn             | Car turns left          | Success         |
| Right Turn            | Car turns right         | Success         |
| Stop                  | Car stops               | Success         |
| Obstacle $\leq 12$ cm | Car stops automatically | Success         |
| Obstacle $> 20$ cm    | Car moves forward       | Success         |



|                   |                         |         |
|-------------------|-------------------------|---------|
| Base rotation     | Base rotates left/right | Success |
| Shoulder movement | Arm moves up/down       | Success |
| Elbow movement    | Elbow moves up/down     | Success |
| Gripper control   | Open and close gripper  | Success |
| Pick and place    | Pick and place object   | Success |

The functional test table indicates successful completion of all listed test cases in the supplied demonstration. The prototype therefore achieved the intended basic mobility, obstacle-response, robotic-arm actuation, gripper control, and pick-and-place functions.

### 19. PICK-AND-PLACE DEMONSTRATION

The pick-and-place operation is performed in four stages: approaching the object, grasping the object with the gripper, lifting the object, and placing it at a new location. This sequence demonstrates the practical integration of mobile positioning and robotic manipulation on the same platform. The supplied results image shows the object being successfully handled and placed by the final working model.

### 20. APPLICATIONS

- Material Handling: picking and moving lightweight objects.
- Remote Inspection: manually operating the vehicle in difficult-to-reach areas.
- Educational Projects: learning robotics, ESP32, sensors, motors, and wireless control.
- Warehouse Assistance: basic object transportation and handling.
- Surveillance and Monitoring: remote movement with sensor-based assistance.
- Industrial Automation: demonstration of the basic concept of mobile robotic systems.

### 21. ADVANTAGES

- Manual wireless control for easy operation.
- Compact and efficient ESP32-based control platform.
- Robotic arm enables basic pick-and-place operations.
- Ultrasonic sensing supports obstacle detection and safe stopping.
- Low-cost and portable robotic platform.
- Vehicle movement and arm operation are coordinated through a single controller.

### 22. FUTURE SCOPE

- AI/ML-based object recognition for more intelligent operation.
- Improved autonomous navigation using advanced sensors.
- More precise and flexible robotic-arm control.
- IoT/cloud connectivity for remote monitoring and control.

These future directions are proposed extensions identified in the supplied project material and are not claimed as implemented features of the present prototype.

### 23. CONCLUSION

The developed ESP32-based smart mobile manipulator provides a compact and low-cost platform for combining mobile vehicle control with robotic-arm operation. The ESP32 coordinates wireless commands, motor control, servo movements, and ultrasonic sensing within a single prototype. The demonstrated results show successful forward, backward, left, right, and stop operations; obstacle-based stopping; robotic-arm movements; gripper operation; and a complete pick-and-place sequence. The work demonstrates the practical integration of embedded control, wireless communication, sensors, DC motors, servo motors, and robotic manipulation in a mobile robotic platform.

### REFERENCES

- [1] "Design and Implementation of a Smart Car Robot Based on ESP32," referenced in the project material for ESP32-based robotic vehicle control.
- [2] "Design and Development of a Wireless Controlled Robotic Arm," referenced for robotic-arm movement and servo control.
- [3] "Obstacle Avoidance Robot Using Ultrasonic Sensor and Arduino," referenced for sensor-based navigation.
- [4] "Development of a Bluetooth Controlled Robot Using Arduino," referenced for manual wireless control.
- [5] "Design and Implementation of an IoT Based Smart Robot," referenced for wireless communication and remote robotic control.
- [6] "Tower Pro, "MG995 High Speed Servo Actuator," component specification sheet."



- [7] “Directronics, “TowerPro MG995 Servo Motor (180° Rotation),” product specification page, accessed Oct. 2026.”
- [8] “SparkFun Electronics, “HC-SR04 Ultrasonic-Sensor,” image and product information, Wikimedia Commons, CC BY 2.0, 2017.”
- [9] “A. Barua and K. Andersson, “Design and Implementation of an Ultrasonic Sensor-Based Obstacle Avoidance System for Arduino Robots,” Proc. ICICT4SD, 2023, doi: 10.1109/ICICT4SD59951.2023.10303550.”