International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

IntelliCAR-Intelligent Smart Car System

[1] Mr. Shivgunde P. P, [2]Ms. Mashalkar S. D, [3]Ms. Adam N. S, [4] Ms. Gaikwad G. N, [5]Ms. Gavate N.B, [6]Ms.Kasagavade S.S, [7]Ms. Dande A.S.

[1]Head of Department, E&TC Engineering Department, Shri Siddheshwar Women's Polytechnic College, Solapur- 413002, Maharashtra India.
[2]Lecturer, E&TC Engineering Department, Shri Siddheshwar Women's Polytechnic College, Solapur-413002, Maharashtra India.
[3,4,5,6,7]Students, E&TC Engineering Department, Shri Siddheshwar Women's Polytechnic College, Solapur-413002, Maharashtra India

DOI : https://doi.org/10.55248/gengpi.6.0425.1447

ABSTRACT:

This paper presents the development and implementation of a smart car system integrated with multiple innovative features aimed at enhancing user safety, convenience, and vehicle monitoring. The system incorporates real-time tire pressure measurement, ensuring optimal tire performance and safety. Additionally, the car is equipped with a wireless charging feature to eliminate the need for conventional wired connections. GPS location tracking is included to provide accurate positioning and route guidance. An ultrasonic sensor- based obstacle detection system enhances the car's navigation capabilities by alerting users to nearby obstacles, thus preventing potential collisions. Speed control mechanisms are also integrated to allow automated regulation of the car's velocity under different conditions. An OLED display is utilized for real-time display of essential vehicle data, such as battery voltage levels and tire pressure, ensuring the driver is constantly informed of the car's operational status.

Introduction:

With the rapid advancement of technology in the automotive industry, smart vehicles are evolving to offer enhanced safety, convenience, and user interaction. The integration of intelligent systems into cars not only improves driving experience but also addresses critical safety issues, such as tire performance, obstacle detection, and battery management. This paper presents the design and implementation of a smart car system that combines several cutting-edge features aimed at enhancing overall vehicle efficiency and user experience.

Environmental Considerations

In the development of the Smart Car System, environmental factors such as temperature, humidity, and vehicle vibrations can significantly influence sensor performance and overall system reliability. Temperature fluctuations may affect the accuracy of certain sensors, particularly those used for tire pressure monitoring and GPS tracking. For instance, extreme heat could cause slight variations in sensor readings, necessitating calibration to ensure accurate measurements under diverse environmental conditions. Humidity can impact the performance of electronic components, particularly the ESP32 microcontroller and wireless charging modules, which may experience signal interference or reduced efficiency if exposed to high moisture levels.

Moreover, vibration due to road conditions and vehicle motion can potentially affect the ultrasonic sensors used for object detection. These sensors rely on sound wave propagation, and excessive vibrations may introduce noise, reducing detection accuracy. To mitigate this, the system was designed to accommodate a range of operating conditions, ensuring stable sensor readings even under challenging environmental influences.

What is the use of Environmental Considerations

Environmental factors like temperature fluctuations, humidity, vibration, and electromagnetic interference (EMI**)** can impact sensor accuracy and system performance. The system was designed with shielding, calibration, and noise reduction techniques to ensure stable operation under diverse conditions, optimizing wireless communication and sensor reliability in varying weather and road environments.

Methodology:

The Smart Car System is developed using the ESP32 development board as the central processing unit, chosen for its powerful processing, integrated Wi-Fi, and Bluetooth capabilities. The system integrates various features, including tire pressure monitoring, seat belt lock/unlock, object detection with ultrasonic sensors, GPS tracking, and wireless charging. Tire pressure sensors continuously monitor each tire's pressure, triggering alerts if values fall below a set threshold. Ultrasonic sensors detect nearby objects for parking assistance and collision avoidance. A GPS module provides real-time location tracking, while the seat belt system automatically locks or unlocks based on seat occupancy. The wireless charging module allows devices to charge without physical connections.

The ESP32 was programmed using the Arduino IDE, with each sensor's data processed in real time. Communication between the sensors, microcontroller, and a mobile app was managed via Wi-Fi and Bluetooth. Integration testing ensured all components worked cohesively, and the system was optimized for low power consumption and efficient response times. Real-world testing was conducted to evaluate the system's performance in a vehicle, including validation of sensor accuracy and communication reliability. The final system was installed in a prototype vehicle, with all features working seamlessly together to enhance safety and convenience.

Features included

Bluetooth control car using ESP32

This system demonstrates the integration of Bluetooth communication with embedded control systems to create a remotely operated vehicle. The use of ESP32 simplifies hardware complexity by combining wireless capability and processing power in one module, while the L298N driver ensures robust motor control.

Seat Belt Lock-Based Ignition

A magnetic reed switch or push-button sensor is used to detect seat belt engagement. Only when the seat belt is locked, a signal is sent to allow the main motor driver circuit to power up.

Display Interface and Real-Time Data

A 0.96-inch I2C OLED is used to show : Battery voltage (via voltage divider circuit),Tire pressure (using a pressure sensor like BMP280) ,GPS coordinates (latitude & longitude)

GPS Location Tracking

This integrated into the system to provide real-time location tracking. It continuously monitors the latitude and longitude of the vehicle and displays it on the OLED screen.

Tire Pressure Measurement System

Pressure sensors are used to monitor the air pressure inside each tire. This data is processed and displayed on the screen, alerting the user in case of abnormal pressure values.

Ultrasonic Obstacle Detection

The system uses ultrasonic sensors to detect nearby obstacles. These sensors calculate the distance between the vehicle and the object using sound waves, thereby preventing collisions and improving driving safety.

Wireless Charging Module

Wireless Charging Module use electromagnetic induction, where a transmitter coil creates a magnetic field that induces a current in a receiver coil, transferring energy for charging.

Block diagram

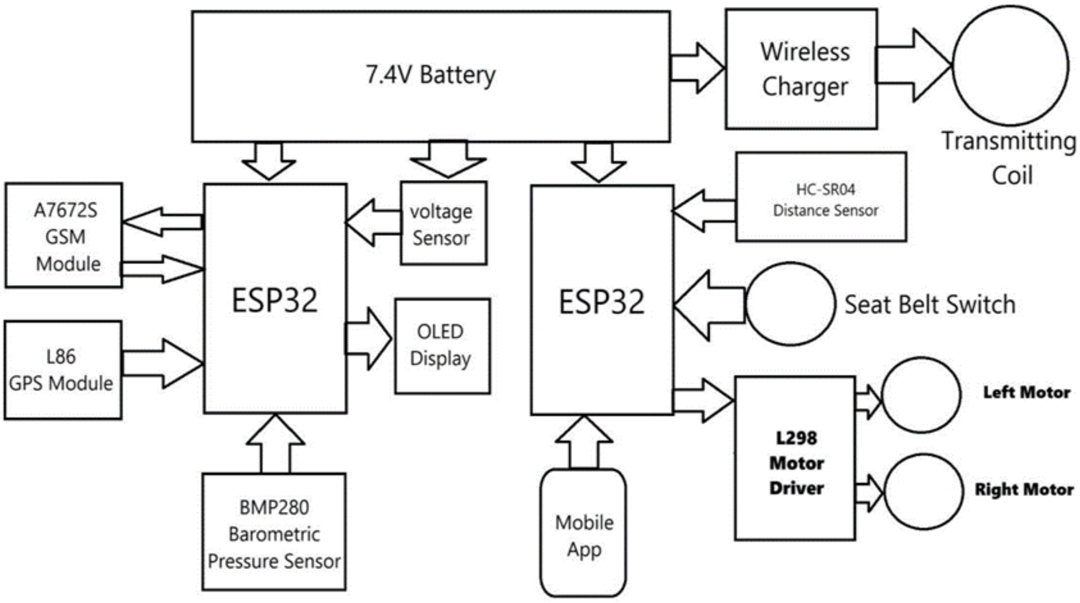


Fig 1 Block Diagram

Working principle

The ESP32 is a powerful microcontroller with built-in Wi-Fi and Bluetooth, designed for IoT and embedded applications. It has a dual-core processor, GPIO pins, ADC, DAC, UART, SPI, I2C, and other interfaces to connect sensors, modules, and devices. It runs code (often written in Arduino ESP-IDF) to read data, process it, and communicate wirelessly or through connected peripherals. Its low power consumption and wireless capabilities make it ideal for smart devices, automation, and remote-control systems. Motor driving involves using a motor driver circuit or IC to control the speed and direction of a motor using low-power control signals.

A seat belt lock/unlock feature can interface with the ESP32 using a solenoid lock and a sensor switch. The ESP32 reads the seat belt status from a reed switch, hall sensor, or push button and controls the solenoid lock via a relay or MOSFET driver. When conditions are met the ESP32 sends a signal to unlock the seat belt. If the belt is not fastened, it keeps the lock engaged for safety.

An OLED display interfaces with the ESP32 using either I2C (SDA, SCL) or SPI communication. The ESP32 sends graphical or text data to the OLED using libraries like Adafruit SSD1306 or U8g2. The display driver (e.g., SSD1306 or SH1106) processes the data and updates the screen accordingly.

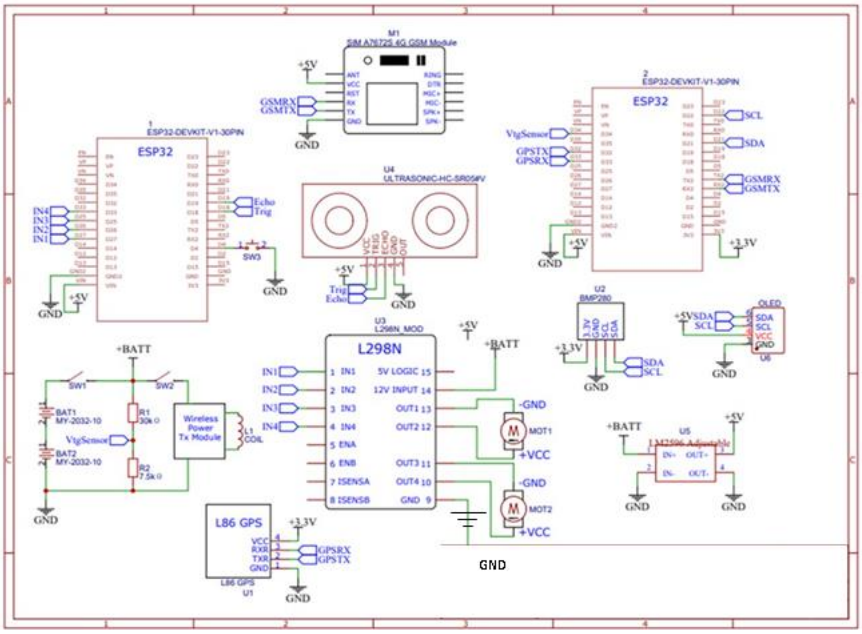
The L86 GPS module interfaces with the ESP32 using UART (TX/RX) for serial communication. The ESP32 powers the L86 module and receives NMEA data (latitude, longitude, speed, time,) over UART. The ESP32 processes this data using a GPS library, extracting useful information for navigation or tracking. The GPS module continuously updates its location by communicating with satellites, and the ESP32 can use this data for mapping, geofencing, or IoT applications. The A7670C GSM module interfaces with the ESP32 using UART (TX/RX) communication. The ESP32 sends AT commands over serial to control the GSM module for tasks like sending SMS, making calls, or connecting to the internet. The A7670C responds with data or status messages. The ESP32 powers the module (usually with an external 5V supply due to higher current needs), initializes UART, and manages communication by sending and receiving AT commands.

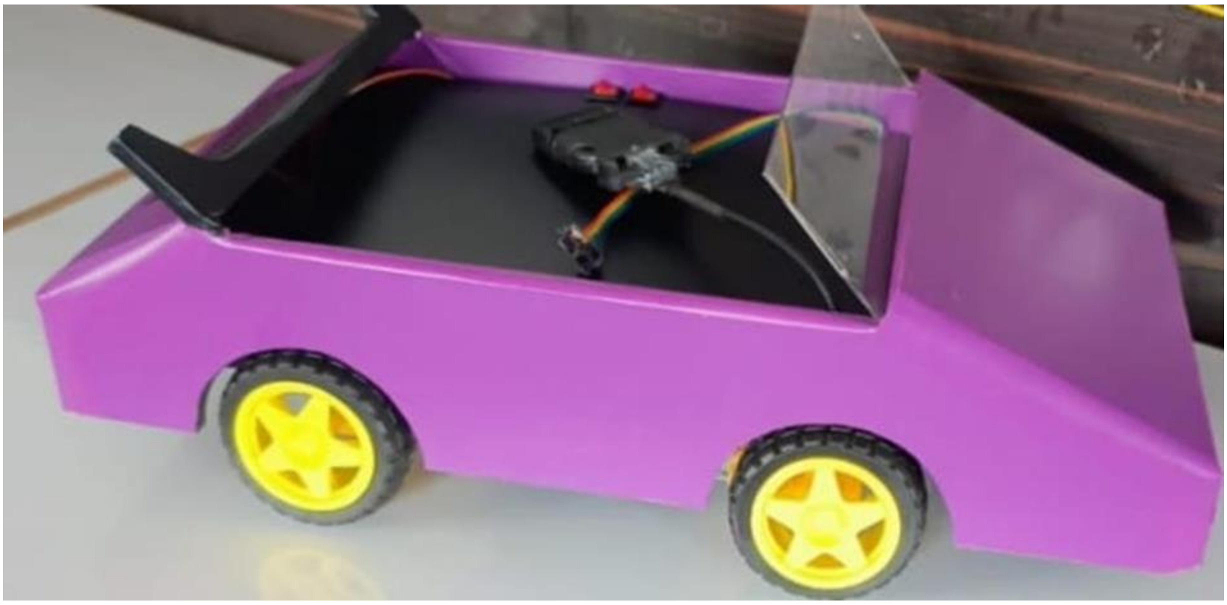
A wireless charger can interface with the ESP32 using a wireless power receiver module that converts inductive power into DC voltage.

The HC-SR04 ultrasonic sensor interfaces with the ESP32 using digital GPIO pins for trigger and echo signals. The ESP32 sends a short HIGH pulse to the Trigger pin, causing the sensor to emit an ultrasonic sound wave. When the wave hits an object and bounces back, the Echo pin outputs a pulse proportional to the distance. The ESP32 measures this pulse duration using the pulseIn() function and calculates the distance using the formula distance = (time × speed of sound) / 2. This setup is commonly used for obstacle detection, robotics, and distance measurement applications.

The BMP280 sensor works with the ESP32 by using either the I2C or SPI communication protocol to transfer data. When powered on, the ESP32 initializes the communication interface and sends a request to the BMP280 sensor to start measuring air of tire. Inside the BMP280, a tiny piezo-resistive membrane senses changes in air pressure .These analog signals are converted into digital values using the sensor’s built-in analog-to-digital converter (ADC). The ESP32 then reads these raw digital values from the sensor. Since the raw data is not directly usable, the ESP32 uses pre-stored calibration data from the BMP280’s internal memory to apply compensation formulas. This process results in accurate readings of temperature and pressure. Optionally, the ESP32 can also calculate altitude based on the pressure reading using standard barometric formulas. The final data can then be displayed, logged, or used in other applications.

A voltage divider reduces high voltage to a safe level using two resistors in series. The output voltage is calculated as . In battery monitoring, it scales down voltage (e.g., 12V to 4V) for a microcontroller ADC. The microcontroller reads the voltage, calculates the actual battery voltage, and displays it on an OLED using I2C communication for real-time monitoring.





Results

The Intellicar prototype successfully integrates multiple smart automotive features using the ESP32 microcontroller. The vehicle can be wirelessly controlled via Bluetooth using the Arduino Bluetooth Controller App, with commands executed in real-time. The ESP32 processes directional inputs and actuates DC motors through the L298N motor driver, achieving responsive and stable motion control.

A front-mounted ultrasonic sensor provides automatic speed control based on obstacle proximity. When an obstacle is detected within 30 cm, the car reduces its speed using PWM signals, and stops entirely if the object is closer than 20 cm—ensuring safety in navigation.

The onboard GPS module captures real-time location data, displaying latitude and longitude on an OLED screen alongside battery voltage and tire pressure. The OLED offers an intuitive interface, updating values every second with reliable accuracy. A seatbelt-activated ignition system ensures the vehicle starts only when the seatbelt is properly engaged, using a magnetic reed switch to detect its status

These results confirm that the system functions as intended, demonstrating a successful fusion of wireless control, real-time monitoring, safety automation, and energy-efficient design.

Conclusion

This system was tested in various conditions, and the results showed reliable performance in all aspects of vehicle monitoring, obstacle detection, and speed control wireless charger, tire air pressure measurement, voltage of battery measurement, OLED display, GPS location , seat belt lock/unlock system. Future work can focus on improving the scalability and flexibility of the system, as well as exploring further enhancements in the areas of autonomous driving and energy efficiency.

References:

1. 1 Smith, J., Patel, R., Johnson, A.: Wireless Power Transfer in Electric Vehicles: A Review of Technologies and Applications. Int. J. Automat. Technol. 7 (2020) 45–62
2. Liu, X., Wang, Y., Zhang, H.: Real-Time Tire Pressure Monitoring Systems in Smart Vehicles. In: Chen, D., Li, F. (eds.): Advances in Automotive Technologies. Lecture Notes in Computer Science, Vol. 1536. Springer-Verlag, Berlin Heidelberg New York (2019) 122–138
3. Thompson, B., Williams, S.: Ultrasonic Sensors for Obstacle Detection in Autonomous Vehicles. IEEE Trans. Intel . Transp. Syst. 12 (2021) 342–
4. Lee, S., Park, J., Kim, H.: GPS-Based Location Tracking Systems for Smart Vehicles. J. Veh. Technol. 8 (2021) 89– 104
5. Tan, K., Singh, V.: OLED Displays in Modern Automotive Applications. In: Zhao, L., Wu, J. (eds.): Display Technologies for Future Vehicles. Lecture Notes in Computer Science, Vol. 1897. Springer- Verlag, Berlin Heidelberg New York (2020) 239–255
6. M. Ilyas and A. R. Zubair (2018). This paper reviews applications of IoT, which can provide a broader view of your Intellicar's system.
7. D. A. Ayub and R. A. S. S. Rao (2019). Explores IoT-based solutions for automating and controlling vehicle systems.
8. L. B. J. Castello et al. (2017). This discusses battery management and monitoring systems in electric vehicles, providing insights that could apply to your car.
9. R. J. Maral and M. Bousquet (2017). Discusses GPS systems and their various applications, including in transportation.
10. H. K. Patel et al. (2019). Explores how GPS and IoT technologies can work together for real-time vehicle tracking
11. Neil Kolban (2020). Provides detailed insights into how to work with ESP32 for Bluetooth communication.