IJMECIE International Journal of Mechanical, Electrical and Civil Engineering
E-ISSN: 3047-4531
P-ISSN: 3047-4523

Research Article

Design and Implementation MINIOT Car Using Esp8266 and L298

Diyajeng Luluk Karlina*

Vocational Education in Electrical Engineering; Faculty of Teacher Training and Education; Universitas Sultan Ageng Tirtayasa; Serang; Indonesia; e-mail: diyajeng@untirta.ac.id
* Corresponding Author: Diyajeng Luluk Karlina

Abstract: This research presents the design, development, and implementation of a mini smart car prototype that operates using Internet of Things (IoT) technology. The system is built around the ESP8266 microcontroller (Amica version), which functions as the core processing unit responsible for handling Wi-Fi communication and data processing. The motion of the car is controlled by an L298 motor driver module that regulates the operation of DC motors. The entire system is powered by a 3.7-volt rechargeable battery, ensuring portability and energy efficiency. The study discusses in detail the hardware configuration, software programming, and integration of IoT-based control through a web or mobile interface. Functional testing of the prototype, named MINIOT, focuses on evaluating the responsiveness, stability, and reliability of remote control operations. The results are expected to show that the system can effectively receive and execute user commands while transmitting real-time telemetry data, such as motor status and connection indicators. This project demonstrates the feasibility of low-cost IoT-based automation for small-scale robotic applications.

Keywords: Design; Internet Of Things; Microcontroller ESP 8266; Modul L298; Smart Car

1. Introduction

The advancement of digital technology and embedded systems has driven the development of various automated devices that are increasingly sophisticated and efficient. One interesting implementation of this technology is in smart vehicles (smart cars), which are capable of moving and making decisions independently based on data from their surroundings. In educational and prototyping contexts, the Arduino Uno combined with IoT has become a popular choice due to its ease of programming and flexibility in integrating with various sensors and actuators, thus enabling the design of an automatic vehicle system that is simple yet functional(Chaudry, 2020).

The Internet of Things, or IoT, is the concept in which all objects in the world can communicate with one another as part of an integrated system that uses the internet as a connector(Swathi et al., 2018). The rapid development of the Internet of Things (IoT) has opened opportunities for innovation in various fields, including robotics. One particularly interesting area is the development of intelligent and connected mini robotic vehicles(Adi Winarno & Affandi, 2022).

This research focuses on the design of an IoT-based mini smart car, "MINIOT" which utilizes the ESP8266 (Amica) microcontroller as the control center and the L298 motor driver module for movement (Nuratch, 2017). The selection of the ESP8266 Amica is based on its built-in Wi-Fi capability, which facilitates remote communication with the control device. Its processing power and available GPIO pins make it ideal for managing the car's control logic and data communication (Rajunaik & Amareswer, 2023).

The L298 module was chosen for its ability to control two DC motors independently, enabling forward, backward, and turning movements on the MINIOT. An IoT-based mini smart car like MINIOT has potential applications in education, allowing interactive learning about electronics, programming, and IoT concepts. In addition, this platform is also appealing for hobbies and experimentation, enabling users to develop personalized control and integration with other sensors or IoT platforms (Zinkevich, 2021).

Received: August 30, 2025
Revised: September 19, 2025
Accepted: September 30, 2025
Published: October 11, 2025
Curr. Ver.: October 11, 2025

Although previous studies have explored the use of ESP8266 and L298 in robotics, research that specifically documents the design and implementation of an IoT-based mini smart car using the combination of both, with a focus on system architecture, control software, and functional testing, still needs to be further explored (Fransiska et al., 2013).

This research aims to present in detail the design and implementation process of the MINIOT prototype, focusing on the integration of the ESP8266 Amica for IoT connectivity and the L298 for motor control. The results of this study are expected to provide a meaningful contribution and serve as a reference for the development of IoT-based mini robotic systems using affordable and accessible components, as well as to validate the effectiveness of combining the ESP8266 and L298 in creating a responsive and interconnected mini smart car platform.

2. Proposed Method

In this research, several stages are applied to design a system based on engineering technology. The purpose of this system is to develop a prototype of a mini smart car named MINIOT, which can be controlled through the Internet of Things (IoT) network. The research method is carried out in several stages as follows:

Study Literature and Identification

The initial stage involves a study of the ESP8266 microcontroller (Amica version), the L298 motor driver, as well as the basic principles of IoT communication and DC motor operation. In addition, an analysis of the system requirements is carried out, focusing on movement functions and remote control.

Design System

The system design in this research includes:

Hardware Components

The system is developed using the ESP8266 Amica as the control center and Wi-Fi connection, the L298 motor driver to control two DC motors for wheel movement, and 3.7-volt lithium battery as the main power source.

Software Components

Programming is carried out using the Arduino IDE with a Blynk-base application interface, which enables users to control the car via the internet.

Prototype Design

After the design phase is completed, all components are assembled into a mini car prototype. Integration is carried out between the ESP8266, motor driver, DC motors, wheels, and battery, in accordance with the system's functional design.

Prototype System Testing

Testing is conducted to evaluate the system's performance in responding to movement commands from the application, the stability of the network connection, and the efficiency of power consumption. The car's response to directional commands (forward, backward, turn) is directly observed.

Prototype Evaluation and Analysis

The test results are analyzed to assess the feasibility of the system from both technical and operational aspects. The evaluation includes response speed, connection stability, and battery endurance during operation.

3. Results and Discussion

Integration of ESP8266 Microcontroller & Blynk

The design results of the MINIOT smart car using the ESP8266 microcontroller require integration with the Blynk software to control the movement of the MINIOT smart car, allowing it to be operated via a mobile phone.

```
// Include the library files
#define BLYNK_TEMPLATE_ID "TMPL6EerbseSu"
#define BLYNK_TEMPLATE_NAME "Mobil keren"
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
```

(*Source*: *Personal Documentation*, 2025)

Figure 1. Initialization of Blynk Template and ESP8266 Library

Figure 1 explains BLYNK_TEMPLATE_ID & BLYNK_TEMPLATE_NAME: These are the Blynk template ID and name used to connect the Blynk application with the device (ESP8266). BLYNK_PRINT Serial: Specifies that Blynk output will be displayed through the Serial Monitor. #include <ESP8266WiFi.h> & #include <BlynkSimpleEsp8266.h>. Import the libraries required for Wi-Fi connection and communication with the Blynk platform via ESP8266.

```
// Define the motor pins
#define ENA D0
#define IN1 D1
#define IN2 D2
#define IN3 D3
#define IN4 D4
#define ENB D5
```

(*Source*: *Personal Documentation*, 2025)

Figure 2. Motor Control Pin Definition Commands on ESP8266

Figure 2 explains the ESP8266 microcontroller pins used to control the motors through a motor driver (such as L298N). The ENA and ENB pins are used to control motor speed using PWM signals, while pins IN1 through IN4 are used to set the rotation direction of the two motors, with two pins assigned to each motor. This combination allows the microcontroller to control both the direction and speed of the motors independently, which is essential for controlling the movement of robotic vehicles.

```
// Variables for the Blynk widget values
int x = 50;
int y = 50;
int Speed;
```

(*Source*: *Personal Documentation*, 2025)

Figure 3. Joystick Variable Definition Commands and Blynk Connection Parameters

The code section in Figure 3 defines variables to store the values from the joystick widget in Blynk, namely x and y for movement direction and Speed for motor speed.

```
char auth[] = "hIkVaUuUfENoC9IBbJXnK4JEJYZNMiMR"; //Enter your Blynk auth token
char ssid[] = "Aldi Firdaus"; //Enter your WIFI name
char pass[] = "11111111"; //Enter your WIFI passowrd
```

Figure 4. Blynk and Wi-Fi Connection Variables

The code section in Figure 4 contains three important variables for connecting the microcontroller to the Blynk platform via a Wi-Fi network. The auth variable stores the authentication token obtained from the Blynk application, while ssid and pass are used to store the Wi-Fi network name and password, respectively. These three variables enable the ESP8266 device to connect to the internet and communicate with the Blynk server to perform remote control functions.

```
void setup() {
  Serial.begin(9600);
  //Set the motor pins as output pins
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
```

(*Source*: *Personal Documentation*, 2025)

Figure 5. Serial Initialization and Motor Pin Configuration

Figure 5 explains the part of the setup() function used to initialize serial communication and set pin modes as outputs. The command Serial.begin(9600); is used to start serial communication at a baud rate of 9600, which is useful for debugging purposes. Meanwhile, the pinMode() command is used to configure all motor control pins (ENA, ENB, IN1–IN4) as outputs, allowing these pins to control the motors according to the program commands.

```
  // Initialize the Blynk library
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
}
```

(*Source*: *Personal Documentation*, 2025)

Figure 6. Initialization of Connection to Blynk via Wi-Fi

Figure 6 shows the command Blynk.begin(auth, ssid, pass, "blynk.cloud", 80); which functions to connect the ESP8266 microcontroller to the Wi-Fi network and the Blynk platform. This function uses the authentication token (auth), Wi-Fi name (ssid), and Wi-Fi password (pass) to establish the connection. With this initialization, the device can connect to the Blynk server, allowing it to be controlled remotely through the Blynk application.

```
// Get the joystick values
BLYNK_WRITE(V1) {
  y = param[0].asInt();
}
//Get the slider values
BLYNK_WRITE(V2) {
  Speed = param.asInt();
}
```

(*Source*: *Personal Documentation*, 2025)

Figure 7. Input Function Initialization

Figure 7 shows the functions BLYNK_WRITE(V0), BLYNK_WRITE(V1), and BLYNK_WRITE(V2), which are used to handle input from widgets in the Blynk application. Each function captures the values sent by the user through the application and stores them into variables:
V0 retrieves the horizontal joystick value and stores it in the variable *x*.
V1 retrieves the vertical joystick value and stores it in the variable *y*.
V2 retrieves the value from the slider widget to set the motor speed and stores it in the variable *Speed*.

```
void loop() {
  Blynk.run();// Run the blynk function
  smartcar();// Call the main function
}
```

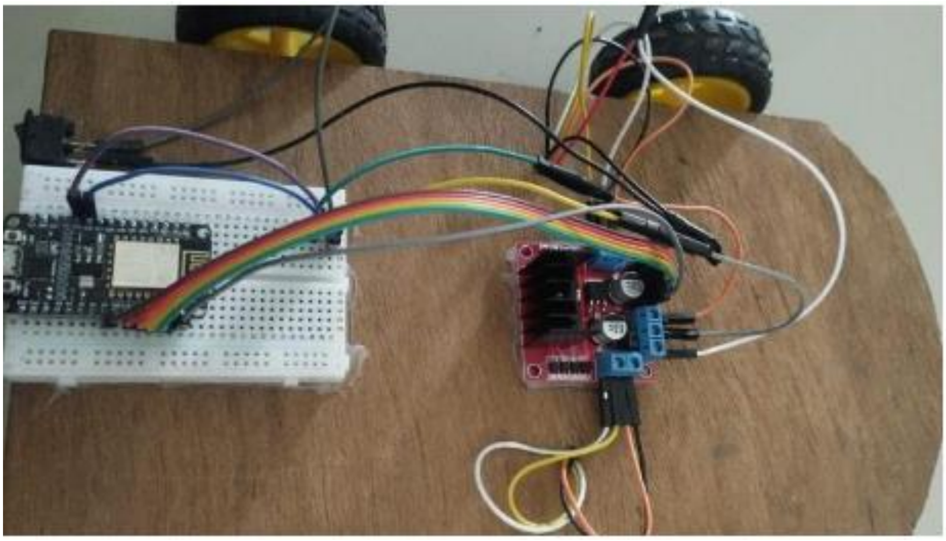
(*Source*: *Personal Documentation*, 2025)

Figure 8. Loop Function for Running Blynk Connection and Car Control

Figure 8 explains two main function lines:
Blynk.run() is maintains the active connection with the Blynk platform and handles incoming events from the application.
Smartcar() is calls the main function to read joystick input and move the car according to the input conditions.

Hardware Testing

After all the circuits were integrated into the MINIOT IoT-based mini smart car using ESP8266 and L298, the next step was the assembly of all the completed circuits. Below is an image of the finished IoT-based mini smart car built with ESP8266 and L298, as shown in Figure 9.



(*Source*: *Personal Documentation*, 2025)

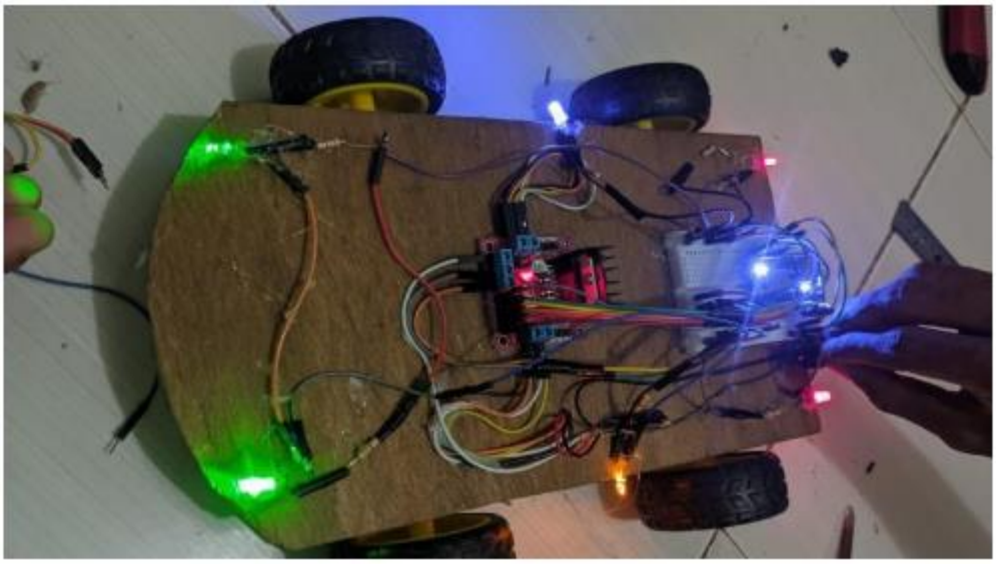
Figure 9. MINIOT System Design Results

Programmer Downloader Testing

Testing of the downloader system was carried out by transferring program data from the computer to the NodeMCU ESP8266 microcontroller. The downloader was first connected to the laptop via the USB port. The program, which had been written using the Arduino software, was then compiled and uploaded to the microcontroller. If the downloading process runs without errors, it indicates that both the downloader and the microcontroller are functioning properly.

Hardware Testing Results

After the hardware has been programmed into the microcontroller and executed using the downloader, the program will be automatically integrated into the microcontroller. The result of this process can be seen in Figure 10.



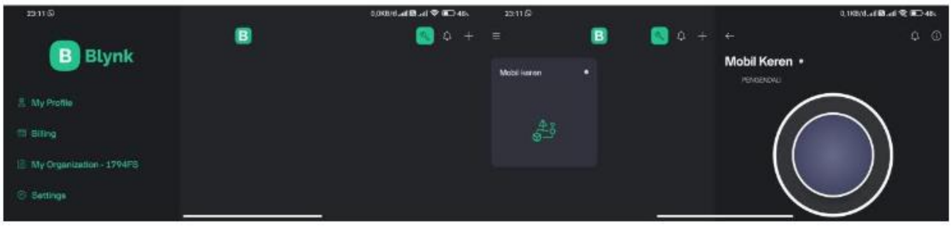
(*Source: Personal Documentation, 2025*)

Figure 10. Design Hardware MINIOT

From Figure 10 regarding the MINIOT hardware design, it can be explained that the first stage is controlling the car using a mobile phone with the Blynk Android application. The second stage is moving the car forward, backward, turning right, and turning left using the joystick feature configured in Blynk, and the final stage is turning the car on and off, which is done using a switch located at the back of the car.

Software Testing Results

To obtain the results of testing the MINIOT: Mini Smart Car Design based on IoT, the Blynk application was used as a medium to monitor control remotely. The results of this testing can be seen in Figure 11.



(*Source*: *Personal Documentation*, 2025)

Figure 11. Car Testing Results Using Blynk

4. Conclusion

This research successfully designed and implemented a prototype of a mini smart car based on the Internet of Things (IoT), named MINIOT. The system uses the ESP8266 (Amica) microcontroller as the control center, handling Wi-Fi connectivity and command processing, and the L298N module as the DC motor driver for car maneuvers. The prototype was successfully controlled remotely through the Blynk application, featuring full movement capabilities (forward, backward, turn right, turn left) and good responsiveness to user input. The test results showed that the system operates stably, is efficient in terms of power consumption, and is easy to operate. This platform has high potential as an educational tool in the STEM field, as well as being suitable for experimentation and technology hobbies.

References

Adi Winarno, & Affandi, M. (2022). *Design and construction of smart house prototype based Internet of Things (IoT) using ESP8266. BEST: Journal of Applied Electrical, Science, & Technology, 4*(1), 11–14. https://doi.org/10.36456/best.vol4.no1.5447

Chaudry, A. M. (2020). *Using Arduino Uno microcontroller to create interest in physics. The Physics Teacher, 58*(6), 418–421. https://doi.org/10.1119/10.0001841

Duhi, S. (2025). *Sistem monitoring lokasi dan data gangguan satu fase menggunakan ESP8266 via Blynk dan Telegram [Location and single-phase fault data monitoring system using ESP8266 via Blynk and Telegram]*. [Unpublished manuscript].

Fransiska, R. W., Septia, E. M. P., Vessabhu, W. K., Frans, W., Abednego, W., & Hendro. (2013). *Electrical power measurement using Arduino Uno microcontroller and LabVIEW*. In *Proceedings of the 2013 3rd International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering: Science and Technology for Improvement of Health, Safety, and Environment (ICICI-BME 2013)* (pp. 226–229). IEEE. https://doi.org/10.1109/ICICI-BME.2013.6698497

Gao, C., Luo, L., Zhang, Y., Pearson, B., & Fu, X. (2019). *Microcontroller-based IoT system firmware security: Case studies*. In *Proceedings of the IEEE International Conference on Industrial Internet Cloud (ICII 2019)* (pp. 200–209). IEEE. https://doi.org/10.1109/ICII.2019.00045

Juang, H. S., & Lurrr, K. Y. (2013). *Design and control of a two-wheel self-balancing robot using the Arduino microcontroller board*. In *Proceedings of the IEEE International Conference on Control and Automation (ICCA)* (pp. 634–639). IEEE. https://doi.org/10.1109/ICCA.2013.6565146

Kabilan, M., Manikandan, V., & Dharshini, S. K. (2025). *Automatic solar tracking system: A comprehensive overview of an advanced automatic solar tracking system using ESP8266. Asian Journal of Applied Science and Technology, 9*(1), 11–20. https://doi.org/10.38177/ajast.2025.9102

Martínez-Santos, J. C., Acevedo-Patino, O., & Contreras-Ortiz, S. H. (2017). *Influence of Arduino on the development of advanced microcontrollers courses. Revista Iberoamericana de Tecnologías del Aprendizaje, 12*(4), 208–217. https://doi.org/10.1109/RITA.2017.2776444

Nuratch, S. (2017). *A universal microcontroller circuit and firmware design and implementation for IoT-based real-time measurement and control applications.* In *Proceedings of the 2017 International Electrical Engineering Congress (IEECON 2017).* IEEE. https://doi.org/10.1109/IEECON.2017.8075906

Rajunaik, M., & Amareswer, E. (2023). *Irrigation system using IoT. AIP Conference Proceedings, 2492,* 2–6. https://doi.org/10.1063/5.0119809

Revansa Arya Pradhana, D., Ichsan Pradana, A., & Hartanti, D. (2024). *Implementasi Internet of Things pada sistem kontrol RC-car pemantau area berbasis ESP32-CAM. Jurnal Teknologi, 11*(2), 149–159. https://doi.org/10.31479/jtek.v11i2.306

Sari, I. P., Novita, A., Al-Khowarizmi, A.-K., Ramadhani, F., & Satria, A. (2024). *Pemanfaatan Internet of Things (IoT) pada bidang pertanian menggunakan Arduino Uno R3. Blend Sains: Jurnal Teknik, 2*(4), 337–343. https://doi.org/10.56211/blendsains.v2i4.505

Swathi, K., Sandeep, T. U., & Ramani, A. R. (2018). *Performance analysis of microcontrollers used in IoT technology. International Journal of Scientific Research in Science, Engineering and Technology, 4*(4), 1268–1273.

Wadly, F., Ramadhan, Z., & Rosan, K. (2025). *Integrating the NodeMCU ESP8266 microcontroller with the DHT11 and MQ-135 sensors enables IoT-based air quality monitoring. International Conference on Artificial Intelligence, Navigation, Engineering, and Aviation Technology, 2*(1), 272–277.

Zinkevich, A. V. (2021). *ESP8266 microcontroller application in wireless synchronization tasks.* In *Proceedings of the 2021 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM 2021)* (pp. 670–674). IEEE. https://doi.org/10.1109/ICIEAM51226.2021.9446411