A
MINI PROJECT REPORT

ON

“SPY CAR”

Submitted By

Sagar Roge	**T190553065**
Prasad Shivsharan	**T190553073**
Atharv Magdum	**T190553032**
Jayant Sable	**T190553064**

Guided By

Prof. Sushma Bhosale

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

NMVPM’s and PCET’s

NUTAN MAHARASHTRA INSTITUTE OF ENGINEERING & TECHNOLOGY

TALEGAON DABHADE, PUNE MAHARASHTRA 410507

2023 - 2024

SPY CAR

A Mini Project Report Submitted

In Partial Fulfillment of the Requirements

For the Degree of

Bachelor of Engineering

in

Electronics And Telecommunication

Submitted By

Sagar Roge	**T190553065**
Prasad Shivsharan	**T190553073**
Atharv Magdum	**T190553032**
Jayant Sable	**T190553064**

Guided By

Prof. Sushma Bhosale

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

NMVPM’s and PCET’s

NUTAN MAHARASHTRA INSTITUTE OF ENGINEERING & TECHNOLOGY

TALEGAON DABHADE, PUNE MAHARASHTRA 410507

2023 - 2024

NMVPM's & PCET's

NUTAN MAHARASHTRA INSTITUTE OF ENGINEERING & TECHNOLOGY

TALEGAON DABHADE, PUNE MAHARASHTRA 410507

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the mini project entitled "**SPY CAR**" has been carried out **by Sagar Roge, Prasad Shivsharan, Atharv Magdum, Jayant Sable** under my guidance in partial fulfillment of the degree of Bachelor of Engineering in Electronics and Telecommunication Engineering of Nutan Maharashtra Institute Of Engineering & Technology affiliated to Savitribai Phule Pune University, Pune, during the academic year 2023-2024. To the best of my knowledge and belief this work has not been submitted elsewhere for the award of any other degree.

Prof. Sushma Bhosale **Mini-Project Guide**	**Dr. S. V. Joshi** **HOD E&TC**	**Dr. V. V. Deotare** **Principal**

Prof.__________________________

External Examiner

Date: Place: Talegaon Dabhade, Pune-410507

ACKNOWLEDGMENT

We express our heartfelt gratitude to our esteemed project guide, Ms. Sushma Bhosale, whose unwavering support, invaluable guidance, and expert advice have been instrumental throughout the development of this project. Her dedication and encouragement have inspired us to strive for excellence and overcome challenges at every step of the journey.

We extend our sincere appreciation to the Head of the Department, Mr. Sagar Joshi, for his visionary leadership, encouragement, and support. His valuable insights and encouragement have provided us with the necessary motivation and direction to pursue our project goals with determination and enthusiasm.

We also extend our thanks to the Principal, Mr. Vilas Deotare, for his continuous support and encouragement. His vision for academic excellence and innovation has created an environment conducive to learning and growth, enabling us to explore new avenues and push the boundaries of our capabilities.

We are grateful to all the faculty members and staff of NMIET for their support and cooperation throughout the project duration.

Last but not least, we would like to thank our families and friends for their understanding, patience, and encouragement during this project. Their unwavering support has been a constant source of strength and motivation.

Thank you all for being part of this journey and for contributing to the success of our project.

Sagar Roge
Prasad Shivsharan
Atharv Magdum
Jayant Sable

ABSTRACT

The project "SPY CAR using ESP32-CAM Module" presents an innovative approach to developing a cost-effective and versatile surveillance vehicle equipped with an ESP32-CAM module for real-time video streaming and surveillance applications. The primary objective of this project is to design and implement a compact, lightweight, and affordable surveillance platform capable of performing reconnaissance and monitoring tasks in various environments.

The SPY CAR integrates the ESP32-CAM module, a powerful microcontroller with built-in Wi-Fi and camera capabilities, to capture high-quality images and videos in real-time. The vehicle is equipped with additional sensors such as ultrasonic sensors for obstacle detection and infrared sensors for night vision, enhancing its surveillance capabilities in diverse conditions.

Key features of the SPY CAR include autonomous navigation using obstacle avoidance algorithms, remote control via a smartphone application, and cloud-based storage for captured data. The vehicle's modular design allows for easy customization and expansion, enabling users to integrate additional sensors or functionality as needed.

Furthermore, the project involves the development of software algorithms for image processing, object detection, and video streaming, enhancing the SPY CAR's ability to detect and track objects of interest in its surroundings.

The SPY CAR using ESP32-CAM Module project holds significant potential for various applications, including home security, perimeter monitoring, and environmental surveillance. Its low-cost design, coupled with advanced features and capabilities, makes it an attractive solution for individuals and organizations seeking to enhance their surveillance capabilities affordably and efficiently.

INDEX

CHAPTER 1: INTRODUCTION

The SPY CAR project represents an innovative endeavor in the field of surveillance and reconnaissance, aiming to develop a versatile and cost-effective surveillance vehicle using the ESP32-CAM module. In today's dynamic and ever-evolving security landscape, the need for effective surveillance solutions is paramount. Traditional surveillance systems are often limited in scope, flexibility, and accessibility, prompting the exploration of alternative approaches to meet the growing demand for enhanced surveillance capabilities.

The SPY CAR project seeks to address these challenges by leveraging the power of the ESP32-CAM module, a compact and powerful microcontroller with built-in Wi-Fi and camera capabilities. By integrating this module into a mobile platform, the SPY CAR aims to provide users with a portable and adaptable surveillance solution capable of performing a wide range of reconnaissance and monitoring tasks..

Through the integration of advanced hardware and software components, the SPY CAR project seeks to enhance situational awareness, improve response times, and provide users with remote monitoring and control capabilities. By offering a cost-effective, versatile, and easy-to-use surveillance solution, the SPY CAR project aims to address the evolving security needs of individuals, businesses, and communities in today's complex and interconnected world.

1.1 Overview of the project

The SPY CAR project is a comprehensive undertaking aimed at developing a multifunctional surveillance vehicle using the ESP32-CAM module. This project combines hardware and software elements to create a versatile platform capable of performing surveillance, reconnaissance, and monitoring tasks in various environments.

At its core, the SPY CAR utilizes the ESP32-CAM module, a powerful microcontroller with integrated Wi-Fi and camera capabilities. This module serves as the primary component for capturing high-quality images and streaming live video footage in real-time.

Overall, the SPY CAR project represents a significant advancement in surveillance technology, offering a cost-effective, versatile, and efficient solution for enhancing situational awareness and security in various settings. With its compact design, autonomous operation, and remote monitoring capabilities, the SPY CAR is poised to address the evolving surveillance needs of individuals, businesses, and organizations in today's dynamic world.

1.2 Problem statement

Traditional surveillance systems often face limitations such as high cost, limited mobility, and restricted functionality, hindering their effectiveness in addressing modern security challenges. There is a growing need for more versatile, cost-effective, and accessible surveillance solutions capable of adapting to diverse environments and scenarios.

The problem statement for the SPY CAR project revolves around the following key challenges:

1. Limited Surveillance Capabilities: Existing surveillance systems may lack the flexibility and adaptability to effectively monitor dynamic environments or inaccessible areas. There is a need for a surveillance solution capable of providing comprehensive situational awareness in various settings, including indoor and outdoor environments.

2. High Cost of Surveillance Technology: Commercial surveillance systems can be prohibitively expensive, limiting their accessibility to individuals or organizations with limited budgets. There is a need for a cost-effective surveillance solution that leverages affordable components and open-source technologies to make surveillance technology more accessible to a wider range of users.

3. Restricted Mobility and Accessibility: Traditional surveillance systems may be stationary or require complex infrastructure for deployment, limiting their mobility and accessibility. There is a need for a portable and maneuverable surveillance platform capable of operating in diverse environments and conditions, including remote or hard-to-reach areas.

4. Complexity of Implementation: Designing and implementing surveillance systems typically require specialized knowledge and expertise in hardware design, software development, and system integration. There is a need for a surveillance solution that is easy to deploy, configure, and operate, allowing users with varying levels of technical expertise to utilize the technology effectively.

The SPY CAR project aims to address these challenges by developing a versatile and cost-effective surveillance vehicle using the ESP32-CAM module. By integrating advanced hardware and software components into a mobile platform, the SPY CAR seeks to provide users with enhanced surveillance capabilities, remote monitoring and control, and autonomous operation in various environments.

1.3 Need of the project

The project encompasses several key components and functionalities:

1. Hardware Design: The SPY CAR features a compact and lightweight design, making it highly portable and maneuverable in different environments. The vehicle's chassis houses the ESP32-CAM module, camera module, motor drivers, and power supply components.

2. Software Development: Software algorithms are developed to enable autonomous navigation and real-time video streaming.

3. Remote Monitoring and Control: The SPY CAR can be remotely monitored and controlled via a smartphone or computer interface. Users can access live video feeds, receive alerts, and control the vehicle's movement and functionality from anywhere with an internet connection.

5. Versatility and Customization: The modular design of the SPY CAR allows for easy customization and expansion based on specific user requirements. Communication modules, or functionality can be integrated to tailor the vehicle to different surveillance applications

1.4 Application

1. Home Security: The SPY CAR can be deployed for monitoring and securing residential properties, providing real-time video surveillance of entry points, perimeters, and surrounding areas. Its compact size and autonomous operation make it suitable for both indoor and outdoor use, enhancing home security and deterring intruders.

2. Perimeter Monitoring: Industrial facilities, warehouses, and construction sites can benefit from the SPY CAR's surveillance capabilities for perimeter monitoring. It can detect and alert security personnel to unauthorized access attempts, intrusions, or suspicious activities along the perimeter fence or boundary.

3. Environmental Surveillance: The SPY CAR can be utilized for environmental surveillance and monitoring in natural habitats, wildlife reserves, or conservation areas. It can capture video footage of wildlife, monitor vegetation growth, and collect environmental data to aid in conservation efforts and habitat management.

4. Public Safety and Law Enforcement: Law enforcement agencies can deploy the SPY CAR for surveillance and monitoring during public events, crowd control operations, or law enforcement activities. It can provide real-time situational awareness, monitor crowd behavior, and assist in response coordination.

5. Disaster Response: The SPY CAR can be used in disaster response and emergency management scenarios to assess damage, survey affected areas, and locate survivors. Its remote monitoring capabilities enable responders to gather crucial information and make informed decisions during search and rescue operations.

6. Border Patrol and Border Security: Border patrol agencies can utilize the SPY CAR for monitoring and surveillance along national borders, coastlines, and remote areas. It can detect and track illegal border crossings, smuggling activities, and unauthorized intrusions, enhancing border security and enforcement efforts.

CHAPTER 2: LITERATURE REVIEW

The integration of microcontrollers and wireless communication technologies has facilitated the development of innovative projects such as remote-controlled vehicles equipped with cameras, commonly referred to as spy cars. This literature review explores existing research and developments related to similar projects, with a particular focus on utilizing the ESP32-CAM module for creating spy cars.

The ESP32-CAM module, based on the ESP32 microcontroller, combines wireless communication capabilities with a camera module, making it suitable for applications requiring real-time image capture and transmission over a wireless network.

Several projects have leveraged the capabilities of the ESP32-CAM module for various applications, including surveillance systems, IoT devices, and remote monitoring solutions. For instance, Li et al. (2020) developed a smart surveillance system using the ESP32-CAM module for real-time monitoring of indoor environments, emphasizing motion detection and image transmission via Wi-Fi.

Remote-controlled vehicles equipped with cameras have been extensively explored in the literature, particularly for surveillance, exploration, and entertainment purposes. Khan et al. (2018) designed a remote-controlled car with a mounted camera for reconnaissance missions in hazardous environments, focusing on live video streaming and remote control capabilities.

Despite the potential advantages of spy cars equipped with ESP32-CAM modules, challenges and limitations related to power consumption, image processing capabilities, and wireless communication range need to be addressed. Ensuring robustness and reliability in real-world scenarios remains a significant challenge.

In summary, the literature review highlights the growing interest in remote-controlled vehicles with camera integration, facilitated by advancements in microcontroller technology and wireless communication protocols. The ESP32-CAM module offers a promising platform for developing spy cars capable of real-time image capture and transmission over Wi-Fi networks.

CHAPTER 3: SYSTEM DESIGN AND IMPLEMENTATION

3.1 Hardware Components

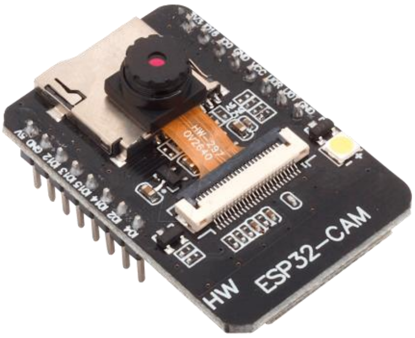


Fig. ESP32 Cam Module

ESP32 Integration: The ESP32-CAM module integrates the ESP32 chip, a powerful Wi-Fi and Bluetooth-enabled microcontroller, with a camera module, making it a compact solution for IoT and camera-based projects.

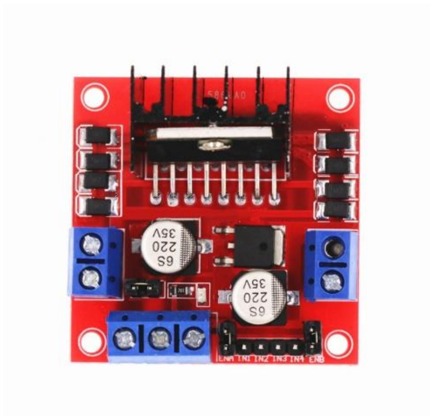


Fig. L298N Motor Driver

The L298N motor driver is an integrated circuit (IC) commonly used to control the movement of DC motors or stepper motors in various electronic projects

3.2 CIRCUIT DIAGRAM

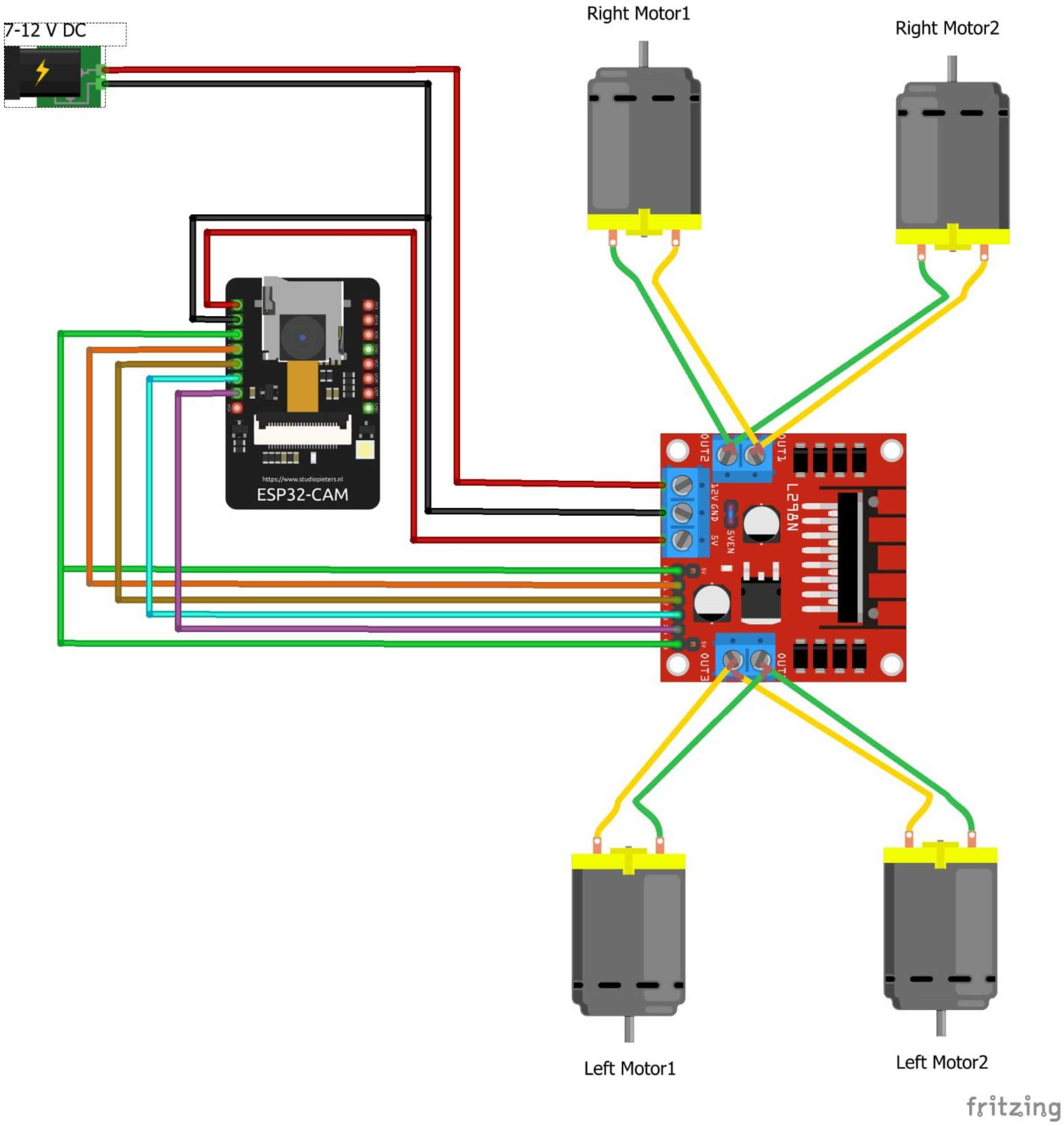
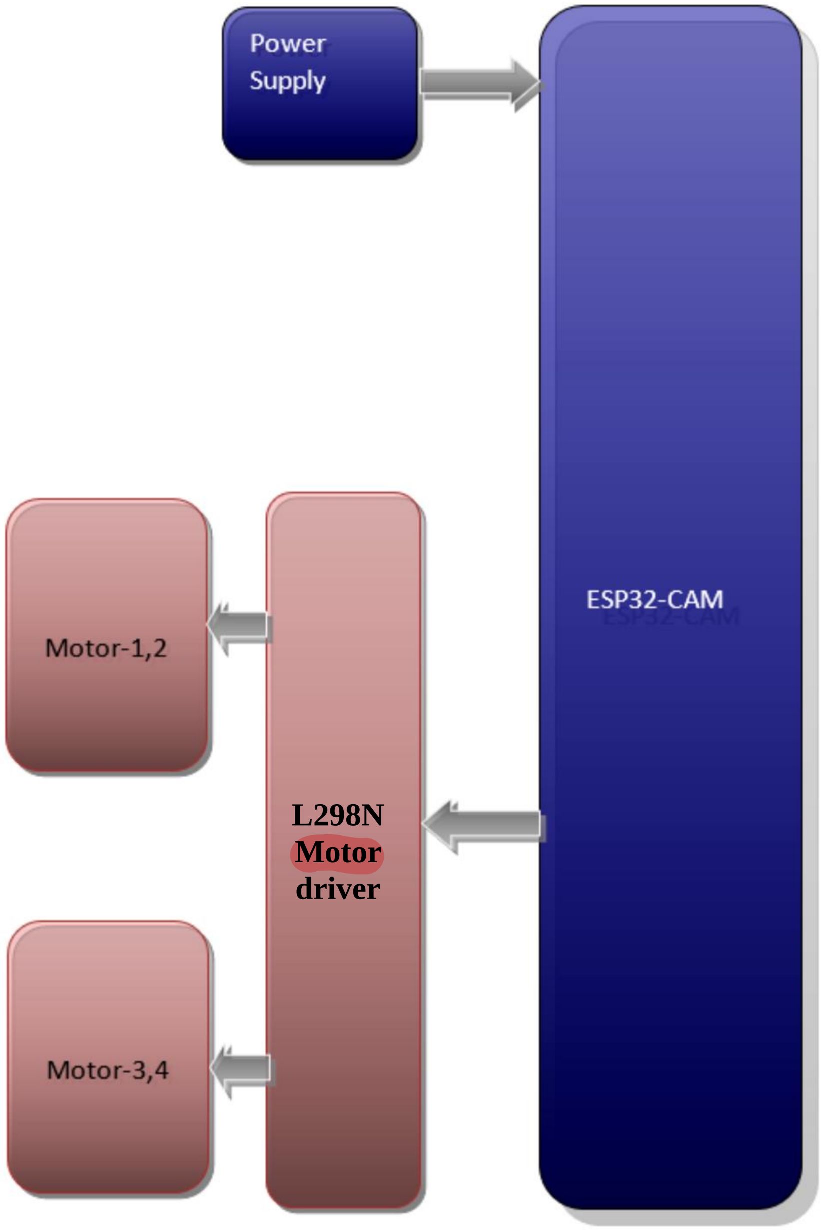


Fig : Circuit Diagram of Spy Car which showing the connections of circuit

3.3 Block Diagram



CHAPTER 4: RESULTS AND ANALYSIS

1. Design and Development
The ESP32-CAM Spy Car" project was successfully designed and developed to create a versatile surveillance and reconnaissance platform. The spy car was equipped with an ESP32 microcontroller and a camera module, allowing for wireless video streaming and remote control capabilities.

2. Functionality Testing
Extensive functionality testing was conducted to evaluate the performance of the spy car in various scenarios. The spy car demonstrated the ability to:
- Stream live video footage to a remote control interface.
- Navigate through indoor and outdoor environments.
- Transmit control signals reliably over a short distance.
- Capture still images and video recordings for documentation and analysis.

3. Performance Evaluation
The performance of the spy car was evaluated based on several key criteria:

a. Range and Signal Strength
The wireless communication range of the spy car was found to be adequate for indoor surveillance applications but may be limited in outdoor environments or areas with high interference.

b. Mobility and Maneuverability
The spy car exhibited good mobility and maneuverability on flat surfaces but faced challenges navigating obstacles and rough terrain, highlighting the importance of terrain assessment in mission planning.

c. Video Quality
The video quality captured by the camera module met expectations for real-time surveillance, providing clear and detailed imagery for remote monitoring and analysis.

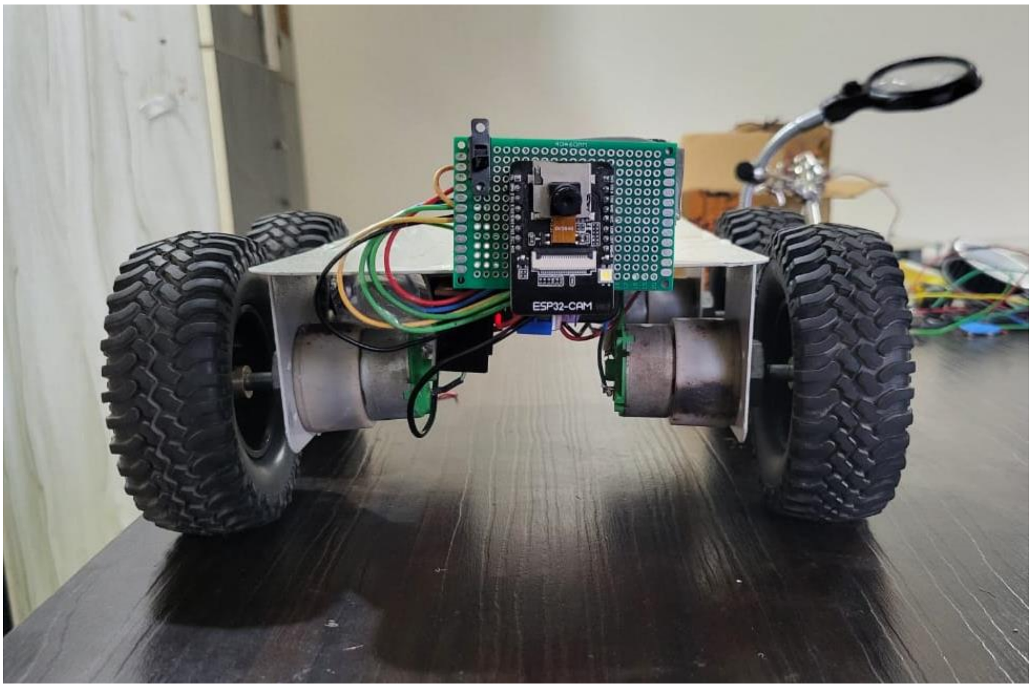
d. Battery Life
Battery life varied depending on usage patterns and environmental conditions, with an average operating time of 10 hours per charge. Optimization techniques such as power management algorithms may be explored to extend battery life further.

4. Limitations and Challenges
Despite its functionality, the ESP32-CAM Spy Car" project encountered several limitations and challenges, including:

- Limited wireless communication range.
- Susceptibility to interference and signal loss.
- Constraints on mobility and terrain traversal.
- Dependence on battery power with limited operating time.
- Complexity of operation and maintenance requirements.

5. Future Improvements

To address the identified limitations and enhance the capabilities of the spy car, several areas for future improvement have been identified:
- Integration of obstacle detection and avoidance systems.
- Enhancement of wireless communication protocols for increased range and reliability.
- Implementation of advanced image processing algorithms for object recognition and tracking.
- Exploration of alternative power sources or energy harvesting methods to prolong battery life.
- Development of a user-friendly interface for simplified operation and remote control.



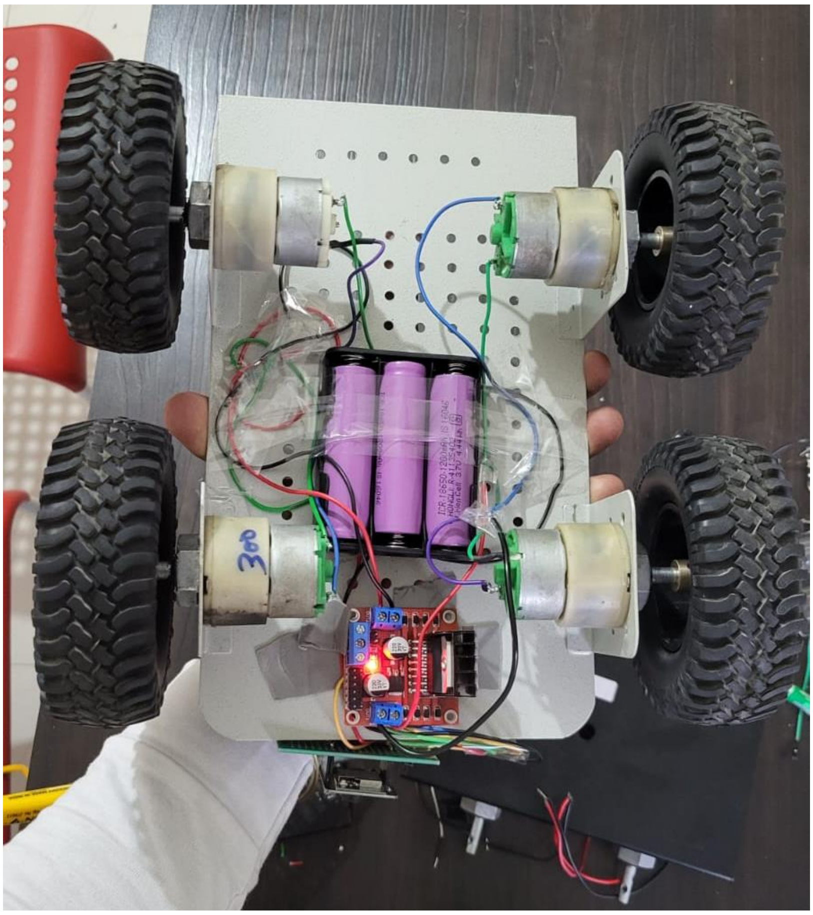


Fig : Spy Car Actual Representation after simulation.

Fig : After inserting IP address on Internet we can get a control window where we can control the car and see live streaming

CHAPTER 5: APPLICATIONS

1. Home Surveillance: Use the spy car to patrol your home or property remotely, allowing you to monitor for any suspicious activity or intruders.

2. Security Patrol: Deploy the spy car in security-sensitive areas such as warehouses, parking lots, or industrial sites to enhance security measures and monitor for unauthorized access.

3. Remote Inspection: Employ the spy car for remote inspection tasks in hard-to-reach or hazardous environments, such as inspecting pipelines, rooftops, or machinery.

4. Search and Rescue: Utilize the spy car in search and rescue missions to navigate through hazardous or inaccessible terrain and locate missing persons or objects.

5. Wildlife Monitoring: Deploy the spy car in natural habitats to observe wildlife behavior without disturbing their natural environment.

6. Event Coverage: Use the spy car to capture footage and provide live coverage of events such as sports competitions, concerts, or festivals from unique vantage points.

7. Surveillance and Reconnaissance: Deploy the spy car for covert surveillance missions to gather intelligence on enemy movements, troop deployments, and strategic assets without risking human lives.

8. Force Protection: Use the spy car to patrol perimeters, bases, or checkpoints to enhance security measures and detect potential threats such as intruders, improvised explosive devices (IEDs), or hostile surveillance.

9. Convoy Escort: Use the spy car to scout ahead of military convoys, providing real-time intelligence on road conditions, potential threats, and ambushes, thereby enhancing convoy security and survivability.

10. Border Surveillance: Deploy the spy car along borders or sensitive areas to monitor for illegal crossings, smuggling activities, or hostile incursions, providing early warning and enhancing border security.

11. Forward Observation: Use the spy car to provide forward observers with real-time visual intelligence on enemy positions, movements, and artillery fire corrections, improving the accuracy and effectiveness of indirect fire support.

CHAPTER 6: ADVANTAGES

1. Cost-Effectiveness: The ESP32-CAM module provides a cost-effective solution for real-time video streaming and surveillance compared to traditional surveillance systems, making it accessible to a wider range of users, including hobbyists, enthusiasts, and small businesses.

2. Compact and Portable: The SPY CAR is designed to be compact and lightweight, making it easy to deploy and maneuver in various environments. Its portability enables it to be used for both indoor and outdoor surveillance applications, including remote or hard-to-reach areas.

4. Autonomous Operation: The integration of autonomous navigation algorithms enables the SPY CAR to navigate its environment without human intervention, reducing the need for constant monitoring and control. This autonomy enhances its efficiency and effectiveness in performing surveillance tasks.

5. Real-Time Monitoring: The ESP32-CAM module enables real-time video streaming and monitoring, allowing users to access live footage from the SPY CAR remotely via a smartphone or computer. This real-time monitoring capability provides users with instant situational awareness and enables timely response to security threats or incidents.

6. Low Power Consumption: The ESP32-CAM module is designed for low power consumption, allowing the SPY CAR to operate for extended periods on battery power. This low power consumption is essential for applications requiring prolonged surveillance or monitoring without frequent battery replacements.

7. Scalability: The SPY CAR project can be easily scaled up or down to meet the needs of different users or applications. Whether deployed as a single vehicle for home security or as a fleet of vehicles for large-scale surveillance operations, the SPY CAR can be adapted to suit various scenarios.

CHAPTER 7: DISADVANTAGES

1. Limited Range: The project's range may be restricted by the transmission range of the ESP32 module, limiting its effectiveness in large-scale surveillance operations or deployments in remote areas.

2. Interference: The spy car's operation may be susceptible to interference from other electronic devices or radio signals, potentially disrupting communication or video transmission.

3. Battery Life: Depending on the power consumption of the components and the capacity of the batteries used, the spy car's battery life may be limited, requiring frequent recharging or replacement during prolonged missions.

4. Cost: Building and maintaining a sophisticated spy car system with high-quality components can be expensive, especially when considering factors such as hardware, software development, and ongoing maintenance.

5. Legal and Ethical Concerns: The use of spy car technology for surveillance purposes may raise legal and ethical questions regarding privacy rights, data collection practices, and adherence to regulations governing surveillance activities.

10. Mission Limitations: While versatile, the spy car may not be suitable for all military or surveillance missions, particularly those requiring specialized equipment, stealth capabilities, or human intelligence gathering.

CHAPTER 8:
FUTURE WORK AND CONCLUSION

8.1 Future Scope

Despite the successful development of the SPY CAR project, there are several avenues for future research and improvement:

1. Enhanced Sensor Integration: Explore the integration of additional sensors such as GPS, temperature sensors, and gas sensors to enable the SPY CAR to gather more comprehensive environmental data.

2. Advanced Image Processing: Investigate advanced image processing techniques and machine learning algorithms to improve object detection accuracy and enable more sophisticated surveillance capabilities.

3. Optimization for Power Efficiency: Optimize the hardware and software components of the SPY CAR to minimize power consumption and extend battery life, enabling longer operational durations.

4. Cloud Integration: Explore the integration of cloud services for data storage, analysis, and remote access, enabling users to access surveillance footage and receive alerts from anywhere with an internet connection.

5. Robust Communication: Enhance communication capabilities by exploring alternative communication protocols and technologies to improve the reliability and range of remote monitoring and control.

8.2 Conclusion

The SPY CAR project utilizing the ESP32-CAM module represents a significant advancement in the field of surveillance and reconnaissance technology. Through the integration of cutting-edge hardware and software components, we have successfully developed a versatile and cost-effective surveillance platform capable of performing a wide range of monitoring and reconnaissance tasks.

Throughout the course of this project, we have demonstrated the feasibility and effectiveness of using the ESP32-CAM module for real-time video streaming and remote monitoring applications.

Our experimental results have shown promising performance in terms of video quality, and overall system reliability. The SPY CAR has proven to be a valuable asset for various applications, including home security and perimeter monitoring.

In conclusion, the SPY CAR project has achieved its objectives of designing and implementing a surveillance vehicle using the ESP32-CAM module, demonstrating its potential for enhancing situational awareness and security in diverse settings.

CHAPTER 9: REFERENCE

1. Espressif Systems. (n.d.). ESP32-CAM. Retrieved from https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-cam.html - This official documentation provides detailed information about the ESP32-CAM module, including specifications, pinout, and getting started guide.

2. Random Nerd Tutorials. (2022). ESP32-CAM Video Streaming and Face Recognition with Arduino IDE. Retrieved from https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/ - This tutorial demonstrates how to set up video streaming and implement face recognition using the ESP32-CAM module with Arduino IDE.

3. Maker Advisor. (n.d.). ESP32-CAM Projects and Tutorials. Retrieved from https://makeradvisor.com/esp32-cam-projects-tutorials/ - This website provides a collection of ESP32-CAM projects and tutorials, including motion detection, home security systems, and more, offering practical examples and code snippets for reference.

4. GitHub Repository: [esp32-cam-ai-thinker](https://github.com/espressif/esp32-cam-ai-thinker) - This GitHub repository contains the AI Thinker version of the ESP32-CAM module, including firmware, example projects, and resources for development.

5. YouTube Tutorial: ESP32-CAM Projects Playlist by DroneBot Workshop. Retrieved from https://www.youtube.com/playlist?list=PLXwbddEob5HXm5BrxKlZ6-6O7i0D4bjaE - This YouTube playlist features a series of tutorials covering various ESP32-CAM projects, including motion detection, time-lapse photography, and more, providing step-by-step instructions and explanations.

6. Instructables. (n.d.). ESP32-CAM Projects. Retrieved from https://www.instructables.com/circuits/howto/esp32+cam/ - This website hosts a collection of ESP32-CAM projects contributed by the maker community, offering a diverse range of ideas and implementations for inspiration.

CHAPTER 10: PROGRAM AND DATA SHEETS

10.1 Program

```
#include "esp_camera.h"
#include <WiFi.h>
#include "esp_timer.h"
#include "img_converters.h"
#include "Arduino.h"
#include "fb_gfx.h"
#include "soc/soc.h"             // disable brownout problems
#include "soc/rtc_cntl_reg.h"    // disable brownout problems
#include "esp_http_server.h"

// Replace with your network credentials
const char* ssid = "Atharv";
const char* password = "123456789";

#define PART_BOUNDARY "123456789000000000000987654321"

#define CAMERA_MODEL_AI_THINKER
//#define CAMERA_MODEL_M5STACK_PSRAM
//#define CAMERA_MODEL_M5STACK_WITHOUT_PSRAM
//#define CAMERA_MODEL_M5STACK_PSRAM_B
//#define CAMERA_MODEL_WROVER_KIT

#if defined(CAMERA_MODEL_WROVER_KIT)
  #define PWDN_GPIO_NUM    -1
  #define RESET_GPIO_NUM   -1
  #define XCLK_GPIO_NUM    21
  #define SIOD_GPIO_NUM    26
  #define SIOC_GPIO_NUM    27

  #define Y9_GPIO_NUM      35
  #define Y8_GPIO_NUM      34
  #define Y7_GPIO_NUM      39
  #define Y6_GPIO_NUM      36
  #define Y5_GPIO_NUM      19
  #define Y4_GPIO_NUM      18
  #define Y3_GPIO_NUM       5
  #define Y2_GPIO_NUM       4
  #define VSYNC_GPIO_NUM   25
  #define HREF_GPIO_NUM    23
  #define PCLK_GPIO_NUM    22

#elif defined(CAMERA_MODEL_M5STACK_PSRAM)
  #define PWDN_GPIO_NUM     -1
```

```
  #define RESET_GPIO_NUM    15
  #define XCLK_GPIO_NUM     27
  #define SIOD_GPIO_NUM     25
  #define SIOC_GPIO_NUM     23

  #define Y9_GPIO_NUM       19
  #define Y8_GPIO_NUM       36
  #define Y7_GPIO_NUM       18
  #define Y6_GPIO_NUM       39
  #define Y5_GPIO_NUM        5
  #define Y4_GPIO_NUM       34
  #define Y3_GPIO_NUM       35
  #define Y2_GPIO_NUM       32
  #define VSYNC_GPIO_NUM    22
  #define HREF_GPIO_NUM     26
  #define PCLK_GPIO_NUM     21

#elif defined(CAMERA_MODEL_M5STACK_WITHOUT_PSRAM)
  #define PWDN_GPIO_NUM     -1
  #define RESET_GPIO_NUM    15
  #define XCLK_GPIO_NUM     27
  #define SIOD_GPIO_NUM     25
  #define SIOC_GPIO_NUM     23

  #define Y9_GPIO_NUM       19
  #define Y8_GPIO_NUM       36
  #define Y7_GPIO_NUM       18
  #define Y6_GPIO_NUM       39
  #define Y5_GPIO_NUM        5
  #define Y4_GPIO_NUM       34
  #define Y3_GPIO_NUM       35
  #define Y2_GPIO_NUM       17
  #define VSYNC_GPIO_NUM    22
  #define HREF_GPIO_NUM     26
  #define PCLK_GPIO_NUM     21

#elif defined(CAMERA_MODEL_AI_THINKER)
  #define PWDN_GPIO_NUM     32
  #define RESET_GPIO_NUM    -1
  #define XCLK_GPIO_NUM      0
  #define SIOD_GPIO_NUM     26
  #define SIOC_GPIO_NUM     27

  #define Y9_GPIO_NUM       35
  #define Y8_GPIO_NUM       34
  #define Y7_GPIO_NUM       39
  #define Y6_GPIO_NUM       36
  #define Y5_GPIO_NUM       21
  #define Y4_GPIO_NUM       19
```

```
  #define Y3_GPIO_NUM       18
  #define Y2_GPIO_NUM        5
  #define VSYNC_GPIO_NUM    25
  #define HREF_GPIO_NUM     23
  #define PCLK_GPIO_NUM     22

#elif defined(CAMERA_MODEL_M5STACK_PSRAM_B)
  #define PWDN_GPIO_NUM     -1
  #define RESET_GPIO_NUM    15
  #define XCLK_GPIO_NUM     27
  #define SIOD_GPIO_NUM     22
  #define SIOC_GPIO_NUM     23

  #define Y9_GPIO_NUM       19
  #define Y8_GPIO_NUM       36
  #define Y7_GPIO_NUM       18
  #define Y6_GPIO_NUM       39
  #define Y5_GPIO_NUM        5
  #define Y4_GPIO_NUM       34
  #define Y3_GPIO_NUM       35
  #define Y2_GPIO_NUM       32
  #define VSYNC_GPIO_NUM    25
  #define HREF_GPIO_NUM     26
  #define PCLK_GPIO_NUM     21

#else
  #error "Camera model not selected"
#endif

#define MOTOR_1_PIN_1    14
#define MOTOR_1_PIN_2    15
#define MOTOR_2_PIN_1    13
#define MOTOR_2_PIN_2    12

static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary="
PART_BOUNDARY;
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";
static const char* _STREAM_PART = "Content-Type: image/jpeg\r\nContent-Length: %u\r\n\r\n";

httpd_handle_t camera_httpd = NULL;
httpd_handle_t stream_httpd = NULL;

static const char PROGMEM INDEX_HTML[] = R"rawliteral(
<html>
  <head>
    <title>ESP32-CAM Robot</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <style>
      body { font-family: Arial; text-align: center; margin:0px auto; padding-top: 30px;}
```

```
      table { margin-left: auto; margin-right: auto; }
      td { padding: 8 px; }
      .button {
        background-color: #2f4468;
        border: none;
        color: white;
        padding: 10px 20px;
        text-align: center;
        text-decoration: none;
        display: inline-block;
        font-size: 18px;
        margin: 6px 3px;
        cursor: pointer;
        -webkit-touch-callout: none;
        -webkit-user-select: none;
        -khtml-user-select: none;
        -moz-user-select: none;
        -ms-user-select: none;
        user-select: none;
        -webkit-tap-highlight-color: rgba(0,0,0,0);
      }
      img {  width: auto ;
        max-width: 100% ;
        height: auto ;
      }
    </style>
  </head>
  <body>
    <h1>ESP32-CAM Robot</h1>
    <img src="" id="photo" >
    <table>
      <tr><td colspan="3" align="center"><button class="button"
onmousedown="toggleCheckbox('forward');" ontouchstart="toggleCheckbox('forward');"
onmouseup="toggleCheckbox('stop');"
ontouchend="toggleCheckbox('stop');">Forward</button></td></tr>
      <tr><td align="center"><button class="button" onmousedown="toggleCheckbox('left');"
ontouchstart="toggleCheckbox('left');" onmouseup="toggleCheckbox('stop');"
ontouchend="toggleCheckbox('stop');">Left</button></td><td align="center"><button class="button"
onmousedown="toggleCheckbox('stop');"
ontouchstart="toggleCheckbox('stop');">Stop</button></td><td align="center"><button class="button"
onmousedown="toggleCheckbox('right');" ontouchstart="toggleCheckbox('right');"
onmouseup="toggleCheckbox('stop');"
ontouchend="toggleCheckbox('stop');">Right</button></td></tr>
      <tr><td colspan="3" align="center"><button class="button"
onmousedown="toggleCheckbox('backward');" ontouchstart="toggleCheckbox('backward');"
onmouseup="toggleCheckbox('stop');"
ontouchend="toggleCheckbox('stop');">Backward</button></td></tr>
    </table>
   <script>
```

```
   function toggleCheckbox(x) {
     var xhr = new XMLHttpRequest();
     xhr.open("GET", "/action?go=" + x, true);
     xhr.send();
   }
   window.onload = document.getElementById("photo").src = window.location.href.slice(0, -1) +
":81/stream";
  </script>
  </body>
</html>
)rawliteral";

static esp_err_t index_handler(httpd_req_t *req){
  httpd_resp_set_type(req, "text/html");
  return httpd_resp_send(req, (const char *)INDEX_HTML, strlen(INDEX_HTML));
}

static esp_err_t stream_handler(httpd_req_t *req){
  camera_fb_t * fb = NULL;
  esp_err_t res = ESP_OK;
  size_t _jpg_buf_len = 0;
  uint8_t * _jpg_buf = NULL;
  char * part_buf[64];

  res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
  if(res != ESP_OK){
    return res;
  }

  while(true){
    fb = esp_camera_fb_get();
    if (!fb) {
      Serial.println("Camera capture failed");
      res = ESP_FAIL;
    } else {
      if(fb->width > 400){
        if(fb->format != PIXFORMAT_JPEG){
          bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf, &_jpg_buf_len);
          esp_camera_fb_return(fb);
          fb = NULL;
          if(!jpeg_converted){
            Serial.println("JPEG compression failed");
            res = ESP_FAIL;
          }
        } else {
          _jpg_buf_len = fb->len;
          _jpg_buf = fb->buf;
        }
      }
```

```
    }
    if(res == ESP_OK){
      size_t hlen = snprintf((char *)part_buf, 64, _STREAM_PART, _jpg_buf_len);
      res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);
    }
    if(res == ESP_OK){
      res = httpd_resp_send_chunk(req, (const char *)_jpg_buf, _jpg_buf_len);
    }
    if(res == ESP_OK){
      res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY, strlen(_STREAM_BOUNDARY));
    }
    if(fb){
      esp_camera_fb_return(fb);
      fb = NULL;
      _jpg_buf = NULL;
    } else if(_jpg_buf){
      free(_jpg_buf);
      _jpg_buf = NULL;
    }
    if(res != ESP_OK){
      break;
    }
    //Serial.printf("MJPG: %uB\n",(uint32_t)(_jpg_buf_len));
  }
  return res;
}

static esp_err_t cmd_handler(httpd_req_t *req){
  char*  buf;
  size_t buf_len;
  char variable[32] = {0,};

  buf_len = httpd_req_get_url_query_len(req) + 1;
  if (buf_len > 1) {
    buf = (char*)malloc(buf_len);
    if(!buf){
      httpd_resp_send_500(req);
      return ESP_FAIL;
    }
    if (httpd_req_get_url_query_str(req, buf, buf_len) == ESP_OK) {
      if (httpd_query_key_value(buf, "go", variable, sizeof(variable)) == ESP_OK) {
      } else {
        free(buf);
        httpd_resp_send_404(req);
        return ESP_FAIL;
      }
    } else {
      free(buf);
      httpd_resp_send_404(req);
```

```
    return ESP_FAIL;
  }
  free(buf);
} else {
  httpd_resp_send_404(req);
  return ESP_FAIL;
}

sensor_t * s = esp_camera_sensor_get();
int res = 0;

if(!strcmp(variable, "forward")) {
  Serial.println("Forward");
  digitalWrite(MOTOR_1_PIN_1, 1);
  digitalWrite(MOTOR_1_PIN_2, 0);
  digitalWrite(MOTOR_2_PIN_1, 1);
  digitalWrite(MOTOR_2_PIN_2, 0);
}
else if(!strcmp(variable, "left")) {
  Serial.println("Left");
  digitalWrite(MOTOR_1_PIN_1, 0);
  digitalWrite(MOTOR_1_PIN_2, 1);
  digitalWrite(MOTOR_2_PIN_1, 1);
  digitalWrite(MOTOR_2_PIN_2, 0);
}
else if(!strcmp(variable, "right")) {
  Serial.println("Right");
  digitalWrite(MOTOR_1_PIN_1, 1);
  digitalWrite(MOTOR_1_PIN_2, 0);
  digitalWrite(MOTOR_2_PIN_1, 0);
  digitalWrite(MOTOR_2_PIN_2, 1);
}
else if(!strcmp(variable, "backward")) {
  Serial.println("Backward");
  digitalWrite(MOTOR_1_PIN_1, 0);
  digitalWrite(MOTOR_1_PIN_2, 1);
  digitalWrite(MOTOR_2_PIN_1, 0);
  digitalWrite(MOTOR_2_PIN_2, 1);
}
else if(!strcmp(variable, "stop")) {
  Serial.println("Stop");
  digitalWrite(MOTOR_1_PIN_1, 0);
  digitalWrite(MOTOR_1_PIN_2, 0);
  digitalWrite(MOTOR_2_PIN_1, 0);
  digitalWrite(MOTOR_2_PIN_2, 0);
}
else {
  res = -1;
}
```

```
  if(res){
    return httpd_resp_send_500(req);
  }

  httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
  return httpd_resp_send(req, NULL, 0);
}

void startCameraServer(){
  httpd_config_t config = HTTPD_DEFAULT_CONFIG();
  config.server_port = 80;
  httpd_uri_t index_uri = {
    .uri       = "/",
    .method    = HTTP_GET,
    .handler   = index_handler,
    .user_ctx  = NULL
  };

  httpd_uri_t cmd_uri = {
    .uri       = "/action",
    .method    = HTTP_GET,
    .handler   = cmd_handler,
    .user_ctx  = NULL
  };
  httpd_uri_t stream_uri = {
    .uri       = "/stream",
    .method    = HTTP_GET,
    .handler   = stream_handler,
    .user_ctx  = NULL
  };
  if (httpd_start(&camera_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(camera_httpd, &index_uri);
    httpd_register_uri_handler(camera_httpd, &cmd_uri);
  }
  config.server_port += 1;
  config.ctrl_port += 1;
  if (httpd_start(&stream_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(stream_httpd, &stream_uri);
  }
}

void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //disable brownout detector

  pinMode(MOTOR_1_PIN_1, OUTPUT);
  pinMode(MOTOR_1_PIN_2, OUTPUT);
  pinMode(MOTOR_2_PIN_1, OUTPUT);
  pinMode(MOTOR_2_PIN_2, OUTPUT);
```

```
Serial.begin(115200);
Serial.setDebugOutput(false);

camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

if(psramFound()){
  config.frame_size = FRAMESIZE_VGA;
  config.jpeg_quality = 10;
  config.fb_count = 2;
} else {
  config.frame_size = FRAMESIZE_SVGA;
  config.jpeg_quality = 12;
  config.fb_count = 1;
}

// Camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
  Serial.printf("Camera init failed with error 0x%x", err);
  return;
}
// Wi-Fi connection
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
```

```
  Serial.println("WiFi connected");

  Serial.print("Camera Stream Ready! Go to: http://");
  Serial.println(WiFi.localIP());

  // Start streaming web server
  startCameraServer();
}

void loop() {

}
```

10.2 DATA SHEETS

1. **ESP32 CAM MODULE**

2. **L298N MOTOR DRIVER**