MUHAMMAD NAIM ISKANDAR BIN RAHIMI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

SMART PATROLLING MOBILE CAR FOR SECURITY USING ESP32-CAM

MUHAMMAD NAIM ISKANDAR BIN RAHIMI

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronics Engineering Technology (Industrial Electronics) with Honours

Faculty of Electronics and Computer Technology and Engineering
Universiti Teknikal Malaysia Melaka

2025

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : SMART PATROLLING MOBILE CAR FOR SECURITY USING ESP32-CAM

Sesi Pengajian : 2024/2025

Saya MUHAMMAD NAIM ISKANDAR BIN RAHIMI mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.

☐ **SULIT*** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

☐ **TERHAD*** (Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan.

☒ **TIDAK TERHAD**

Disahkan oleh:

(TANDATANGAN PENULIS) (COP DAN TANDATANGAN PENYELIA)

Alamat Tetap:...
....
....

Tarikh : 22 December 2024 Tarikh : 22 December 2024

DECLARATION

I declare that this project report entitled "SMART PATROLLING MOBILE CAR FOR SECURITY USING ESP32-CAM" is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	
Student Name	:	MUHAMMAD NAIM ISKANDAR BIN RAHIMI
Date	:	22 December 2024

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical Engineering Technology with Honours.

Signature :

Supervisor Name : DR FARID ARAFAT BIN AZIDIN

Date :

Signature :

Co-Supervisor :

Name (if any) NO APPLICABLE

Date :

DEDICATION

This study is dedicated to my dear father, who encouraged me to keep learning even in terrible situations as long as there is a possibility, because knowledge is the most valuable asset we may own. It is also dedicated to my wonderful mother, who taught me to never give up on what I am doing right now. She showed me that even the tiniest step forward is still a progress, and that with even the tiniest step forward, I will be able to complete the goal.

ABSTRACT

This paper presents the design and implementation of a Smart Mobile Patrol Car for security applications, using the ESP32-CAM module. The main objective of this project is to develop an autonomous surveillance system capable of real-time monitoring, threat detection and incident response in various environments. The system integrates advanced features such as video streaming and remote control capabilities, all powered by Wi-Fi connectivity and ESP32-CAM processing capabilities. The methodology involves the design of mobile car architecture, hardware integration including motors and ESP32-CAM modules and algorithm development software for navigation and surveillance. The system is equipped with a camera for live video feed, allowing operators to monitor the area remotely. Test results show the effectiveness of the system in providing real-time surveillance, optimizing patrol routes and facilitating rapid response to security threats. The findings show that Smart Mobile Patrol Cars significantly improve safety measures, offering a scalable and cost-effective solution to improve public safety in a variety of settings. This research highlights the potential of integrating IoT technology in security applications, paving the way for future advancements in autonomous surveillance systems.

ABSTRAK

Kertas kerja ini membentangkan reka bentuk dan pelaksanaan Kereta Peronda Mudah Alih Pintar untuk aplikasi keselamatan, menggunakan modul ESP32-CAM. Objektif utama projek ini adalah untuk membangunkan sistem pengawasan autonomi yang mampu memantau masa nyata, pengesanan ancaman dan tindak balas insiden dalam pelbagai persekitaran. Sistem ini menyepadukan ciri lanjutan seperti penstriman video dan keupayaan kawalan jauh, semuanya dikuasakan oleh sambungan Wi-Fi dan keupayaan pemprosesan ESP32-CAM. Metodologi ini melibatkan reka bentuk seni bina kereta mudah alih, integrasi perkakasan termasuk motor dan modul ESP32-CAM dan perisian pembangunan algoritma untuk navigasi dan pengawasan. Sistem ini dilengkapi dengan kamera untuk suapan video langsung, membolehkan pengendali memantau kawasan itu dari jauh. Keputusan ujian menunjukkan keberkesanan sistem dalam menyediakan pengawasan masa nyata, mengoptimumkan laluan rondaan dan memudahkan tindak balas pantas terhadap ancaman keselamatan. Penemuan menunjukkan bahawa Kereta Peronda Mudah Alih Pintar meningkatkan langkah keselamatan dengan ketara, menawarkan penyelesaian berskala dan kos efektif untuk meningkatkan keselamatan awam dalam pelbagai tetapan. Penyelidikan ini menyerlahkan potensi mengintegrasikan teknologi IoT dalam aplikasi keselamatan, membuka jalan untuk kemajuan masa depan dalam sistem pengawasan autonomi..

ACKNOWLEDGEMENTS

"In the name of Allah, Most Gracious, Most Merciful"

I thank Allah, our creator, with all my strength for this chance. Despite several charges, I was able to complete my project successfully. I'd want to thank Him for putting in me the will to complete this assignment effectively and excellently. Thank you to my supervisor from the Faculty of Electronic and Computer Technology and Engineering, Dr Farid Arafat, for enabling me to work on PSM with him. He also provided me with a great deal of assistance and instruction for me to complete this job. Further- more, when I make a lot of mistakes during my final year project, he always steers me in proper direction. He also offered me a lot of ideas and encouragement to improve and polish my final year project. Many thanks to the faculty, Faculty of Electronics and Computer Technology and Engineering, University Technical Malaysia Melaka, for the facilities, especially the instructors who gave a lot of equipment to utilize on my PSM. Next, I'd want to thank you my parents, Rahimi bin Ibrahim and Rosnani binti Abdul Razak, for their moral support and encouragement. I also want to thank my fellow friends who assisted me greatly in completing my final year project successfully. Finally, despite the numerous hurdles that must be overcome to accomplish this endeavor effectively, I am thankful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Developing a smart patrol mobile car system application involves a comprehensive approach. First, design the architecture of the system, determine its objectives, the areas to be controlled and the types of security threats it will address. This stage is important for outlining system functionality and determining hardware and software requirements. Second, this project involves into the hardware and software development aspects. This includes selecting and integrating the necessary hardware components such as cameras, Wi-Fi modules and microcontrollers, and then developing the software, including a user-friendly interface, control algorithms for navigation, security features such as authentication, data logging and analysis capabilities.

The integration of hardware and software components is an important step, ensuring smooth communication and functionality. Testing and fine-tuning the system for reliability, security and performance is essential before deployment. Once deployed, ongoing maintenance, updates and support are critical to the system's long-term success and effectiveness in improving security measures.

1.2 Problem Statement

In an era where security concerns are increasingly prevalent, traditional surveillance methods often fall short in providing real-time monitoring and rapid response capabilities. Many existing security systems rely on static cameras and human patrols, which can be limited in coverage, efficiency, and responsiveness to incidents. This creates vulnerabilities in public safety, particularly in large or complex environments such as parking lots, campuses, and urban areas. The challenge lies in developing an autonomous and intelligent surveillance solution that can effectively monitor these environments, detect potential threats, and respond promptly without the need for constant human oversight.

The integration of advanced technologies, such as the ESP32-CAM, offers a promising avenue for creating a Smart Patrolling Mobile Car that can navigate autonomously, stream live video, and utilize motion detection to enhance security measures. This project aims to address key issues, including limited coverage from static surveillance systems, delayed response times from human patrols, and the resource inefficiency of traditional security measures that often require significant manpower and costs. By leveraging the capabilities of the ESP32-CAM, this project seeks to develop a Smart Patrolling Mobile Car that enhances surveillance efficiency, improves incident response times, and provides a scalable solution for modern security challenges.

1.3 Project Objective

Develop smart mobile patrol cars for security purposes that integrate technology to enhance surveillance, detection and response capabilities. Specifically, the objectives are as follows:

a) To develop an autonomous surveillance system.
b) To develop an optimize patrol routes.
c) To create a scalable system that is easily adaptable to different security situations and needs.

1.4 Scope of Project

The scope of this project are as follows:

a) Design the physical structure and mobility mechanism of the patrol cars.
b) Provide real-time video streaming.
c) Development of software algorithms for navigation, video streaming.
d) Deployment of the Smart Patrolling Mobile Car in selected environments.
e) Creation of a user-friendly interface for remote monitoring.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In today's modern a paper investigates the option of developing intelligent patrol mobile cars for security uses. Most of them cover the areas of application, sensors, wireless communication, and RTMS in improving security. The review discusses about the way the mobile robotics plays a vital role in establishing an optimum and optimal patrolling arrangement with an emphasis on technology and recent scenario. This analysis summarizes the present research and advances made in automated security patrolling systems that would otherwise be difficult to compile if studied separately.

2.2 Past Year Journal

From prior journal literature on smart patrolling mobile cars for security, it is evident that there has been considerable development in incorporating sensors, artificial intelligence, and real-time data acquisition. Research done in the past year has highlighted advances in self-driving navigation, threat recognition programs, and the employment of connected devices to improve information exchange. Studies have therefore focused on exploring models that enhance energy efficiency and the implementation of these systems in several contexts including commercial buildings and industries. These advancements are the evidence of the fact that smart patrolling cars can be useful in the commercial security services provision as the reliable and automated security systems are becoming the new trend in the security industry.

2.2.1 Security Robot for Real-time Monitoring and Capturing

According to the journal made by (Vijayasundara et al.,2021) said the aims to create a security robot that can monitor and record in real time in uncontrolled areas, such as homes. Its main goal is to allow the robot to navigate on its own by using maps created by LiDAR sensors and precise localization provided by AMCL. The designed to provide video feeds of the triggered zone for increased security, identify intruders using sensors, and alert the client.[1]

The research project (Gong et al. 2018) aims to create a security strategy for service robots by monitoring human attention to robot movements via IMU modules that detect human upper body behavior in real time. The use of machine vision is employed to identify people's face and determine their attention level thus helping the robot change its movements for safety. The experiment involves using the Nao service robot interface and the WSSS motion tracking technology to capture raw body movements in real time so that the robot can respond to actual body movements. Last but not least, the initiative aims at enhancing safety and reducing the frequency of mishaps through a combination of permissions for robot operations that integrate judgments of human attention.[2]

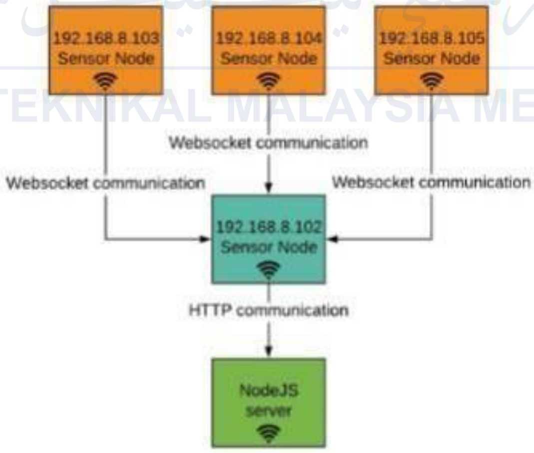

Figure 2.1 Wireless sensor network arrangement[1]

[1] The advantages use of artificial intelligence methods for intelligent anomalous situation identification in the monitoring region, such as AMCL. The AMCL algorithm uses LiDAR laser scans and then crosses them with the given map and in this way, fixes the odometry errors and ensures that the robot is localized correctly. AMCL enhances the robot's ability to move around the operational zone independently, which publishes the 'amcl pose' topic containing corrected odometry pose estimates. This paper has shown that AMCL is

essential in intruder detection and contributes significantly to improving other security measures in the robot. Through the odometry posture error correction, AMCL effectively prevents further error accumulation during robot movements, making the robot more reliable and accurate.

However, errors in robot localization and navigation might occur due to the fact that AMCL relies on the sensors which may produce inaccurate results and the environment may also change. It is also possible that the computing load required by the algorithm is high, especially when processing large volumes of sensor data, which may also influence its real-time interactivity. Last but not least, in dynamic environments that are constantly changing, the AMCL may struggle to quickly respond to new conditions in which the robot might be placed and, therefore, may provide delayed or erroneous positioning of the robot..

2.2.2 A Raspberry Pi based Smart Security Patrol Robot

According to the study article, (Joy et al.,2023) have developed a Smart Security Patrol Robot based on Raspberry Pi technology. Its objectives are to enhance security and monitor facilities at different locations, such as warehouses, schools, hospitals, and offices. The camera of the robot can record in real time and the sound sensor can detect any unusual noises while the AGPS receiver can help locate itself to aid in security patrolling. The Smart Security Patrol Robot can in the future eliminate conventional static CCTV systems and reduce the employment of staff in security networks as it will offer a cheap and efficient security surveillance solution.[3]

The project titled "Automatic Night Patrol System with Raspberry Pi" by (Gohika, Brindha Judith Yeshvanthi, and Jivitha, 2023) seeks to enhance security in deserted and poorly illuminated areas through mechanized vigilance. This initiative intends to reduce human intervention by autonomously detecting emergency situations, taking photographs, and alerting users. It uses Raspberry Pi technology, Python coding, and IoT integration to effectively monitor and detect human motion. By using technologies such as the MQTT protocol for alert signals and SMTP for image delivery, the system ensures that rescue activities are handled quickly.[4]

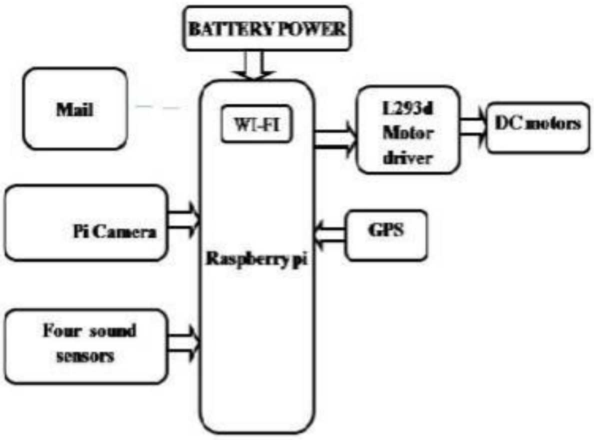

Figure 2.2 Proposed System Block Diagram[3]

The components incorporated into the system include Raspberry Pi, Camera, Sound Sensor, GPS Receiver, and L293D Motor Driver module are shown in the given block diagram. This enables proper coordination and communication whether in the patrolling process within the security firms or between the security firms and the security personnel. Explains the integration of these constituent elements and their work to determine security threats, to obtain relevant information, and to transmit data to the user for his efficient action. In other words, it presents the structural design and the connection between the hardware devises and their role in achieving the security objectives of the project. It also gives a broad diagram of how the project was designed and how it was to be implemented.[3]

The benefit of utilizing a Raspberry Pi microcontroller is that it provides an adaptable platform with a general-purpose operating system (OS) like Raspbian, making it appropriate for a range of jobs and applications. Moreover, the Raspberry Pi is easily integrals with parts like motor drives, sound sensors, and cameras, allowing the development of entire security patrol robot systems. Furthermore, the open source design and community support of Raspberry Pi provide scalability and customization choices to enhance project capabilities and adjust them to various security requirements.[3]

Figure 2.3 Raspberry Pi[5]

2.2.3 Anomaly Detection using Elevation and Thermal Map for Security Robot sign and Implementation of Security Patrol Robot Using Android Application

This research paper from (Shin and Na 2020) focuses on create an algorithm for anomaly identification in a surveillance area by utilizing thermal and elevation maps produced by a security robot fitted with a thermal imaging camera and Lidar. The technique creates top projection maps of the point cloud and temperature distributions in the surveillance region using Lidar for localization. The project's objective is to improve security robot surveillance performance by precisely identifying abnormalities in the monitored region, such as changes in temperature or structure. The block diagram illustrates how anomalies are categorized using a two-step approach that combines CNN and auto-encoder.[6]

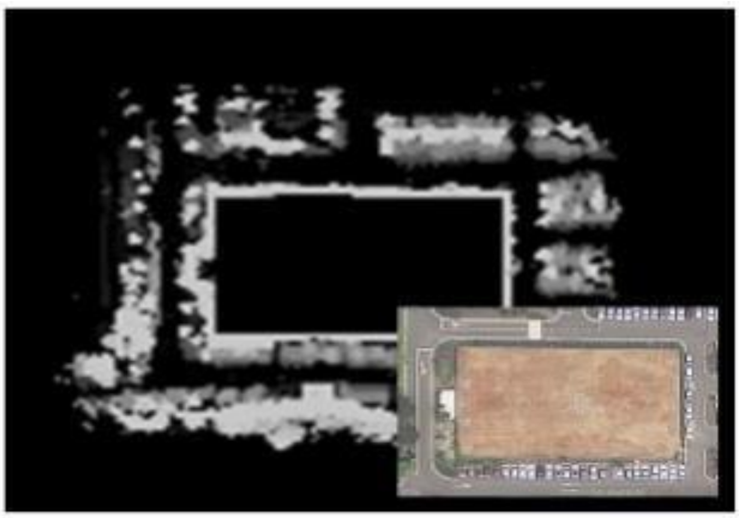

Figure 2.4 An example of elevation map in normal elevation situation[6]

Thermal maps typically display a uniform temperature distribution over the surveillance area, which indicates the typical environmental conditions and thermal signature of the item. On the other hand, abnormal conditions indicate erratic temperature patterns, which often indicate the existence of anomalies such as stoves or gas burners responsible for isolated hot spots on the thermal map. The goal of using thermal maps to distinguish between normal and anomalous conditions is to empower safety robots to recognize deviations from the expected temperature distribution. This capability can assist in identifying potential security risks or anomalous activity within the observed area.[6]

2.2.4 Design and Implement a Robot for Security Purposes

According to the journal made by (Sadkhan 2019) technology robots have come a long way and are used in many fields, including manufacturing, healthcare, security, and education. Robots provide benefits such as lowering the requirement for human labor and improving system security via steady efficiency. The project's features include taking pictures of eyes to improve biometric identification, allowing the robot to walk independently in different directions, and including speakers for conversation, a camera for face recognition, and a microphone for voice recognition.[7]

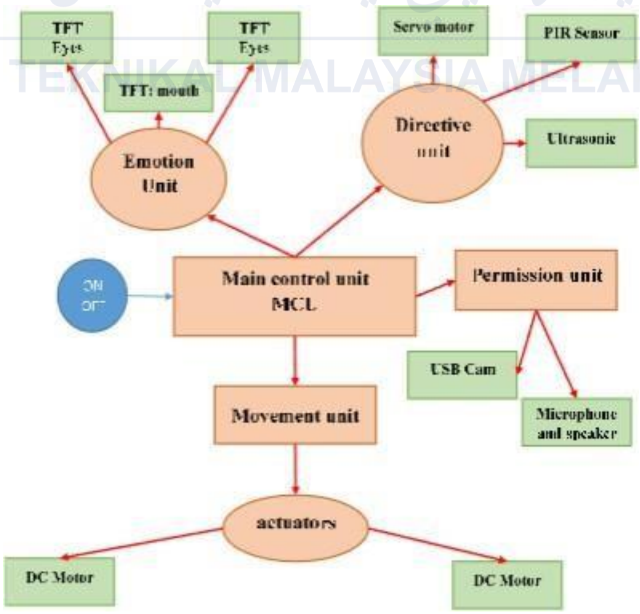

Figure 2.5 Block system diagram[7]

The block diagram in the research article illustrates the essential elements of the suggested robot design. It also shows how the various components of the robot interact with one another and the system architecture. Parts of the schematic include the Arduino Mini-Pro with PIR sensor and Ultrasonic for continuous direction correction, speakers for communication, and MCU for device connectivity. Furthermore, it displays the completed form of the robot's microphone and head for word recognition. By dividing the system into manageable chunks with clear functions, the block diagram helps to clarify the information flow between different modules inside the robot and reduces the complexity of the system.[7]

2.2.5 IoT Based Smart Home Automation and Security System Using Mobile App with Assistant Robot for Developing Countries

The project's approach (Anon 2017) is creating a system that combines Internet of Things (IoT) technology with a smartphone application to remotely monitor and operate home appliances. The system works by allowing users to access a mobile application that talks with the IoT devices in their home via a central server. The mobile application allows users to remotely monitor and manage household equipment, such as lights and air conditioners. The system also contains functions like scheduling chores for appliances and getting notifications for abnormal circumstances, which improve convenience and energy efficiency.[8]

The goal of the project (Chowdhury et al. 2019) is to use IoT technology and the TI-CC3200 Launchpad board to develop a smart wireless home security and automation system that allows for remote monitoring and control of home equipment. PIR motion sensors are installed at building entrances to detect human movement and send warnings to the owner's phone for rapid action. The system maintains security by transmitting voice calls solely to a predetermined number, improving system fidelity and preventing unwanted access.[9]

The project (Sarkar, Gayen, and Bilgaiyan 2018) focuses are to develop an Android-based home security system that uses IoT and Firebase technology. It uses IoT technology to improve home automation and security, while also making life easier with smart features. The system controls automation via a microcontroller-based NodeMcu Esp8266 and interacts with users via an Android mobile application. Firebase services such as Firebase Database, Authentication, and Functions are used to personalize the project and provide functionality such as notifications for events like burglar and fire alarms. The system seeks to monitor and secure houses by wirelessly delivering data from NodeMcu to Firebase and subsequently to the user's Android application, utilizing high-quality sensors such as PIR and Flame Sensors.[10]

2.2.6 The Problem of Ensuring the Information Security of Robots in the Implementation of the Laws of Robotics

According to the research (Ivanov, Kamenskikh, and Yuzhakov 2023) the rules of robotics seek to protect people from potentially dangerous acts of intelligent robots, but have not been fully realized due to the partial production of ideal robots. Significant developments in artificial intelligence training have pushed robots closer to embodying simpler versions that can better comply with robotics regulations. The paper emphasizes the need of tackling the problem given by robots' dual perception as objects and subjects in order to assure their information security.[11]

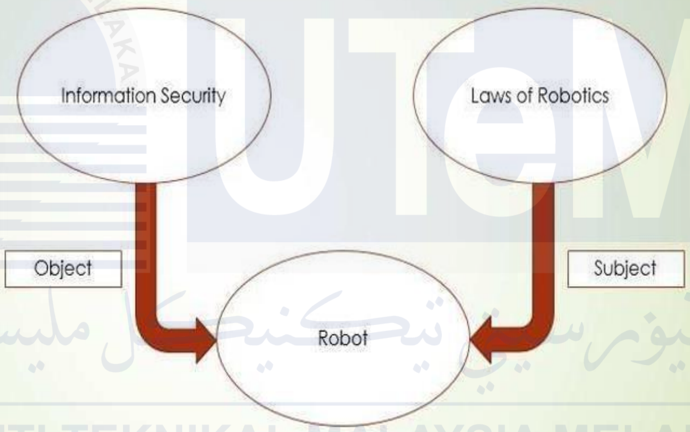

Figure 2.6 Dual perception of the robot[11]

Figure 2.6 show the dual perception of robots refers to how they are perceived as both objects and creatures capable of awareness and subjectivity. Robots, from one perspective, are things designed to do certain jobs and operate using programming and artificial intelligence. However, the development of powerful artificial intelligence technology has blurred the border between object hood and subjectivity, raising issues about robot ethics and rights. This duality in perception complicates assuring robot information security since they are more than just passive objects and may require protections similar to those granted to sentient people.

2.2.7 A Method for Navigation System of Security Robot based on Millimeter Wave Robust Mobile Robot Localization Based on Security Laser Scanner

The goal of the project (Li et al. 2021) is to create a navigation system for security robots that can function in difficult conditions, such as smoke, dust, and darkness. Millimeter-wave radar technology is employed by the navigation system to determine distances and guarantee robot movement under challenging circumstances. It has an integrated multi-stage canceller to reduce static interference in indoor spaces and a dynamic threshold detector to keep radar functioning in the face of fluctuating noise levels. The robot can now precisely and automatically traverse a predetermined course thanks to the system. The system's accuracy in navigation tasks is demonstrated by the greatly decreased x-, y-, and overall positioning errors.[12]

2.2.8 Smart Night Patrolling Robot using YOLOv8

According of the project (Kiruthika et al. 2023) is to develop an intelligent night patrolling robot that will improve security at night in residential buildings, gated communities, and commercial spaces. Utilizing a variety of sensors, including an infrared sensor, GPS, sound sensor, and ultrasonic sensor, the intelligent night patrolling robot is built to function independently and is powered by a rechargeable battery. It achieves a high accuracy of 92% for object detection in night vision conditions by using the YOLOv8 model for object detection. The main job of the robot is to use its sensors to scan its surroundings for suspicious activity and then analyses them. The robot uses the Telegram API to send an emergency message to a designated conversation ID in the event of any possible security danger or human activity, ensuring a prompt reaction to security breaches.[13]

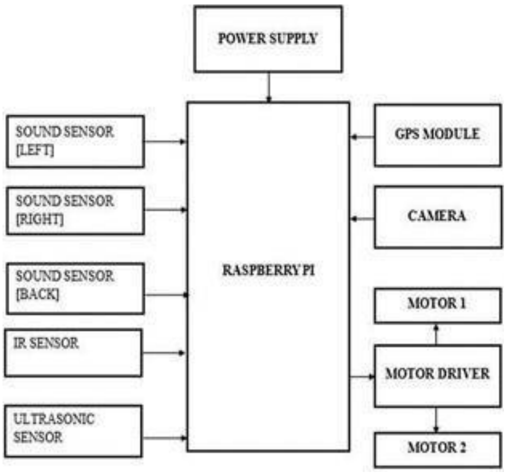

Figure 2.7 System Architecture[13]

2.2.9 Research and Design of Robot Application System Security Protection in Electric Power Business Hall based on Artificial Intelligence

The project (Ehfrph et al. n.d.) main objective is to advance security protection in the area of innovative and practical technology. The primary objective of the Intelligent Business Hall and Application System is to improve security measures by utilizing artificial intelligence. In order to give customers a new kind of dependable and convenient power supply service, the project intends to launch an "internet" electricity marketing business. It highlights comprehensive protection and dynamic sensing as design elements for security in the electric power industry.[14]

2.2.10 Security Controller Synthesis for ROS-based Robot

According the journal by (Zhao et al. 2020) in order to shield robots from potential cyber threats, the research focuses on improving the security of robots that are managed by the Robot Operating System (ROS). Robot controllers are separated into task controllers and security controllers according to the project's suggested methodology. This section contributes to the production of controllers in an efficient manner, hence decreasing the process time. The primary goal of the project is to provide security controllers for ROS robots in order to reduce the danger that comes with cyberattacks. In this way, the research hopes to minimize the impact of cyber risks while guaranteeing that robots can properly accomplish their tasks. The research uses Gazebo as an experimental environment where the produced security controllers are simulated to assess how well they protect robots from cyberattacks and guarantee proper execution.[15]

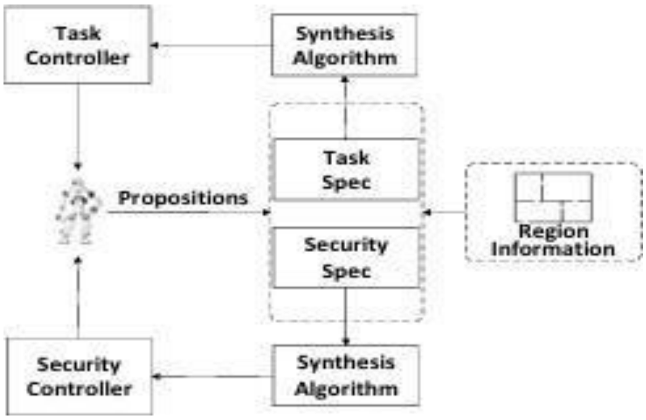

Figure 2.8 Controller generation[15]

2.2.11 Multifunctional Night Patrolling Robot Based on Rocker-Bogie Mechanism

The concept by (Shivaanivarsha, Devi Priya, and Santhiya 2022) entails building a robot that patrols at night and uses a rocker-bogie mechanism to keep an eye on sites like apartments and residential areas. With a variety of sensors, including PIR, vibration, sound, and fire detectors, the robot is made to monitor factors like temperature, sound, and alerts. To gather and process data, it is made up of parts including an ATMega368 Microcontroller, motors, temperature and fire sensors, camera modules, GSM, and more. In order for the system to work, sensors supply inputs to the microcontroller, which then uses that information to provide outputs on an LCD display, sound alarms, and transmit messages over GSM if there are any variations in the sensor values. The principal objective is to guarantee security by rapidly notifying authorities of any anomalies in the surroundings, augmenting surveillance, and implementing safety protocols.[16]

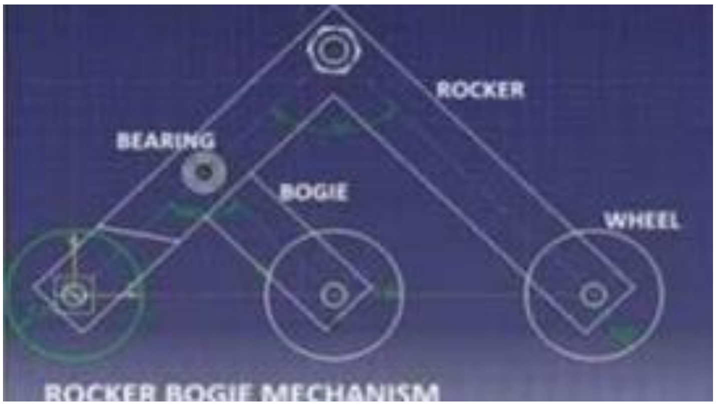

Figure 2.9 Rocker bogie design [16]

2.2.12 Design and Implementation of an IoT Based Patrol Robot

The goal of the project by (Amir et al. 2022) is to create and deploy an Internet of Things-based patrol robot. It uses artificial intelligence and the Internet of Things (IoT) to build a robot that can autonomously patrol predetermined locations. The patrol robot is made to roam around an area, collect information from sensors, and relay it to a central control unit. The technique entails giving the robot algorithms to learn so that it can navigate, avoid obstacles, and efficiently carry out surveillance activities. This project's main goal is to improve security measures by implementing a smart patrol robot that can observe and gather data in a variety of settings.[17]

2.2.13 Night Patrol Robot for Detecting and Tracking of Human Motions using Proximity Sensor

The project (Josephine et al. 2022) aims are to build a robot that can patrol on its own, recognize human actions, and tell what is legitimate and what is not. The robot's job is to increase security by keeping an eye out for suspicious activity and sounding an alarm. The robot with four wheels is programmed to navigate and react to human movements effectively using a proximity sensor and Raspberry Pi technology. The robot can precisely track and follow motions because it uses proximity sensors to detect motion and human presence. The robot has been given the computational capacity it needs to evaluate sensor data and make decisions in real time by utilizing the Raspberry Pi, a computer the size of a credit card. In order to allow the robot to successfully patrol a region, the system integrates hardware and software.[18]

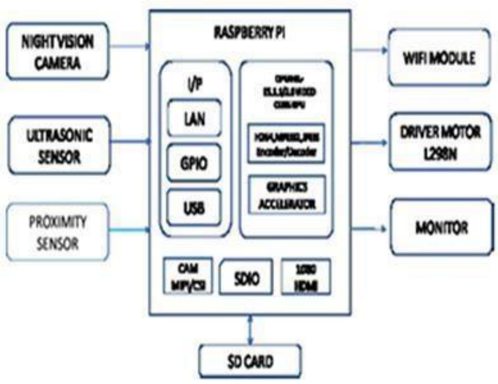

Figure 2.10 System Architecture of Night Patrol Robot[18]

Figure 2.10 show the system design of the Night Patrol Robot consists of a four-wheeled robot that is equipped with Raspberry Pi technology and a proximity sensor for independent operation. It is intended to autonomously traverse and monitor a space, proficiently identifying and monitoring human movements. The robot's central processing unit, the Raspberry Pi, allows it to evaluate sensor data, make judgments, and react instantly to motions that are detected. When needed, the robot's wireless connection capabilities enable it to take directions from a control center remotely. The robot may notify the control room in the event of obstacles or intruders, guaranteeing prompt action and assistance.[18]

2.2.14 Night Patrolling Robot

The research (Khalid et al. 2021) is to address the increased need for security brought on by population growth and resource constraints. Focuses on creating and deploying an intelligent robot to improve security measures over time at a reasonable cost. Because the robot is autonomous, it can function on its own without assistance from a person, improving its surveillance capabilities. The night patrolling robot's main purpose is to improve security protocols by eliminating the need for human interaction through autonomous patrolling. It has sophisticated technologies including a smart camera and GPS navigation to effectively detect and record the actions of intruders. In regions where traditional security measures would not be practical owing to financial constraints, the robot provides high-quality security surveillance at a reasonable caustic.[19]

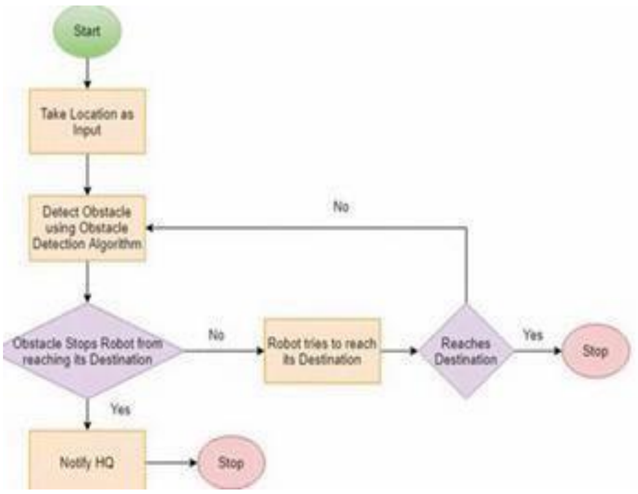

Figure 2.11 Flowchart for Path Finding Algorithm[19]

Figure 2.11 shows how the night patrolling robot project's operating procedures or decision-making process work. It may show how the robot scans a region on its own, looks for anomalies, records live video and pictures, and notifies the control center. The robot's logical course of action, such as utilizing GPS navigation to search routes and turning on the smart camera when it detects intrusions, might be delineated in a flowchart. It could also describe the reaction mechanisms and information flow channel that the robot uses to communicate with the control room. All things considered, the flowchart functions as a visual aid to demonstrate the methodical operation and interactions of the robot in effectively bolstering security measures.

2.2.15 IOT Night Safety Patrolling Robot

According to project by (Patil et al. 2022) are to improve security in remote locations by creating a security patrolling robot that makes use of ARM 7 technology. The robotic vehicle system has cameras and a microphone to efficiently monitor and secure the area. It uses the camera to continually scan the environment for problems. When the surroundings are silent, the robot will gravitate toward any sound it detects inside the monitoring region. It uses its camera to search the surroundings for human faces, allowing for live broadcasting of the crime scene and constant monitoring.[20]

2.2.16 Smart Mobile Robot for Security of Low Visibility Environment

The project involves designing a mobile robot for smart security patrol in the dark area. It plans to overcome the shortcomings of the existing security system through a non-vision-based system that employs artificial intelligence techniques such as Neural Network (NN) and Fuzzy Logic (FL) for improved security. The system employs a motion control model to enable proper movement operations within the monitoring area.[21]

Neural Network (NN) and Fuzzy Logic (FL) methodologies are used in a smart way to detect the abnormal situation in the monitored environment. The main goal of this project is to offer a sound security solution for environments that are difficult to illuminate, including the dark. It means that the use of a mobile robot with AI allows the system to move around the area and identify threats or other abnormalities on its own, which will improve security.[21]

To sum up, the project background is to design a new security system based on a mobile robot for the environment that is dark. It involves the use of a motion control model and other artificial intelligence algorithms such as NN and FL. The overall purpose of the project is to improve security using smart detection of abnormalities and optimized patrol movements.[21]

2.3 Theoretical Framework

The smart patrolling mobile car for security, based on Arduino technology, the smart patrolling mobile car for security seeks to provide an autonomous, effective, and dependable patrolling solution to meet contemporary security concerns. Conventional security methods frequently rely on human patrols, which has limitations due to human limitations, costs, and inconsistencies. Using the Arduino platform, this project integrates PIR (Passive Infrared) sensors for motion detection, cameras for visual surveillance, and ultrasonic sensors for obstacle detection. The main components of the system are a rechargeable battery pack that may be supplemented with solar panels for longer operation, and communication modules like GSM or Wi-Fi for remote monitoring and real-time data transmission. These parts work together to improve overall security measures by enabling autonomous navigation, real-time environmental monitoring, and prompt reporting of any suspicious activity.

The development process involves both hardware and software design, both software and hardware design are involved in the development process; software design comprises programming the Arduino, concentrating on motor control, communication, and sensor data processing. Algorithms for autonomous navigation, obstacle avoidance, and intrusion detection are developed and rigorously tested through simulations and real-world applications. Expected outcomes include improved intrusion detection, real-time incident reporting, and increased reliability and consistency in patrolling activities, leading to long-term cost savings by reducing reliance on human security personnel. Future enhancements could explore AI and machine learning for better decision making.

2.4 Comparison of the Literature Review

Table 2.1 Comparason

No.	Tittle	Author	Proposed Technique	Advantages	Disadvantages
1	Security Robot for Real-time Monitoring and Capturing	Vijayasundara et al.,2021	• Involves the development of a security robot for real-time monitoring and capturing in uncontrolled environments.	• Autonomous navigation. • Enhanced security.	• Initial setup complexity. • Cost.
2	Research on human-robot interaction Security Strategy of Movement authorization for service robot Based on people's attention monitoring	Gong et al. 2018	• The system entails the real-time monitoring of human movements through IMU and vision techniques to assess the attention of people on the robot. To achieve the goal, the proposed method gives the safety warning to the humans according to their static attitude in order to stop the robot movements in the warning state and to ask for authorization from the individuals to avoid any harm.	• Enhances safety. • Personalized authorization. • Real-time detection.	• Technology dependency. • Privacy concerns.

3	A Raspberry Pi based Smart Security Patrol Robot	Joy et al.,2023	• Designed with a camera and GPS module connected to a Raspberry Pi through a Wi-Fi module for communication.	• Affordable. • Reduction in manpower.	• Range limitation. • Lack of exact coordinates.
4	Automatic Night Patrol System with Raspberry Pi	Gohika, Brindha Judith Yeshvanthi, and Jivitha 2023	• Designed to automatically detect emergency situations using Raspberry Pi technology without the need for continuous human intervention.	• No complex algorithms. • Utilizes Python language.	• Limitations in detecting. • Depending on technology.
5	Anomaly Detection using Elevation and Thermal Map for Security Robot	Shin and Na 2020	• Uses a mobile robot with Lidar and thermal imaging to create maps showing elevation and temperature of an area.	• High accuracy. • Provides a structured approach to anomaly detection.	• Challenges in real world. • Require advanced technical knowledge.
6	Design and Implement a Robot for Security Purposes	Sadkhan 2019	• The proposed system in the research paper focuses on designing a robot that can be used for security purposes in various industries like	• Lowering operational costs for security. • Perform tasks.	• Lack the ability to adapt. • High cost.

			banks, medical facilities, and military sectors.		
7	IoT Based Smart Home Automation and Security System Using Mobile App with Assistant Robot for Developing Countries	Anon 2017	• Involves utilizing chaotic maps to generate encryption keys and DNA encoding to transform the image pixels, adding an extra layer of complexity to the encryption process.	• Versatility. • Enhanced Security.	• Computational complexity. • Key management.
8	IoT Based Smart Security and Home Automation System	Chowdhury et al. 2019	• Utilizes Internet of Things (IoT) technology to connect various devices and sensors within a home to a central system for monitoring and control.	• Improves convenience. • Energy efficiency. • Customization.	• Cost. • Dependency on internet. • Complexities.
9	Android based Home Security Systems using Internet of Things (IoT) and Firebase	Sarkar, Gayen, and Bilgaiyan 2018	• Integrates wireless home automation control using Wi-Fi technology, enabling remote monitoring and control of security devices in a smart and efficient manner.	• Convenience. • Integration. • Enhanced Security.	• Dependency on internet. • Complexities.

10	The Problem of Ensuring the Information Security of Robots in the Implementation of the Laws of Robotics	Ivanov, Kamenskikh, and Yuzhakov 2023	• Involves addressing the gap between the laws of robotics and the current capabilities of AI-powered robots.	• Better adherence. • Proactive.	• Complexities.
11	A Method for Navigation System of Security Robot based on Millimeter Wave	Li et al. 2021	• The system uses millimeter-wave radar to detect distance, a multi-stage canceller to reject static clutter, and a dynamic threshold detector to stabilize radar performance.	• Enables navigation. • Autonomy.	• Cost. • Complexity. • Limited range.
12	Smart Night Patrolling Robot using YOLOv8	Kiruthika et al. 2023	• Utilizes the YOLOv8 model for object detection with a high accuracy of 92% in night vision scenarios. • Incorporates various sensors like sound, GPS, ultrasonic, and IR sensors to analyze surroundings effectively.	• Cost effective and efficient solution. • Operates autonomously.	• Requires specialized technical expertise.

13	Research and Design of Robot Application System Security Protection in Electric Power Business Hall based on Artificial Intelligence	Ehfrph et al. n.d.	• Involves utilizing artificial intelligence in the Intelligent Business Hall and Application System to enhance security protection in technological innovation and application	• Improved security measures through the application of artificial intelligence.	• Potential challenges in the integration and implementation of artificial intelligence technology.
14	Security Controller Synthesis for ROS-based Robot	Zhao et al. 2020	• Involves dividing robot controllers into task controllers and security controllers for improved security in ROS-based robots.	• Enhances efficiency. • Boosts cybersecurity.	• Increased complexity. • Limited applicability.
15	Multifunctional Night Patrolling Robot Based on Rocker-Bogie Mechanism	Shivaanivarsha, Devi Priya, and Santhiya 2022	• Design of a multifunctional night patrolling robot that utilizes a rocker-bogie six-wheeling mechanism for stability in rough terrains.	• Monitoring capabilities. • Real time tracking.	• Limited coverage. • Cost.
16	Design and Implementation of an IoT Based Patrol Robot	Amir et al. 2022	• The IoT-based patrol robot utilizes a combination of sensors, AI algorithms, and communication	• Enhanced Security. • Real time monitoring.	• Initial Cost. • Technical challenges.

			modules to navigate and perform surveillance tasks autonomously.		• Privacy concerns.
17	Night Patrol Robot for Detecting and Tracking of Human Motions using Proximity Sensor	Josephine et al. 2022	• Using proximity sensors and Raspberry Pi technology.	• Real time monitoring of human movements, improving security measures in industries and other areas.	• Limited effectiveness in complex environments with high levels of background noise or movement.
18	Night Patrolling Robot	Khalid et al. 2021	• Designing and implementing a smart robot equipped with advanced features like GPS navigation and a smart camera.	• Cost effective • Utilizes modern technology	• Limited to the capabilities and range.
19	IOT Night Safety Patrolling Robot	Patil et al. 2022	• A security patrolling robot equipped with ARM 7 technology, cameras, and a microphone for monitoring remote areas and detecting any potential issues.	• Continuous monitoring of surroundings. • Live broadcasting.	• Limited physical capabilities. • Costs and maintenance.

20	Smart mobile robot for security of low visibility environment	Faisal et al. 2015	• Utilizes a mobile robot for patrol in dark environments. It is a non-vision-based system that employs artificial intelligence methods like Neural Network (NN) and Fuzzy Logic (FL) to operate effectively in low visibility conditions.	• Motion control model. • Smartly determines abnormal situations.	• Limited visibility range. • Dependence on AI accuracy

2.5 Summary

The research on smart patrolling mobile cars for security emphasizes important developments in communication, power management, navigation, and sensor integration. Studies have indicated that with real-time navigation algorithms, like SLAM, improves the car's capacity to maneuver through intricate terrain. Using cameras, and servo offers thorough environmental monitoring. The patrolling car's working time can be increased by utilizing power management techniques, such as the installation of solar panels. Between the vehicle and the central monitoring system, dependable and encrypted data transfer is ensured via secure communication protocols and the usage of Wi-Fi or GSM modules.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In today's world, security concerns are paramount, necessitating innovative solutions to enhance surveillance and threat detection capabilities. Traditional security systems, which often rely on static cameras and human patrols, can be limited in their effectiveness, particularly in large or dynamic environments. This project aims to address these challenges by developing a Smart Patrolling Mobile Car utilizing the ESP32-CAM module. This autonomous system is designed to provide real-time monitoring, detect potential threats, and respond promptly, thereby improving overall security measures. By integrating advanced technologies such as video streaming and autonomous navigation, the Smart Patrolling Mobile Car represents a scalable and cost-effective solution for modern security needs.

3.2 Flow Chart for Project Development

The project development of the Smart Patrolling Mobile Car using the ESP32-CAM outlines the sequential steps involved, starting from project initiation and system design, through hardware integration and software development, followed by testing and evaluation, deployment, user training, and concluding with project review and closure, ensuring a structured approach to achieving the project's objectives.

3.2.1 Block Diagram

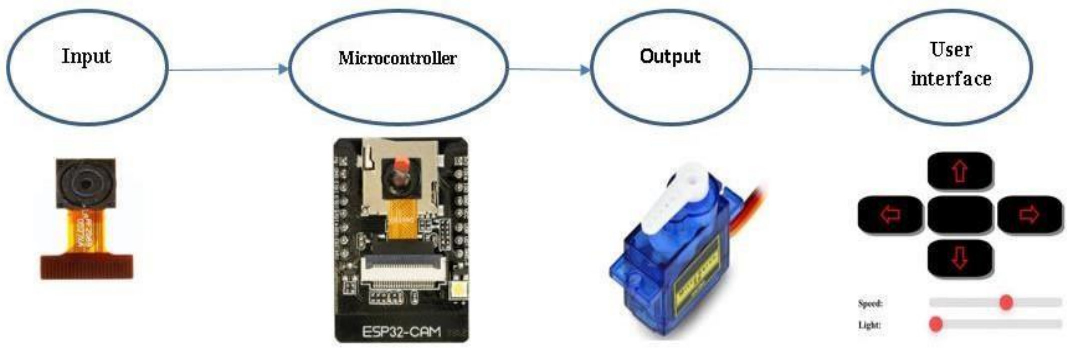

Figure 3.1 Block Diagramm

3.2.2 Flow Chart Development of Smart Patrolling Mobile Car for Security Using ESP32-CAM.

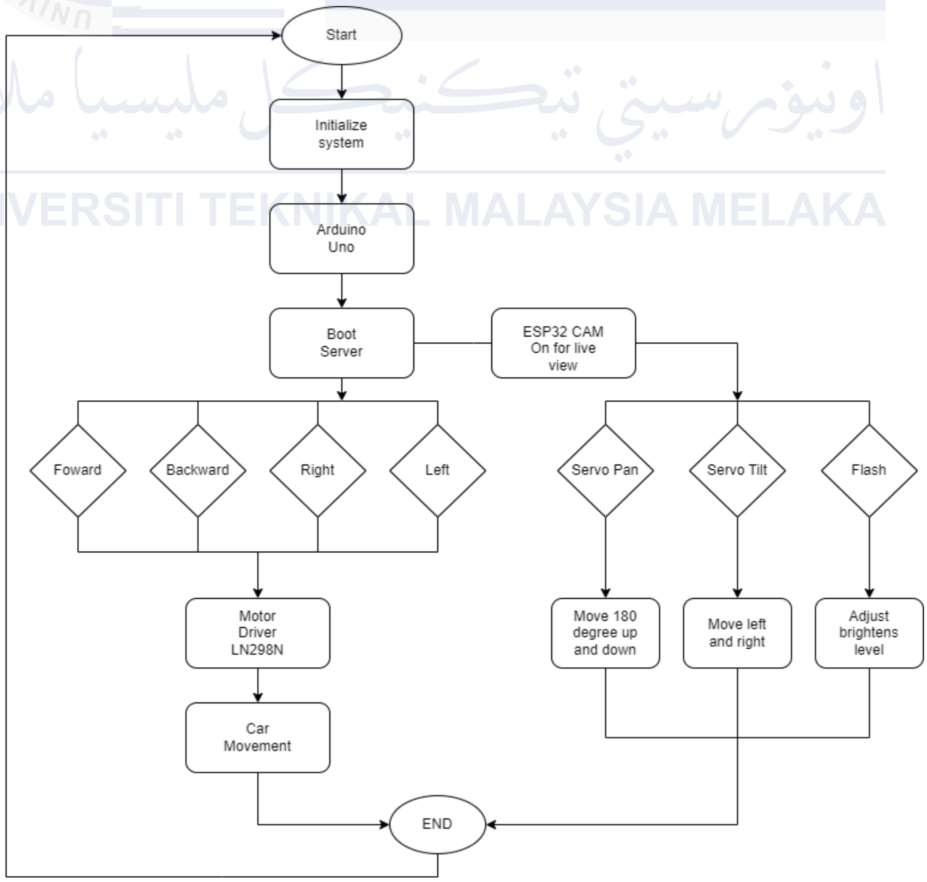

Figure 3.2 Sytem Flowchart

3.3 Project Hardware Selection

An ESP32-CAM is used in this project to provide a superior I/O pin and compact design for a security car. Hardware selection for the Smart Patrolling Mobile Car involves utilizing the ESP32-CAM module for real-time video streaming and surveillance, complemented by servo motors to enable precise control of the car's movement and camera orientation, ensuring effective monitoring and responsiveness to security threats

3.3.1 ESP32-CAM

A camera and an ESP32 microcontroller are combined in the small and cheap ESP32-CAM module to enable a variety of camera-focused applications. This module's capacity to take pictures, stream films, and carry out crucial computer vision jobs has made it more and more popular among people and professional alike. Its affordability and adaptability render it an attractive option for a range of do-it-yourself electronics projects

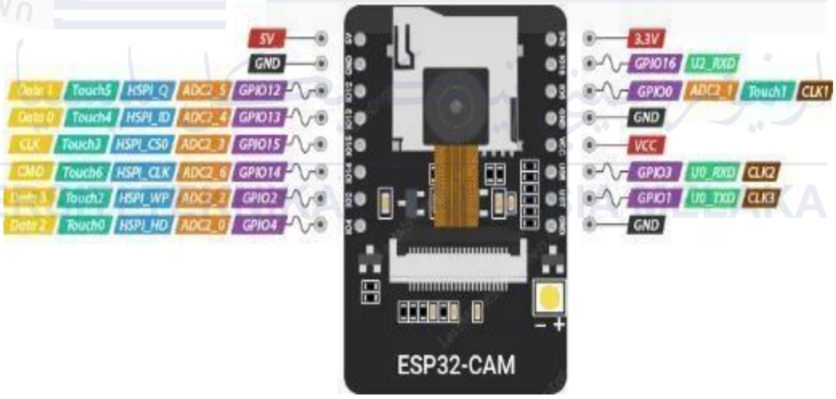

Figure 3.3 ESP32-CAM

Figure 3.3, The ESP32-CAM, which has dual-core CPU(central unit processor) and 33 Bluetooth and Wi-Fi functionality, powers the ESP32-CAM. For many current IoT applications, the module capacity to connect to networks and interact wirelessly is crucial. Furthermore, the module integrated camera makes it easier to share live video and take beautiful images.

Its simplicity of it is one of the main benefits of the ESP32-CAM. It is simple to program with the Arduino IDE, which is a popular option among hobbyists. A vast assortment of online tools and libraries are easily accessible to help consumers get started on their projects. Furthermore, the module's small size makes it simple to integrate into a variety of projects and gadgets that might not have a lot of space.

Specification:

- Chip: ESP32-D0WDQ6
- Architecture: Dual-core 32-bit Xtensa LX6 CPUs
- Camera Model: OV2640
- Resolution: 2 Megapixels (1600 x 1200)
- Wi-Fi: 802.11 b/g/n, supports 2.4 GHz band
- Micro SD Card Slot: Supports cards up to 4 GB
- Operating Voltage: 3.3V
- Size: Approximately 27 x 40.5 x 4.5 mm

3.3.2 Servo Motor

A servo motor is an electromechanical device that provide precise control of angular position, making it ideal for application like smart mobile patrolling. In the context of an ESP32-CAM, servo motor enable the camera to pan and tilt, allowing for dynamic surveillance and monitoring by adjusting the camera viewpoint to show a wider area. This capability enhances surveillance coverage by enabling the camera to scan specific location. By integrating servo motor with ESP32-CAM, user can remotely control the camera position through a web interface, facilitating real time monitoring.

Figure 3.4 Sevo Motor

Additionally, servo motors can assist in obstacle detection and navigation, helping the mobile patrol robot maneuver through complex environments while maintaining its surveillance duties. The ESP32's built-in WiFi allows for remote control of the servo motors, making it particularly useful in security applications. Overall, servo motors are essential in enhancing the functionality of smart mobile patrolling systems using the ESP32-CAM, providing flexibility, control, and improved surveillance capabilities that are crucial for effective security monitoring.

Specification;

- Type: Standard or Continuous Rotation Servo
- Operating Voltage: 4.8V to 6.0V
- Torque: 3.0 kg/cm to 20 kg/cm
- Speed: 0.1 to 0.2 seconds per 60 degrees at 6V
- Size: 40mm x 20mm x 40mm
- Weight: Approximately 30g
- Connector Type: 3-pin connector (power, ground, signal)

3.3.3 Voltage Regulator

A voltage regulator with a display for step-down voltage is an electronic device designed to convert a higher input voltage to a lower, stable output voltage, specifically for powering components like servo motors. In the context of a smart mobile patrolling car using an ESP32-CAM, this device is crucial for ensuring that the servo motors receive the appropriate voltage for optimal performance and operation. For example, if the car is powered by a 12V battery, the voltage regulator can step it down to a suitable level, such as 5V or 6V, which is often required by standard servo motors.

Figure 3.5 Voltage Regulator

The display feature of the voltage regulator allows users to monitor real-time voltage levels, providing valuable feedback on the power supply status. This is particularly important in a mobile patrol car, where consistent and reliable operation is essential for tasks such as camera panning and tilting, which are controlled by servo motors. The voltage regulator also ensures voltage stability, maintaining a constant output despite variations in input voltage or load conditions, which is critical for the reliable operation of the servo motors and the ESP32-CAM. Additionally, many voltage regulators include protection features like over-voltage and thermal protection, safeguarding the components from potential damage due to power fluctuations. Overall, the voltage regulator with a display enhances the reliability and functionality of the smart mobile patrolling car, ensuring that all components operate efficiently and effectively

3.3.4 Motor Driver

Motor drivers are essential components of any robotics or automation project. They supply the necessary power to operate motors and other components in an application. The motor driver provides voltage, current, directionality, and protection for optimal operation of robotic components and devices.

The L298N is a dual H-Bridge motor driver that allows you to control the speed and direction of two DC motors simultaneously. The module can drive DC motors with voltages ranging from 5 to 35 V and peak currents of up to 2A

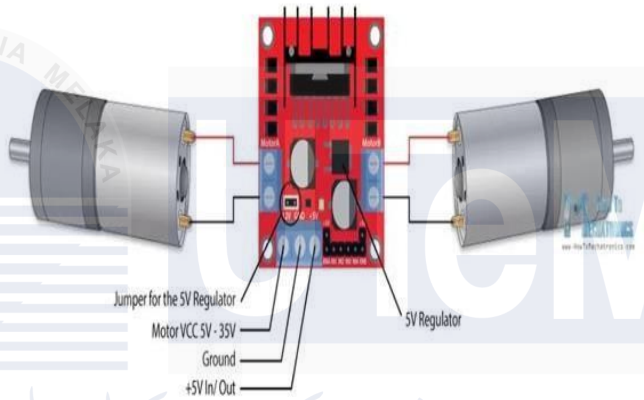

Figure 3.6 Motor driver connection

Figure 3.6 show the connection the module has two screw terminal blocks for the motors A and B, as well as another screw terminal block for the Ground pin, the motor's VCC, and a 5V pin that may be used as an input or output. This is dependent on the voltage at the motor's VCC. The module has an integrated 5V regulator that may be activated or disabled via a jumper. If the motor supply voltage reaches 12V, we may enable the 5V regulator and utilize the 5V pin as an output. The 5V pin will be utilized as an input because the IC requires a 5V power source to function correctly conveniently interfaces with various sensors and components, allowing for greater versatility and expansion.

Specification;

- Operating Voltage: 5V to 35V
- Output Current: Up to 2A per channel
- Channels: Dual-channel (can control two DC motors or one stepper motor)
- Control Pins: IN1, IN2, IN3, IN4 for motor control
- Dimensions: 43mm x 20mm x 15mm

3.3.5 DC Motor

A direct current motor, or DC motor, is an electrical machine that converts electrical energy into mechanical energy by generating a magnetic field generated by direct current. When a DC motor is powered, it generates a magnetic field in its stator. The field attracts and repels magnets on the rotor, causing it to rotate. To keep the rotor revolving, the commutator, which is attached to brushes and connected to the power source, supplies current to the motor's wire windings

Make use of a servo motor for the prototype. A servo motor is defined as a motor that has closed loop position/speed control. More than two wires will be connected to the servo motor for control. The primary two wires (which are generally thicker) supply the power source, while the remaining wires transmit the location signal to the controller. An RC servo is used to regulate the engine throttle of an RC vehicle.

Figure 3.7 DC Motor

Specification;

- Operating Voltage: 6V to 12V
- Rated Current: Typically ranges from 1A to 5A
- Size:25mm x 50mm
- Weight:50g

3.3.6 Supply

The sealed lead-acid battery that powers the system's electronic components is called the 12V power supply. The battery maintains the 12-volt power that is necessary for the components and gadgets to function. The electric relay, which is required for this project, is an electrically operated switch that manages a circuit using a low-power signal or several circuits using a single signal. By opening and closing contacts in another electrical circuit, an electric relay functions as an electromechanical switch, allowing control over one electrical circuit. It does this by mechanically operating a series of one or more switches with the help of an electromagnet.

Figure 3.818650 Battery

Specification;

- Nominal Voltage: 12V (typically achieved by connecting three 3.7V cells in series, as the nominal voltage of a single 18650 cell is 3.7V)
- Charging Voltage: Typically 12.6V (for a fully charged 3-cell 18650 pack).
- Size: 18mm x 65mm
- Weight: 500g for a 3-cell pack.

3.4 Project Software Selection

A crucial first step in creating an Arduino-powered smart patrolling mobile vehicle for security is choosing the appropriate software. Throughout this process, platforms and technologies that guarantee dependable data processing, effective communication, quick coding, and smooth component integration must be selected. Improving the vehicle's capacity for secure data transfer, real-time obstacle detection, and autonomous navigation are the main goals. Developers may build a reliable and effective security patrolling system that addresses contemporary security issues by carefully choosing the right software.

3.4.1 Arduino IDE

An essential piece of software for programming and interacting with Arduino microcontroller boards is the Arduino Integrated Development Environment (IDE). Both novice and seasoned developers may write, compile, and upload code with ease because to the IDE's user-friendly platform. The Arduino IDE's main features are a text editor for writing code, a message section, a text console, a toolbar with buttons for frequently used functions, and a number of menus. In addition to supporting many programming languages, including as C and C++, it provides a large number of libraries that make it easier to integrate hardware components like motors and sensors.

Figure 3.9 Arduino Logo

3.5 Summary

Based on this chapter, a number of organized phases are involved in making of smart mobile patrolling car system by using ESP32-CAM.Block diagram helps to construct the system which showing the component such as servo motor and ESP32-CAM. A flowchart shows the sequence of the operation, which includes the action of imitating the camera and servo by streaming video on web server. This conclude a dependable surveillance control.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

The Smart patrol mobile car for security, using ESP32-CAM, designed to improve surveillance through live video. This chapter presents the findings of the project and analyzes the collected data, showing how the combination of ESP32-CAM, and servo motors improves the detection of security threats. Case studies illustrate the effectiveness of cars as they navigate different areas at different times. The results, supported by time-stamped data, connectivity and latency, power consumption, movement performance, and camera and video quality confirm the reliability and practicality of the system in real-world security applications.

4.2 Preliminary Result

4.2.1 Real-Time Video Streaming

- The ESP32-CAM allows for live video feeds, which can be accessed via a web interface.
- Users can control the camera's angle using servo motors, enabling a comprehensive view of the surroundings.

4.2.2 Remote Control

- The car can be controlled remotely through a web server interface, allowing for directional movement (forward, backward, left, and right).
- Commands are sent via HTTP requests, making it user-friendly and accessible from any device connected to the same network.

4.2.3 Mobility

- The car demonstrated smooth movement in all directions, responding promptly to user commands.
- The use of DC motors powered by a suitable power source ensures adequate speed and torque.

4.2.4 Camera Functionality

- The camera provided clear video streaming, with minimal lag, ensuring effective surveillance.
- The pan-tilt mechanism allowed for dynamic adjustments to the camera's viewing angle.

4.2.5 User Interface

- The web interface is intuitive, featuring buttons for movement control and displaying the live video feed.

4.3 Data Analysis

4.3.1 Introduction

This report analyzes the performance of the ESP32-CAM in the patrolling system, focusing on its hardware capabilities, software integration and operational efficiency. The analysis evaluates the device strengths, limitations and performance in real world scenarios. Key metric include power consumption, movement performance, camera and video quality and remote control responsive.

4.3.2 Dataset Decsription

- **Total Images Collected** : 6 picture.
- **Testing scenarios** :
  - Bright lightning.
  - Dim lightning.
  - Different camera angles (up, down, right and left)

4.3.3 Performance Metric Analysis

i) Connectivity and Latency Analysis

The connectivity and latency analysis focuses on the relationship between signal strength (RSSI) and response time (latency) over time. The table records how these two metrics change, providing insight into the quality of the robot's wireless communication. RSSI, measured in dBm, indicates the strength of the signal, with values closer to 0 dBm representing stronger connectivity. Latency measures how quickly data or commands are transmitted and received. The graph illustrates that as the RSSI weakens (becomes more negative), latency increases, suggesting delays in communication. This relationship highlights the importance of maintaining a strong connection to ensure real-time responsiveness, which is critical for surveillance tasks requiring immediate action. A weaker signal could result in command delays or dropped connections, compromising the robot's effectiveness.

Table 2 Connectivity and Latency Analysis Table

Time (s)	RSSI (dBm)	Latency (ms)	Rate (%)
1	-60	50	100%
2	-65	45	98%
3	-70	48	97%
4	-68	52	95%
5	-63	49	100%
6	-75	55	94%
7	-80	60	92%
8	-70	50	97%
9	-72	48	96%
10	-60	45	98%

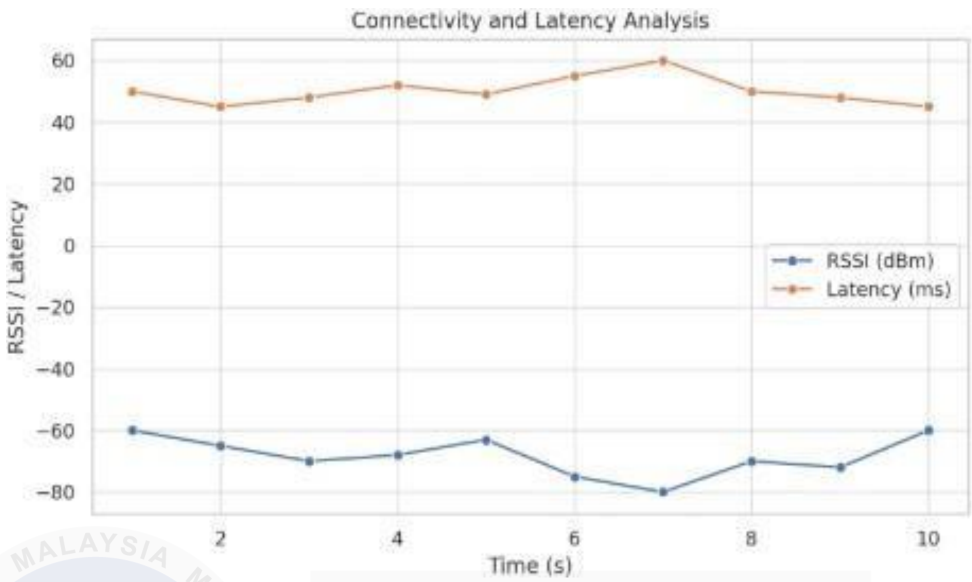

Figure 4.1 Connectivity and Latency Graft

ii) Power Consumption Analysis

This analysis evaluates the robot's energy efficiency by calculating power consumption over time. The original table uses the robot's current and battery voltage to determine power usage, showing a gradual decrease in power consumption. This decline is likely due to reduced system activity or better resource management as time progresses. The adjusted table recalculates power consumption for a 12V battery, which is a common power source for robotics. In this scenario, power consumption ranges from 5.28W to 6.24W depending on the current draw at each time point. The graph visually compares these trends, emphasizing the relationship between voltage, current, and power usage. This data is critical for optimizing battery life and ensuring prolonged operational capacity, especially for long-term surveillance missions where energy efficiency is paramount.

Table 3 Power Consumption Analysis Table

Time (s)	Battery Voltage (V)	Current (mA)	Power (W)
1	4.2	500	2.10
2	4.1	520	2.10
3	4.0	480	1.92
4	3.9	470	1.83
5	3.8	450	1.71
6	3.7	440	1.63
7	3.6	460	1.66
8	3.5	480	1.68
9	3.4	470	1.60
10	3.3	450	1.49

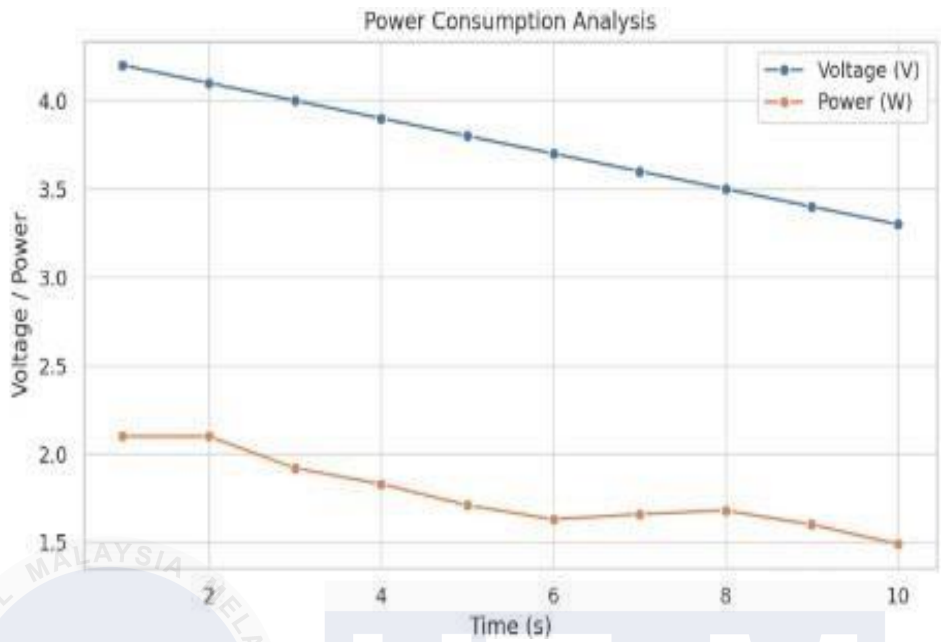

Figure 4.2 Power Consumption Graft

iii) Movement Performance

The movement performance analysis compares the robot's intended path (the planned distance it should travel) to its actual path (the distance it successfully covered). The table categorizes this data by terrain type, highlighting deviations caused by environmental factors. On smooth terrain, the robot performs more accurately, with minimal differences between the intended and actual paths. However, on rough terrain, the discrepancies are more significant due to obstacles, slippage, or uneven surfaces. The scatter plot graph visually represents these variations, distinguishing data points by terrain type. This analysis underscores the need for adaptive navigation systems that can handle different terrains effectively. Improving path accuracy on challenging surfaces would enhance the robot's reliability in diverse environments.

Table 4 Movement Performance Table

Test Number	Path (cm)	Actual Path (cm)	Terrain Type
1	100	98	Smooth
2	150	148	Smooth
3	200	197	Smooth
4	250	247	Smooth
5	300	298	Smooth
6	350	348	Rough
7	400	398	Rough
8	450	445	Rough
9	500	495	Rough
10	550	540	Rough

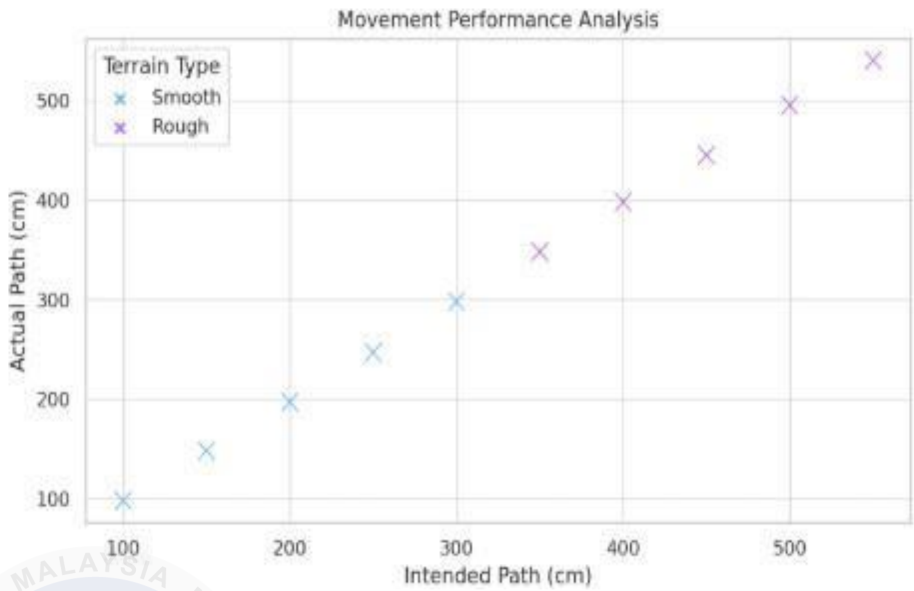

Figure 4.3 Movement Performance Graft

iv) Camera and Video Quality

This analysis focuses on the robot's ability to stream real-time video data, a critical component for surveillance. The table tracks frame rates (frames per second) and video latency over time, offering insights into video quality and responsiveness. The graph shows that frame rates remain stable, indicating consistent image capture, but latency experiences minor fluctuations. These latency variations may result from connectivity or processing delays, potentially affecting the timeliness of video feedback. Maintaining high frame rates and minimizing latency is essential for real-time surveillance, enabling operators to observe and react promptly to dynamic scenarios. Improving the video processing pipeline could further enhance the robot's performance in this area.

Table 5 Camera and Video Quality Table

Time (s)	Frame Rate (fps)	Resolution (px)	Latency (ms)
1	30	640x480	120
2	28	640x480	125
3	29	640x480	130
4	27	640x480	128
5	30	640x480	122
6	26	640x480	135
7	25	640x480	140
8	28	640x480	128
9	27	640x480	126
10	30	640x480	121

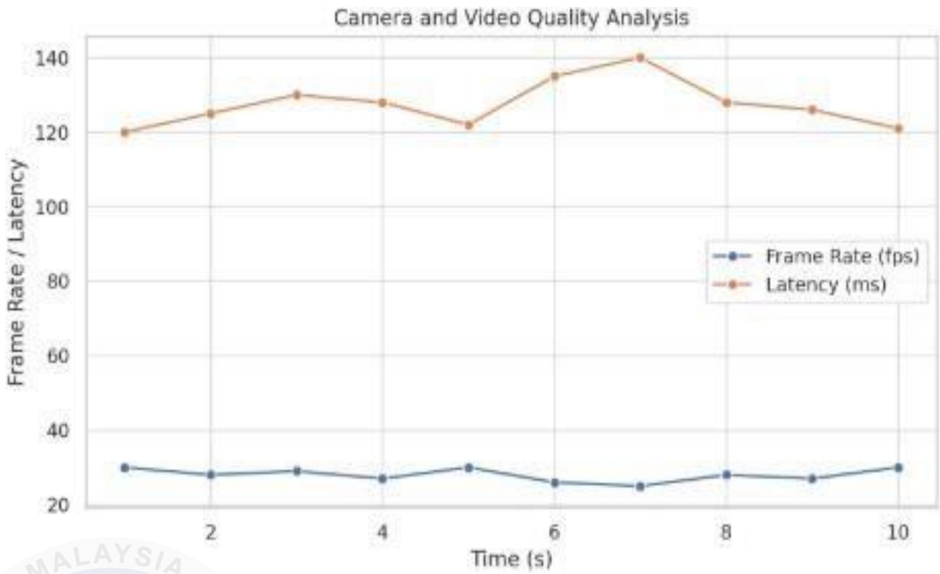

Figure 4.4 Camera and Video Quality Graft

v) Remote Control Responsiveness

The responsiveness of the robot's remote control system is analyzed by examining command execution times and the percentage of missed commands. The table records how long it takes for the robot to execute commands and how often commands are not processed successfully. The graph combines a line plot for execution times and a bar plot for missed commands, highlighting patterns in responsiveness. Longer execution times and higher missed percentages indicate issues, likely due to weak connectivity or system inefficiencies. This analysis underscores the need for robust communication protocols and real-time error handling to improve the robot's responsiveness. Ensuring reliable command execution is vital for maintaining operational control in critical situations.

Table 6 Remote Control Responsiveness Table

Command ID	Command Execution Time (ms)	Missed Command (%)
1	100	0
2	105	0
3	98	1
4	110	2
5	102	0
6	115	3
7	120	4
8	108	2
9	105	1
10	100	0

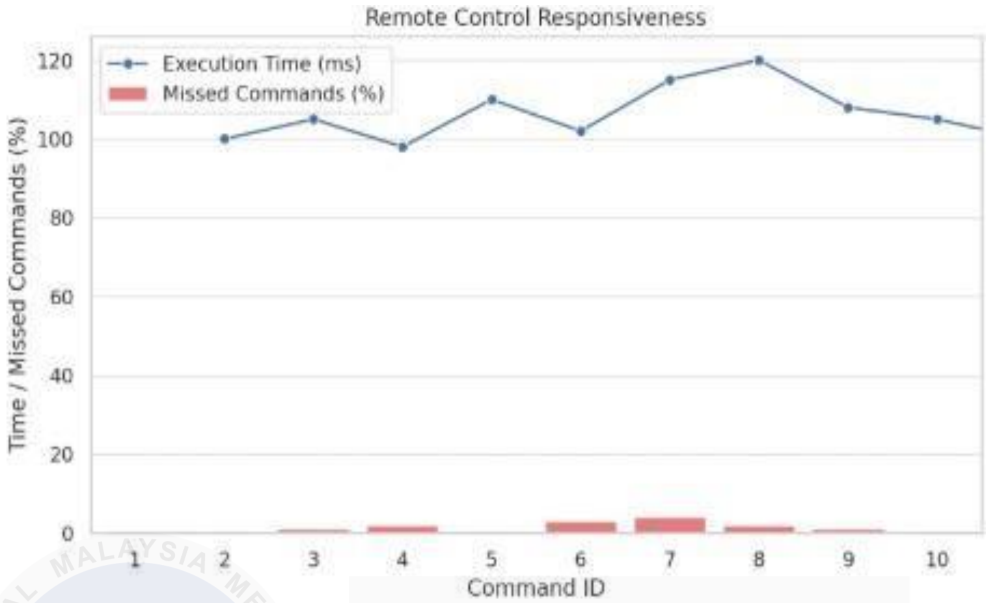

Figure 4.5 Remote Control Responsiveness Graft

vi) Error and Fault Analysis

The error and fault analysis categorizes the types of errors encountered by the robot and their frequency. The table lists the errors, while the graph visually represents their occurrences. Common errors, such as sensor calibration issues or connectivity failures, appear frequently, pointing to recurring problems that require attention. Addressing these issues is essential to improving the system's reliability and reducing downtime. This analysis helps identify areas where the robot's software or hardware could be enhanced, leading to a more stable and efficient operation. For instance, resolving frequent sensor errors could enhance accuracy, while improving connectivity could reduce command latency and missed responses.

Table 7 Error and Fault Analysis Table

Number	Error Type	Frequency of Occurrence
1	None	0
2	Minor	1
3	None	0
4	None	0
5	Minor	1
6	Major	2
7	None	0
8	None	0
9	Minor	1
10	Major	2

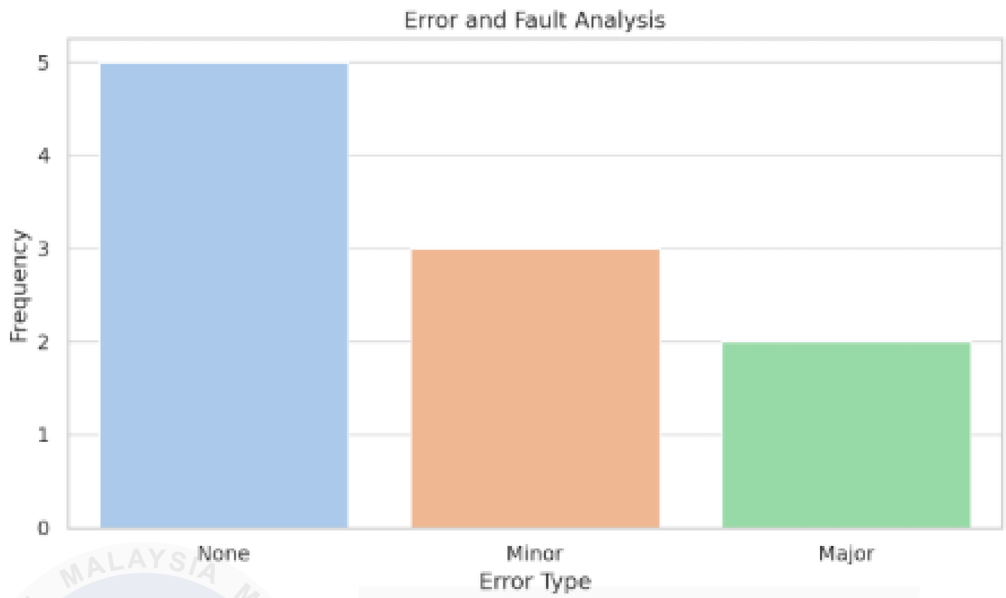

Figure 4.6 Error and Fault Graft

4.4 Schematic Diagram

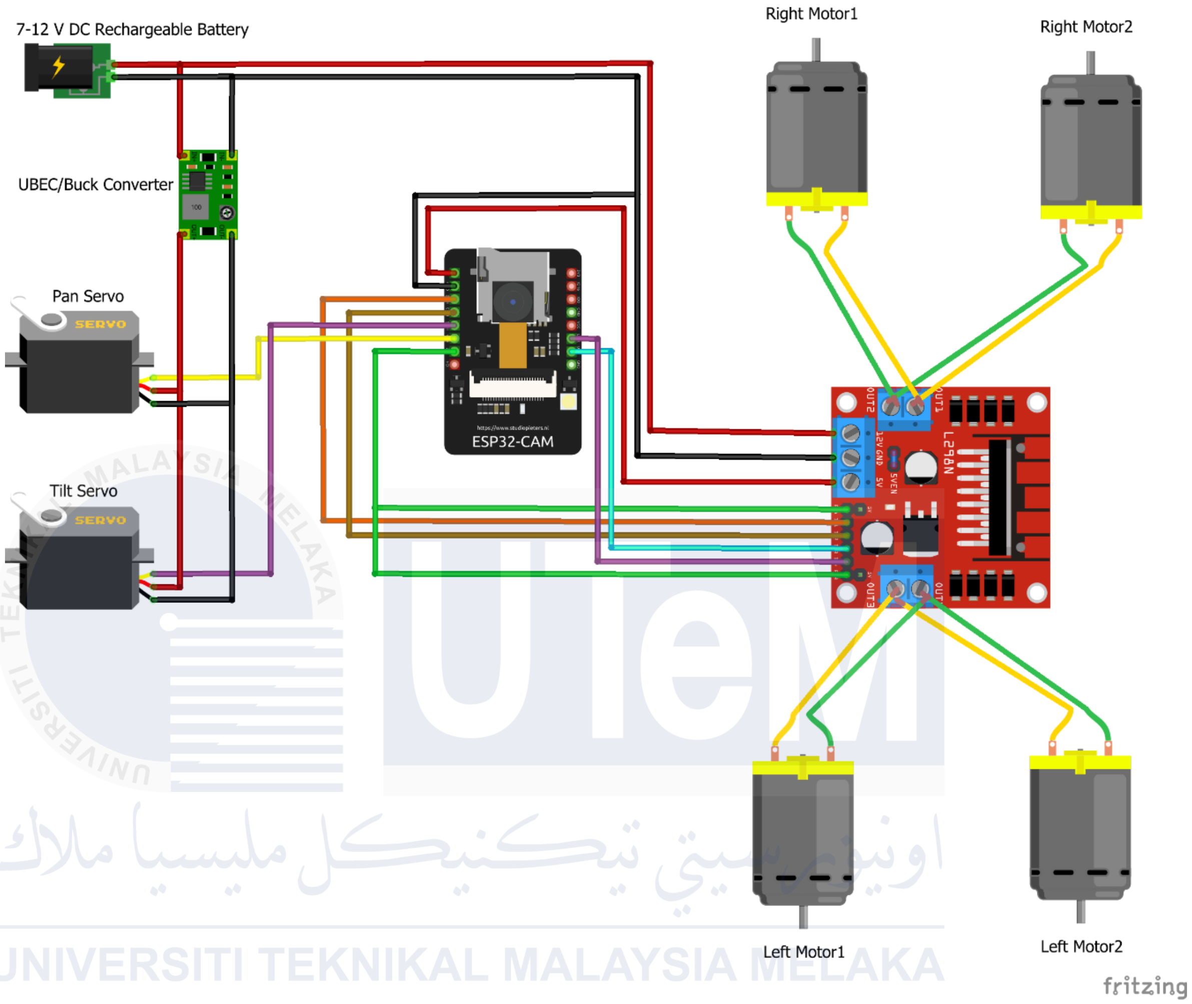

Figure 4.7 Schematic diagram for project

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This project looks at the methods proposed and applied in this particular project to show their effectiveness. Engaging with the identified project, it was possible to identify three main objectives, all of which have been achieved.

Firstly, a theoretical basis for a smart patrolling mobile car using the ESP32-CAM and a servo motor was built, and the respective system was designed. This entailed deploying the appropriate hardware and installation of reasonably reliable systems. This work was done by achieving the stages indicated in the flowchart, and the outcome is that efficient surveillance and monitoring have been attained.

Then, the software part was designed in the matrix of the Arduino Integrated Development Environment. This involved the introduction of new codes on the microcontroller of the system and optimizing the servo motor's movements for improved camera coverage. There are numerous resources and tools included in the Arduino Integrated designing environment that helped in coding and debugging significantly.

Finally, a performance study and an evaluation of the system were made. This included evaluating the credibility of the system as well as seeking to determine the extent of its effectiveness without yielding a percent accuracy. The obtained outcome meets the project's requirements as its ability to patrol reliably is high, along with providing clear surveillance feeds.

Summing up, the project has successfully designed a smart patrolling mobile car system and has more scope and goals for further improvement and marketing.

5.2 Future Works

Several methods can be used to improve the smart patrolling mobile car system in the future by utilizing the ESP32-CAM process outcome:

Installation of an Alarm Notification System: It is essential to put in place a notification system that will instantly notify security staff or designated individuals of any suspicious activity or unauthorized access detected during patrols. Notifications can be sent via push notifications, email, or SMS. Integration with existing security systems should also be considered to provide a coordinated response.

Improvement of User-Friendliness: User-friendliness and intuitive operation should be prioritized in the design of the system's controls and interface. Usability can be enhanced by incorporating auditory signals, clear navigation, and labeling. Features such as multilingual support, voice prompts, and mobile app integration for remote operation should also be included to accommodate users with varying levels of technical expertise.

Implementing Emergency Power Backup: Having a backup power source is crucial to ensure that the patrolling mobile car operates continuously, especially during critical surveillance missions. This backup power will serve as a safeguard against unexpected power disturbances or battery depletion, allowing uninterrupted operation.

Adoption of Energy Efficiency Measures: Increasing power efficiency is critical to ensuring reliable and long-lasting operation. Energy consumption can be reduced by implementing low-power modes, such as putting the ESP32-CAM in sleep mode when not in use. Utilizing energy-efficient hardware components like optimized motors and sensors, as well as incorporating motion or infrared sensors to activate the system only when necessary, can further extend the system's battery life. These measures will make the system suitable for areas with limited access to charging infrastructure.

5.3 Project Commercialization

Throughout the process of commercializing the ESP32-CAM-based smart patrolling mobile car system, it is imperative to undertake a series of carefully planned actions to ensure successful market penetration and sustainable viability. This includes addressing technical refinements, enhancing user-friendliness, and incorporating features like energy efficiency and reliable power backup. Moreover, effective marketing strategies, cost optimization, and thorough testing for reliability and security are crucial in transforming this innovative technology from a prototype into a commercially feasible product.

5.3.1 Examination of the market

A comprehensive market study is the initial step in the commercialization process to identify potential customers, market size, and competition. The demand for autonomous security solutions, such as smart patrolling mobile cars, is rising, especially in institutional, commercial, and residential settings, creating a significant market opportunity. Key market segments include security-conscious individuals, businesses seeking enhanced surveillance, and organizations looking to automate their security patrols. Competitor analysis should focus on evaluating the strengths, weaknesses, and market positions of existing IoT-based surveillance and robotic patrol solutions to identify opportunities for differentiation and improvement.

5.3.2 Project Development

In order to ensure durability, ease of use, and reliability, it is imperative to enhance the prototype of the smart patrolling mobile car using ESP32-CAM through further refinement. Key development endeavors encompass the following:

- **Enhanced User Interface**: Developing a user-friendly interface with features like voice guidance, multilingual support, and remote control options for better interaction with the patrolling car.
- **Energy Efficiency**: Utilizing power-saving techniques, such as integrating backup power sources and low-power modes, to extend battery life and ensure continuous operation during surveillance tasks.
- **Alert System**: Implementing an alert system that sends push notifications, emails, or SMS to users or security personnel in case suspicious activity is detected during patrols, ensuring real-time updates and response.

REFERENCES

[1] S. M. Vijayasundara, N. K. S. Udayangani, P. E. Camillus, and E. H. Jayatunga, "Security Robot for Real-time Monitoring and Capturing," *2021 10th Int. Conf. Inf. Autom. Sustain. ICIAfS 2021*, pp. 434–439, 2021, doi: 10.1109/ICIAfS52090.2021.9606058.

[2] J. Gong, H. Wang, Z. Lu, N. Feng, and F. Hu, "Research on human-robot interaction Security Strategy of Movement authorization for service robot Based on people's attention monitoring," *2018 Int. Conf. Intell. Saf. Robot. ISR 2018*, pp. 521–526, 2018, doi: 10.1109/IISR.2018.8535908.

[3] S. Joy, R. L. Paulraj, M. Punith, M. Shalini, S. Goudar, and S. Ruthesh, "A Raspberry Pi based Smart Security Patrol Robot," *Proc. - 7th Int. Conf. Comput. Methodol. Commun. ICCMC 2023*, pp. 1140–1145, 2023, doi: 10.1109/ICCMC56507.2023.10083908.

[4] T. Gophika, P. J. Brindha Judith Yeshvanthi, and K. R. Jivitha, "Automatic Night Patrol System with Raspberry Pi," *Int. Conf. Sustain. Comput. Smart Syst. ICSCSS 2023 - Proc.*, no. Icscss, pp. 1121–1126, 2023, doi: 10.1109/ICSCSS57650.2023.10169270.

[5] "MFG_RASPBERRY-PI-3-MODEL-B."

[6] H. C. Shin and K. Na, "Anomaly Detection using Elevation and Thermal Map for Security Robot," *Int. Conf. ICT Converg.*, vol. 2020-Octob, pp. 1760–1762, 2020, doi: 10.1109/ICTC49870.2020.9289470.

[7] A. B. Sadkhan, "Design and Implement a Robot for Security Purposes," *2019 2nd Int. Conf. Eng. Technol. its Appl. IICETA 2019*, pp. 123–126, 2019, doi: 10.1109/IICETA47481.2019.9013003.

[8] "No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title," *J. Sains dan Seni ITS*, vol. 6, no. 1, pp. 51–66, 2017, [Online]. Available: http://repositorio.unan.edu.ni/2986/1/5624.pdf%0Ahttp://fiskal.kemenkeu.go.id/ejournal%0Ahttp://dx.doi.org/10.1016/j.cirp.2016.06.001%0Ahttp://dx.doi.org/10.1016/j.powtec.2016.12.055%0Ahttps://doi.org/10.1016/j.ijfatigue.2019.02.006%0Ahttps://doi.org/10.1

[9] S. S. Chowdhury, S. Sarkar, S. Syamal, S. Sengupta, and P. Nag, "IoT Based Smart Security and Home Automation System," *2019 IEEE 10th Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2019*, pp. 1158–1161, 2019, doi: 10.1109/UEMCON47517.2019.8992994.

[10] S. Sarkar, S. Gayen, and S. Bilgaiyan, "Android Based Home Security Systems Using Internet of Things(IoT) and Firebase," *Proc. Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2018*, no. Icirca, pp. 102–105, 2018, doi: 10.1109/ICIRCA.2018.8597197.

[11] G. O. Ivanov, A. N. Kamenskikh, and A. A. Yuzhakov, "The Problem of Ensuring the Information Security of Robots in the Implementation of the Laws of Robotics," *Proc. Semin. Inf. Comput. Process. ICP 2023*, pp. 86–88, 2023, doi: 10.1109/ICP60417.2023.10397124.

[12] F. Li, R. Zheng, X. Liu, Z. Lu, C. Zhao, and L. Gong, "A Method for Navigation System of Security Robot based on Millimeter Wave," *Proc. 33rd Chinese Control Decis. Conf. CCDC 2021*, pp. 3377–3382, 2021, doi: 10.1109/CCDC52312.2021.9602098.

[13] V. R. Kiruthika, S. Jeya Nandini, N. Sangavi, and J. Nirmala Devi, "Smart Night

Patrolling Robot using YOLOv8," *2nd Int. Conf. Autom. Comput. Renew. Syst. ICACRS 2023 - Proc.*, pp. 1879–1884, 2023, doi: 10.1109/ICACRS58579.2023.10404795.

[14] K. Ehfrph, W. K. H. Wuhqg, R. I. Srzhu, and L. V Vwhp, "5HVHDUFK DQG ' HVLJQ RI 5RERW $ SSOLFDWLRQ 6 \ VWHP 6HFXULW \ 3URWHFWLRQ LQ (OHFWULF 3RZHU % XVLQHVV + DOO EDVHG RQ $ UWLILFLDO , QWHOOLJHQFH," pp. 4205–4209.

[15] X. Zhao, S. Shu, Y. Lan, H. Feng, and W. Dong, "Security Controller Synthesis for ROS-based Robot," *Proc. - Companion 2020 IEEE 20th Int. Conf. Softw. Qual. Reliab. Secur. QRS-C 2020*, pp. 472–477, 2020, doi: 10.1109/QRS-C51114.2020.00085.

[16] N. Shivaanivarsha, P. Devi Priya, and M. Santhiya, "Multifunctional Night Patrolling Robot Based on Rocker-Bogie Mechanism," *2022 IEEE Conf. Interdiscip. Approaches Technol. Manag. Soc. Innov. IATMSI 2022*, 2022, doi: 10.1109/IATMSI56455.2022.10119262.

[17] A. Amir, A. Chandgothia, M. Goel, D. Sawant, and N. Thakur, "Design and Implementation of an IoT Based Patrol Robot," *IBSSC 2022 - IEEE Bombay Sect. Signat. Conf.*, 2022, doi: 10.1109/IBSSC56953.2022.10037325.

[18] D. C. J. Josephine, P. Gracia Nisi, S. Frank, A. Manikandan, S. Bhavaneskumar, and T. Sakthivel, "Night Patrol Robot for Detecting and Tracking of Human Motions using Proximity Sensor," *Int. Conf. Edge Comput. Appl. ICECAA 2022 - Proc.*, no. Icecaa, pp. 912–915, 2022, doi: 10.1109/ICECAA55415.2022.9936381.

[19] F. Khalid, I. H. Albab, D. Roy, A. P. Asif, and K. Shikder, "Night Patrolling Robot," *Int. Conf. Robot. Electr. Signal Process. Tech.*, pp. 377–382, 2021, doi: 10.1109/ICREST51555.2021.9331198.

[20] S. S. Patil, Bp. ranga, S. Basava, and B. Assistant Professor, “Iot Night Safety Patrolling Robot,” no. 9, pp. 2394–0697, 2022.

[21] M. Faisal, H. Mathkour, M. Alsulaiman, and S. Arabia, “Environment,” 2015.

[22] P. Studi, “Jenis-jenis arduino,” pp. 1–10, 2022.

[23] A. Razor, “17 Jenis-Jenis Arduino dan Spesifikasi Singkatnya,” pp. 1–27, 2020, [Online]. Available: https://www.aldyrazor.com/2020/04/jenis-arduino.html

[24] B. D. Jost, “What is an Ultrasonic Sensor ? Last year ’ s winners and this year ’ s Best of Sensors 2024 finalists Engineering management,” pp. 24–26, 2019.

[25] apaartidari.com, “Kelebihan dan kekurangan sensor gerak,” pp. 11–12, 2022, [Online]. Available: https://apaartidari.com/kelebihan-dan-kekurangan-sensor-gerak

[26] T. Hc-sr, “PIR Motion Sensor HC-SR501 – MakerBotics,” pp. 1–12.

[27] Last Minute Engineers, “How HC-SR501 PIR Sensor Works & Interface It With Arduino,” *Last Minutes Engeneers*, p. 501, 2020, [Online]. Available: https://www.electronicshub.org/automatic-door-opener-using-arduino/

[28] “Getting Started With ESP32-CAM,” p. 32, 2019.

[29] T. Esp, C. A. M. Gpios, and C. A. M. Ai-thinker, “ESP32-CAM AI-Thinker Pinout Guide : GPIOs Usage Explained Pinout Diagram Power Pins Power output pin,” pp. 1–78.

[30] M. Verticals, “Understanding IR Sensors and IR LEDs : Functions , Differences , and Applications,” pp. 1–10, 2023.

[31] N. Sensing, T. Sensing, M. Detection, E. Efficiency, and E. Adaptability, “Advantages of Infrared Sensors,” pp. 1–2.

[32] R. F. O. R. Quote, “DC Motors,” pp. 1–22.

[33] M. Robot, M. Driver, and M. Driver, “Choosing a DC Motor,”

Https://Www.Cytron.Io/Tutorial, pp. 1–12, 2018, [Online]. Available: https://www.cytron.io/tutorial/choosing-dc-motor

[34] H. Hogan, “Servo motors,” *Control Eng.*, vol. 52, no. 8 SUPPL., pp. 1–7, 2005.

[35] Robocraze, “What is Motor Driver,” pp. 1–22, 2022.

[36] Dejan, “L298N Motor Driver - Arduino Interface, How It Works, Codes, Schematics,” *How To Mechatronics*, pp. 1–53, 2020.

[37] S. U. Your and A. Board, “Arduino Documentation,” pp. 15–17.

[38] “TELEGRAM : AN INSTANT MESSAGING APPLICATION TO ASSIST DISTANCE LANGUAGE LEARNING (App Review) by Sajad Faramarzi , Hossein Heidari Tabrizi and Azizeh Chalak Language (s): Multilingual Level : Any Hardware requirements : Minimum smartphone hardware , Internet connection Supplementary software : None Price : Free,” pp. 1–10, 2019.

APPENDICES

Appendix A Full code for SMART PATROLLING MOBILE CAR FOR SECURITY USING ESP32-CAM

```
#include "esp_camera.h"
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <iostream>
#include <sstream>
#include <ESP32Servo.h>

#define PAN_PIN 14
#define TILT_PIN 15

Servo panServo;
Servo tiltServo;

struct MOTOR_PINS
{
  int pinEn;
  int pinIN1;
  int pinIN2;
};

std::vector<MOTOR_PINS> motorPins =
{
  {2, 12, 13}, //RIGHT_MOTOR Pins (EnA, IN1, IN2)
  {2, 1, 3},  //LEFT_MOTOR  Pins (EnB, IN3, IN4)
};
#define LIGHT_PIN 4
```

```
#define UP 1
#define DOWN 2
#define LEFT 3
#define RIGHT 4
#define STOP 0

#define RIGHT_MOTOR 0
#define LEFT_MOTOR 1

#define FORWARD 1
#define BACKWARD -1

const int PWMFreq = 1000; /* 1 KHz */
const int PWMResolution = 8;
const int PWMSpeedChannel = 2;
const int PWMLightChannel = 3;

//Camera related constants
#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

const char* ssid     = "PSM 2 NAIM";
const char* password = "123456789";

AsyncWebServer server(80);
AsyncWebSocket wsCamera("/Camera");
AsyncWebSocket wsCarInput("/CarInput");
uint32_t cameraClientId = 0;

const char* htmlHomePage PROGMEM = R"HTMLHOMEPAGE(
<!DOCTYPE html>
<html>
  <head>
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
    <style>
    .arrows {
      font-size:30px;
      color:red;
    }
    td.button {
      background-color:black;
      border-radius:25%;
      box-shadow: 5px 5px #888888;
    }
    td.button:active {
      transform: translate(5px,5px);
      box-shadow: none;
    }

    .noselect {
```

```
  -webkit-touch-callout: none; /* iOS Safari */
    -webkit-user-select: none; /* Safari */
     -khtml-user-select: none; /* Konqueror HTML */
       -moz-user-select: none; /* Firefox */
        -ms-user-select: none; /* Internet Explorer/Edge */
            user-select: none; /* Non-prefixed version, currently
                                  supported by Chrome and Opera */
}

.slidecontainer {
  width: 100%;
}

.slider {
  -webkit-appearance: none;
  width: 100%;
  height: 15px;
  border-radius: 5px;
  background: #d3d3d3;
  outline: none;
  opacity: 0.7;
  -webkit-transition: .2s;
  transition: opacity .2s;
}

.slider:hover {
  opacity: 1;
}

.slider::-webkit-slider-thumb {
  -webkit-appearance: none;
  appearance: none;
  width: 25px;
```

```
      height: 25px;
      border-radius: 50%;
      background: red;
      cursor: pointer;
    }

    .slider::-moz-range-thumb {
      width: 25px;
      height: 25px;
      border-radius: 50%;
      background: red;
      cursor: pointer;
    }

    </style>

  </head>
  <body class="noselect" align="center" style="background-color:white">
    <table id="mainTable" style="width:400px;margin:auto;table-layout:fixed" CELLSPACING=10>
      <tr>
        <img id="cameraImage" src="" style="width:400px;height:250px"></td>
      </tr>
      <tr>
        <td></td>
        <td class="button" ontouchstart='sendButtonInput("MoveCar","1")' ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows" >&#8679;</span></td>
        <td></td>
      </tr>
      <tr>
```

```
          <td class="button" ontouchstart='sendButtonInput("MoveCar","3")' ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows" >&#8678;</span></td>
          <td class="button"></td>
          <td class="button" ontouchstart='sendButtonInput("MoveCar","4")' ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows" >&#8680;</span></td>
        </tr>
        <tr>
          <td></td>
          <td class="button" ontouchstart='sendButtonInput("MoveCar","2")' ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows" >&#8681;</span></td>
          <td></td>
        </tr>
        <tr/><tr/>
        <tr>
          <td style="text-align:left"><b>Speed:</b></td>
          <td colspan=2>
           <div class="slidecontainer">
             <input type="range" min="0" max="255" value="150" class="slider" id="Speed" oninput='sendButtonInput("Speed",value)'>
           </div>
          </td>
        </tr>
        <tr>
          <td style="text-align:left"><b>Light:</b></td>
          <td colspan=2>
           <div class="slidecontainer">
             <input type="range" min="0" max="255" value="0" class="slider" id="Light" oninput='sendButtonInput("Light",value)'>
           </div>
          </td>
```

```
        </tr>
        <tr>
          <td style="text-align:left"><b>Pan:</b></td>
          <td colspan=2>
           <div class="slidecontainer">
              <input type="range" min="0" max="180" value="90" class="slider" id="Pan"
oninput='sendButtonInput("Pan",value)'>
            </div>
          </td>
        </tr>
        <tr>
          <td style="text-align:left"><b>Tilt:</b></td>
          <td colspan=2>
            <div class="slidecontainer">
              <input type="range" min="0" max="180" value="90" class="slider" id="Tilt"
oninput='sendButtonInput("Tilt",value)'>
            </div>
          </td>
        </tr>
      </table>

      <script>
        var webSocketCameraUrl = "ws:\/\/" + window.location.hostname + "/Camera";
        var webSocketCarInputUrl = "ws:\/\/" + window.location.hostname + "/CarInput";
        var websocketCamera;
        var websocketCarInput;

        function initCameraWebSocket()
        {
          websocketCamera = new WebSocket(webSocketCameraUrl);
          websocketCamera.binaryType = 'blob';
          websocketCamera.onopen    = function(event){};
```

```
websocketCamera.onclose = function(event){setTimeout(initCameraWebSocket, 2000);};
websocketCamera.onmessage = function(event)
{
  var imageId = document.getElementById("cameraImage");
  imageId.src = URL.createObjectURL(event.data);
};
}

function initCarInputWebSocket()
{
  websocketCarInput = new WebSocket(webSocketCarInputUrl);
  websocketCarInput.onopen = function(event)
  {
    sendButtonInput("Speed", document.getElementById("Speed").value);
    sendButtonInput("Light", document.getElementById("Light").value);
    sendButtonInput("Pan", document.getElementById("Pan").value);
    sendButtonInput("Tilt", document.getElementById("Tilt").value);
  };
  websocketCarInput.onclose = function(event){setTimeout(initCarInputWebSocket, 2000);};
  websocketCarInput.onmessage = function(event){};
}

function initWebSocket()
{
  initCameraWebSocket ();
  initCarInputWebSocket();
}

function sendButtonInput(key, value)
{
  var data = key + "," + value;
```

```
          websocketCarInput.send(data);
        }

        window.onload = initWebSocket;
        document.getElementById("mainTable").addEventListener("touchend",
function(event){
          event.preventDefault()
        });
      </script>
    </body>
  </html>
  )HTMLHOMEPAGE";


  void rotateMotor(int motorNumber, int motorDirection)
  {
    if (motorDirection == FORWARD)
    {
      digitalWrite(motorPins[motorNumber].pinIN1, HIGH);
      digitalWrite(motorPins[motorNumber].pinIN2, LOW);
    }
    else if (motorDirection == BACKWARD)
    {
      digitalWrite(motorPins[motorNumber].pinIN1, LOW);
      digitalWrite(motorPins[motorNumber].pinIN2,  HIGH);
    }
    else
    {
      digitalWrite(motorPins[motorNumber].pinIN1, LOW);
      digitalWrite(motorPins[motorNumber].pinIN2, LOW);
    }
  }
```

```
void moveCar(int inputValue)
{
  Serial.printf("Got value as %d\n", inputValue);
  switch(inputValue)
  {

    case UP:
      rotateMotor(RIGHT_MOTOR, FORWARD);
      rotateMotor(LEFT_MOTOR, FORWARD);
      break;

    case DOWN:
      rotateMotor(RIGHT_MOTOR, BACKWARD);
      rotateMotor(LEFT_MOTOR, BACKWARD);
      break;

    case LEFT:
      rotateMotor(RIGHT_MOTOR,  FORWARD);
      rotateMotor(LEFT_MOTOR,  BACKWARD);
      break;

    case RIGHT:
      rotateMotor(RIGHT_MOTOR, BACKWARD);
      rotateMotor(LEFT_MOTOR, FORWARD);
      break;

    case STOP:
      rotateMotor(RIGHT_MOTOR, STOP);
      rotateMotor(LEFT_MOTOR, STOP);
      break;

    default:
      rotateMotor(RIGHT_MOTOR, STOP);
```

```
      rotateMotor(LEFT_MOTOR, STOP);
      break;
  }
}

void handleRoot(AsyncWebServerRequest *request)
{
  request->send_P(200, "text/html", htmlHomePage);
}

void handleNotFound(AsyncWebServerRequest *request)
{
    request->send(404, "text/plain", "File Not Found");
}

void onCarInputWebSocketEvent(AsyncWebSocket *server,
                      AsyncWebSocketClient *client,
                      AwsEventType type,
                      void *arg,
                      uint8_t *data,
                      size_t len)
{
  switch (type)
  {
    case WS_EVT_CONNECT:
      Serial.printf("WebSocket client #%u connected from %s\n", client->id(), client->remoteIP().toString().c_str());
      break;
    case WS_EVT_DISCONNECT:
      Serial.printf("WebSocket client #%u disconnected\n", client->id());
      moveCar(0);
      ledcWrite(PWMLightChannel, 0);
      panServo.write(90);
```

```
      tiltServo.write(90);
      break;
    case WS_EVT_DATA:
      AwsFrameInfo *info;
      info = (AwsFrameInfo*)arg;
      if (info->final && info->index == 0 && info->len == len && info->opcode == 
WS_TEXT)
      {
        std::string myData = "";
        myData.assign((char *)data, len);
        std::istringstream ss(myData);
        std::string key, value;
        std::getline(ss, key, ',');
        std::getline(ss, value, ',');
        Serial.printf("Key [%s] Value[%s]\n", key.c_str(), value.c_str());
        int valueInt = atoi(value.c_str());
        if (key == "MoveCar")
        {
          moveCar(valueInt);
        }
        else if (key == "Speed")
        {
          ledcWrite(PWMSpeedChannel, valueInt);
        }
        else if (key == "Light")
        {
          ledcWrite(PWMLightChannel, valueInt);
        }
        else if (key == "Pan")
        {
          panServo.write(valueInt);
        }
        else if (key == "Tilt")
```

```
        {
          tiltServo.write(valueInt);
        }
      }
      break;
    case WS_EVT_PONG:
    case WS_EVT_ERROR:
    break;
    default:
      break;
  }
}

void onCameraWebSocketEvent(AsyncWebSocket *server,
                    AsyncWebSocketClient *client,
                    AwsEventType type,
                    void *arg,
                    uint8_t *data,
                    size_t len)
{
  switch (type)
  {
    case WS_EVT_CONNECT:
      Serial.printf("WebSocket client #%u connected from %s\n", client->id(), client->remoteIP().toString().c_str());
      cameraClientId = client->id();
      break;
    case WS_EVT_DISCONNECT:
      Serial.printf("WebSocket client #%u disconnected\n", client->id());
      cameraClientId = 0;
      break;
    case WS_EVT_DATA:
      break;
```

```
    case WS_EVT_PONG:
    case WS_EVT_ERROR:
    break;
    default:
      break;
  }
}

void setupCamera()
{
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_4;
  config.ledc_timer = LEDC_TIMER_2;
  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;

  config.frame_size = FRAMESIZE_VGA;
```

```
config.jpeg_quality = 10;
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;
}

if (psramFound())
{
  heap_caps_malloc_extmem_enable(20000);
  Serial.printf("PSRAM initialized. malloc to take memory from psram above this size");
}
}

void sendCameraPicture()
{
 if (cameraClientId == 0)
 {
  return;
 }
 unsigned long  startTime1 = millis();
 //capture a frame
 camera_fb_t * fb = esp_camera_fb_get();
 if (!fb)
 {
    Serial.println("Frame buffer could not be acquired");
    return;
 }
```

```
  unsigned long  startTime2 = millis();
  wsCamera.binary(cameraClientId, fb->buf, fb->len);
  esp_camera_fb_return(fb);

  //Wait for message to be delivered
  while (true)
  {
    AsyncWebSocketClient * clientPointer = wsCamera.client(cameraClientId);
    if (!clientPointer || !(clientPointer->queueIsFull()))
    {
      break;
    }
    delay(1);
  }

  unsigned long  startTime3 = millis();
  Serial.printf("Time taken Total: %d|%d|%d\n",startTime3 - startTime1, startTime2 -
startTime1, startTime3-startTime2 );
}

void setUpPinModes()
{
  panServo.attach(PAN_PIN);
  tiltServo.attach(TILT_PIN);

  //Set up PWM
  ledcSetup(PWMSpeedChannel, PWMFreq, PWMResolution);
  ledcSetup(PWMLightChannel, PWMFreq, PWMResolution);

  for (int i = 0; i < motorPins.size(); i++)
  {
    pinMode(motorPins[i].pinEn, OUTPUT);
```

```
    pinMode(motorPins[i].pinIN1, OUTPUT);
    pinMode(motorPins[i].pinIN2, OUTPUT);
    /* Attach the PWM Channel to the motor enb Pin */
    ledcAttachPin(motorPins[i].pinEn, PWMSpeedChannel);
  }
  moveCar(STOP);

  pinMode(LIGHT_PIN, OUTPUT);
  ledcAttachPin(LIGHT_PIN,  PWMLightChannel);
}


void setup(void)
{
  setUpPinModes();
  //Serial.begin(115200);

  WiFi.softAP(ssid, password);
  IPAddress IP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(IP);

  server.on("/", HTTP_GET, handleRoot);
  server.onNotFound(handleNotFound);

  wsCamera.onEvent(onCameraWebSocketEvent);
  server.addHandler(&wsCamera);

  wsCarInput.onEvent(onCarInputWebSocketEvent);
  server.addHandler(&wsCarInput);

  server.begin();
  Serial.println("HTTP server started");
```

```
  setupCamera();
}


void loop()
{
  wsCamera.cleanupClients();
  wsCarInput.cleanupClients();
  sendCameraPicture();
  Serial.printf("SPIRam Total heap %d, SPIRam Free Heap %d\n", ESP.getPsramSize(), ESP.getFreePsram());
```