

WIRELESS VIDEO SURVEILLANCE ROBOT USING RASPBERRY PI

A Project report phase -II

Submitted in partial fulfilment of the
requirements for the award of
Bachelor of Engineering degree in Electronics and Communication Engineering

By

Sushma Sagarika Midathana (39130457)

Tara Umasai Uttara Tadi (39130466)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI – 600119**

APRIL – 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with – All Grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai 600 119
www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Final Year Project Report is the bonafide work of TARA UMASAI UTTARA TADI (REG NO: 39130466) AND SUSHMA SAGARIKA MIDATHANA (REG NO: 39130457) who have done the project work as an individual who carried out the project entitled “WIRELESS VIDEO SURVEILLANCE ROBOT USING RASPBERRY PI” under my supervision from Jan 2023 to April 2023.

Internal Guide

Dr. S. Emalda Roslin, M.E., Ph.D.

Head of the Department

Dr. T. Ravi, M.E., Ph.D.

Submitted for the viva voice held on _____
24.4.23

Internal Examiner

External Examiner

DECLARATION

I, **TARA UMASAI UTTARA TADI** (REG NO: 39130466) and **SUSHMA SAGARIKA MIDATHANA** (REG NO: 39130457), hereby declare that the Project Report on “**WIRELESS VIDEO SURVEILLANCE ROBOT USING RASPBERRY PI**” done by us under the guidance of **Dr. S. EMALDA ROSLIN** at Sathyabama Institute of Science and technology is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in **Electronics and Communication Engineering**.

DATE: 24.4.23

1.



2.



PLACE: Chennai

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. N.M. NANDHITHA M.E., Ph. D, Dean**, School of Electronics, **Dr. T. RAVI M.E., Ph.D.**, Head of the Department of Electronics and Communication Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. S. Emalda Roslin, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Robots are being used in variety of industrial applications for various activities like pulse detection, material handling and in hazardous places etc. Robots can be replaced by humans when the human life is at stake thus reducing the suffering of humans. Nowadays ensuring safety and security has become an inevitable essentially. Surveillance system, is being installed in every security critical area. These systems generally have high quality cameras, multiple computers for monitoring, servers for storing these videos and many security personnel for monitoring these videos. Tremendous development of the Technology yields the unassailable security systems. Video Surveillance is the process of monitoring a particular range of area, persons, etc. Security plays the most pre-eminent role in day-to-day life. This wireless video surveillance robot can be accessed easily anywhere. The video surveillance has played supreme role in the research of last few decades. The application has wide range of purposes like traffic monitoring, understanding human activity. It is known that, different kind of cameras is used for surveillance like fixed cameras, and pan and tilt cameras. These kinds of cameras are generally used for indoor security. In indoor security system multiple cameras are mounted on wall with different angels to track object. These types of systems need a computer or a laptop for monitoring. Nowadays, most of the system uses a mobile robot with a camera for surveillance. The camera mounted on the robot can move to different locations. These types of robots are more flexible than the fixed cameras. Also mostly used surveillance robots are wheel robot. The wheel-based robots are more suitable for flat platform. With the development in wireless communication and internet, the videos captured by wheel robot can be seen remotely on computer or laptop.

TABLE OF CONTENTS

S.No.	TITLE	Page No.
	ABSTRACT	lv
	LIST OF FIGURES	vii
1.	INTRODUCTION	1
1.1	Domain Introduction	1
1.2	Outline of The Project	11
1.3	Motivation	12
1.4	Objectives	13
1.5	Applications	14
2.	LITERATURE SURVEY	15
2.1	Overview of Literature Survey	15
2.2	Comparison of Existing Work	21
2.3	Open Problems in Existing System	23
2.4	Summary of Existing Work	25
3.	WIRELESS VIDEO SURVEILLANCE ROBOT	26
3.1	Block Diagram of the Proposed System	26
3.2	Requirements	28
3.2.1	Functional Requirements	28
3.2.2	Non-Functional Requirements	28
3.3	Software Requirements	29
3.3.1	Notepad++	29
3.3.2	C++ Language	30
3.4	Constraints	32
3.5	Standards	33
4.	RESULTS AND DISCUSSION	34
4.1	Transmitter Section	34
4.2	Receiver Section	35

4.3	Hardware Implementation	38
5.	CONCLUSION AND FUTURE WORK	39
5.1	Conclusion	39
5.2	Future Work	39
	REFERENCES	41
	APPENDIX	42

LIST OF FIGURES

Figure No.	Figure Name	Pg.no
1.1	Wireless communication framework	3
1.2	T.V and Radio Broadcasting	6
1.3	Satellite communication system	7
3.1	Block Diagram	26
3.2	Architecture of Proposed System	27
4.1	Hardware of the Robot	34
4.2	Robot	35
4.3	Device Connection	35
4.4	Captured Video	36
4.5	Receiver Device	37

CHAPTER 1

INTRODUCTION

1.1 DOMAIN INTRODUCTION

Wireless communication seems destined to make a large and continuing impact on our lives. Recent developments in wireless technologies provide a new channel for implementation of embedded systems with remote access for mobile and non-mobile products and services. Several wireless technologies are available with their own advantages and disadvantages. Virtually every electronic device designed and manufactured today, whether be a domestic, medical, consumer, industrial or military product, is an embedded system. Complexity in design varies from low, with a single microcontroller chip, to very high with multiple units, peripherals, and networks. In such a scenario wireless connectivity is one of the solutions that offers good amount of freedom and convenience. Short Range Wireless Systems are specialized programmable systems that are part of a larger system or a machine.

Wireless communication (or just wireless, when the context allows) is the transfer of information (*telecommunication*) between two or more points without the use of an electrical conductor, optical fibre or other continuous guided medium for the transfer. The most common wireless technologies use radio waves. With radio waves, intended distances can be short, such as a few meters for Bluetooth or as far as millions of kilometres for deep-space radio communications. It encompasses various types of fixed, mobile, and portable applications, including two-way radios, cellular telephones, personal digital assistants (PDAs), and wireless networking.

The most dynamic and rapidly expanding technology area in the communication industry is wireless communication. Information can be transmitted wirelessly from one location to another without using physical mediums like wires, cables, or other connections. In a communication system, information is typically sent across a short distance from a transmitter to a receiver. With wireless communication, the transmitter and receiver can be positioned anywhere from a few

meters (similar to a TV remote control) to a few thousand kilometres away (Satellite Communication).

Wireless Networks

The wireless communication revolution is bringing fundamental changes to data networking, telecommunication, and is making integrated networks a reality. By freeing the user from the cord, personal communications networks, wireless LAN's, mobile radio networks and cellular systems, harbour the promise of fully distributed mobile computing and communications, anytime, anywhere.

A wireless network is a computer network that uses wireless data connections between network nodes.

Wireless networking is a method by which homes, telecommunications networks and business installations avoid the costly process of introducing cables into a building, or as a connection between various equipment locations. Admin telecommunications networks are generally implemented and administered using radio communication. This implementation takes place at the physical level (layer) of the OSI model network structure.

Wireless Communication Domain

Wireless communications are the transmission of voice and data without cable or wires. In place of a physical connection, data travels through electromagnetic signals broadcast from sending facilities to intermediate and end-user devices. Several wireless techniques exist for short-range communication with a goal of delivering wireless connectivity. Following details provide an overview of some of the standard technologies for short-range wireless communication.

Bluetooth

Bluetooth is a high-speed, low-power microwave wireless link technology, intended to exchanging data over short distances from fixed and mobile devices, creating personal area networks (PANs). Bluetooth is omni-directional and does not necessitate line-of-sight positioning of connected devices. When Bluetooth-enabled devices come within range of another such devices, they instantly exchange address information and create small ad-hoc networks (known as piconet) among each other, without user involvement or initiative. Piconets are established

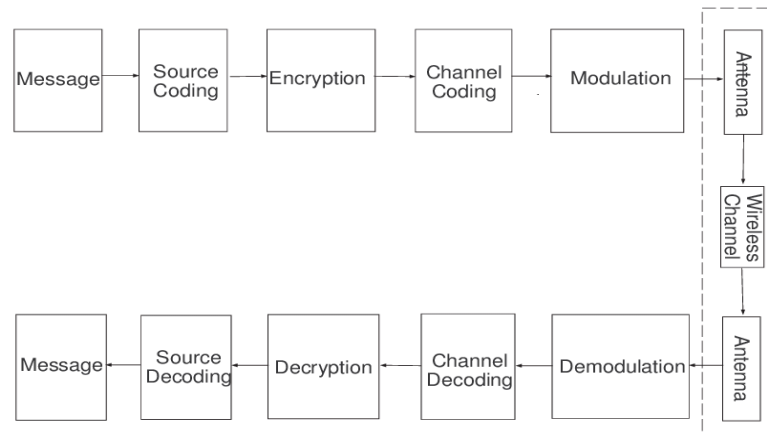


Fig: 1.1 Wireless Communication Framework
(Courtesy: Researchgate)

dynamically and automatically as Bluetooth enabled devices enter and leave radio proximity.

Bluetooth specifications were developed in 1994 and licensed by the Bluetooth Special Interest Group (SIG) that consists of companies in the areas of telecommunication, computing, networking, and consumer electronics.

Infrared Data Association (IrDA)

Infrared Data Association, founded in 1993, is an industry-based group of device manufacturers that developed a standard for transmitting data via infrared light waves. Infrared transmissions are inherently localized and governments do not regulate the infrared portion of the light spectrum.

The standard is especially suited for low-cost, shortrange and point-to-point communication and is used for providing wireless connectivity for devices that normally would use a cable-oriented solution. It provides a set of protocols including all layers of data transfer, moreover some network management capabilities and interoperability designs features.

Institute of Electrical and Electronics Engineers 802.11 standard

IEEE 802.11 is a set of standards carrying out wireless local area network (WLAN) computer communication in the 2.4, 3.6 and 5 GHz frequency bands.

The original modulation used in 802.11 was phase-shift keying (PSK). However, other schemes, such as complementary code keying (CCK), are used in some of the newer specifications. The newer modulation methods provide higher data speed and reduced vulnerability to interference.

This is used to connect two relatively large devices that have lots of power at high speeds e.g., laptops. This technology is also useful to extend LAN to places where it is either expensive or inconvenient to run cables.

Institute of Electrical and Electronics Engineers 802.16 standard

IEEE 802.16 is a series of Wireless Broadband standards with current version IEEE 802.16-2009 amended by IEEE 802.16j-2009. It is designed to develop standards for the global deployment of broadband Wireless Metropolitan Area Networks.

Though officially called Wireless MAN in IEEE, it has been commercialized under the name “WiMAX” (from Worldwide Interoperability for Microwave Access) by the industry alliance called the WiMAX Forum.

The objective of WiMAX is to deliver wireless broadband services to remote access places without highspeed cable or digital subscriber line (DSL) Internet service.

Zigbee

ZigBee is an established set of specifications for wireless personal area networking (WPAN), which are based on the IEEE 802.15.4-2003 standards. This is proposed by ZigBee Alliance. ZigBee is targeted at radio-frequency (RF) applications that require a low data rate, long battery life, secure networking, and simple implementation.

It is a standards-based wireless technology developed to enable low-cost, low-power wireless machine-to-machine (M2M) and internet of things (IoT) networks. Zigbee is for low data rate, low power applications and is an open standard.

Wireless Communication Approaches

Standards like Bluetooth and IrDA address high data rate applications such as voice, video, and LAN communications, whereas, ZigBee targets low data rates, consume very low power and are thus characterized by long battery life.

Types Of Wireless Communication Systems

Today, people need Mobile Phones for many things like talking, internet, multimedia etc. All these services must be made available to the user on the go i.e., while the user is mobile. With the help of these wireless communication services, we can transfer voice, data, videos, images etc.

Wireless Communication Systems also provide different services like video conferencing, cellular telephone, paging, TV, Radio etc. Due to the need for variety of communication services, different types of Wireless Communication Systems are developed. Some of the important Wireless Communication Systems available today are:

- Television and Radio Broadcasting
- Satellite Communication
- Radar
- Mobile Telephone System (Cellular Communication)
- Global Positioning System (GPS)
- Infrared Communication
- WLAN (Wi-Fi)
- Bluetooth
- ZigBee
- Paging
- Cordless Phones
- Radio Frequency Identification (RFID)

There are many other systems with each being useful for different applications. Wireless Communication systems can be again classified as Simplex, Half Duplex and Full Duplex. Simplex communication is one way communication. An example is Radio broadcast system. Half Duplex is two-way communication but not simultaneous one. An example is walkie – talkie (civilian

band radio). Full Duplex is also two-way communication and it is a simultaneous one. Best example for full duplex is mobile phones.

The devices used for Wireless Communication may vary from one service to other and they may have different size, shape, data throughput and cost. The area covered by a Wireless Communication system is also an important factor. The wireless networks may be limited to a building, an office campus, a city, a small regional area (greater than a city) or might have global coverage.

Television and Radio Broadcasting

Radio is the first wireless service to be broadcast. It is an example of a Simplex Communication System where the information is transmitted only in one direction and all the users receiving the same data.



Fig; 1.2 TV and Radio Broadcasting

(Courtesy: Electronicshub)

Satellite Communication

Satellite Communication System is an important type of Wireless Communication. Satellite Communication Networks provide worldwide coverage independent to population density.

Satellite Communication Systems offer telecommunication (Satellite Phones), positioning and navigation (GPS), broadcasting, internet, etc. Other

wireless services like mobile, television broadcasting and other radio systems are dependent of Satellite Communication Systems.

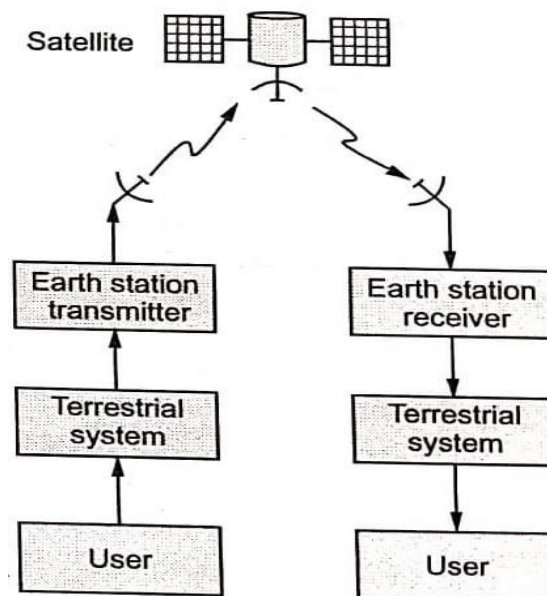


Fig: 1.3 Satellite Communication System
(Courtesy: Thetech.com)

Mobile Telephone Communication System

Perhaps, the most used wireless communication system is the Mobile Phone Technology. The development of mobile cellular device changed the World like no other technology. Today's mobile phones are not limited to just making calls but are integrated with numerous other features like Bluetooth, Wi-Fi, GPS, and FM Radio.

The latest generation of Mobile Communication Technology is 5G (which is indeed successor to the widely adapted 4G). Apart from increased data transfer rates (technologists claim data rates in the order of Gbps), 5G Networks are also aimed at Internet of Things (IoT) related applications and future automobiles.

Global Positioning System

GPS is solely a subcategory of satellite communication. GPS provides different wireless services like navigation, positioning, location, speed etc. with the help of dedicated GPS receivers and satellites.

Bluetooth

Bluetooth is another important low range wireless communication system. It provides data, voice, and audio transmission with a transmission range of 10 meters. Almost all mobile phones, tablets and laptops are equipped with Bluetooth devices. They can be connected to wireless Bluetooth receivers, audio equipment, cameras etc.

Paging

Although it is considered an obsolete technology, paging was a major success before the wide spread use of mobile phones. Paging provides information in the form of messages and it is a simplex system i.e., the user can only receive the messages.

Wireless Local Area Network (WLAN)

Wireless Local Area Network or WLAN (Wi-Fi) is an internet related wireless service. Using WLAN, different devices like laptops and mobile phones can connect to an access point (like a Wi-Fi Router) and access internet.

Wi-Fi is one of the widely used wireless network, usually for internet access (but sometimes for data transfer within the Local Area Network). It is very difficult to imagine the modern World without Wi-Fi.

Infrared Communication

Infrared Communication is another commonly used wireless communication in our daily lives. It uses the infrared waves of the Electromagnetic (EM) spectrum. Infrared (IR) Communication is used in remote controls of Televisions, cars, audio equipment etc.

Features Of Wireless Communication

The evolution of wireless technology has brought many advancements with its effective features.

- The transmitted distance can be anywhere between a few meters (for example, a television's remote control) and thousands of kilometres (for example, radio communication).

- Wireless communication can be used for cellular telephony, wireless access to the internet, wireless home networking, and so on.
- Other examples of applications of radio wireless technology include GPS units, garage door openers, wireless computer mice, keyboards and headsets, headphones, radio receivers, satellite television, broadcast television and cordless telephones.

Wireless Advantages

Wireless communication involves transfer of information without any physical connection between two or more points. Because of this absence of any 'physical infrastructure', wireless communication has certain advantages. This would often include collapsing distance or space.

Wireless communication has several advantages; the most important ones are discussed below –

- ***Cost Effectiveness***

Wired communication entails the use of connection wires. In wireless networks, communication does not require elaborate physical infrastructure or maintenance practices. Hence the cost is reduced.

- ***Flexibility***

Wireless communication enables people to communicate regardless of their location. It is not necessary to be in an office or some telephone booth in order to pass and receive messages.

- ***Convenience***

Wireless communication devices like mobile phones are quite simple and therefore allow anyone to use them, wherever they may be. There is no need to physically connect anything in order to receive or pass messages.

- ***Speed***

Improvements can also be seen in speed. The network connectivity or the accessibility were much improved in accuracy and speed.

- ***Accessibility***

The wireless technology helps easy accessibility as the remote areas where ground lines cannot be properly laid, are being easily connected to the network.

- ***Constant Connectivity***

Constant connectivity also ensures that people can respond to emergencies relatively quickly.

Advantages Of Wireless Technology

Wireless communications offer several cost and connectivity benefits for organizations. The rapid rise of remote work -- fuelled in part by the pandemic that began in 2020 -- has created an environment where employees demand high-performance access to network resources regardless of where they might be.

Wireless networks can be less expensive to build and maintain, requiring fewer switches and other components required by wired infrastructures. New wireless standards -- chief among them multiuser, multiple input and multiple output -- enable physical APs to serve a greater number of users, further reducing costs.

Wireless technologies also support novel services, among them location tracking, IoT deployments and remote diagnostics. Finally, advances in wireless performance enable organizations to stress mobility as a primary benefit, thus helping them redefine how they design their offices and workgroups.

Challenges Of Wireless Technology

Setting up a business Wi-Fi network requires careful planning and design strategies to ensure APs and other components are installed correctly and that users get access to the services and applications they need. Wireless site surveys are a must to ensure adequate coverage and signal strength, as well as to determine that any interference across the network is minimized as much as possible. The different types of site surveys include predictive, passive and active. A comprehensive wireless network evaluation usually requires all three types of these surveys.

Wireless security presents particular challenges, requiring organizations to put together a comprehensive strategy to protect WLANs and cellular connections from data loss. The sheer number of devices connecting to a wireless network --

from laptops to specialized IoT devices -- dramatically increases the network's exposure to threats from attackers. Although security standards, such as Wi-Fi Protected Access, provide substantial protection against hackers, some vulnerabilities remain. Organizations routinely face different types of wireless network attacks and should take proper proactive measures to prevent them.

Wireless network components, such as Wi-Fi 6 APs, require more power; enterprises will need to replace Power over Ethernet switches and other underlying infrastructure to support the newest standards, even as they maintain legacy components. Price points and carrier pricing will remain in flux as suppliers roll out new equipment.

Additionally, troubleshooting wireless networks can be complicated. This is in part because users are so distributed and use a wide variety of devices. Organizations must also craft strategies and devote resources to ensure that remote workers can safely and reliably access corporate services and applications to do their jobs.

1.2 OUTLINE OF THE PROJECT

In today's world the robotics field is growing exponentially and some of the popular robotic products are used largely by the industries, defence, academic and research communities. The design and implementation cost of a robot is very less than hiring a human caregiver. The robots can be reprogrammed faster and more efficient. The robot has sufficient intelligence to cover the largest area to provide a secured space. The intelligent robots can perform preferred tasks in unstructured environments with or without human direction. The real time object detection is required because safety and security are essential in the remote monitoring and control systems such as intelligent home environments, consumer surveillance system, etc. The real-time human body detection is essential for various fields like home security systems, surveillance systems, communication systems and more. Basically, the surveillance systems are building up with multiple cameras which are placed in different angles of view to track human objects. Particularly the tracking task is needed on cameras for dynamic objects which increases the number of cameras used in the system. Earlier the robots were controlled through wired networks but now to make robot more users friendly, they are framed to make user

commanded work. There are no distance limitation issues in this project. The robot is capable to work everywhere where there is a wireless connection.

This project can be used for security purposes where we need to get information about some suspicious area/people. We can do this by sitting at a far secure place and safely devise a plan to tackle their activities. It acquires image from cameras through a web browser. The robot contains a PC with a web server. It receives real-time uploaded image from cameras. As the internet of things is the concept, newly introduced in the field of electronics. The concept is about handling the things with the use of internet and the best model for these applications is raspberry pi.

1.3 MOTIVATION

In the current scenario, ensuring safety and security has become an inevitable essentiality. Traditionally, surveillance systems have been installed in every security critical area. These systems generally have high quality cameras, multiple computers for monitoring, servers for storing these videos and many security personnel for monitoring these videos. Wireless system has been under rapid development in recent years. Communication without wiring makes control systems very compact and robust. The key feature of this technology is that it reads a signal (Transmitting/Encoding Circuit) and processes it into a wave (low frequency) form which is then Decrypt into a specified signal receiver (Receiver/encoding) which then decrypt the signal into the previously transmitted signal resulting in a successful wireless data communication. In robotics, this technology is of heavy demand. By the application of a wireless signal communication, a robotic system can be made more efficient and compact.

In our proposed system the surveillance robot uses raspberry pi 3 model B as the controller. The unusual movement due to human interventions is detected by the USB camera. The camera live streams the video to the web page using Internet of Things. The robot's movement is controlled using Bluetooth connection to the android or laptop. The camera used here is a night vision camera. Thus, the system is more flexible to monitor even during night.

The field of surveillance robots is very popular. A lot of work was done in the algorithms and the navigation control System wireless surveillance robots. A

common subject is also the use of a camera on the robot to receive live video. As seen in all documents and previous research they use webcams. Webcams require Internet, one of our applications is under the supervision of the tragedies and terrorist attacks. Communication fails in these situations because of traffic so there is no use of webcam so here we use an in-built camera that does not depend on third party network but depends on its own network. For a long-distance communication according to previous documents, they used the internet or Bluetooth control. The surveillance can be made effective with the help of high-quality video transmission. The quality of video is improved in the proposed system. This ground bot can move on the various surfaces such as muddy areas, staircases etc. It will detect an object and if the object is detected, it will change directions. It has a camera which can be used for surveillance of human beings. This project's main functionality is to deal with tough situations where human beings cannot handle situations like darkness, entering narrow and small places and detecting hidden bombs etc.

Today's video surveillance technologies let homeowners remotely view footage from their desktops and smartphones. The result is real-time insight into what is happening at home, which can provide peace of mind or actionable intelligence. While this capability alone is extremely valuable, the benefits and applications of remote video surveillance extend much further.

1.4 OBJECTIVES

- It enables the connection of billions of people to the Internet so that they can reap the benefits of today's digital economy.
- Every sector of the economy now relies upon wireless technologies in fundamental ways – from banking and agriculture to transportation and healthcare.
- Hold great promise to improve lives at an unprecedented pace and scale.
- People across the world are connected and able to benefit from the widening range of digital services offered via wireless communication.
- Wireless Communications enables people to communicate regardless of their location.
- It is a type of data communication that is performed and delivered wirelessly.
- The wireless network has afforded opportunities to industrial solutions and provides global coverage by fostering short and long-distance communication.

- It has unquestionably added convenience and comfort at a personal and professional level

1.5 APPLICATIONS

- Remote video monitoring: to protect against theft, burglaries, and dishonest employees.
- Facility protection: to protect the perimeter of the property or the perimeter of buildings.
- Monitor operations: to monitor day-to-day operations and as a tool to streamline operations.
- Loss prevention: to protect assets.
- Vandalism deterrence: visible cameras may be a deterrent to vandals because of the possibility that they can be identified on the video. High-definition cameras with facial recognition in a durable, vandal-proof housing can be used.
- Employee safety: for compliance with safety regulations and to protect the employer in civil proceedings.
- Parking lots: to monitor for theft or damage to vehicles as well as accidents.
- Event video surveillance: for crowd control as well as crime prevention.
- Public safety: routinely used for city streets, parks, communities, and neighbourhoods to help deter crime and enhance public safety.
- Traffic monitoring: commonly used to improve the flow of traffic and monitor for accidents.
- Outdoor perimeter security: security can be maintained in different ways by utilizing video surveillance with security/protection officers on patrol, fences, vehicle, and pedestrian gates and intrusion detection.

CHAPTER 2

LITERATURE SURVEY

2.1 OVERVIEW OF LITERATURE SURVEY

Zeeshan Shaikh, Priyanka Gaikwad, Nikhil Kare, Swapnali Kapade (2017) has presented a Surveillance Robot scaling horizontal and vertical surfaces while automatically controlling surface transitions, and provides the controlling user with surveillance of its location. Unlike other wall climbing robots currently available using vacuum suction technique and magnetic prosthesis. The proposed model is also capable of capturing real-time images, video and audio to provide surveillance for a specific person or area. This robot is suitable for military applications like monitoring a person or area of interest, provide tactical advantage in hostile grounds or during hostage situations. It is capable of walking on any surface and providing monitoring over a area with the help of image processing which is capable to analysis and manipulation of a digitized image, video. The proposed system consists of a single unit, which will monitor the Environment for various hazardous conditions and provide video feedback.

Mahesh Totad, R. Naveen Kumar Reddy, Mahindhar Remata, Vinay A (2020) have designed a system that will be useful in military real time applications. The proposed Raspberry pi based robotic vehicle system is designed to save the precious human life in some dangerous fields such as military during hostage situations. During war or when military people enter uncharted territory (hostage situations) they will sometimes become victims of surprise enemy attack. This robotic vehicle is a new method to trace out the enemies and use that information to make a tactical move. The system can substitute the soldier in border to provide surveillance as well for reconnaissance circumstances.

Ashish U. Bokade; V. R. Ratnaparkhe (2016) have proposed a method for controlling a wireless robot for surveillance using an application built on Android

platform. The Android application will open a web-page which has video screen for surveillance and buttons to control robot and camera. Android Smartphone and Raspberry pi board is connected to Wi-Fi. An Android Smartphone sends a wireless command which is received by Raspberry pi board and accordingly robot moves. The Video Streaming is done using MJPG streamer program that gets mjpeg data and sends it through a HTTP session. The Raspberry pi programming is done in python language. The experimental result shows that the video streamed up to 15 frames per second.

Keerthi Premkumar; K Gerard Joe Nigel (2015) have proposed method for controlling a Robotic arm using an application build in the android platform. The android phone and raspberry pi board is connected through Wi-Fi. As the name suggests the robotic arm is designed as it performs the same activity as a human hand works. A signal is generated from the android app which will be received by the raspberry pi board and the robotic arm works according to the predefined program. The android application is the command centre of the robotic arm. The program is written in the python language in the raspberry board. the different data will control the arm rotation.

Anushree Deepak More¹, Rakshita Devendra Godiyal², Hiral Gachake, Ashwini Balaji Ghodake (2022) Remote surveillance of our house, as well as office areas, has become an important aspect. Human surveillance is achieved by moving (troop or equipment) personnel near sensitive areas to constantly monitor the area. But humans have their limitations, and deployment in inaccessible places is always impossible. They have developed a robot that can be used for video surveillance & monitoring which can be controlled through a GUI. Using this project, they put forward a surveillance robot with an Interface Board of the Raspberry Pi and Software to fulfil Real Time requirements. The Raspberry Pi Robot has a personalized digital device with peripherals such as Raspberry PI, LED lights, a Personalized Camera, Speaker for text to voice conversion and security or threat alarms, a range sensor, etc. The base controller of the robot will be the Raspberry Pi 3. A webcam attached to the Pi monitors the area with Live Streaming. The live

streaming ability of the Pi allows the camera feed to be analysed from any location using the internet. For detecting the obstacles, they had used an ultrasonic range sensor along with it to give commands or pass on messages we have installed a speaker which will convert text to voice which is an E-speak technique. With such a system, every user will feel more safe, secure, and sheltered while they are not at their place of residence or when they've left their kids and geriatric alone at home.

Bhavyalakshmi R, B. P. Harish (2019) The proposed work aims to develop an automatic solution to detect the presence of an enemy or any hostile events such as fire/gas leakage in targeted places without loss of human life. It consists of a robotic vehicle for spying the pre-allocated area by continuously monitoring it. Whenever a sensor gets activated, the surveillance system checks for the presence of humans and then runs a face recognition algorithm. If the facial data of the person detected does not match with the pre-stored personal data of soldiers, the system recognizes him as an intruder and activates the laser gun to target him. In case of detection of unusual/dangerous events such as someone carrying a knife or a gun in high security zone, an alert message is sent to the operator. The system also provides the live streaming of surveillance data to the operator using Raspberry Pi and VNC Viewer.

Rui Chen, Rong Liu, Jifan Chen, Jin Zhang (2014) have presented the design, analysis and fabrication of a quadruped wall-climbing robot. Inspired by the climbing gaits of geckos, the robot has kinematics like a gecko's motion. Unlike geckos stick on the wall surface with Van der Waals force, the robot is based on electrostatic adhesion force induced by a specifically designed electrostatic adhesive footpad. By combining the bionic gait design of the gecko and the unique properties of electrostatic adhesion mechanism, the robot has advantages of light weight, low energy consumption, flexible movement and high adaptability to different wall surfaces. Climbing experiments on the surface of a high-rise glass window are demonstrated, and the robot can achieve straight climbing and turning robustly and agilely. The potential applications of this robot are reconnaissance, inspection, cleaning and repairing.

Y.S.S.N. Udaya Laxmi, S. Jakeera, G. Naresh, E. Praveen Babu (2018) They have proposed a method for controlling a wireless robot for surveillance using an application built on Android platform. The Android application will open a webpage which has video screen for surveillance and buttons to control robot and camera. Android Smartphone and Raspberry pi board is connected to Wi-Fi. An Android Smart phone sends a wireless command which is received by Raspberry pi board and accordingly robot moves. The Video Streaming is done using MJPG streamer program that gets mjpeg data and sends it through a HTTP session. The Raspberry pi programming is done in python language. The experimental result shows that the video streamed up to 15 frames per second.

G.R. Sri Sudharsna, A. Srinivasan (2019) The proposed video surveillance robot can be moved in any direction according to the instructions from the webpage and it streams the captured video lively over the webpage. Raspberry –pi is the heart of this robot. Two software modules, “MOTION” and “FLASK” are incorporated to get the live streaming and to transfer instructions from the webpage to Raspberry-pi. It can be used in small level military applications, monitoring systems in residence, hotels. This robot is designed in a way that it can be accessed anywhere. The coverage range for the video surveillance is not limited as the robot is mobile. The system consists of two segments, hardware segment and software segment.

M.Faesa Fathima, S. Divya, S. Mouriya (2019) develop a cost effective easy to control surveillance vehicle through remote desktop for the implementation of military purpose. Since the risk factor in military border is too high causing threats to the lives of soldiers at time patrol by both climatic conditions and enemy nation which needs a replacement, that is done effectively by the surveillance vehicle that comprises the Raspberry Pi (small single-board computer), pi camera and sensors. The information regarding the detection of living objects by *PIR sensor* and image capture of moving objects by *pi camera* capture is posted inside the webpage simultaneously. The movement of a robot is also controlled automatically through obstacle detecting sensors to avoiding the collision, and the system causes an alert at the time of metal detection through metal detecting sensor. Since the system does multitask this can be used in surveillance purpose.

R. Karthikeyan, S. Karthik, Prasanna Vishal, S. Vignesh (2015) have designed a robot named snitch capable of climbing wall using Micro suction cups, it uses a raspberry pi processor to manage the robot by a wi-fi network for a better and uninterrupted data processing and conveyance. The robot is suitable in military applications like keeping a track on a person or place of interest, provides skilful advantage in hostile grounds or during unwanted solutions.

KunWang Zhiqiang and Houxiang Zhang (2011) have verified the locomotion mechanism and crawling gait of their flexible wall-climbing caterpillar robot. Two different kinds of modules are used, which are connecting module and joint module, were developed. Due to predefined constraints between the inhalation cups and wall, the motion of the caterpillar robot engages a changing genetic set which form open group to closed group, and then to an open group orderly.

Elliott, M. Morris, W. Jizhong Xiao (2007) They have proposed a design obtains both fast motion of each module on uninterrupted surfaces and smooth progress between the available surface. The video also displays the eloquent simulation results of the aerodynamic tendency which has the aim to optimize the design. DSP – based control system is introduced into surveillance robot using Raspberry-pi technology which starts the robot to operate both manually and autonomously.

Tushar Maheshwari, Upendra Kumar, Chaitanya Nagpae, Chandrakant Ojha, and V.K Mitta (2015) Spy-robot which are controlled wireless can be truly useful if they can be controlled remotely over a long-distance operating range. Availability of multiple procedures for their wireless control operation can be improvised their capabilities and the range of applications. The spy-robot can be inhabited using a smart phone based DTMF, distantly control application, audio commands and constantly changing gesture control function. DTMF uses alpha-numeric keypad of mobile phone. The remote function is developed for the android platform based smart phone.

Chen, Rong Liu, Jifan Chen and Jin Zhang (2014) have shown the design, analysis, and forgery of a quadruped wall climbing robot. Inspired by the climbing

bearing of geckos, the robot has genetics like a gecko's motion. Geckos stick on the wall surface with force. Rui, the robot is based on electrostatic adhesion force induced by a specially designed electrostatic adhesive footpad. By adding the bionic bearing design of the gecko and unique assets of electrostatic adhesion mechanism, the robot has movement, and high versatility to different wall surfaces. Climbing practice on the surface of a high-rise glass window are exhibited, and the robot achieve the straight climbing and turning robustly and agilely.

Deepika R, Prathyusha K, Amulya P (2014) They showed the frame work on vision based interfaced that has designed to instruct a humanistic robot through jesters using image processing. Image predefined and blob detection techniques are used to obtain sign language. Then we evaluating the images to recognize the gesture given by the user in front of a web camera and take relevant movements (like clicking picture, moving bot, etc). The application is developed using OpenCV libraries and Microsoft visual c++. The movements obtained by processing the live images are used to command a humanistic robot with simple capabilities. A commercial robotic human toy robot Robosapiens were used as the o/p module of the system. The robot was consolidating to computer by USB-UIRT (Universal Infrared Receiver and Transmitter) module.

Osumi H, Yokohama K, Takeuchi K, Nakamura R (2012) This paper, first a method using force redundancy of the robot is been proposed for the purpose of speeding Surveillance Robot Using Raspberry-Pi Technology Synopsis up of collateral legs with the constraints due to ZMP, the limits of connecting actuators and the limits of available friction forces. Then, an algorithm for getting the fastest walking pattern in a trot gait specific time span is developed by combining the result of the collateral legs and that of the waving legs. The obtained walking stencils is installed in a quadruped robot SONY ERS-7 and its performance is varied by experiments.

S. Wu, M. Li, S. Xiao (2006) They have proposed a wireless dispersed wall climbing robot system for exploration purpose. Firstly, it introduces the work of distributed wall climbing electronic system. The mother wall-climbing robot of one single inhalation cup with two wheels loco-motion system that enables the fast

motion and can accommodate nearly any kind of vertical wall surface in interrupted environment. The child wall climbing robot is inchworm like civilized mechanical structure with the advantages of compact size and light weight, which enables the robot turn from one surface to another and can avoid exposure. Secondly, embedded bureaucrat of wall climbing robotic system is designed. With the Li-battery energy supply and wireless transmission system, robots have the calibre to explore the world semi-autonomously.

M Fujita (2004) They have described effects interventions with a pet-type robot in this paper, with AIBO. First, we portray a design concept for AIBO based on how the "natural" appearance can be improved. By launching statistical results of marketing and practice involving the human-robot interactions using AIBO, we show that pet-type robot launches the human feelings adequately. Furthermore, the experiments validate that AIBO helps in human-human communication. We discuss the anomaly of interaction with AIBO, and attempt to explain why this happened.

2.2 COMPARISION OF EXISTING WORK

S. No	Authors	Title	Description	Year
1	Zeeshan Shaikh, Priyanka Gaikwad, Nikhil Kare, Swapnali Kapade	Surveillance robot using raspberry Technology	The proposed system can capture real time images	(2017)
2	Mahesh Totad, R. Naveen Kumar Reddy, Mahindhar Remata, Vinay A	Military surveillance robot	This system will be useful in military real time applications It is designed to be used as a robotic vehicles system it uses in dangerous fields	(2020)

3	Ashish U. Bokade, V. R. Ratnaparkhe	Robust pi Surveillance Robot	It is a wireless surveillance robot using an application built on android platform	(2016)
4	Keerthi Premkumar, K Gerard Joe Nigel	Smart phone based Robotic arm Control using Raspberry Pi	It tells a method for controlling a robotic arm using an application built on android application	(2015)
5	Anushree Deepak More, Rakshita Devendra Godiyal, Hiral Gachake, Ashwini Balaji Ghodake	Robust Pi Surveillance Robot	It is a robot that can be used for video surveillance and monitoring which can be controlled through a GUI.	(2022)
6	Bhavyalakshmi R, B. P Harish	Surveillance Robot with face recognition using raspberry pi	It aims to develop an automatic solution to detect the presence of an enemy or any hostile events	(2019)
7	Rui Chen, Rong Liu, Jifan Chen, Jim Zhang	A gecko inspired wall-climbing robot based on Electrostatic adhesion mechanism	They have presented the design, analysis and fabrication of a quadruped wall-climbing have presented by the	(2014)
8	Y S. S. N Udaya Laxmi, S Jakeera, G. Naresh, E. Praveen Babu	Wireless video surveillance robot using raspberry pi	They have proposed a method for controlling a wireless robot for surveillance using an application built on android platform	(2018)

9	R. Karthikeyan, S. Karthik, Prasanna Vishal, S. Vignesh	Snitch robot capable of climbing walls.	The robot is suitable in military applications like keeping a track on a person or place of interest, provides skilful advantage in hostile grounds or during unwanted solutions.	(2015)
---	---	---	---	--------

2.3 OPEN PROBLEMS IN EXISTING SYSTEM

Protecting homes, businesses and communities from external threats is a top priority for many. One of the easiest solutions is installing a video surveillance system – video cameras attached to watched monitors that track intruders – so that any threats may be witnessed and potentially even stopped before they take place. We know that surveillance can deter theft and other crimes, and that these cameras can provide evidence when perpetrators do act. There is a certain amount of trust we place in surveillance, to operate effectively and efficiently, to provide valued data and to be vigilantly watched. Indeed, most have become so used to video surveillance it's almost second nature to rely on the camera that sees all. Given the dependency on these systems, however, it's worth taking a look at a few challenges that still exist in recording and using video surveillance and how to eliminate them.

Few of the challenges that are now faced are as follows:

1. Data loss
2. Operator loss
3. Power failure

Data Loss:

Because there is so much reliance on surveillance systems, data loss in the event of system failure or external circumstances can be detrimental. Residential and commercial use of security cameras requires the recording to be well documented and secure, so that information is not easily lost.

Operator Loss:

Those who are responsible for watching video surveillance are only human, which means there can be opportunities for criminals to sneak into their blind spots or attack when operators are not alert. In these cases, video surveillance can fail to accurately capture movement and track crimes as they occur.

Power Failure:

If the power goes out, and a surveillance system operates on Wi-Fi, a camera's chances of catching anything on video are compromised. While there is a dependency on many power systems, there are some operational inefficiencies that are simply out of our control.

2.2.1 Problems faced with public video surveillance

Video cameras, or closed-circuit television (CCTV), are becoming a more and more widespread feature of American life. Fears of terrorism and the availability of ever-cheaper cameras have accelerated the trend even more.

Video Surveillance has not been proven effective:

The implicit justification for the recent push to increase video surveillance is the threat of terrorist attacks. But suicide attackers are clearly not deterred by video cameras - and may even be attracted to the television coverage cameras can ensure - and the expense of an extensive video surveillance system such as Britain's - which sucks up approximately 20 percent of that nation's criminal justice budget - far exceeds the limited benefits that the system may provide in investigating attacks or attempted attacks after the fact.

The real reason cameras are usually deployed is to reduce much pettier crimes. But it has not even been demonstrated that they can do that. In Britain, where cameras have been extensively deployed in public places, sociologists studying the issue have found that they have not reduced crime.

CCTV is susceptible to abuse:

One problem with creating such a powerful surveillance system is that experience tells us it will inevitably be abused. There are five ways that surveillance-camera systems are likely to be misused:

1. Criminal abuse
2. Institutional abuse
3. Abuse for personal purposes
4. Discriminatory targeting
5. Voyeurism

The lack of limits or controls on cameras use: Advanced surveillance systems such as CCTV need to be subject to checks and balances. Because the technology has evolved so quickly, however, checks and balances to prevent the kinds of abuses outlined above do not exist.

Video surveillance will have a chilling effect on public life: The growing presence of public cameras will bring subtle but profound changes to the character of our public spaces. When citizens are being watched by the authorities - or aware they might be watched at any time - they are more self-conscious and less free-wheeling.

2.4 SUMMARY OF EXISTING WORK

Already existing systems use robots that have limited range of communication as they are based on RF Technology, Zigbee and Bluetooth. Some existing projects use short range wireless camera. Some existing robots can only be controlled with a manual mode which needs human supervision throughout the whole surveillance process.

In existing system, the Surveillance is not that effective. It only captures the real time images or videos but cannot take any actions for the suspicious activity. So, for that purpose humans are send to such places hence their lives are not that secure. So proposed system deals with the development of a mobile robot capable of capturing real-time images and videos for surveillance. In the proposed system, the surveillance is more effective efficient.

CHAPTER 3

WIRELESS VIDEO SURVEILLANCE SYSTEM

At this moment many video camera systems have been installed to detect unwanted behaviour of some people. Unwanted behaviour such as aggression or violence acts against people or material. Military areas should be protected against unwanted intruders aiming at stealing weapons from the arsenal/armoury or violence acts by bombing or whatever. At this moment most systems are only used for recording of data. In case of an incident the video recordings will be analysed after the incident. Real time monitoring of video data by human operators takes a lot of time and money.

3.1 BLOCK DIAGRAM OF THE PROPOSED SYSTEM

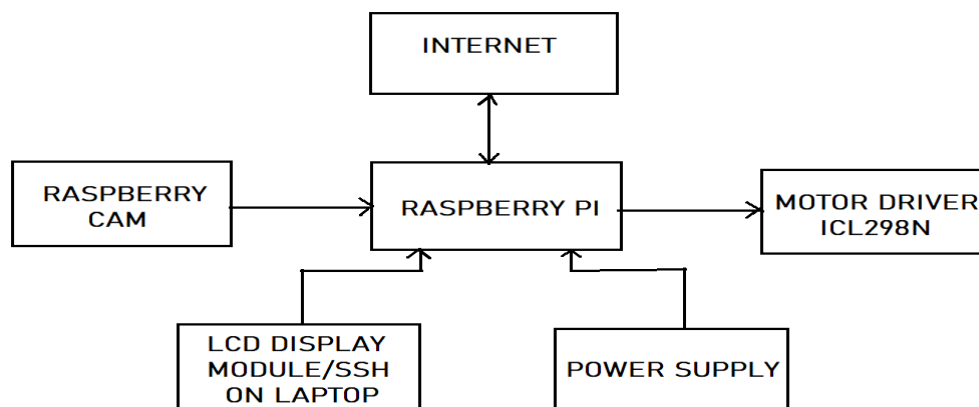


Fig: 3.1 Block Diagram of the Robot

Camera surveillance systems monitored by human operators is currently used for surveillance of limited areas such as industrial areas, areas around buildings or military areas. In this paper we study a military area composed of streets, building, squares and grass fields. The area is surrounded by a fence or by water. To enter the area a visitor must pass a guarded barrier, or use a valid access card. During daytime we can observe traffic from visitors moving to the lecture halls, dormitories etc. In exceptional cases we can observe people moving to the arsenal. Of special interest are the unusual rarely observed tracks, people climbing the gate,

hitting the barrier, or walking on the grass fields instead of using the regular roads. In this paper we want to analyse the tracks people take in the military area and detect unusual and possible suspicious behaviour.

When developing smart monitoring systems, the factors influencing acceptance such as usefulness, risks and privacy should be determined. In this way, the acceptance of these systems can be achieved by adapting it to the fears and doubts of those affected and not vice versa. Possible ways to address this issue could be to censor, pixelate or to ensure an increased level of data-privacy by omitting information that allows the identification of people. The surveillance can be made effective with the help of high-quality video transmission. The quality of video is improved in the proposed system. This ground bot can move on the various surfaces such as muddy areas, staircases etc.

This project's main functionality is to deal with tough situations where human beings cannot handle situations like darkness, entering narrow and small places and detecting hidden bombs etc. Such hostile situation is occurring day by day in different parts of the world through terrorist attack or in natural catastrophes. This designed system is connected to a remote computer wirelessly through which the whole controlling of the system response is done. The project is also designed to search invisible metal stuffs from where people are not capable to reach and it is so designed to work in environment where visible light will not be available.

The proposed system consists of two units mainly a robotic unit and a remotely control unit. The robotic unit is consisting of the webcam, the heart of the project, raspberry pi along with the PCB containing motor driven IC and voltage regulator circuitry

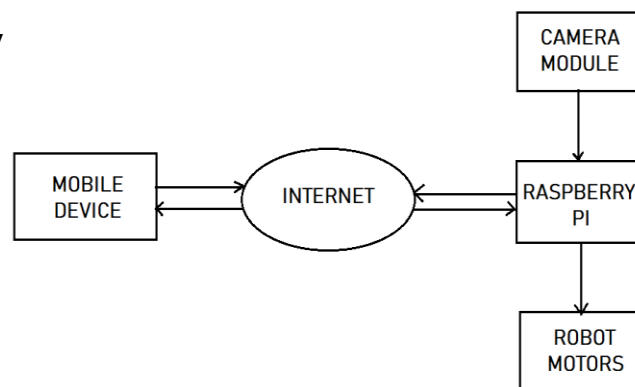


Fig: 3.2 Architecture of the Proposed System

The robot is built using mechanical and electrical components. The main controlling unit of the system is Raspberry Pi. Power supply circuit is designed to provide power to Pi board and motor driver IC (L293D). Two dc motors are used to control the forward, backward, left and right movement of robot. Camera module is used to take picture frames for video. Tilt motion of camera is controlled using servomotors to provide the wide capturing area. Wi-Fi dongle is used to connect Raspberry Pi to the Wi-Fi Router.

3.2 REQUIREMENTS

Requirements are the basic constraints that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

- Functional requirements
- Non-Functional requirements
- Software requirements

Robotic control device is constructed on a platform designed for security and monitoring tasks. This means it can be remotely controlled by a human as a sentinel surveillance camera movement. Secret surveillance closely confined spaces is required in many military and civil activities, such as landslides and raiding enemy within hostage rescue missions.

3.2.1 Functional Requirements

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system.

3.2.2 Non-Functional Requirements

Process of functional steps,

1. Problem defines
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction the result

3.3 Software Requirements

Software: Notepad++

Programming Language: C++

3.3.1 Notepad++

Notepad++ is a free and open-source text and source code editor for use with Microsoft Windows. It supports tabbed editing, which allows working with multiple open files in a single window. The product's name comes from the C postfix increment operator. Notepad++ is distributed as free software. At first, the project was hosted on SourceForge.net, from where it was downloaded over 28 million times and twice won the Source Forge Community Choice Award for Best Developer Tool. The project was hosted on Tux Family [fr] from 2010 to 2015; since 2015, Notepad++ has been hosted on GitHub. It uses Scintilla editor component.

Notepad++ is a source code editor. It features syntax highlighting, code folding and limited autocompletion for programming, scripting, and markup languages, but not intelligent code completion or syntax checking. As such, it may properly highlight code written in a supported schema, but whether the syntax is internally sound or compliable, cannot be verified.

The language list also displays two special-case items for ordinary plain text: "Normal text" (default) or "MS-DOS Style", which tries to emulate DOS-era text editors.

Notepad++ has features for consuming and creating cross-platform plain text files. It recognizes three newline representations (CR, CR+LF and LF) and can convert between them on the fly. In addition, it supports reinterpreting plain text files in various character encodings and can convert them to ASCII, UTF-8 or UCS-2. As such, it can fix plain text that seem gibberish only because their character encoding is not properly detected.

Notepad++ also has features that improve plain text editing experience in general, such as:

- Autosave
- Finding and replacing strings of text with regular expressions

- Searching text strings within opened tabs
 - Searching text strings in a directory
 - Guided indentation
 - Line bookmarking
 - Macros
 - Simultaneous editing
 - Split screen editing and synchronized scrolling
 - Line operations, including sorting, case conversion (Uppercase, lowercase, camel case, sentence case), and removal of redundant whitespace
- Tabbed document interface

Plugins:

Notepad++ has support for macros and plugins and has been remarked for its robust plugin architecture which enabled various new features to be integrated into the program.^[26] Currently, over 140 compatible plugins are developed for Notepad++, 10 of which are included by default in the program. The first plugin to be included in the program was "*TextFX*", which includes W3C validation for HTML and CSS, text sorting, character case alteration and quote handling.

Internationalization:

Notepad++ supports internationalization through XML files in an application-specific format containing all internationalized strings (dialog captions, menu titles and items, etc.) in a certain language; this file can be reloaded from the application settings. Translations to new languages can thus be written by simply editing an existing file.

3.3.2 C++ Language

C++ is a high-level, general-purpose programming language created by Danish computer scientist Bjarne Stroustrup. First released in 1985 as an extension of the C programming language, it has since expanded significantly over time; modern C++ currently has object-oriented, generic, and functional features, in addition to facilities for low-level memory manipulation. It is almost always implemented as a compiled language, and many vendors provide C++ compilers,

including the Software Foundation, LLVM, Microsoft, Intel, Embarcadero, Oracle, and IBM.

C++ was designed with systems programming and embedded, resource-constrained software and large systems in mind, with performance, efficiency, and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, video games, servers (e.g. e-commerce, web search, or databases), and performance-critical applications (e.g. telephone switches or space probes).

C++ is standardized by the International Organization for Standardization (ISO), with the latest standard version ratified and published by ISO in December 2020 as *ISO/IEC 14882:2020* (informally known as C++20). The C++ programming language was initially standardized in 1998 as *ISO/IEC 14882:1998*, which was then amended by the C++03, C++11, C++14, and C++17 standards. The current C++20 standard supersedes these with new features and an enlarged standard library. Before the initial standardization in 1998, C++ was developed by Stroustrup at Bell Labs since 1979 as an extension of the C language; he wanted an efficient and flexible language similar to C that also provided high-level features for program organization. Since 2012, C++ has been on a three-year release schedule with C++23 as the next planned standard.

The C++ language has two main components: a direct mapping of hardware features provided primarily by the C subset, and zero-overhead abstractions based on those mappings. Stroustrup describes C++ as "a light-weight abstraction programming language [designed] for building and using efficient and elegant abstractions"; and "offering both hardware access and abstraction is the basis of C++. Doing it efficiently is what distinguishes it from other languages."

Object storage:

As in C, C++ supports four types of memory management: static storage duration objects, thread storage duration objects, automatic storage duration objects, and dynamic storage duration objects.

- Static storage duration objects
- Thread storage duration objects

- Automatic storage duration objects
- Dynamic storage duration objects

Hardware Requirements

The components for the proposed system are Raspberry pi, Robot Chassis, Motor Driver Circuit, Raspberry pi cam, Power Supply.

3.4 CONSTRAINTS

- Raspberry Pi's performances are compared with some popular boards and development platforms on a general level by computing power, size, and overall costs of the solutions.
- The detail analyses of Raspberry Pi have shown that as ultra-cheap-yet-serviceable computer board.
- The Raspberry Pi brings the advantages of a PC to the domain of sensor network, what makes it the perfect platform for interfacing with wide variety of external peripherals.
- The advantage of Raspberry Pi lies in its flexibility and endless possibility of its usage enabling at the same time end-users to program it according their needs and budgets.
- Security: Utilizing wireless networks raises serious security concerns. An improperly installed or managed wireless network could pose serious security risks. A wireless network does not require physical connections like wires. They merely require a wireless adaptor, which raises the risk of hacking because hackers may easily enter the network. A worsening of the situation may occur if the network has password protection.
- Speed: The wireless network's speed is less rapid than that of wired networks. In a wireless network, file transfers and file sharing are much slower. The location of the user on the network affects performance as well. The connection degrades more rapidly the further the user is away from the network. For vast areas or buildings, this is a major issue.
- Prone to Interference: There is a high chance of interference and jamming in wireless networks because of environmental variables like dust storms or fog.

When too many users are in one area, the air band that the signals are transported across can become overloaded. A wireless network could become unreliable due to fog, radiation, radio waves, or any other type of interference since wireless networks are particularly prone to interference.

- **Requires Basic Computer Knowledge:** A basic understanding of computers is necessary to set up a wireless network. Installing a wireless network might be difficult for people with little computer skills. Security is risky, and those networks are easy targets for hackers.

3.5 STANDARDS

- There has been an emerging need for remote surveillance and monitoring of our houses in the view of safety and security purpose.
- A surveillance robot has been proposed which is integrated into any type of household devices.
- It watches the premises.
- Camera will be always in active mode in order to capture the intruder movement.
- This system made every user to feel safer using such a kind of device while the Authorised person is away from home or when they have left their children and elderly relatives alone at home.
- This device plays vital role in surveillance.
- It provides privacy on both sides as only one person is viewing.
- This smart surveillance robot can be in places where human movement is not a possibility.
- This robot reduces human intervention directly in a hazardous place where continuous supervision and security is necessary.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 TRANSMITTER SECTION

This robot is designed in a way that it can be accessed anywhere. The coverage range for the video surveillance is not limited as the robot is mobile. The system consists of two segments, hardware segment and software segment. the Internet Of Things concept. This wireless video surveillance robot can be accessed and the live streaming can be observed in the webpage by feeding the IP address of the Raspberry-pi. The Raspberry-pi and the PC/Mobile through which the robot to be accessed should be connected to the same network (WI-FI).

In the transmitter section, that is the robot is connected to the device hotspot or both the device and the robot must be connected to the same network (Wi-fi). When it is connected, the camera attached will be activated and it then captures the video of the surrounding area. This captured video will then be sent to the device with the built in android web-page.

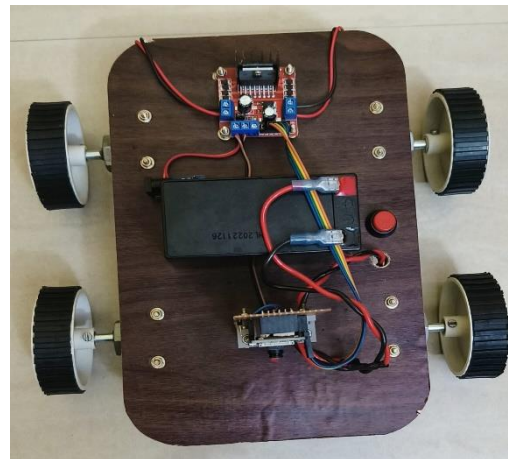
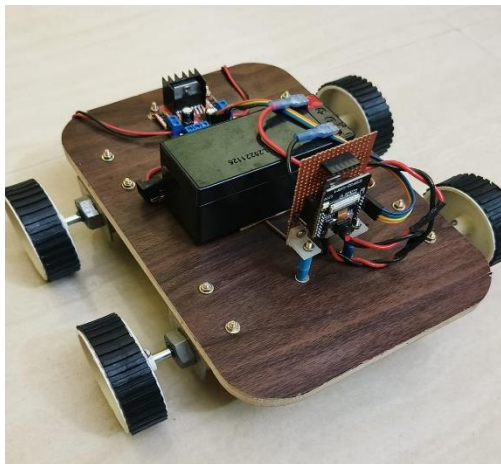


Fig: 4.1 Hardware of the Robot

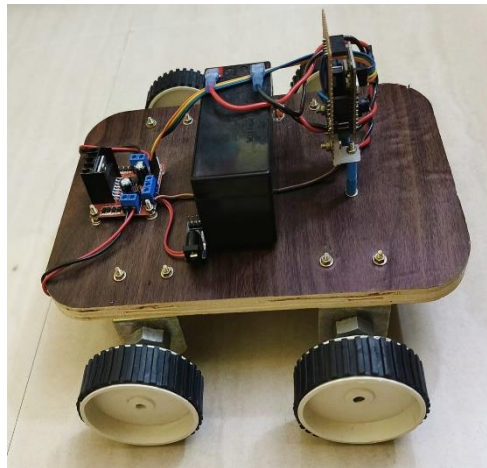


Fig: 4.2: Robot

4.2 RECEIVER SECTION

In the receiver section, that is either a mobile or pc, which be connected to the same network as the robot or gives Wi-Fi hotspot connection to the robot.

To open the webpage, the IP address of the Raspberry pi connected to the robot have to be used. Once the webpage is opened, we can see the video captured by the robot. This receiver section or the device used should be controlled by human to control and operate the robot.

The camera attached to the robot can be tilted so it can capture video of the surroundings in all directions.

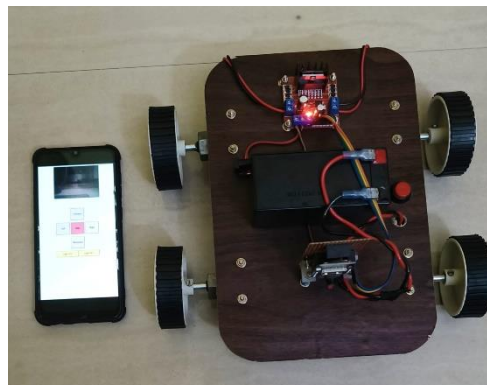


Fig: 4.3: Device connected to the robot



Fig: 4.4: Captured Video

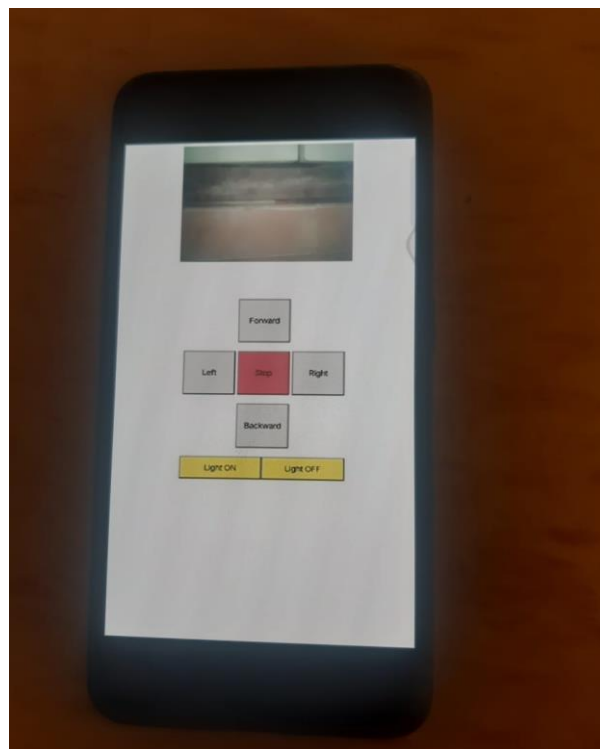
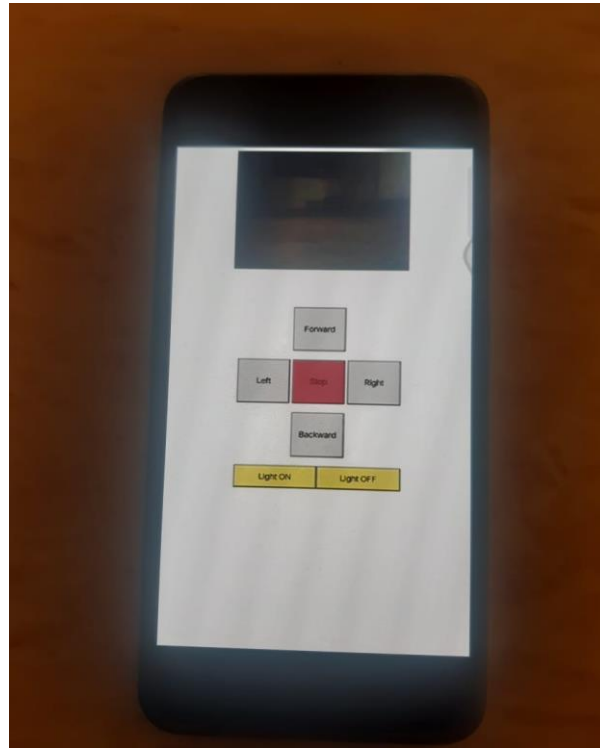


Fig: 4.5 Receiver Device

4.3 HARDWARE IMPLEMENTATION

Video surveillance is the process of monitoring a situation, an area or a person. This generally occurs in a military scenario where surveillance of borderlines and enemy territory is essential to a country's safety. Human surveillance is achieved by deploying personnel near sensitive areas in order to constantly monitor for changes. But humans do have their limitations, and deployment in inaccessible places is not always possible.

There are also added risks of losing personnel in the event of getting caught by the enemy. With advances in technology over the years, it is possible to remotely monitor areas of importance by using robots in place of humans.

Today most of the Smart phones are based on the Android platform. It has almost all the features that are available in computer or laptop. So the controlling of robot using smart phone makes the system efficient and reliable. A Wi-Fi dongle is used for connecting Raspberry Pi to Wi-Fi router. The Wi-Fi router is connected to local area network (LAN) for providing internet connectivity to the Pi.

This project's main functionality is to deal with tough situations where human beings cannot handle situations like darkness, entering narrow and small places and detecting hidden bombs etc. Such hostile situation is occurring day by day in different parts of the world through terrorist attack or in natural catastrophes. This designed system is connected to a remote computer wirelessly through which the whole controlling of the system response is done. The project is also designed to search invisible metal stuffs from where people are not capable to reach and it is so designed to work in environment where visible light will not be available.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

Thus, we have designed a smart surveillance system capable of recording video and transmitting to any portable device like smart phones and laptops. It is advantageous as it offers quick transmission of the video through Internet of Things. Necessary action can be taken in short span of time in the case of emergency such as the presence of the unwanted people in war areas who are not allowed in such areas can be determined by the PIR sensor which sends a signal to the Raspberry Pi when a human - being is in the ambience of the Robot. In turn, the Pi triggers the camera immediately to capture an image and send it to the web page. The PIR sensor and proximity sensors are activated depend on external stimuli via Internet of Things. The control room collects this information for later reference.

It is advantageous as it offers reliability and privacy on both sides. It is authenticated and encrypted on the receiver side; hence it offers the sensitive information only to the authorized personnel. Also because of its small size and portability, it can be placed in any kind of surrounding for surveillance. Areas where movement is restricted such as sensitive military/nuclear sites or in banks, this kind of surveillance and monitoring systems can be implemented. This will not only omit the requirement of physical presence for surveillance, but will also capture all information in the storage which will be proving very beneficial.

The surveillance can be made effective with the help of high-quality video transmission. The quality of video is improved in the proposed system. This ground bot can move on the various surfaces such as muddy areas, staircases etc.

5.2 FUTURE WORK

The development & innovation in technological field is moving at the rocket speed. Every new day rises with some new innovation and the rate at which the

technology is developing and has spread its wings wide it is almost near to impossible for a single person to keep the track of all the development been done till now. With development of technology in modern world gaining pace it is about time that some importance should be given to the department of surveillance technology which is still working on age old technology to get its work done. Consider an example of traffic signal. Since the establishment of embedded technology, the prototype for traffic signal has remained same even though the technology surrounding it has been revised couple of times. The technology surrounding traffic signals is mostly vehicles which depend on it for its smooth functioning. The earlier day's vehicles were just installed with simple machinery which is required for working of the four or two wheels which are installed on it and some lights surrounding the vehicles. But with the advancement of technology the changes are made and now we have reached to the point where we don't even need to drive our own cars! This proves that how far we have moved with technology and how long the development in surveillance field is overdue. There are various work and researches carried on with the surveillance technology but none has been brought into real world; that too in abundance to be a noticeable change. Similarly various robotic technologies have been proposed and brought into implementation in the robotic field but none worked as fine as it should and has been rejected very soon than expected. The Wireless Video Surveillance Robot is used to overcome all these technological lacking and unavailability in the technological field to bring about some overdue changes in this field.

REFERENCES

1. R. Karthikeyan, S. Karthik, Prasanna Vishal, S. Vignesh "Snitch design and development of a mobile robot for surveillance and reconnaissance" IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication systems ICIECS'2015.
2. KunWang, ZhiqiangWang and Houxiang Zhang, "Development of modular wall-climbing robot inspired by natural caterpillar" IEEE 2nd International Conference on computing, Control and Industrial Engineering (CCIE),2011.
3. Elliott, M., Morris, W., Jizhong Xiao, "City-Climbers at work" IEEE International Conference on Robotics and Automation,2007.
4. Deepika Ravipati, Prathyusha Karreddi, Amulya Patlola, "Real-time Gesture Recognition and Robot control through Blob Tracking" IEEE Students' Conference on Electrical, Electronics.
5. Surveillance Robot with Face Recognition using Raspberry Pi – IJERT
6. (PDF) Video surveillance robot powered by raspberry pi (researchgate.net)
7. Robust Pi Surveillance Robot (ijraset.com)
8. Y.S.S.N. Udaya Laxmi, S. Jakeera,G. Naresh ,E. Praveen Babu, Wireless Video Surveillance Robot Using Raspberry Pi, International Journal of Research, Vol ume 05, I s s ue 07 ,Ma rc h 2018
9. A.Yaamini , R.S. Sanchhali and G.Nivashni Devi, IOT Based Surveillance Robot Using Raspberry-Pi 3, International Journal of Advanced Research in Electrical, Vol. 9, Issue 2, February 2020
10. G.R.Sri Sudharsna, A.Srinivasan, Design of Wireless Video Surveillance Robot using Raspberry-Pi, International Journal of Recent Technology and Engineering (IJRTE) , Volume-8 Issue-3, September 2019

APPENDIX

SOURCE CODE

Code:

```
#include "esp_camera.h"

#include <WiFi.h>

#define CAMERA_MODEL_AI_THINKER

const char* ssid = "MSJ Broadband";
const char* password = "MSI19911993";

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

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

```

#define HREF_GPIO_NUM  23
#define PCLK_GPIO_NUM  22

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

#define Y9_GPIO_NUM  35
#define Y8_GPIO_NUM  34
#define Y7_GPIO_NUM  39
#define Y6_GPIO_NUM  36
#define Y5_GPIO_NUM  21
#define Y4_GPIO_NUM  19
#define Y3_GPIO_NUM  18
#define Y2_GPIO_NUM   5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM  23
#define PCLK_GPIO_NUM  22

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

extern int gpLb = 2;
extern int gpLf = 14;
extern int gpRb = 15;
extern int gpRf = 13;

```

```

extern int gpLed = 4;
extern String WiFiAddr = "";

void startCameraServer();

void setup()
{
    Serial.begin(115200);
    Serial.setDebugOutput(true);
    Serial.println();

    pinMode(gpLb, OUTPUT); //Left Backward
    pinMode(gpLf, OUTPUT); //Left Forward
    pinMode(gpRb, OUTPUT); //Right Forward
    pinMode(gpRf, OUTPUT); //Right Backward
    pinMode(gpLed, OUTPUT); //Light

    digitalWrite(gpLb, LOW);
    digitalWrite(gpLf, LOW);
    digitalWrite(gpRb, LOW);
    digitalWrite(gpRf, LOW);
    digitalWrite(gpLed, LOW);

    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;

```

```

config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

```

```

if(psramFound())
{
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
}

```

```

else
{
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}

```

```

esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK)
{

```

```

    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}

sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_CIF);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP());
WiFiAddr = WiFi.localIP().toString();
Serial.println("' to connect");
}

void loop()
{

}

```

Code:

```
#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "img_converters.h"
#include "camera_index.h"
#include "Arduino.h"

extern int gpLb;
extern int gpLf;
extern int gpRb;
extern int gpRf;
extern int gpLed;
extern String WiFiAddr;

void WheelAct(int nLf, int nLb, int nRf, int nRb);

typedef struct {
    size_t size; //number of values used for filtering
    size_t index; //current value index
    size_t count; //value count
    int sum;
    int * values; //array to be filled with values
} ra_filter_t;

typedef struct {
    httpd_req_t *req;
    size_t len;
} jpg_chunking_t;

#define PART_BOUNDARY "1234567890000000000000987654321"
```

```
static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary=" PART_BOUNDARY;
```

```
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";
```

```
static const char* _STREAM_PART = "Content-Type: image/jpeg\r\nContent-Length: %u\r\n\r\n";
```

```
static ra_filter_t ra_filter;
```

```
httpd_handle_t stream_httpd = NULL;
```

```
httpd_handle_t camera_httpd = NULL;
```

```
static ra_filter_t * ra_filter_init(ra_filter_t * filter, size_t sample_size){
```

```
    memset(filter, 0, sizeof(ra_filter_t));
```

```
    filter->values = (int *)malloc(sample_size * sizeof(int));
```

```
    if(!filter->values){
```

```
        return NULL;
```

```
    }
```

```
    memset(filter->values, 0, sample_size * sizeof(int));
```

```
    filter->size = sample_size;
```

```
    return filter;
```

```
}
```

```
static int ra_filter_run(ra_filter_t * filter, int value){
```

```
    if(!filter->values){
```

```
        return value;
```

```
    }
```

```
    filter->sum -= filter->values[filter->index];
```

```
    filter->values[filter->index] = value;
```

```
    filter->sum += filter->values[filter->index];
```

```
    filter->index++;
```

```

filter->index = filter->index % filter->size;
if (filter->count < filter->size) {
    filter->count++;
}
return filter->sum / filter->count;
}

```

```

static size_t jpg_encode_stream(void * arg, size_t index, const void* data, size_t
len){
    jpg_chunking_t *j = (jpg_chunking_t *)arg;
    if(!index){
        j->len = 0;
    }
    if(httpd_resp_send_chunk(j->req, (const char *)data, len) != ESP_OK){
        return 0;
    }
    j->len += len;
    return len;
}

```

```

static esp_err_t capture_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    int64_t fr_start = esp_timer_get_time();

    fb = esp_camera_fb_get();
    if (!fb) {
        Serial.printf("Camera capture failed");
        httpd_resp_send_500(req);
        return ESP_FAIL;
    }
}

```

```

httpd_resp_set_type(req, "image/jpeg");
httpd_resp_set_hdr(req, "Content-Disposition", "inline; filename=capture.jpg");

size_t fb_len = 0;
if(fb->format == PIXFORMAT_JPEG){
    fb_len = fb->len;
    res = httpd_resp_send(req, (const char *)fb->buf, fb->len);
} else {
    jpg_chunking_t jchunk = {req, 0};
    res = frame2jpg_cb(fb, 80, jpg_encode_stream,
&jchunk)?ESP_OK:ESP_FAIL;
    httpd_resp_send_chunk(req, NULL, 0);
    fb_len = jchunk.len;
}
esp_camera_fb_return(fb);
int64_t fr_end = esp_timer_get_time();
Serial.printf("JPG:  %uB  %ums", (uint32_t)(fb_len), (uint32_t)((fr_end -
fr_start)/1000));
return res;
}

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

    static int64_t last_frame = 0;
    if(!last_frame) {

```

```

    last_frame = esp_timer_get_time();
}

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

while(true){
    fb = esp_camera_fb_get();
    if (!fb) {
        Serial.printf("Camera capture failed");
        res = ESP_FAIL;
    } else {
        if(fb->format != PIXFORMAT_JPEG){
            bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf, &_jpg_buf_len);
            esp_camera_fb_return(fb);
            fb = NULL;
            if(!jpeg_converted){
                Serial.printf("JPEG compression failed");
                res = ESP_FAIL;
            }
        } else {
            _jpg_buf_len = fb->len;
            _jpg_buf = fb->buf;
        }
    }
}
if(res == ESP_OK){
    size_t hlen = snprintf((char *)part_buf, 64, _STREAM_PART, _jpg_buf_len);
    res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);
}

```

```

if(res == ESP_OK){
    res = httpd_resp_send_chunk(req, (const char *)_jpg_buf, _jpg_buf_len);
}
if(res == ESP_OK){
    res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY,
strlen(_STREAM_BOUNDARY));
}
if(fb){
    esp_camera_fb_return(fb);
    fb = NULL;
    _jpg_buf = NULL;
} else if(_jpg_buf){
    free(_jpg_buf);
    _jpg_buf = NULL;
}
if(res != ESP_OK){
    break;
}
int64_t fr_end = esp_timer_get_time();

int64_t frame_time = fr_end - last_frame;
last_frame = fr_end;
frame_time /= 1000;
uint32_t avg_frame_time = ra_filter_run(&ra_filter, frame_time);
Serial.printf("MJPG: %uB %ums (%.1ffps), AVG: %ums (%.1ffps)"
    ,(uint32_t)(_jpg_buf_len),
    (uint32_t)frame_time, 1000.0 / (uint32_t)frame_time,
    avg_frame_time, 1000.0 / avg_frame_time
);
}

```

```

    last_frame = 0;
    return res;
}

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

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

```

```

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

int val = atoi(value);
sensor_t * s = esp_camera_sensor_get();
int res = 0;

if(!strcmp(variable, "framesize")) {
    if(s->pixformat == PIXFORMAT_JPEG) res = s->set_framesize(s,
(framesize_t)val);
}
else if(!strcmp(variable, "quality")) res = s->set_quality(s, val);
else if(!strcmp(variable, "contrast")) res = s->set_contrast(s, val);
else if(!strcmp(variable, "brightness")) res = s->set_brightness(s, val);
else if(!strcmp(variable, "saturation")) res = s->set_saturation(s, val);
else if(!strcmp(variable, "gainceiling")) res = s->set_gainceiling(s,
(gainceiling_t)val);
else if(!strcmp(variable, "colorbar")) res = s->set_colorbar(s, val);
else if(!strcmp(variable, "awb")) res = s->set_whitebal(s, val);
else if(!strcmp(variable, "agc")) res = s->set_gain_ctrl(s, val);
else if(!strcmp(variable, "aec")) res = s->set_exposure_ctrl(s, val);
else if(!strcmp(variable, "hmirror")) res = s->set_hmirror(s, val);
else if(!strcmp(variable, "vflip")) res = s->set_vflip(s, val);
else if(!strcmp(variable, "awb_gain")) res = s->set_awb_gain(s, val);
else if(!strcmp(variable, "agc_gain")) res = s->set_agc_gain(s, val);
else if(!strcmp(variable, "aec_value")) res = s->set_aec_value(s, val);
else if(!strcmp(variable, "aec2")) res = s->set_aec2(s, val);
else if(!strcmp(variable, "dcw")) res = s->set_dcw(s, val);

```

```

else if(!strcmp(variable, "bpc")) res = s->set_bpc(s, val);
else if(!strcmp(variable, "wpc")) res = s->set_wpc(s, val);
else if(!strcmp(variable, "raw_gma")) res = s->set_raw_gma(s, val);
else if(!strcmp(variable, "lenc")) res = s->set_lenc(s, val);
else if(!strcmp(variable, "special_effect")) res = s->set_special_effect(s, val);
else if(!strcmp(variable, "wb_mode")) res = s->set_wb_mode(s, val);
else if(!strcmp(variable, "ae_level")) res = s->set_ae_level(s, val);
else {
    res = -1;
}

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

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

static esp_err_t status_handler(httpd_req_t *req){
    static char json_response[1024];

    sensor_t * s = esp_camera_sensor_get();
    char * p = json_response;
    *p++ = '{';

    p+=sprintf(p, "\"framesize\":%u,", s->status.framesize);
    p+=sprintf(p, "\"quality\":%u,", s->status.quality);
    p+=sprintf(p, "\"brightness\":%d,", s->status.brightness);
    p+=sprintf(p, "\"contrast\":%d,", s->status.contrast);
    p+=sprintf(p, "\"saturation\":%d,", s->status.saturation);

```

```

p+=sprintf(p, "\"special_effect\":%u,", s->status.special_effect);
p+=sprintf(p, "\"wb_mode\":%u,", s->status.wb_mode);
p+=sprintf(p, "\"awb\":%u,", s->status.awb);
p+=sprintf(p, "\"awb_gain\":%u,", s->status.awb_gain);
p+=sprintf(p, "\"aec\":%u,", s->status.aec);
p+=sprintf(p, "\"aec2\":%u,", s->status.aec2);
p+=sprintf(p, "\"ae_level\":%d,", s->status.ae_level);
p+=sprintf(p, "\"aec_value\":%u,", s->status.aec_value);
p+=sprintf(p, "\"agc\":%u,", s->status.agc);
p+=sprintf(p, "\"agc_gain\":%u,", s->status.agc_gain);
p+=sprintf(p, "\"gainceiling\":%u,", s->status.gainceiling);
p+=sprintf(p, "\"bpc\":%u,", s->status.bpc);
p+=sprintf(p, "\"wpc\":%u,", s->status.wpc);
p+=sprintf(p, "\"raw_gma\":%u,", s->status.raw_gma);
p+=sprintf(p, "\"lenc\":%u,", s->status.lenc);
p+=sprintf(p, "\"hmirror\":%u,", s->status.hmirror);
p+=sprintf(p, "\"dcw\":%u,", s->status.dcw);
p+=sprintf(p, "\"colorbar\":%u", s->status.colorbar);
*p++ = '};
*p++ = 0;
httpd_resp_set_type(req, "application/json");
httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "");
return httpd_resp_send(req, json_response, strlen(json_response));
}

```

```

static esp_err_t index_handler(httpd_req_t *req){
    httpd_resp_set_type(req, "text/html");
    String page = "";
    page += "<meta name=\"viewport\" content=\"width=device-width, initial-
scale=1.0, maximum-scale=1.0, user-scalable=0\">\n";
    page += "<script>var xhttp = new XMLHttpRequest();</script>";
}

```

```
page += "<script>function getsend(arg) { xhttp.open('GET', arg +'?' + new
Date().getTime(), true); xhttp.send() } </script>";
```

```
//page += "<p align=center><IMG SRC='http://" + WiFiAddr + ":81/stream'
style='width:280px;'></p><br/><br/>";
```

```
page += "<p align=center><IMG SRC='http://" + WiFiAddr + ":81/stream'
style='width:300px; transform:rotate(180deg);'></p><br/><br/>";
```

```
page      +=      "<p      align=center>      <button      style=background-
color:lightgrey;width:90px;height:80px      onmousedown=getsend('back')
onmouseup=getsend('stop')      ontouchstart=getsend('go')
ontouchend=getsend('stop')><b>Forward</b></button> </p>";
```

```
page += "<p align=center>";
```

```
page += "<button style=background-color:lightgrey;width:90px;height:80px;
onmousedown=getsend('right')      onmouseup=getsend('stop')
ontouchstart=getsend('right')
ontouchend=getsend('stop')><b>Left</b></button>&nbsp;";
```

```
page += "<button style=background-color:indianred;width:90px;height:80px
onmousedown=getsend('stop')
onmouseup=getsend('stop')><b>Stop</b></button>&nbsp;";
```

```
page += "<button style=background-color:lightgrey;width:90px;height:80px
onmousedown=getsend('left')      onmouseup=getsend('stop')
ontouchstart=getsend('left') ontouchend=getsend('stop')><b>Right</b></button>";
```

```
page += "</p>";
```

```
page      +=      "<p      align=center><button      style=background-
color:lightgrey;width:90px;height:80px      onmousedown=getsend('back')
onmouseup=getsend('stop')      ontouchstart=getsend('back')
ontouchend=getsend('stop')><b>Backward</b></button></p>";
```

```
page += "<p align=center>";
```

```
page += "<button style=background-color:yellow;width:140px;height:40px
onmousedown=getsend('ledon')><b>Light ON</b></button>";
```

```
page += "<button style=background-color:yellow;width:140px;height:40px
onmousedown=getsend('ledoff')><b>Light OFF</b></button>";
```

```
page += "</p>";
```

```
return httpd_resp_send(req, &page[0], strlen(&page[0]));
```

```
}
```

```
static esp_err_t go_handler(httpd_req_t *req){  
    WheelAct(HIGH, LOW, HIGH, LOW);  
    Serial.println("Go");  
    httpd_resp_set_type(req, "text/html");  
    return httpd_resp_send(req, "OK", 2);  
}
```

```
static esp_err_t back_handler(httpd_req_t *req){  
    WheelAct(LOW, HIGH, LOW, HIGH);  
    Serial.println("Back");  
    httpd_resp_set_type(req, "text/html");  
    return httpd_resp_send(req, "OK", 2);  
}
```

```
static esp_err_t left_handler(httpd_req_t *req){  
    WheelAct(HIGH, LOW, LOW, HIGH);  
    Serial.println("Left");  
    httpd_resp_set_type(req, "text/html");  
    return httpd_resp_send(req, "OK", 2);  
}
```

```
static esp_err_t right_handler(httpd_req_t *req){  
    WheelAct(LOW, HIGH, HIGH, LOW);  
    Serial.println("Right");  
    httpd_resp_set_type(req, "text/html");  
    return httpd_resp_send(req, "OK", 2);  
}
```

```
static esp_err_t stop_handler(httpd_req_t *req){  
    WheelAct(LOW, LOW, LOW, LOW);  
    Serial.println("Stop");  
}
```

```

    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, "OK", 2);
}

static esp_err_t ledon_handler(httpd_req_t *req){
    digitalWrite(gpLed, HIGH);
    Serial.println("LED ON");
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, "OK", 2);
}

static esp_err_t ledoff_handler(httpd_req_t *req){
    digitalWrite(gpLed, LOW);
    Serial.println("LED OFF");
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, "OK", 2);
}

void startCameraServer(){
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();

    httpd_uri_t go_uri = {
        .uri      = "/go",
        .method    = HTTP_GET,
        .handler   = go_handler,
        .user_ctx  = NULL
    };

    httpd_uri_t back_uri = {
        .uri      = "/back",
        .method    = HTTP_GET,
        .handler   = back_handler,

```

```

        .user_ctx = NULL
};

httpd_uri_t stop_uri = {
    .uri      = "/stop",
    .method   = HTTP_GET,
    .handler  = stop_handler,
    .user_ctx = NULL
};

httpd_uri_t left_uri = {
    .uri      = "/left",
    .method   = HTTP_GET,
    .handler  = left_handler,
    .user_ctx = NULL
};

httpd_uri_t right_uri = {
    .uri      = "/right",
    .method   = HTTP_GET,
    .handler  = right_handler,
    .user_ctx = NULL
};

httpd_uri_t ledon_uri = {
    .uri      = "/ledon",
    .method   = HTTP_GET,
    .handler  = ledon_handler,
    .user_ctx = NULL
};

```

```
httpd_uri_t ledoff_uri = {  
    .uri      = "/ledoff",  
    .method   = HTTP_GET,  
    .handler  = ledoff_handler,  
    .user_ctx = NULL  
};
```

```
httpd_uri_t index_uri = {  
    .uri      = "/",  
    .method   = HTTP_GET,  
    .handler  = index_handler,  
    .user_ctx = NULL  
};
```

```
httpd_uri_t status_uri = {  
    .uri      = "/status",  
    .method   = HTTP_GET,  
    .handler  = status_handler,  
    .user_ctx = NULL  
};
```

```
httpd_uri_t cmd_uri = {  
    .uri      = "/control",  
    .method   = HTTP_GET,  
    .handler  = cmd_handler,  
    .user_ctx = NULL  
};
```

```
httpd_uri_t capture_uri = {  
    .uri      = "/capture",  
    .method   = HTTP_GET,
```

```

        .handler = capture_handler,
        .user_ctx = NULL
};

```

```

httpd_uri_t stream_uri = {
    .uri      = "/stream",
    .method   = HTTP_GET,
    .handler  = stream_handler,
    .user_ctx = NULL
};

```

```

ra_filter_init(&ra_filter, 20);
Serial.printf("Starting web server on port: '%d'", config.server_port);
if (httpd_start(&camera_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(camera_httpd, &index_uri);
    httpd_register_uri_handler(camera_httpd, &go_uri);
    httpd_register_uri_handler(camera_httpd, &back_uri);
    httpd_register_uri_handler(camera_httpd, &stop_uri);
    httpd_register_uri_handler(camera_httpd, &left_uri);
    httpd_register_uri_handler(camera_httpd, &right_uri);
    httpd_register_uri_handler(camera_httpd, &ledon_uri);
    httpd_register_uri_handler(camera_httpd, &ledoff_uri);
}

```

```

config.server_port += 1;
config.ctrl_port += 1;
Serial.printf("Starting stream server on port: '%d'", config.server_port);
if (httpd_start(&stream_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(stream_httpd, &stream_uri);
}

```

```
}
```

```
void WheelAct(int nLf, int nLb, int nRf, int nRb)
```

```
{
```

```
    digitalWrite(gpLf, nLf);
```

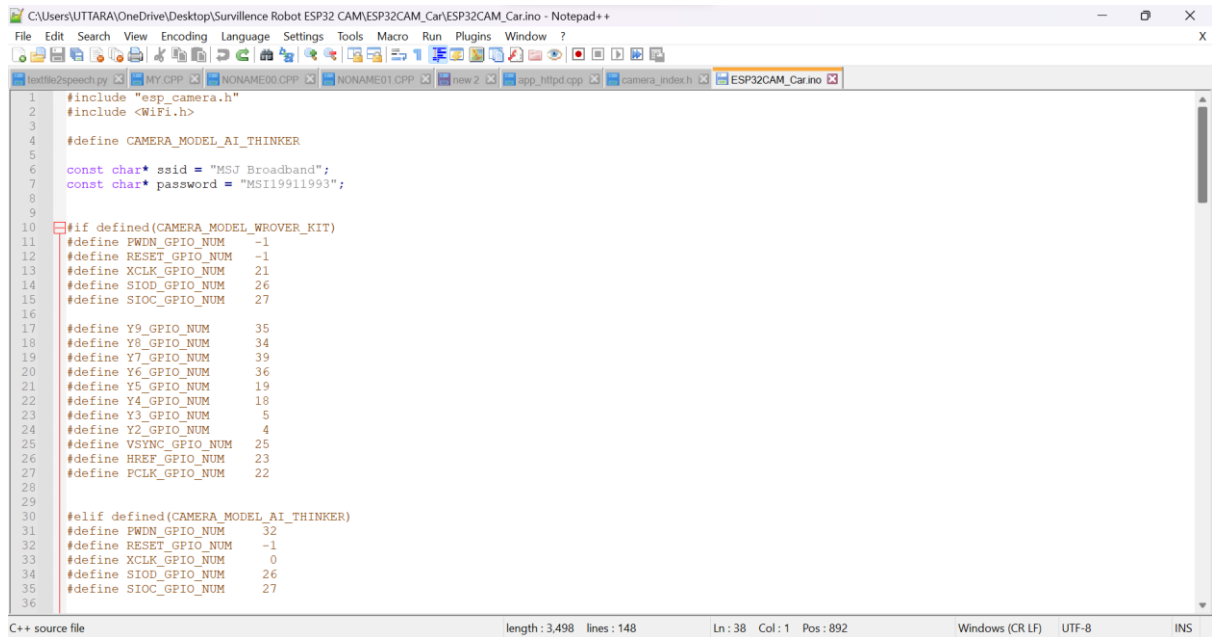
```
    digitalWrite(gpLb, nLb);
```

```
    digitalWrite(gpRf, nRf);
```

```
    digitalWrite(gpRb, nRb);
```

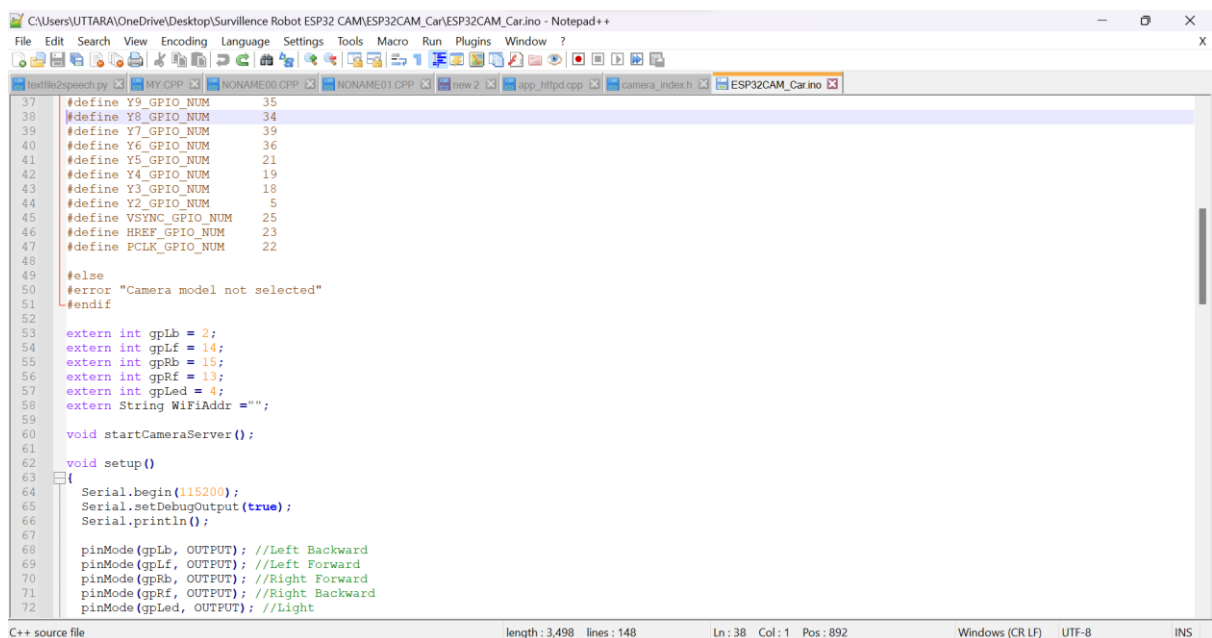
```
}
```

SCREENSHOTS



```
1 #include "esp_camera.h"
2 #include <WiFi.h>
3
4 #define CAMERA_MODEL_AI_THINKER
5
6 const char* ssid = "MSJ_Broadband";
7 const char* password = "MSI19911993";
8
9
10 #if defined(CAMERA_MODEL_WROVER_KIT)
11 #define PWDN_GPIO_NUM    -1
12 #define RESET_GPIO_NUM  -1
13 #define XCLK_GPIO_NUM    21
14 #define SIOD_GPIO_NUM    26
15 #define SIOC_GPIO_NUM    27
16
17 #define Y9_GPIO_NUM       35
18 #define Y8_GPIO_NUM       34
19 #define Y7_GPIO_NUM       39
20 #define Y6_GPIO_NUM       36
21 #define Y5_GPIO_NUM       19
22 #define Y4_GPIO_NUM       18
23 #define Y3_GPIO_NUM       5
24 #define Y2_GPIO_NUM       4
25 #define VSYNC_GPIO_NUM    25
26 #define HREF_GPIO_NUM     23
27 #define PCLK_GPIO_NUM     22
28
29
30 #elif defined(CAMERA_MODEL_AI_THINKER)
31 #define PWDN_GPIO_NUM     32
32 #define RESET_GPIO_NUM    -1
33 #define XCLK_GPIO_NUM     0
34 #define SIOD_GPIO_NUM     26
35 #define SIOC_GPIO_NUM     27
36
```

C++ source file length: 3,498 lines: 148 Ln: 38 Col: 1 Pos: 892 Windows (CR LF) UTF-8 INS



```
37 #define Y9_GPIO_NUM       35
38 #define Y8_GPIO_NUM       34
39 #define Y7_GPIO_NUM       39
40 #define Y6_GPIO_NUM       36
41 #define Y5_GPIO_NUM       21
42 #define Y4_GPIO_NUM       19
43 #define Y3_GPIO_NUM       18
44 #define Y2_GPIO_NUM       5
45 #define VSYNC_GPIO_NUM    25
46 #define HREF_GPIO_NUM     23
47 #define PCLK_GPIO_NUM     22
48
49 #else
50 #error "Camera model not selected"
51 #endif
52
53 extern int gpLb = 2;
54 extern int gpLf = 14;
55 extern int gpRb = 15;
56 extern int gpRf = 13;
57 extern int gpLed = 4;
58 extern String WiFiAddr = "";
59
60 void startCameraServer();
61
62 void setup()
63 {
64   Serial.begin(115200);
65   Serial.setDebugOutput(true);
66   Serial.println();
67
68   pinMode(gpLb, OUTPUT); //Left Backward
69   pinMode(gpLf, OUTPUT); //Left Forward
70   pinMode(gpRb, OUTPUT); //Right Forward
71   pinMode(gpRf, OUTPUT); //Right Backward
72   pinMode(gpLed, OUTPUT); //Light

```

C++ source file length: 3,498 lines: 148 Ln: 38 Col: 1 Pos: 892 Windows (CR LF) UTF-8 INS

```
C:\Users\UTTARA\OneDrive\Desktop\Surveillance Robot ESP32 CAM\ESP32CAM_Car\ESP32CAM_Car.ino - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
textile2speech.py MY.CPP NONAME00.CPP NONAME01.CPP new 2 app_httpd.cpp camera_index.h ESP32CAM_Car.ino
73
74 digitalWrite(gpLb, LOW);
75 digitalWrite(gpLf, LOW);
76 digitalWrite(gpRb, LOW);
77 digitalWrite(gpRf, LOW);
78 digitalWrite(gpLed, LOW);
79
80 camera_config_t config;
81 config.ledc_channel = LEDC_CHANNEL_0;
82 config.ledc_timer = LEDC_TIMER_0;
83 config.pin_d0 = Y2_GPIO_NUM;
84 config.pin_d1 = Y3_GPIO_NUM;
85 config.pin_d2 = Y4_GPIO_NUM;
86 config.pin_d3 = Y5_GPIO_NUM;
87 config.pin_d4 = Y6_GPIO_NUM;
88 config.pin_d5 = Y7_GPIO_NUM;
89 config.pin_d6 = Y8_GPIO_NUM;
90 config.pin_d7 = Y9_GPIO_NUM;
91 config.pin_xclk = XCLK_GPIO_NUM;
92 config.pin_pclk = PCLK_GPIO_NUM;
93 config.pin_vsync = VSYNC_GPIO_NUM;
94 config.pin_href = HREF_GPIO_NUM;
95 config.pin_sscb_sda = SIOD_GPIO_NUM;
96 config.pin_sscb_scl = SIOC_GPIO_NUM;
97 config.pin_pwdn = PWDN_GPIO_NUM;
98 config.pin_reset = RESET_GPIO_NUM;
99 config.xclk_freq_hz = 20000000;
100 config.pixel_format = PIXFORMAT_JPEG;
101
102 if(paramFound())
103 {
104     config.frame_size = FRAMESIZE_UXGA;
105     config.jpeg_quality = 10;
106     config.fb_count = 2;
107 }
108
C++ source file length: 3,498 lines: 148 Ln: 38 Col: 1 Pos: 892 Windows (CR LF) UTF-8 INS
```

```
C:\Users\UTTARA\OneDrive\Desktop\Surveillance Robot ESP32 CAM\ESP32CAM_Car\ESP32CAM_Car.ino - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
textile2speech.py MY.CPP NONAME00.CPP NONAME01.CPP new 2 app_httpd.cpp camera_index.h ESP32CAM_Car.ino
109 else
110 {
111     config.frame_size = FRAMESIZE_SVGA;
112     config.jpeg_quality = 12;
113     config.fb_count = 1;
114 }
115
116 esp_err_t err = esp_camera_init(&config);
117 if (err != ESP_OK)
118 {
119     Serial.printf("Camera init failed with error 0x%x", err);
120     return;
121 }
122
123 sensor_t * s = esp_camera_sensor_get();
124 s->set_framesize(s, FRAMESIZE_CIF);
125
126 WiFi.begin(ssid, password);
127
128 while (WiFi.status() != WL_CONNECTED)
129 {
130     delay(500);
131     Serial.print(".");
132 }
133 Serial.println("");
134 Serial.println("WiFi connected");
135
136 startCameraServer();
137
138 Serial.print("Camera Ready! Use 'http://'");
139 Serial.print(WiFi.localIP());
140 WiFiAddr = WiFi.localIP().toString();
141 Serial.println(" to connect");
142
143
144 void loop()
```

C:\Users\UTTARA\OneDrive\Desktop\Surveillance Robot ESP32 CAM\ESP32CAM_Car\ESP32CAM_Car.ino - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

ESP32CAM_Car.ino

```
136 startCameraServer();
137
138 Serial.print("Camera Ready! Use 'http://");
139 Serial.print(WiFi.localIP());
140 WiFiAddr = WiFi.localIP().toString();
141 Serial.println("' to connect");
142 }
143
144 void loop()
145 {
146
147 }
148
```

C++ source file length: 3,498 lines: 148 Ln: 38 Col: 1 Pos: 892 Windows (CR LF) UTF-8 INS

