ISSN 2063-5346

SURVEILLANCE ROBOT USING ESP32 CAM

Menaka M[1], Stella K[2], Ramya D[3], Sakthi Sri R[4], Thirisha M[5]

Article History: Received: 01.02.2023 Revised: 07.03.2023 Accepted: 10.04.2023

Abstract

A surveillance robot using ESP32-CAM is a system that utilizes the ESP32-CAM board and a robot chassis to create a mobile surveillance device. The ESP32-CAM is alow-cost development board that integrates a small camera module and Wi-Fi connectivity. The robot chassis allows the device to move around and capture video in different locations. The system can be controlled through a web interface hosted on the ESP32-CAM board. The web interface allows the user to control the robot's movement, view live video streams, and take snapshots of the video feed. Additionally, the system can be programmed to detect motion using computer vision algorithms, such as object detection and tracking, and send alerts to the user. The surveillance robot using ESP32-CAM has potential applications in home security, monitoring of remote locations, and industrial surveillance. With its low cost and easy-to-use interface, it provides a convenient solution for anyone who needs to monitor their surroundings remotely

Keyword- ESP 32cam, motor driver, camera.

m.menaka@velhightech.com[1], stellaakk16@gmail.com[2], ramyasuja4@gmail.com[3], rajakumar72196@gmail.com[4], thirishamohanasundaram@gmail.com[5]

[1]Assistant Professor,[2]Associate Professor, Department of Electronics and Communication Engineering,

Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai, India.

[3,4,5]UG Student of Department of Electronics and Communication Engineering,

Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai, India

DOI:10.31838/ecb/2023.12.s1-B.316

.

I. INTRODUCTION

The development of surveillance robots has been a growing area of interest due to its applications in various fields such as security, industrial monitoring, and home automation. The surveillance robot using the ESP32-CAM is a project that focuses on developing a low-cost and efficient surveillance robot for indoor and outdoor use.

Surveillance robots are becoming increasingly popular in modern times due to their ability to monitor and collect information from a remote location. The use of the ESP32 CAM module, which is an integrated camera module with Wi-Fi and Bluetooth connectivity, has made it possible to create surveillance robots that can be controlled remotely.

The ESP32 CAM module is a small-sized camera module that can be easily integrated into a robot. The module provides high-quality images and can be controlled using the ESP32 microcontroller. The ESP32 CAM module also has Wi-Fi and Bluetooth connectivity, which allows for easy communication with remote control station or a computer.

With the ESP32 CAM module, it is possible to create a surveillance robot that can be controlled remotely from a computer or a mobile device. The robot can be programmed to move around a specific area and capture images or videos of the surroundings. The images and videos can be transmitted wirelessly to the remote-control station, where they can be viewed and analyzed in real-time.

The use of the ESP32 CAM module in surveillance robots has many advantages, including the ability to capture high quality images and videos, the ability to control the robot remotely, and the ability to transmit data wirelessly. Additionally, the ESP32 CAM module is easy to use and can be integrated into a robot without requiring any additional hardware.

Overall, the ESP32 CAM module is an excellent tool for creating surveillance robots that can be used in a variety of applications, including security, monitoring, and inspection.

II. METHODOLOGY:

The ESP32-CAM is a low-cost microcontroller that integrates a camera module and Wi-Fi capabilities. In this project, we use the ESP32-CAM to control the movement of the robot and capture video and images that can be viewed remotely. The robot is equipped with a set of motors that enable it to move in different directions.

The methodology for developing a surveillance robot using ESP32-CAM can be divided into several steps:

Hardware setup:

The first step is to assemble the hardware components required for the project. This includes the ESP32-CAM module, motor drivers, motors, wheels, chassis, battery, and infrared sensor. The ESP32-CAM module is connected to the motor drivers and the infrared sensor using jumper wires. The motors are connected to the motor drivers, and the wheels are attached to the motors. The battery is connected to the ESP32-CAM module and the motor drivers to provide power to the robot.

Programming:

The next step is to program the ESP32-CAM module to control the movement of the robot and capture video and images. The programming is done using the Arduino IDE and the ESP32-CAM library. The code includes functions to control the motors, capture video and images, and send them to a remote server using Wi-Fi.

Testing:

Once the programming is completed, the robot is tested to ensure that it is functioning as intended. The robot is placed in different environments to test its movement and obstacle detection capabilities. The video and images captured by the robot are also checked for quality and clarity.

Remote access:

The final step is to enable remote access to the robot. This is done by setting up a server that can receive the video and images captured by the robot. The server can be accessed using a smartphone or a computer, allowing users to view the surveillance footage in real-time.

In summary, the methodology for developing a surveillance robot using ESP32-CAM involves assembling the hardware components, programming the ESP32-CAM module, testing the robot, and enabling remote access to the surveillance footage.

III. HARDWARE COMPONENTS

3.1. ESP32cam

The ESP32-CAM shown in Fig.3.1.1 is a small-sized camera module that is based on the ESP 32 microcontroller and OV2640 sensor. It is capable of capturing images, streaming video, and performing various image processing tasks. The ESP32CAM module also features Wi-Fi and Bluetooth connectivity, making it ideal for 10T and surveillance applications.

Here are some key features of the ESP32-CAM:

1. ESP32 microcontroller
2. OV2640 2 MP camera sensor
3. Wi-Fi and Bluetooth connectivity
4. GPIO pins for interfacing with other devices
5. Micro SD card slot for storage
6. 5 V DC power supply

To use the ESP32-CAM module, you need to have some knowledge of programming, especially with the Arduino IDE, as it is commonly used to program the ESP32-CAM. You will also need to have some basic electronics knowledge, such as how to connect wires and components to the GPIO pins of the module shown in Fig 3.5.1.

There are many projects that you can do with the ESP32CAM, such as building a smart security camera, a remote-controlled car with a camera, or even a face recognition system. With its small size and powerful features, the ESP32-CAM is a versatile and powerful tool for makers and hobbyists.

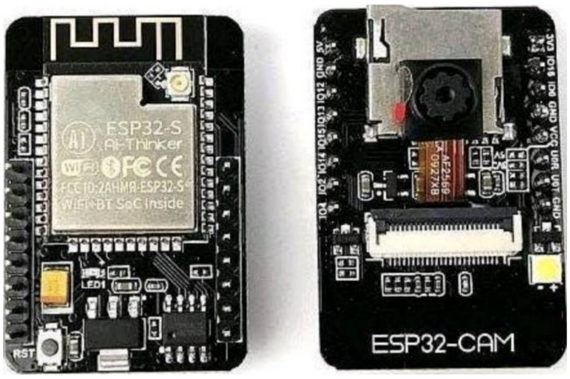


Fig.3.1.1 ESP32 cam

3.2. MOTOR DRIVER

The L298N is a popular motor driver IC (integrated circuit) that can control the direction and speed of DC motors and stepper motors. It can handle up to 2 amps of continuous current per channel, and has a wide input voltage range of 5V to 35V.

The L298N has two H-bridge circuits, which are used to control the direction of the motor. Each H-bridge consists of four transistors that can be controlled independently. By switching these transistors on and off in the correct sequence, the motor can be driven in either direction.

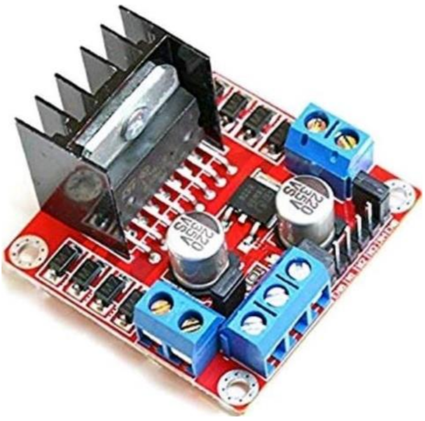


Fig.3.2.1 L298N Motor driver

To use the L298N, you need to connect it to your microcontroller or other control circuitry. There are several pins on the L298N shown in Fig 3.2.1 that you need to connect to:

ENA and ENB: These are the enable pins for channels A and B, respectively. You can connect these to digital output pins on your microcontroller to enable or disable the motor channels.

IN 1, IN2, IN3, and IN4: These are the control pins for the H-bridge circuits. By setting the correct combination of high and low signals on these pins, you can control the direction and speed of the motor.

OUTI, OUT2, OUT3, and OUT4: These are the output pins that connect to the motor.

To connect the L298N to a motor, you need to connect the motor to the OUT pins, and also connect the motor power supply to the L298N's V+ and GND pins.

It's important to note that the L298N can get quite hot when driving high-current motors, so you should use a heat sink to dissipate the heat. You should also be careful not to exceed the maximum current ratings of the L298N, or you may damage the IC

3.3 FDTI programmer

FTDI stands for Future Technology Devices International, and an FTDI programmer is a device used to program and communicate with electronic devices that use FTDI chips for their USB communication.

An FTDI programmer in Fig 3.3.1 typically consists of a USB-to serial converter chip from FTDI and a connector that can be used to interface with the target device. The programmer provides a simple and reliable way to communicate with the device, allowing for the transfer of data and programming instructions between the device and a computer.

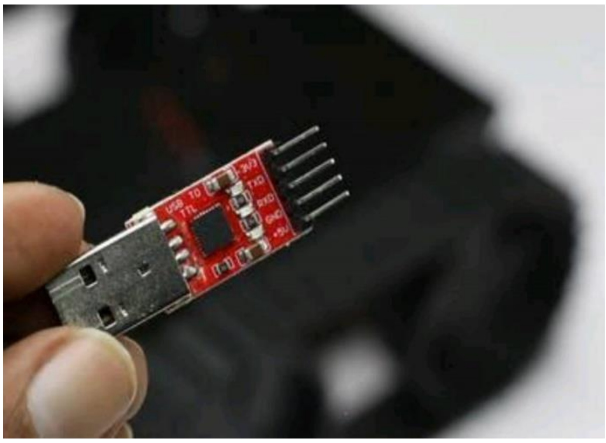


Fig.3.3.1 FTDI Programmer

FTDI programmers are commonly used to program and debug microcontrollers, such as those based on the Atmel AVR and Microchip PIC architectures. They are also used in other applications where a USB-to-serial interface is needed, such as in GPS receivers, data loggers, and other electronic devices.

In addition to providing a USB-to-serial interface, FTDI programmers often include additional features such as voltage regulation, programmable clock sources, and other features that make them useful for a wide range of applications.

Overall, FTDI programmers are an essential tool for many electronics designers and developers, providing a reliable and easy-to-use method for programming and communicating with a wide range of electronic devices.

i) Pin configuration of FDTI Programmer

FTDI chips and programmers typically use a 6-pin connector called a "FTDI cable" or "FTDI header". The pinout of the FTDI cable or header is as follows:

- DTR (Data Terminal Ready)
- RXD (Receive Data)
- TXD (Transmit Data)
- VCC (Voltage Supply)
- CTS (Clear to send)
- GND (Ground)

Here's a brief description of each pin:

DTR - This pin is used to reset the microcontroller or target device. The programmer sends a signal to this pin to reset the device before uploading new code.

RXD - This pin is used to receive data from the target device. Data is transmitted from the target device to the programmer through this pin.

TXD - This pin is used to transmit data to the target device. Data is transmitted from the programmer to the target device through this pin.

VCC - This pin provides power to the target device. The voltage level can vary depending on the programmer and the target device, but it's typically 3.3 V or 5 V.

CTS - This pin is used for flow control. It tells the target device when it's safe to transmit data.

GND - This pin provides a common ground reference for the programmer and the target device.

It's important to note that the pinout may vary depending on the specific FTDI programmer or cable being used. It's always a good idea to consult the documentation for the specific device you're using to ensure you're connecting it correctly.

3.4 The interface between esp32 cam and FDTI programmer

The ESP32-CAM is a small camera module based on the ESP 32 microcontroller, which includes Wi-Fi and Bluetooth connectivity. The module features an FTDI chip that allows it to communicate with a computer through a USB port using an FTDI programmer.

To interface an ESP32-CAM with an FTDI programmer, you need to connect the FTDI programmer to the module's UART pins. Here's how to do it:

- Connect the FTDI programmer's GND pin to the ESP32-CAM's GND pin.
- Connect the FTDI programmer's TX pin to the ESP32-CAM's RX pin.
- Connect the FTDI programmer's RX pin to the ESP32-CAM's TX pin.
- Connect the FTDI programmer's DTR pin to the ESP32-CAM's GPIOO pin.
- Connect the FTDI programmer's VCC pin to the ESP32-CAM's 3.3V pin.

Once the connections are made, you can use a program like the Arduino IDE or the ESP-IDF (ESP32 10T Development Framework) to upload code to the ESP32CAM module. To do this, you'll need to put the module in programming mode by pulling its GPIOO pin low while resetting it. The FTDI programmer's DTR pin can be used to reset the module automatically by pulling the GPIOO pin low for a short time when programming begins.

It's worth noting that the specific pinout and programming process may vary depending on the ESP32CAM module and FTDI programmer you're using. Always consult the documentation for your specific hardware to ensure you're connecting it correctly and following the appropriate programming steps.

3.5 Interface between esp32 cam and 1298N

The L298N is a dual H-bridge motor driver that can be used to control two DC motors or a single bipolar stepper motor. It can also be used to drive other types of loads that require bidirectional current control.

To interface an ESP 32-CAM with an L298N, you can use the following steps:

- Connect the ESP32-CAM's 5V pin to the L298N's VCC pin.
- Connect the ESP32-CAM's GND pin to the L298N's GND pin.
- Connect the ESP32-CAM's GPIO pins to the L298N's control pins as follows:

i. GPIOx to INI

ii. GPIOy to IN2

iii. GPIOz to ENA

The specific GPIO pins used will depend on the code you're using to control the L298N. For example, if you're using the Arduino IDE, you can use the pin-Mode() and digital-Write() functions to control the GPIO pins

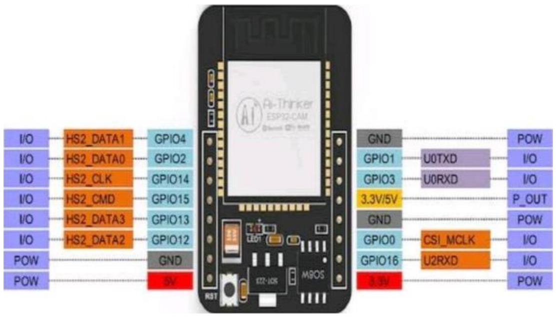


Fig.3.5.1 GPIO Pins

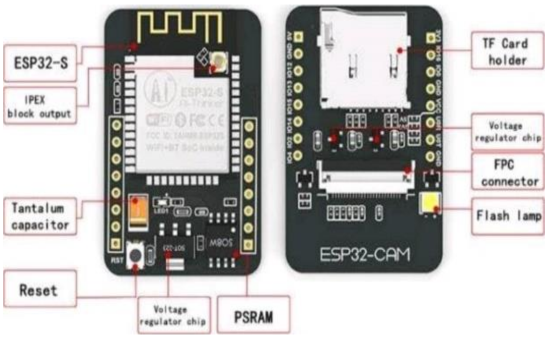


Fig.3.5.2 Part of ESP32 CAM

Connect the L298N's motor terminals to the DC motor or bipolar stepper motor you want to control.

Once the connections are made, you can program the ESP32-CAM to control the L298N and the attached motor. The specific code required will depend on the motor you're using and the specific control scheme you want to implement. For example, if you're using a DC motor, you can use PWM signals on the ENA pin to control its speed and the INI/IN2 pins to control its direction.

It's important to note that the L298N can draw a significant amount of current, so you may need to use an external power supply to provide sufficient power to the motor. Always consult the datasheet for your specific motor and motor driver to ensure you're using them correctly and safely.

3.6 The power supply to be provided

The power supply required for an ESP32-CAM and an L298N motor driver will depend on the specific motors you're using and their voltage and current requirements. However, here are some general guidelines to help you determine the appropriate power supply:

The ESP32-CAM can be powered by a 5 V power supply. The recommended operating voltage range is 2.7 V to 3.6V, but it can be powered up to 5.5 V. You can use a USB cable connected to a computer or a USB power adapter to power the ESP32-CAM.

The L298N motor driver can be powered by a separate power supply from the ESP32-CAM. The recommended operating voltage range is 5V to 35 V, and the maximum current per channel is 2A. You should choose a power supply with a voltage and current rating that is appropriate for the motors you're using.

If you're using a high-power motor, you may need to use an external power supply with a higher voltage and current rating to ensure the motor operates correctly. In this case, you'll need to ensure that the ground of the external power supply is connected to the ground of the ESP32-CAM and L298N to ensure they share a common ground.

Be sure to check the power consumption of your circuit and the power rating of your power supply to ensure it can supply enough power to all components.

It's important to note that incorrect power supply connections or voltage/current ratings can damage the ESP32-CAM and L298N or even cause a safety hazard. Always refer to the datasheets of your components and consult an expert if you're unsure about the appropriate power supply.

3.7. Gear Motor

A gear motor is a type of motor that incorporates a gearhead, which is a gear assembly that is designed to reduce the speed of the motor while increasing its torque. Gear motors are widely used in a variety of industrial, commercial, and consumer applications that require precise and controlled movement.

The motor in a gear motor can be an AC motor or a DC motor, and the gearhead can consist of various types of gears, such as spur gears, worm gears, helical gears, or planetary gears. The gearhead is typically attached to the motor shaft and is designed to convert the high-speed, low torque output of the motor into a lower-speed, higher-torque output.

Gear motors are used in many applications, such as robotics, conveyor systems, industrial machinery, automotive equipment, and many more. They are preferred over regular motors because they provide greater control, accuracy, and efficiency. In addition, gear motors are generally more reliable and have a longer lifespan than regular motors.

When selecting a gear motor, it is important to consider the application and the specific requirements, such as the Srequired torque and speed, the duty cycle, the environment, and the power source.

3.8. Interface between gear motor and L298N

The L298N is a dual H-bridge motor driver IC that can be used to drive DC motors, stepper motors, and other types of motors. A gear motor is a type of motor that has a gearbox attached to it, which allows for higher torque and slower speed.

To interface a gear motor with the L298N, you will need to connect the motor to the output pins of the L298N and provide power to the L298N to control the motor. Here are the steps you can follow:

Connect the motor to the output pins of the L298N. The L298N has two output channels, so you can connect one motor to each channel. Each output channel has two pins: an enable pin and two direction control pins. Connect the motor leads to the two output pins of one channel, and repeat for the other channel if you are using two motors.

Provide power to the L298N. The L298N requires a separate power supply to drive the motors. Connect a 5 V or 12V power supply to the VCC pin of the L298N, and connect the ground of the power supply to the ground pin of the L298N(Refer the Fig.3.8.1 Description Table).

Connect the control signals to the L298N. The L298N has two input pins for each motor channel: an enable pin and two direction control pins. Connect the enable pin to a PWM output pin on your microcontroller or other control circuit. The direction control pins can be connected to any digital output pins on your microcontroller or control circuit.

	Description	Function
power	Logic Voltage Input	Power supply to the module, connect to 5V
GND	Common Ground	Connect to ground.
vcc	Drive Voltage Input	Power Supply to the motor: 4.8V to 16V.
1M, IN2, IN3, IN4	Motor Control Pin	Control motors rotate direction, either clockwise or anticlockwise.
OUTI: OUT2, OUT3, OUT4	Motor Terminal	Connect to the motors.
Ll.L2, L3,L4	Motor Status LED	Indicate the status ofthe motors.

SWI	Share Power Source Selected Switch	ON: Share the drive power to logic power. OFF: Logic power isolated from drive power:
ENA ENB	Motor PWM Control pin	Control PWM (speed) of the motors.

Fig.3.8.1 Description Table

Write code to control the motor. To control the motor, you will need to write code to generate PWM signals for the enable pins and set the direction control pins to control the direction of the motor. The specific code will depend on your microcontroller or control circuit, and the programming language you are using.

Overall, interfacing a gear motor with the L298N is a fairly simple process that involves connecting the motor to the output pins of the L298N and providing power and control signals to the L298N. With the right code, you can control the speed and direction of the motor to meet your specific needs.

3.9. SOTWARE USED

Arduino IDE (Integrated Development Environment) is a software platform used to program and develop applications for the Arduino microcontroller boards. Arduino is an opensource hardware and software platform used to build electronics projects, which is designed for makers, hobbyists, and professionals.

The Arduino IDE provides a simple and user-friendly interface to write, compile, and upload code to Arduino boards. It supports the C and C++ programming languages,

and also provides a range of libraries and functions that can

be used to interface with various sensors, actuators, and other

components.

With the Arduino IDE, you can easily create custom programs for your Arduino board, such as controlling an LED, reading data from sensors, and communicating with other devices. The IDE provides a built-in serial monitor to debug your code and view the output from your Arduino board.

The Arduino IDE is available for free and can be downloaded from the Arduino website. It is compatible with various operating systems, including Windows, Mac, and Linux. The Arduino IDE is also open-source, which means that the source code is available for anyone to view and modify.

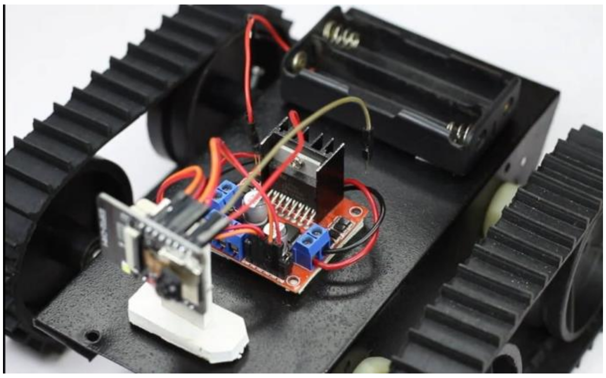
The Arduino IDE (Integrated Development Environment) is a software tool used to write and upload code to a variety of microcontrollers, including the ESP32-CAM.The ESP32-CAM is a development board that combines an ESP32-S chip, a camera module, and other components to create a versatile platform for building Internet of Things (IoT) projects.

To program the ESP32-CAM using the Arduino IDE, you will need to install the ESP32 board package in the IDE and select the appropriate board and port settings. You will also need to install any necessary libraries for your project.

Once you have everything set up, you can use the Arduino IDE to write code for the ESP32-CAM and upload it to the board via a USB cable. The IDE provides a range of useful features, including a code editor, a serial monitor, and a debugger, which can help you develop and debug your code.

IV. RESULT

A surveillance robot using an ESP32 Cam is a powerful and versatile device that can be used in a variety of applications. Whether you want to monitor your home or office, keep an eye on your pets, or even use it in industrial settings, the ESP32 cam has the features and capabilities to get the job done



.

Fig 4.1 Hardware setup

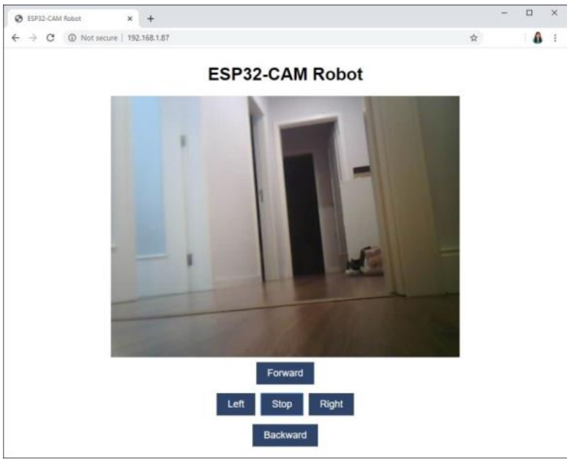


Fig 4.2 Software Output of the live stream

V. CONCLUSION

In conclusion, the surveillance robot using ESP32-CAM is a promising project that can be used for various applications, such as monitoring and surveillance of homes or offices. The ESP32-CAM module provides a compact and cost-effective solution for capturing images and streaming video. By integrating it with a robot platform, it is possible to remotely control the robot and view live video footage from the camera. With the increasing demand for remote monitoring and surveillance, this project has the potential to provide a practical and affordable solution for many different scenarios.

REFERRENCES:

1. A. Arthi., et al. “Smart Spy Surveillance Robot System” International Journal Research & Technology (IJERT) ISSN:2278-0181, Special Issue-2018.
2. “Design and Implementation of SPY Robot”, IJARIIE-ISSN(0)-4396, Vol- 5 Issue-5 2019.
3. Liu, C., et al. "Design and implementation of a wireless surveillance robot based on ESP32-CAM." IEEE Access, vol. 9, 2021, pp. 14582-14591.
4. Kumar, S., et al. "Design and implementation of a low-cost surveillance robot for indoor applications." International Journal of Computer Applications, vol. 180, no. 18, 2018, pp. 9-14.
5. Wang, Y., et al. "Design and implementation of a wireless robot for environmental monitoring." Sensors, vol. 21, no. 1, 2021, p. 36.
6. Nagendra, N., et al. "Design and implementation of a low-cost surveillance robot using Raspberry Pi." International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 12, 2019, pp. 46-50.
7. Noh, J., et al. "Development of a surveillance robot with obstacle avoidance capability." International Journal of Advanced Robotic Systems, vol. 15, no. 1, 2018, pp. 1-8.
8. Singh, A., et al. "Smart security system using ESP32 and Raspberry Pi." International Journal of Engineering and Advanced Technology, vol. 9, no. 1, 2019, pp. 35-38.
9. Waghmare, D. "Implementation of a low-cost security robot using ESP32-CAM." International Journal of Engineering Research and

Applications, vol. 10, no. 2, 2020, pp. 34-38.

10. Zhao, S., et al. "Design and implementation of a remote monitoring system for intelligent robots." Journal of Physics: Conference Series, vol. 1761, 2021, p. 012026.
11. Chen, W., et al. "Design of a mobile robot for environmental monitoring." IEEE Transactions on Instrumentation and Measurement, vol. 67, no. 8, 2018, pp. 1899-1908.
12. Kambli, R., et al. "Implementation of a wireless
13. surveillance robot using ESP32-CAM." International Journal of Advanced Research in Computer Engineering & Technology, vol. 7, no. 5, 2018, pp. 93-96.
14. Zou, J., et al. "A novel design of wireless monitoring robot." Journal of Physics: Conference Series, vol. 1749, 2021, p. 012015.
15. Yarlagadda, P., et al. "Design and implementation of a wireless robot for industrial monitoring." Journal of Robotics and Mechatronics, vol. 33, no. 2, 2021, pp. 399-404.