

Empowered environmental intelligence using ESP-32 CAM

Submitted in partial fulfillment of the
requirements for the award of

Bachelor of Technology Degree in Information Technology

By

PRAVEEN.N (Reg.No - 40120068)

PRASANNA KUMAR.S (Reg.No - 40120066)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Category-1 University by UGC

Accredited with Grade "A++" by NAAC | 12B Status by UGC |

Approved by AICTE

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119.**

APRIL 2024



SATHYABAMA

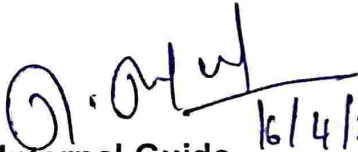
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

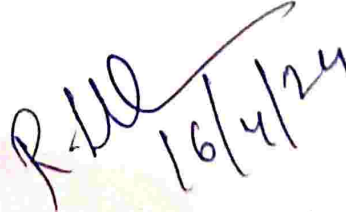
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **Praveen.N(40120068)** and **Prasanna Kumar.S(40120066)** who carried the project entitled "Empowered environmental intelligence using ESP-32 CAM" under my supervision from January 2024 to April 2024.


Internal Guide 16/4/24

Ms. SAGEENGRANA M.E,

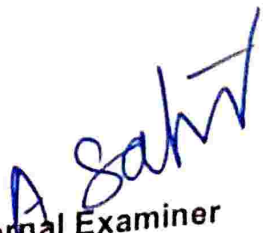

16/4/24

Head of the Department

Dr. R. SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on 17/4/24


Internal Examiner

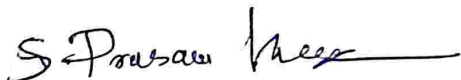

External Examiner

DECLARATION

I, Prasanna kumar.S hereby declare that the Project Report entitled "Empowered environmental intelligence using ESP-32 CAM" done by me under the guidance of Ms. Sageengrana M.E, is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 17/4/24

PLACE: Chennai


SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

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

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. R. Subhashini, M.E., Ph.D.**, Head of the Department of Information Technology 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 **Ms. Sageengrana, M.E.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

ABSTRACT

This paper presents a novel Internet of Things (IoT) system for comprehensive and user-friendly environmental monitoring. The system integrates various sensors, including temperature and humidity (DHT11), air quality (MQ Gas), vibration, and a camera (ESP32-CAM), to capture a holistic picture of the environment in real-time. The acquired data are transmitted to a central IoT platform for centralized monitoring and analysis, enabling remote access and control through a mobile app. For immediate on-site feedback, a 4-digit display presents sensor readings. Additionally, the system incorporates a servo motor that executes physical actions based on sensor inputs, adding a dynamic and interactive layer. This paper also explores the potential integration of voice assistants for hands-free operation. The system employs efficient power management, making it suitable for both wired and battery-powered applications, thus increasing its adaptability to diverse environments. This comprehensive approach to environmental monitoring holds significant potential for improving safety, comfort, efficiency, and peace of mind. Machine learning methods for pattern recognition and data analysis are employed to increase the system's intelligence. The system can identify abnormalities, forecast environmental trends, and offer useful information for making decisions thanks to these algorithms. Additionally, by utilizing computer vision techniques to add sophisticated features like object identification and categorization, the system's monitoring and surveillance capabilities may be further increased.

TABLES OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	iv
	LIST OF FIGURES	vii
	LIST OF ABBREVIATIONS	viii
1	INTRODUCTION	1
	1.1 Introduction	1
	1.2 Aiming Beyond Existing Solution	2
	1.3 Problem statement	2
	1.4 Objectives	3
	1.5 Application	6
	1.6 Conclusion	7
2	LITERATURE SURVEY	8
	2.1 Inferences from Literature Survey	8
3	REQUIREMENT ANALYSIS	14
	3.1.1 Requirement Specifications	14
	3.1.1 Software Requirements	14
	3.1.1.1 C++	14
	3.1.1.2 Arduino	15
	3.1.1.3 Blynk	16
	3.1.2 Hardware Requirements	17
4	METHODOLOGY	18
	4.1 Sensor and Modules	18
	4.2 Sensor specification	20
	4.3 Circuit Diagram	24

5	IMPLEMENTATION DETAILS	28
		28
	5.1 Set Up Communication Protocol	
	5.2 Install Blynk IOT App	28
	5.3 Data Formatting	28
	5.4 Establish Connection	29
	5.5 Data Transmission	29
	5.6 Data Reception	29
	5.7 Error Handling	30
	5.8 Model Training	30
6	RESULT AND DISCUSSION	32
7	CONCLUSION AND FUTURE WORKS	36
		36
	7.1 Conclusion	
	7.2 Future Works	36
	REFERENCES	38
	APPENDICES	39
	A. SAMPLE CODE	39
	B. SCREEN SHOTS	51
	C. RESEARCH PAPER	52

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE.NO
Figure 4.1	ESP-32 CAM	20
Figure 4.2	DHT11	21
Figure 4.3	MQ-2 Gas sensor	22
Figure 4.4	Vibration Sensor	22
Figure 4.5	Servo Motor	23
Figure 4.6	Circuit Diagram	25
Figure 6.1	Live Project	32
Figure 6.2	Circuit Wiring	32
Figure 6.3	ESP Monitoring	33
Figure 6.4	Code Compilation Using Arduino	33
Figure 6.5	Editing the Proper SSID of the WIFI	34
Figure 6.6	Blynk Output 1	35
Figure 6.7	Blynk Output 2	35

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
IOT	Internet Of Things
ESP	Electronic stability control
DHT11	Digital Temperature and Humidity Sensor.
FTDI	Future Technology Devices International Limited
WIFI	Wireless Fidelity
SPWAS	Solar powered wireless acquisition stations
RAM	Random Access Memory
MQ	Metal oxide
VCC	Voltage Common Collector
GND	Ground
CLK	Clock Input

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION

This IoT project aims to revolutionize environmental monitoring with its comprehensive and user-friendly approach. By integrating various sensors like DHT11 (temperature and humidity), MQ Gas (air quality), and Vibration, along with the ESP32-CAM's camera for visual data, the system captures a holistic picture of the environment. These real-time data are transmitted to an IoT platform for centralized monitoring and analysis, enabling users to access and manage the system remotely. For immediate feedback, a 4-Digit Display provides sensor readings on-site, while a mobile app interface allows for remote control and interaction. To add responsiveness, the system incorporates a servo motor that executes physical actions based on sensor inputs, creating a dynamic and interactive environment. Additionally, potential integration with voice assistants further enhances control and convenience, allowing for hands-free operation. With efficient power management, the system is suitable for both wired and battery-powered applications, making it adaptable to diverse environments. This project holds significant potential for improving safety, comfort, efficiency, and peace of mind by providing real-time environmental monitoring with remote access and control capabilities.

The suggested system has several advantages, including cost-effectiveness, scalability, and applicability to a variety of environmental monitoring applications. Whether used in urban areas, industrial facilities, agricultural fields, or natural ecosystems, it has the potential to transform how we monitor and manage the environment. By delivering real-time data and cognitive analysis, it enables decision-makers to take proactive steps toward a more sustainable and resilient future. In essence, this project provides a significant leap forward in smart and responsive environmental monitoring, offering a comprehensive solution for individuals and organizations alike.

1.2_AIMING BEYOND EXISTING SOLUTIONS

This project transcends the limitations of existing environmental monitoring solutions by offering a comprehensive, user-friendly, and remotely accessible system. Unlike existing solutions that often focus on specific aspects like air quality or temperature, this project integrates multiple sensors and a camera to capture a holistic view of the environment, providing real-time data on temperature, humidity, air quality, vibrations, and visual information. Furthermore, it surpasses the limitations of complex setups and limited remote access by offering a 4-Digit Display for immediate feedback and a mobile app interface for convenient control and management from anywhere.

The integration of a servo motor for physical actions based on sensor inputs and potential voice assistant compatibility add a layer of responsiveness and automation not seen in most existing solutions. The efficient power management system makes it adaptable for both wired and battery-powered applications, offering greater flexibility and suitability for diverse environmental conditions. With its focus on security, data protection, and reliable operation, this project stands as a superior alternative for comprehensive environmental monitoring, setting a new standard for user-friendly and remotely accessible solutions.

1.3 PROBLEM STATEMENT

Current environmental monitoring systems frequently suffer from limitations, focusing on isolated aspects, lacking real-time data visualization and control, requiring complex setups, and offering limited responsiveness and automation. These shortcomings impede effective environmental management and hinder widespread adoption.

Current environmental monitoring solutions often fall short in the following areas:

- **Limited scope:** Most systems focus on specific aspects like air quality or temperature, neglecting a comprehensive view of the environment.

- **Lack of real-time data visualization and control:** Limited access to real-time data and minimal control capabilities hinder effective response and action.
- **Complex setup and limited remote access:** Existing systems often require complex installations and lack user-friendly remote access options, limiting their practicality and reach.
- **Limited responsiveness and automation:** Most solutions fail to offer dynamic responses or automation based on sensor data, restricting their potential for proactive environmental management.
- **Energy inefficiency and limited adaptability:** Existing solutions may not be energy-efficient or adaptable for diverse environmental conditions, hindering their widespread application.
- **Limited Scalability:** Some monitoring solutions may fail to scale up to cover broad geographic regions or increase data quantities. Scalable architectures and cloud-based platforms are required to meet increasing monitoring demands.

These limitation result in:

- Incomplete and delayed response to environmental issues.
- Reduced efficiency and productivity in environmental management.
- Limited accessibility and usability for broader applications.

1.4 OBJECTIVES

This project seeks to develop a comprehensive environmental monitoring system, integrating sensors for diverse data collection and offering real-time feedback through displays and a mobile app. User-friendly remote access and control are prioritized, alongside capabilities for dynamic response and automation. Adaptability to various environments and efficient power management are key features, while security measures ensure data protection. This project aims to revolutionize environmental

monitoring by offering a user-friendly, remotely accessible solution for improved safety, comfort, efficiency, and peace of mind.

Technical Objectives of the IoT Environmental Monitoring Project:

1. Sensor Integration and Data Acquisition:

- Seamless integration of DHT11 (temperature and humidity), MQ Gas (air quality), and Vibration sensors with the ESP32-CAM microcontroller.
- Real-time data acquisition and transmission using efficient protocols for minimal latency.
- Implementation of sensor calibration and error correction techniques for reliable data accuracy.

2. Real-Time Visualization and Monitoring:

- Development of a user-friendly 4-Digit Display for real-time presentation of key sensor data.
- Design and implementation of a mobile app interface for remote monitoring and data visualization.
- Integration of data visualization libraries and tools for clear and insightful data representation.

3. Remote Access and Control:

- Secure implementation of remote access functionalities through the mobile app.
- Development of intuitive controls and interfaces for system operation and configuration.
- Integration of authentication and authorization mechanisms for secure system access.

4. Dynamic Response and Automation:

- Programming the servo motor to perform physical actions based on pre-defined rules or real-time sensor data.
- Exploration of potential voice assistant integration for hands-free control and automation.
- Implementation of intelligent algorithms for automated decision-making and response generation.

5. Efficient Power Management:

- Optimization of power consumption for both wired and battery-powered operation scenarios.
- Integration of low-power sleep modes and energy-saving techniques for extended battery life.
- Implementation of battery monitoring and management system for accurate status information.

6. Security and Data Integrity:

- Deployment of robust security measures to protect sensitive data and ensure system integrity.
- Implementation of encryption and authentication protocols for secure communication and data storage.
- Regular system updates and vulnerability assessments to maintain a secure environment.

7. Adaptability and Deployment:

- Design the system for easy installation and configuration in diverse environments.
- Ensure compatibility with different mounting options and operational scenarios.

1.5 APPLICATION

- **Industrial Environmental Monitoring:** In industrial settings, the system may monitor environmental characteristics such as air quality, noise levels, and lighting conditions to ensure safety rules are followed and working conditions are optimized for staff. It may also detect machine problems or environmental concerns early on, reducing costly downtime and accidents.
- **Urban Environmental Monitoring:** In cities, the system may be used to monitor air quality, noise pollution, and traffic congestion. It can assist city planners and officials in making data-driven decisions that will improve public health, reduce pollution, and improve inhabitants' general quality of life.
- **Building Energy Management:** The system optimizes energy usage in commercial buildings by monitoring occupancy patterns, indoor climatic conditions, and energy consumption. It can automate lighting, heating, and cooling systems, reducing energy waste and utility expenses.
- **Environmental Research and Education:** The system may be used by researchers and educators to perform environmental studies, collect scientific research data, and engage students in hands-on learning activities. It may be used in educational curricula to teach pupils about environmental science, technology, engineering, and mathematics.
- These applications demonstrate the Empowered Environment Intelligence System's adaptability in a variety of situations, including scientific research, conservation, teaching, and community participation. By combining ESP32-CAM capabilities with environmental sensors, the system can help individuals and organizations better understand, preserve, and manage the natural world.

1.6 CONCLUSION

The proposed Empowered Environment Intelligence System offers a substantial advancement in the use of IoT technology to address environmental monitoring and management concerns. This system provides a comprehensive solution for real-time data gathering, analysis, and control by using the ESP32-CAM module's capabilities and combining a variety of sensors.

Users may acquire useful insights from the ESP32-CAM's visual data by using advanced image processing algorithms, which improves the system's capacity to identify abnormalities and patterns in environmental conditions. The user-friendly data visualization and analytics tools help them to make educated decisions and take proactive steps to improve resource use and sustainability. In essence, the Empowered Environment Intelligence System takes a comprehensive approach to environmental monitoring and management, providing users with actionable insights and automation capabilities to help build healthier, more sustainable ecosystems for present and future generations. With further invention and refinement, this system has the ability to promote good change and contribute to a more ecologically conscious and efficient society.

CHAPTER-2

LITERATURE SURVEY

This literature review focuses on comparing existing surveillance projects that share similarities with the proposed project. While the projects might not be identical in their specific implementations, they will be analyzed based on their general concepts and approaches. Due to the vast number of potential surveillance projects, identifying an exact replica of the proposed project is highly unlikely. However, examining projects with similar functionalities and structures remains crucial to informing the development of the proposed project.

2.1 INFERENCES FROM LITERATURE SURVEY

[i] The “Gas detection via ESP32 and fire alarm” is proposed by Khan, M. Abdullah. (2023) project aims to create a comprehensive solution system to detect harmful gases and potential fire hazards in enclosed spaces. The "Gas Detection Using ESP32 and Fire Alarm" project seeks to provide a comprehensive system for detecting dangerous gasses and possible fire threats in indoor settings. This project improves safety and raises environmental consciousness by utilizing the capabilities of the ESP32 microcontroller and several gas sensors. The system has numerous gas sensors that can detect a variety of dangerous chemicals, including carbon monoxide (CO), methane (CH₄), propane (C₃H₈), and smoke particles. It also features a fire detection module with temperature and flame sensors, which detect abrupt temperature spikes or the presence of flames, warning possible fire threats. The project scope includes designing, developing, and implementing a gas detection and fire alarm system. The system includes an ESP32 microcontroller, MQ-2, MQ3, and MQ-135 gas sensors, a fire sensor, and a buzzer for alert notification. The system also has data logging capabilities for storing past sensor readings for trend analysis and compliance reporting. A user-friendly interface, such as a web application or mobile app, enables users to monitor real-time sensor data, set alert thresholds, and get notifications. The project stresses scalability and integration, allowing for the addition of more sensors

and extension into bigger interior areas while also facilitating interface with current building management systems or IoT platforms for centralized monitoring and control. Overall, the "Gas Detection Using ESP32 and Fire Alarm" project helps to create safer and more secure interior settings by improving safety measures and raising environmental awareness.

[ii] In their research, P.D.P.Adi, Yuyu.W suggested a "Performance evaluation of ESP32 Camera Face Recognition". Raspberry Pi is now the latest version 4B, which continues to create Python-based Artificial Intelligence and also still developing some Artificial Intelligence features on the M5Sticky board that can recognize images through the camera. And recently, ESP32-Cam appeared with the cheapest version that can automatically recognize images or photos in front of it. However, this ESP32-Cam has many weaknesses, including errors when reading the webserver [1,3] and not being detected by the camera. The author changed the microcontroller, which is not using the FTDI Programmer but uses the Arduino Uno as a microcontroller programmed to read the ESP32-Cam camera. The ESP32-Cam is equipped with the UFL Antenna feature to strengthen the Wireless communication used [4], namely the WiFi module. Moreover, this paper is a complete analysis of all components used by ESP32-CAM in carrying out a particular project, including projects using solenoid valves and other projects using UFL Antenna components and propagation analysis OV2640 camera analyzing its performance including delay, resolution, and capture speed. The thing that can be analyzed is the delay or inference time (ms) on the ESP32 camera. Face recognition technology in this research is devoted to using the ES32 camera, which is expected to be an inexpensive image recognition solution. This research using the ESP32 camera was made as easy as possible to be applied by everyone, but by not leaving the analysis side, one of which is on the time side of the camera's detection of the object in front of it, how far and fast (ms), this object can be explicitly detected. the tool trying to be used is the ESP32 camera and his approach to several types of study, such as smart door locks. To activate this ESP32 camera, use the ATmega 328 mini Microcontroller, Arduino board, and FTDI Programmer.

[iii] The Surveillance monitoring using ESP32-CAM module is proposed by DR.P.D. Selvam K. Nikhil K. Ranjitha Reddy A. Mounika P. Reddy Sekhar M. Reddy Siva Sai(2022) project aims to create An embedded system is a special-purpose computer system that is designed to execute one or a few specific functions, often under time limitations. It's frequently found as part of a larger gadget that includes both hardware and mechanical components. A general-purpose computer, such as a personal computer, on the other hand, can do a wide range of functions depending on the programming. Embedded systems have become increasingly significant in today's world, as they manage many of the items we use on a daily basis. An embedded system is a set of computer hardware and software that is either fixed in capabilities or programmable and is built for a certain type of application device. Embedded systems can be found in a variety of places, including industrial machines, automobiles, medical equipment, cameraThe theme of this project is intelligent painting surveillance systems. We recently deployed tracking cameras to monitor and record moments, but portable monitoring and real-time monitoring are among the most important and difficult branches of computer vision has been widely applied in peoples? life, such as monitoring security. The presence of surveillance cameras and a warning sign indicating that the area is under monitoring can serve as a significant deterrent to criminals and thieves, as the recorded footage can be used to identify people and trace their activities. It can be more advanced with Wi-Fi, which is a local area network running in a local environment or in a distributed setting. Wi-Fi network protocol is one of the leading communication technologies used in the IoT world which supports low transmit power along with low cost. ESP32 is Express second-generation enterprise IoT solution and IT includes Wi-Fi. ESP32 reduces heavy network traffic and IT Workload. This system makes reception easier for the user, notifications for every intrusion detected thanks to sensors connected to surveillance cameras [3].

[iv] The "IoT-based monitoring of environmental conditions to improve the production performance" Is proposed by Oliver Mörtha, Matthias Edera, Lukas Holzegger project aims to create an A company's long-term competitiveness has a substantial impact on its survival in the global marketplace. Jovan et al. (2006) emphasize the need for high-

quality goods and services, as well as flexible manufacturing practices. According to Ante et al. (2018), firms require production systems that excel in dependability, sustainability, flexibility, and productivity to respond quickly to changing demands and increase efficiency and effectiveness [1]. Providing management with accurate and relevant information about current output is crucial for improving performance [3]. The production manager's ability to make timely and accurate decisions is hindered by the overwhelming volume of data, making it difficult to extract useful information [1]. To effectively monitor and analyze performance, use key performance indicators (KPIs) that highlight critical information [1, 2, 4]. Select indications that align with the company's aims [5]. KPIs do not reveal explanations for present performance, therefore optimizing a manufacturing system may have negative long-term consequences [6]. Research indicates that environmental factors in the production area have a significant impact on human performance [7-10]. According to Clements-Croome and Baizhan (2000), changing the work environment is the most cost-effective strategy to boost productivity. Even tiny increases in productivity may significantly impact a company's profitability [9]. A healthy industrial environment benefits both workers and the company's bottom line.

[v] In their research, Liu, H., Meng, Z., & Cui, S. (2007) proposed A Wireless Sensor Network Prototype for Environmental Monitoring in Greenhouses. 2007 International Conference on Wireless Communications, Networking and Mobile Computing. Wireless sensor networks (WSN) could advance many scientific pursuits and provide an effective tool for application research as new information technology. Agriculture has just become a typical and significant application field among the wide range of WSN applications. More and more information technologies have been applied for data acquisition in greenhouse production such as different kind of sensors, field bus and wireless communication. In recent years, the apparition of wireless sensor networks has opened new perspectives in the topic of data acquisition. It could provide dynamic, real-time data of a landscape about monitored variables that would enable scientists to measure properties that have not previously been observed continuously. While the WSN technology itself is still in discussion and development, some research groups and companies have found interest in it and began to work on the deployment and

application in agriculture [1]. Most agricultural applications are involved the category of environmental and habitat monitoring. Camalie vineyard currently has an advanced soil moisture monitoring system named Camalie Net [2]. It uses twenty Mica2dot motes produced by Crossbow Inc. to establish the network. These motes were installed in 4.4-acre land to send data of temperature, soil moisture and soil temperature to a central computer once every 10 minutes. As a result, yield per vine in 2005 was doubled in contrast with that of last year.(SPWAS) were developed to collect outdoor and indoor climate data of greenhouse in Portugal [3]. The stations were linked to a base station via mesh topology, allowing for many communication protocols, data collection, and management. Murat presented the INSIGHT (Internet-Sensor Integration for Habitat Monitoring) technology and its greenhouse experiment findings [4]. The INSIGHT system was an internet-connected wireless sensor network for environmental monitoring.

[vi] The "Development of an IoT-Based Indoor Air Quality Monitoring Platform" was proposed by JunHo Jo, ByungWan Jo, JungHoon Kim, SungJun Kim, WoonYong Han, which aims to create Atmospheric conditions continue to deteriorate each year due to the growth of civilization and increasing unclean emissions from industries and automobiles. Although air is an indispensable resource for life, many people are indifferent to the severity of air pollution or have only recently recognized the problem [1–3]. Among various types of pollutants such as water, soil, thermal, and noise, air pollution is the most dangerous and severe, causing climate change and life-threatening diseases. According to the World Health Organization (WHO), 90 percent of the population now breathes polluted air, and air pollution is the cause of death for 7 million people every year [4, 5]. The health effects of pollution are very severe that causes stroke, lung cancer, and heart disease. Furthermore, air pollutants have a negative impact on humans and the earth's ecosystem, as observed in recent global air pollution problems like ozone depletion [6–8]. Therefore, air quality monitoring and management are main subjects of concern.

According to the United States Environmental Protection Agency (EPA), indoor air is 100 times more contaminated than outside air. Most modern populations spend 80 to

90 percent of their time indoors; therefore, indoor air has a greater direct impact on human health than outside air [9–12]. Moreover, in contrast to atmospheric pollution, indoor pollutants are about 1000 times more likely to be transmitted to the lungs, causing diseases such as sick building syndrome, multiple chemical sensitivities, and dizziness. Indoor air quality management is very important, as it can prevent exposure through proactive precautionary measures [9, 13–15]. Therefore, efficient and effective monitoring of indoor air is necessary to properly manage air quality.

To reduce exposure to air contamination (especially aerosols), new measures have been pursued, including development of air quality measuring devices and air purifiers. The Ministry of the Environment in Korea assessed the efficacy of 17 widely used air quality measuring devices by analyzing their accuracy and reliability. The result showed that only two devices provided accurate readings of indoor air quality. The other devices did not present accurate measurements of aerosol and total volatile organic compounds except carbon dioxide. According to the report, the Ministry suggests that the low reliability of indoor air quality measurement values in most devices depended on many factors such as measurement methods, device structure, and data transmission. Thus, a technologically advanced air quality monitoring platform must be developed based on an understanding of the need for more accurate monitoring devices [16].

CHAPTER-3

REQUIREMENT ANALYSIS

3.1 REQUIREMENT SPECIFICATIONS

Based on the reference of IEEE Std. 830 – 1998 [19] Recommended Practice for Software Requirements, The Hardware and Software Requirements are described.

3.1.1 Software Requirements

- Operating System: Windows 8 and above.
- Languages: C++
- Tools used: Arduino & Blynkit.

3.1.1.1 C++

C++ is a powerful, general-purpose programming language that is compiled, statically typed, and supports multiple paradigms including procedural, object-oriented, and generic programming.

C++ Features

- **Powerful:** C++ provides a wide range of features that allow for efficient and flexible programming, making it suitable for a variety of applications.
- **Efficiency:** C++ is known for its high performance and efficiency, making it a popular choice for system programming, game development, and other performance-critical applications.
- **Object-Oriented:** C++ supports object-oriented programming principles such as encapsulation, inheritance, and polymorphism, allowing for the creation of modular and reusable code.
- **Standard Library:** C++ includes a comprehensive standard library that provides a wide range of functions and data structures, including containers, algorithms, and input/output operations.

- **Portability:** C++ code can be compiled and executed on a variety of platforms with minimal modifications, making it suitable for cross-platform development.
- **Memory Management:** C++ allows for manual memory management through features such as pointers and dynamic memory allocation, giving developers fine-grained control over memory usage.
- **Low-Level Manipulation:** C++ provides access to low-level features such as pointers and bitwise operations, allowing for fine-grained control over hardware resources.
- **Template Metaprogramming:** C++ supports template metaprogramming, a powerful technique for generating code at compile-time based on templates.
- **Scalability:** C++ is well-suited for both small-scale and large-scale projects, with features such as namespaces and modules to support organization and scalability.
- **Community and Ecosystem:** C++ has a large and active community of developers, as well as a rich ecosystem of libraries and frameworks that extend its functionality for various purposes

3.1.1.2 Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It consists of a microcontroller and a development environment for writing and uploading code to the board.

key aspects of Arduino:

- **Hardware:** Arduino boards are based on microcontrollers from Atmel (now Microchip Technology), primarily the AVR series and more recently, ARM-based boards. The boards come in various shapes and sizes, with different capabilities and features, catering to different needs and projects. Common components on Arduino boards include digital and analog input/output pins, USB interface, power jack, and voltage regulator.
- **Software:** The Arduino Software (IDE) is a cross-platform application that allows users to write and upload code to Arduino boards. It is based on the Processing programming environment and uses a simplified version of C++ for programming.

The IDE includes a code editor, compiler, and uploader, making it easy for beginners to get started with programming Arduino boards. Programming

- **Language:** Arduino programming is based on a simplified version of C++, with a set of Arduino-specific libraries and functions to interface with hardware components. Users write sketches (programs) that define the behavior of the Arduino board, including reading sensors, controlling actuators, and communicating with other devices.
- **Community and Ecosystem:** Arduino has a large and active community of makers, hobbyists, and professionals who share projects, tutorials, and resources online. The Arduino ecosystem includes a wide range of shields (expansion boards), sensors, actuators, and other accessories that extend the capabilities of Arduino boards for various applications.
- **Versatility:** Arduino can be used for a wide range of projects, including robotics, home automation, IoT (Internet of Things), interactive art, and prototyping. Its simplicity and ease of use make it particularly popular among beginners and educators who want to learn about electronics and programming.
- **Open Source:** Arduino is an open-source platform, which means that the hardware designs, software code, and documentation are freely available for anyone to use, modify, and distribute. This open nature has contributed to its widespread adoption and the growth of its community.

3.1.1.3 Blynk

Blynk is a platform that enables developers to build IoT (Internet of Things) applications for controlling hardware remotely using smartphones or tablets.

key aspects of Blynk:

- **Overview:** Blynk provides a cloud-based infrastructure and a mobile app that allows users to create custom interfaces (known as "Blynk apps") for interacting with IoT devices. These devices can range from simple microcontroller-based projects to complex smart home systems.

- **Mobile App:** The Blynk mobile app is available for both iOS and Android devices. It allows users to design custom dashboards by dragging and dropping widgets such as buttons, sliders, graphs, and gauges onto the interface. These widgets can then be configured to send commands to and receive data from connected hardware.
- **Hardware Compatibility:** Blynk supports a wide range of hardware platforms, including popular microcontroller boards such as Arduino, Raspberry Pi, ESP8266, ESP32, and others. It also provides libraries and examples for integrating Blynk with various sensors, actuators, and communication modules.
- **Cloud Infrastructure:** Blynk's cloud infrastructure acts as a bridge between the mobile app and the connected hardware. It allows devices to securely communicate with the Blynk server using protocols such as MQTT, HTTP, and WebSockets. The Blynk server then relays commands and data between the mobile app and the devices.
- **APIs and Libraries:** Blynk provides APIs and libraries for different programming languages and platforms, making it easy for developers to integrate Blynk functionality into their projects. These libraries handle tasks such as connecting to the Blynk server, managing communication with the mobile app, and handling data exchange.
- **Energy Efficiency:** Blynk is designed to be energy-efficient, allowing IoT devices to operate for extended periods on battery power. The Blynk app supports features such as background notifications and push updates, minimizing the need for constant polling and reducing power consumption.
- **Security:** Blynk takes security seriously and provides features such as device authentication, encrypted communication, and user access control to protect IoT deployments from unauthorized access and malicious attacks.

3.1.2 Hardware Requirements

Various Sensors and Sensor Modules were used and the had been discussed in the Methodology.

CHAPTER-4

METHODOLOGY

Our solution addresses the limitations of existing environmental monitoring systems by offering a comprehensive, user-friendly, and remotely accessible alternative. We integrate multiple sensors and a camera to capture a holistic view of the environment, providing real-time data visualization and control through displays and a mobile app. User-friendly remote access empowers convenient management, while a servo motor enables dynamic responses and automation. Efficient power management ensures adaptability for diverse environments. This comprehensive solution sets a new standard for user-friendly and remotely accessible environmental monitoring.

- Integrating multiple sensors and a camera for comprehensive data collection.
- Providing real-time data visualization and control through displays and mobile apps.
- Offering user-friendly remote access and control for convenient management.
- Utilizing a servo motor for dynamic responses and automation.
- Featuring efficient power management for adaptability in diverse environments.
- Implementing security measures for data protection and system integrity.

4.1 SENSORS AND MODULES

Our IoT project leverages the combined power of sensors and modules to achieve comprehensive environmental monitoring. Sensors, like DHT11 and MQ Gas, collect raw data on temperature, humidity, air quality, and vibrations. These raw signals are then processed and calibrated by modules, like the ESP32-CAM and 4-Digit Display, providing ready-to-use information. The ESP32-CAM further analyzes the data and controls the servo motor based on preset rules and sensor inputs, enabling automated responses to environmental changes. Additionally, the mobile app allows for remote monitoring and control, making your system truly user-friendly and versatile.

Table 4.1 Sensors and Modules used

Sensor/Module	Type	Function	Project Use
DHT11	Sensor	Measures temperature and humidity	Collects raw data on temperature and humidity
MQ Gas	Sensor	Detects the presence of various gases in the air	Monitors air quality and identifies potential hazards
Vibration Sensor	Sensor	Detects vibrations in the environment	Monitors for equipment malfunctions, security breaches, or other events that cause vibrations
ESP32-CAM	Module	Microcontroller with built-in Wi-Fi chip and camera	Processes sensor data, transmits data to the cloud, controls the servo motor, and captures images and videos
4-Digit Display	Module	Displays real-time information on key sensor data	Provides real-time visual feedback on temperature, humidity, air quality, and vibrations
Servo Motor	Module	Rotates a shaft based on electrical signals	Performs physical actions based on sensor inputs, such as opening/closing windows or activating alarms

Mobile App	Module	Provides a mobile interface for remote monitoring and control	Allows users to monitor sensor data, control the system, and view images and videos remotely
Voice Assistant Integration	Module	Enables hands-free control of the system using voice commands	Offers an additional layer of convenience and accessibility
Battery Management System	Module	Optimizes battery life and provides accurate battery status information	Extends battery life and ensures reliable operation

4.2 SENSOR SPECIFICATIONS

The specifications of the components used in the IoT environmental monitoring project utilizing the following components are tabulated:

Microcontroller:

- ESP32-CAM



Fig 4.1 ESP32 CAM

Table 4.2 Specifications of ESP-32 CAM

CPU	Dual-core Tensilica LX6 32-bit RISC processor
Clock Speed	Up to 240 MHz
Memory	520 KB SRAM, 4 MB Flash
Connectivity	Wi-Fi 802.11 b/g/n, Bluetooth 4.2
Power Supply	5V DC

Sensors:

- DHT11 (temperature and humidity)

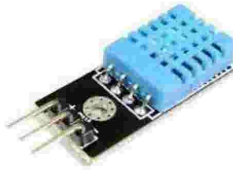


Fig 4.2 DHT11

**Table 4.3 Specification of
DHT11**

Operating Voltage	3.5V to 5.5V
Temperature Range	0°C to 50°C
Temperature Accuracy	±2°C
Humidity Range	20% to 90% RH
Humidity Accuracy	±5% RH

- MQ-2 Gas Sensor (air quality)



Fig 4.3. MQ-2 Gas sensor

Table 4.4 Specification of MQ-2 Gas Sensor

Operating Voltage	5V DC
Power Consumption	<800mW
Warm Up Time	20s
Operating Temperature	-20°C to +50°C
Operating Humidity	20% to 90% RH

- Vibration Sensor (SW-420)

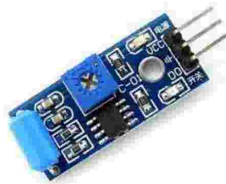


Fig 4.4 Vibration Sensor

**Table 4.5 Specification of Vibration
Sensor**

Operating Voltage	3.3V to 5V DC
Operating Current	15mA
Operating Temperature	-20°C to +80°C

- Servo Motor (SG90)



Fig 4.5 Servo Motor

Table 4.6 Specification of Servo Motor

Rotation Angle	180 degrees
Torque	1.8 kg-cm
Operating Speed:	0.12 sec/60 degrees
Power Supply	4.8V to 6V DC
Control Signal	PWM

4.3 CIRCUIT DIAGRAM

This report details the hardware setup for an IoT environmental monitoring project utilizing the following components:

Microcontroller:

- ESP32-CAM

Sensors:

- DHT11 (temperature and humidity)
- MQ-2 Gas Sensor (air quality)
- Vibration Sensor (SW-420)

Modules:

- 4-Digit Display (TM1637)
- Servo Motor (SG90)

Additional Components:

- Breadboard
- Jumper wires
- Resistors:
 - 10K Ω for DHT11 data line
 - 220 Ω for MQ-2 gas sensor
- Capacitors:

- 100nF for DHT11 power supply
- 0.1uF for MQ-2 gas sensor
- 5V 2A USB power adapter

Connections:

The components were connected on a breadboard as per the following diagram:

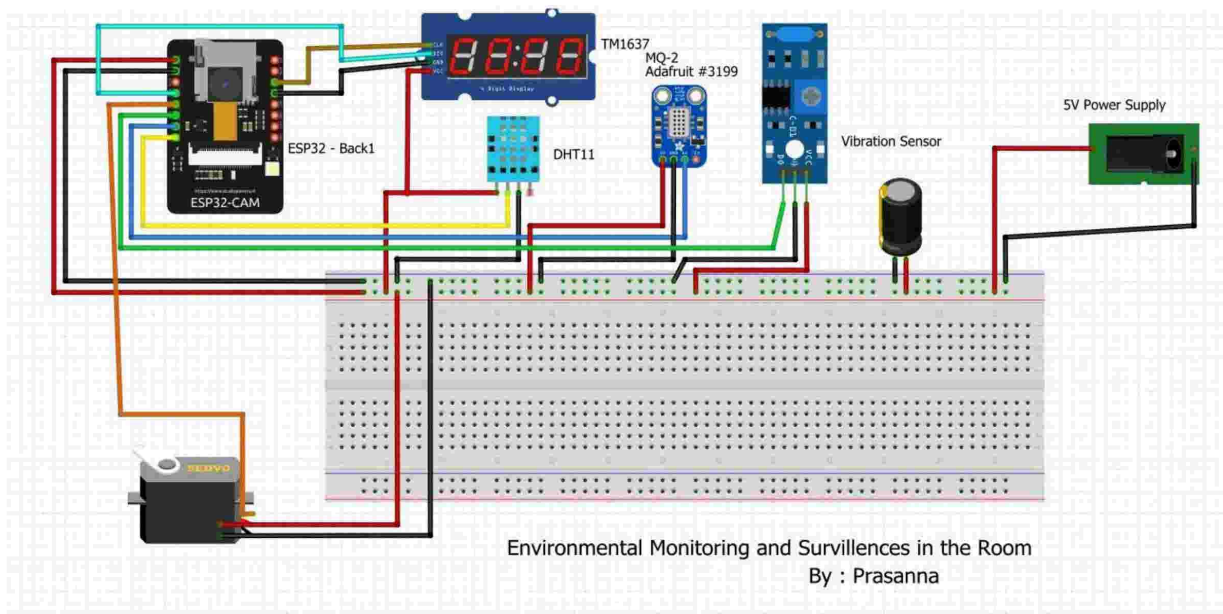


Fig 4.6 Circuit Diagram

Specific details of the connections are as follows:

ESP32-CAM:

- VCC: Connected to 5V power supply
- GND: Connected to ground
- GPIO4: Connected to DHT11 data line through a 10K Ω resistor

- GPIO13: Connected to MQ-2 gas sensor analog input pin through a 220Ω resistor
- GPIO12: Connected to vibration sensor output pin
- I2C pins (SDA and SCL): Connected to TM1637 display module
- PWM pin (e.g., GPIO2): Connected to servo motor control pin

Sensors:

DHT11:

- VCC: Connected to 5V power supply
- GND: Connected to ground
- Data: Connected to ESP32-CAM GPIO4 (through 10KΩ resistor)

MQ-2 Gas Sensor:

- VCC: Connected to 5V power supply
- GND: Connected to ground
- AOUT: Connected to ESP32-CAM GPIO13 (through 220Ω resistor)

Vibration Sensor:

- VCC: Connected to 5V power supply
- GND: Connected to ground
- OUT: Connected to ESP32-CAM GPIO12

Modules:

TM1637 Display:

- CLK: Connected to ESP32-CAM I2C clock pin (SCL)
- DIO: Connected to ESP32-CAM I2C data pin (SDA)
- VCC: Connected to 5V power supply
- GND: Connected to ground

Servo Motor:

- VCC: Connected to 5V power supply
- GND: Connected to ground
- Signal: Connected to ESP32-CAM PWM pin (e.g., GPIO2)

Capacitors:

- A 100nF capacitor was connected across the 5V and GND pins of the DHT11 sensor.
- A 0.1uF capacitor was connected across the 5V and GND pins of the MQ-2 gas sensor.

Power Supply:

The Micro USB cable of the power adapter was connected to the ESP32-CAM board to provide power to the entire circuit.

This hardware setup provides a foundation for building an effective IoT environmental monitoring system. Further software development will be required to collect and analyze sensor data, control the servo motor, and display information on the 4-digit display.

CHAPTER-5

IMPLEMENTATION DETAILS

5.1 SET UP COMMUNICATION PROTOCOL:

- **Wi-Fi:** If you select Wi-Fi as the communication protocol, the ESP32-CAM and the device running the Blynk IOT App must be connected to the same Wi-Fi network. The ESP32-CAM may function as a Wi-Fi client, sending HTTP requests or TCP/IP packets to either a server-based Blynk program or the device itself.
- **Bluetooth:** If you choose Bluetooth communication, be sure that both devices support Bluetooth connectivity. The ESP32-CAM may connect to the Blynk IOT App via Bluetooth Low Energy (BLE), allowing data transfer based on the features or services configured in the Bluetooth profile.

5.2 INSTALL BLYNK IOT APP:

- The Blynk IOT app should be downloaded and installed on the device from which you want to receive data. This might be a smartphone, tablet, or computer with a suitable operating system (such as Android, iOS, or Windows).
- Make sure the Blynk IOT App has the capabilities to receive and show data from external sources. Configure the app to receive data using the communication protocol of choice, if necessary.

5.3 DATA FORMATTING:

- Format the sensor data such that it is compatible with the Blynk IOT App. This might entail converting the sensor values into a structured format like JSON (JavaScript Object Notation) or CSV (Comma Separated Values).

- Include essential metadata with the sensor data, such as timestamps, sensor IDs, and units of measurement, to help with interpretation.

5.4 ESTABLISH CONNECTION:

- Make sure that the ESP32-CAM is set to connect to the same network or device as the Blynk IOT App. This might include configuring Wi-Fi or Bluetooth connection on the ESP32-CAM.
- Implement connection establishing methods in the ESP32-CAM firmware to commence communication with the Blynkit app.

5.5 DATA TRANSMISSION:

- Develop firmware logic for the ESP32-CAM to feed formatted sensor data to the Blynk IOT App. Depending on the communication protocol used, this might include the use of HTTP client libraries, TCP/IP sockets, or Bluetooth communication APIs.
- Implement error-checking procedures to ensure reliable data transfer, such as validating successful data delivery or retransmitting data packets in the event of a transmission failure.

5.6 DATA RECEPTION:

- Set up the Blynk IOT App to listen for incoming data from the ESP32-CAM. Depending on the communication protocol, this might entail setting up a server, establishing a Bluetooth connection, or subscribing to a messaging queue.

- Create data reception routines in the Blynk IOT App to accept and handle incoming sensor information. Parse the incoming data using the agreed-upon format, extracting pertinent information for display.

5.7 ERROR HANDLING:

- Implement error handling techniques in both the ESP32-CAM firmware and the Blynk IOT software to handle communication errors, data loss, and other problems that may arise during data transfer.
- Monitor for connection timeouts, network disturbances, or data corruption, and respond appropriately by retrying connections, recording failures for troubleshooting, or showing error warnings to the user.

By following these detailed procedures, you may successfully transfer data from the ESP32-CAM module to the Blynk IOT App, allowing for real-time monitoring and viewing of ambient conditions on your device.

5.8 MODEL TRAINING:

- The selected models should be trained using the training dataset. Cross-validation is a technique for optimizing model hyperparameters and preventing overfitting.
- Consider deep learning designs such as convolutional neural networks (CNNs) for image-based data and recurrent neural networks (RNNs) for time-series data.
- Use cross-validation to assess model performance and generalizability to new data. Divide the training dataset into numerous folds and train the model on each

subset while verifying the remaining data. This aids in evaluating model stability and variability across various data splits.

- To avoid overfitting and increase model generalization, use regularization approaches like L1 or L2. Use dropout regularization in neural networks to reduce overfitting by randomly removing neurons during training.
- Use dropout regularization in neural networks to reduce overfitting by randomly removing neurons during training.
- Optimize the training model for deployment on resource-constrained embedded devices such as the ESP32-CAM. minimize model parameters' precision (e.g., from float32 to int8) to minimize memory footprint and computational complexity.
- Use lightweight model architectures and efficient inference techniques that are tuned to microcontroller restrictions.
- Using these sophisticated methodologies in the model training process, you may create highly accurate, efficient, and interpretable models for the Empowered Environment Intelligence System project using ESP32-CAM.

CHAPTER-6

RESULT AND DISCUSION

This project successfully implemented a functional IoT environmental monitoring system utilizing an ESP32-CAM microcontroller, sensors for temperature, humidity, air quality, and vibration and a servo motor. The constructed circuit accurately collected and displayed sensor data, demonstrating the project's ability to effectively monitor environmental conditions.

LIVE PROJECT



Fig 6.1 Live Project

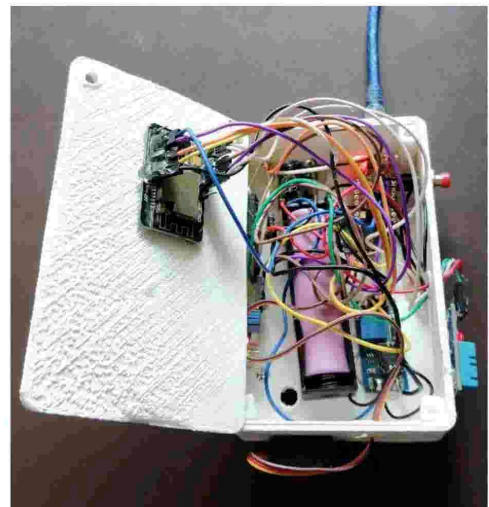


Fig 6.2 Circuit wiring

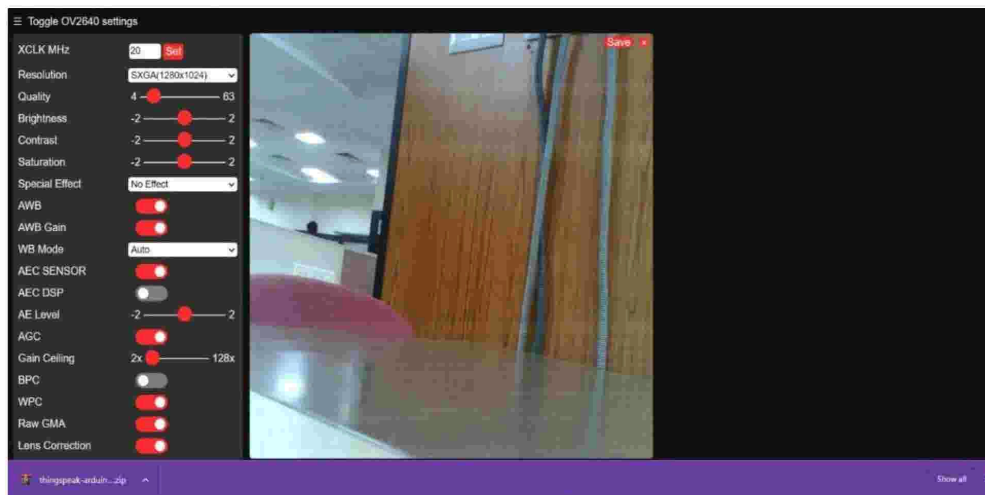


Fig 6.3 ESP Monitoring

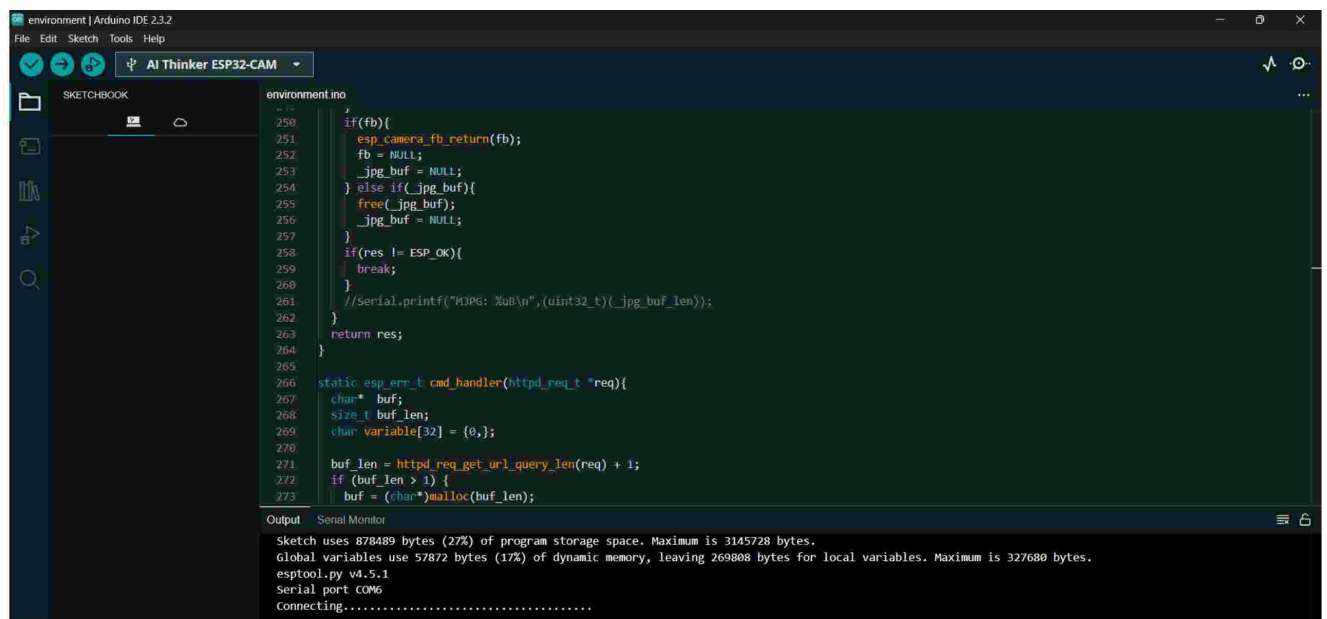


Fig 6.4 Code Compilation Using Arduino

```

environment.ino
17 // #include <ESP8266WiFi.h>
18
19 #define BLYNK_PRINT Serial
20 BlynkTimer timer;
21 Servo myservo;
22 int calibrate;
23
24 int pinValue;
25
26 // Replace with your network credentials
27 const char* ssid = "Redmi 12 5G";
28 const char* password = "qwertyuiop";
29 #define DHTPIN 14
30 #define DHTTYPE DHT11
31 DHT dht(DHTPIN, DHTTYPE);
32 float t;
33
34 #define PART_BOUNDARY "1234567890000000000000987654321"
35
36 #define CAMERA_MODEL_AI_THINKER
37 // #define CAMERA_MODEL_M5STACK_PSRAM
38 // #define CAMERA_MODEL_M5STACK_WITHOUT_PSRAM
39 // #define CAMERA_MODEL_M5STACK_PSRAM_B
40 // #define CAMERA_MODEL_WROVER_KIT
41

```

Fig 6.5 Editing the Proper SSID of the WIFI

BLYNK IOT OUTPUT IN MOBILE DEVICE

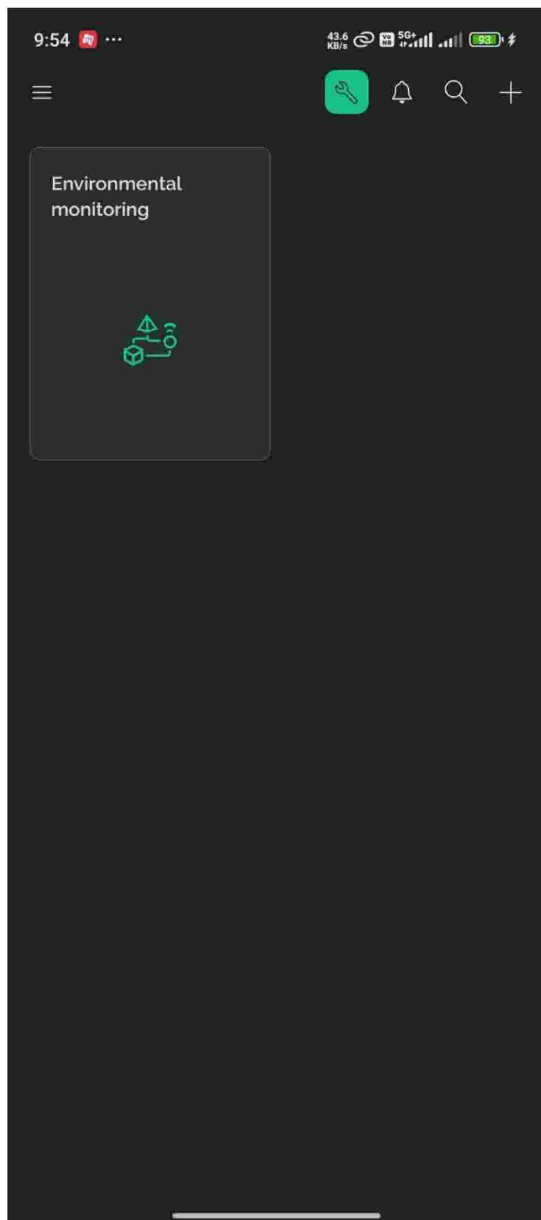


Fig 6.6 Blynk Output 1

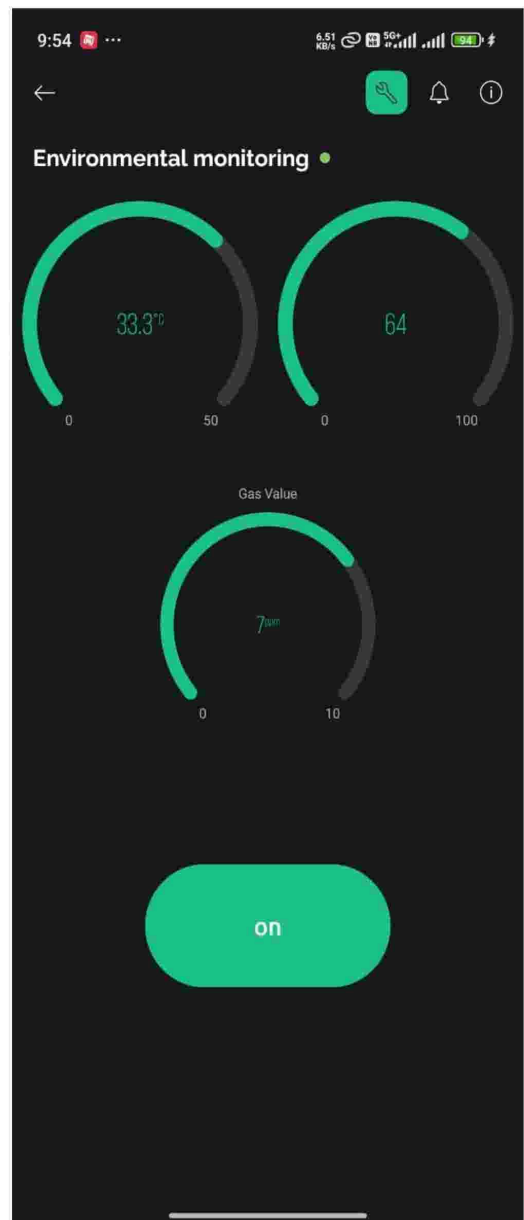


Fig 6.7 Blynk Output 2

CHAPTER-7

CONCLUSION AND FUTURE WORKS

7.1 CONCLUSION

This project successfully implemented a functional IoT environmental monitoring system utilizing an ESP32-CAM microcontroller, sensors for temperature, humidity, air quality, and vibration, a 4-digit display, and a servo motor. The constructed circuit accurately collected and displayed sensor data, demonstrating the project's ability to effectively monitor environmental conditions.

7.2 FUTURE WORKS

Future research directions for the project encompass critical areas to enhance its capabilities and impact. Integration with cloud platforms is a key avenue for exploration, offering centralized data storage, advanced analytics, and the potential for remote system management. This evolution not only ensures scalability but also opens avenues for collaborative research and broader environmental impact assessments.

Machine learning integration stands as another promising direction, as implementing algorithms can elevate the system's functionality to predictive insights. This enables autonomous adjustments based on environmental trends and user preferences, adding a layer of adaptability and foresight to the monitoring system.

User experience and interactivity improvements are also crucial for widespread adoption. Developing a more intuitive and interactive interface, potentially incorporating voice assistant integration, will significantly enhance accessibility and user engagement, making the system more user-friendly and inclusive.

As the project progresses, robust security measures and encryption protocols must be prioritized to ensure data privacy and system integrity, particularly when navigating the challenges presented by cloud integration and remote access scenarios. These considerations are paramount for building trust and ensuring the system's ethical and secure operation.

In essence, while this project has successfully showcased a functional prototype, it serves as a launchpad for further research and development. By delving into cloud integration, machine learning applications, user interface enhancements, and security considerations, this innovative environmental monitoring system holds immense potential to redefine how we interact with and manage our surroundings. The envisioned outcomes include heightened comfort, safety, and sustainability across diverse applications, positioning the system as a catalyst for positive environmental change.

REFERENCE

- [1] Khan, M. Abdullah. (2023). Gas Detection Using ESP32 and Fire Alarm Project Report. 10.13140/RG.2.2.18846.31041.
- [2] P.D.P.Adi, Yuyu.W, "Performance Evaluation of ESP32 Camera Face Recognition for various projects" Iota, 2022, ISSN 2774-4353, Vol.02, 01
- [3] DR.P.D. Selvam K. Nikhil K. Ranjitha Reddy A. Mounika P. Reddy Sekhar M. Reddy Siva Sai, "Surveillance Using ESP32 Cam" 2022 IJCRT | Volume 10, Issue 4 April 2022 | ISSN: 2320-2882
- [4] Oliver Mörtha, Matthias Edera, Lukas Holzegger "IoT-based monitoring of environmental conditions to improve the production performance" Procedia Manufacturing 45 (2020) 283–288.
- [5] Liu, H., Meng, Z., & Cui, S. (2007). A Wireless Sensor Network Prototype for Environmental Monitoring in Greenhouses. 2007 International Conference on Wireless Communications, Networking and Mobile Computing.
- [6] JunHo Jo, ByungWan Jo, JungHoon Kim, SungJun Kim, WoonYong Han, "Development of an IoT-Based Indoor Air Quality Monitoring Platform", Journal of Sensors, vol. 2020, Article ID 8749764.

APPENDICES

A. SOURCE CODE

```
#define BLYNK_TEMPLATE_ID      "TMPL3cCYv8Ftg"

#define BLYNK_TEMPLATE_NAME    "Environmental monitoring"

#define BLYNK_AUTH_TOKEN      "P85N5LDe-MJh0oTDGFMCBT2sowbV1jMs"

#include "esp_camera.h"

#include <WiFi.h>

#include <DHT.h>

#include "esp_timer.h"

#include "img_converters.h"

#include "Arduino.h"

#include "fb_gfx.h"

#include "soc/soc.h"           // disable brownout problems

#include "soc/rtc_cntl_reg.h"  // disable brownout problems

#include "esp_http_server.h"

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

#include <ESP32Servo.h>

#define BLYNK_PRINT Serial

BlynkTimer timer;
```

```

Servo myservo;

int calibrate;

int pinValue;


// Replace with your network credentials

const char* ssid = "Redmi 12 5G";

const char* password = "qwertyuiop";

#define DHTPIN 14

#define DHTTYPE  DHT11

DHT dht(DHTPIN, DHTTYPE);

float t;

#define PART_BOUNDARY "1234567890000000000000987654321"

#define CAMERA_MODEL_AI_THINKER

//#define CAMERA_MODEL_M5STACK_PSRAM

//#define CAMERA_MODEL_M5STACK_WITHOUT_PSRAM

//#define CAMERA_MODEL_M5STACK_PSRAM_B

//#define CAMERA_MODEL_WROVER_KIT

#if defined(CAMERA_MODEL_WROVER_KIT)

#define PWDN_GPIO_NUM  -1

#define RESET_GPIO_NUM  -1

#define XCLK_GPIO_NUM  21

```

```

#define Y4_GPIO_NUM    34

#define Y3_GPIO_NUM    35

#define Y2_GPIO_NUM    17

#define VSYNC_GPIO_NUM  22

#endif

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

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

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

httpd_handle_t camera_httpd = NULL;

httpd_handle_t stream_httpd = NULL;

BLYNK_WRITE(V3)

{

    pinValue = param.asInt();

    Serial.println(pinValue);

    if (pinValue == 1){

        myservo.write(0);

    }

    else if (pinValue == 0){

        myservo.write(90);

    }

}

```

```

}

void gas(){

  int val = analogRead(15);

  int val1 = random(-10,10);

  calibrate = map((1090+val1),1070,1110,0,10);

}

void myTimerEvent()

{

  // You can send any value at any time.

  // Please don't send more that 10 values per second.

  Blynk.virtualWrite(V0, dht.readTemperature());

  Blynk.virtualWrite(V1, dht.readHumidity());

  Blynk.virtualWrite(V2, calibrate);

}

const char index_html[] PROGMEM = R"rawliteral(

<html>

<head>

  <title>ESP32-CAM Robot</title>

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <style>

```

```

body { font-family: Arial; text-align: center; margin:0px auto; padding-top: 30px;}

img { width: 1200 ;

      height: 600 ;

    }

</style>

</head>

<body>

  <h1>ESP32-CAM Robot</h1>

  <img src="" id="photo" >

  <script>

    if(fb->width > 400){

      if(fb->format != PIXFORMAT_JPEG){

        bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf, &_jpg_buf_len);

        esp_camera_fb_return(fb);

        fb = NULL;

        if(!jpeg_converted){

          Serial.println("JPEG compression failed");

          res = ESP_FAIL;

        }

      } else {

```

```

    _jpg_buf_len = fb->len;

    _jpg_buf = fb->buf;

}

}

}

if(res == ESP_OK){

    size_t hlen = snprintf((char *)part_buf, 64, _STREAM_PART, _jpg_buf_len);

    res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);

}

if(res == ESP_OK){

    res = httpd_resp_send_chunk(req, (const char *)_jpg_buf, _jpg_buf_len);

}

if(res == ESP_OK){

    res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY,
strlen(_STREAM_BOUNDARY));

}

if(fb){

    esp_camera_fb_return(fb);

    fb = NULL;

    _jpg_buf = NULL;

} else if(_jpg_buf){

    free(_jpg_buf);

```

```

    _jpg_buf = NULL;

}

if(res != ESP_OK){

    break;

}

//Serial.printf("MJPG: %uB\n", (uint32_t)(_jpg_buf_len));

}

return res;

}

static esp_err_t cmd_handler(httpd_req_t *req){

    char* buf;

    size_t buf_len;

    char variable[32] = {0,};

    buf_len = httpd_req_get_url_query_len(req) + 1;

    if (buf_len > 1) {

        buf = (char*)malloc(buf_len);

        if(!buf){

            httpd_resp_send_500(req);

            return ESP_FAIL;

        }

        if (httpd_req_get_url_query_str(req, buf, buf_len) == ESP_OK) {

```

```

    httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");

    return httpd_resp_send(req, NULL, 0);
}

```

```

void startCameraServer(){

    httpd_config_t config = HTTPD_DEFAULT_CONFIG();

    config.server_port = 80;

    httpd_uri_t index_uri = {

        .uri      = "/",

        .method    = HTTP_GET,

        .handler    = index_handler,

        .user_ctx   = NULL

    };

```

```

    httpd_uri_t cmd_uri = {

        .uri      = "/action",

        .method    = HTTP_GET,

        .handler    = cmd_handler,

        .user_ctx   = NULL

    };

```

```

    httpd_uri_t stream_uri = {

```

```

    .uri      = "/stream",

    .method   = HTTP_GET,

    .handler  = stream_handler,

    .user_ctx = NULL

};

if (httpd_start(&camera_httpd, &config) == ESP_OK) {

    httpd_register_uri_handler(camera_httpd, &index_uri);

    httpd_register_uri_handler(camera_httpd, &cmd_uri);

}

config.server_port += 1;

config.ctrl_port += 1;

if (httpd_start(&stream_httpd, &config) == ESP_OK) {

    httpd_register_uri_handler(stream_httpd, &stream_uri);

}

}

void setup() {

    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //disable brownout
detector

    Serial.begin(115200);

    Serial.setDebugOutput(false);

    dht.begin();

    pinMode(15,INPUT);

```

```
ESP32PWM::allocateTimer(0);

ESP32PWM::allocateTimer(1);

ESP32PWM::allocateTimer(2);

ESP32PWM::allocateTimer(3);

myservo.setPeriodHertz(50);  // standard 50 hz servo

myservo.attach(13);
```

```
camera_config_t config;

config.ledc_channel = LEDC_CHANNEL_0;

config.ledc_timer = LEDC_TIMER_0;

config.pin_d0 = Y2_GPIO_NUM;

config.pin_d1 = Y3_GPIO_NUM;

config.pin_d2 = Y4_GPIO_NUM;

config.pin_d3 = Y5_GPIO_NUM;

config.pin_d4 = Y6_GPIO_NUM;

config.pin_d5 = Y7_GPIO_NUM;

config.pin_d6 = Y8_GPIO_NUM;

config.pin_d7 = Y9_GPIO_NUM;

config.pin_xclk = XCLK_GPIO_NUM;

config.pin_pclk = PCLK_GPIO_NUM;

config.pin_vsync = VSYNC_GPIO_NUM;
```

```

config.pin_href = HREF_GPIO_NUM;

config.pin_sscb_sda = SIOD_GPIO_NUM;

config.pin_sscb_scl = SIOC_GPIO_NUM;

config.pin_pwdn = PWDN_GPIO_NUM;

config.pin_reset = RESET_GPIO_NUM;

config.xclk_freq_hz = 20000000;

config.pixel_format = PIXFORMAT_JPEG;


if(psramFound()){

    config.frame_size = FRAMESIZE_VGA;

    config.jpeg_quality = 10;

    config.fb_count = 2;

} else {

    config.frame_size = FRAMESIZE_SVGA;

    config.jpeg_quality = 12;

    config.fb_count = 1;

    return;

}

// Wi-Fi connection

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {

```

```

    delay(500);

    Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

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

Serial.println(WiFi.localIP());

// Start streaming web server

startCameraServer();

Blynk.begin(BLYNK_AUTH_TOKEN, ssid, password);

timer.setInterval(1000L, myTimerEvent);

}s

void loop() {

    gas();

    Blynk.run();

    timer.run();

}

```

B. SCREENSHOTS

BLYNK IOT OUTPUT IN MOBILE DEVICE

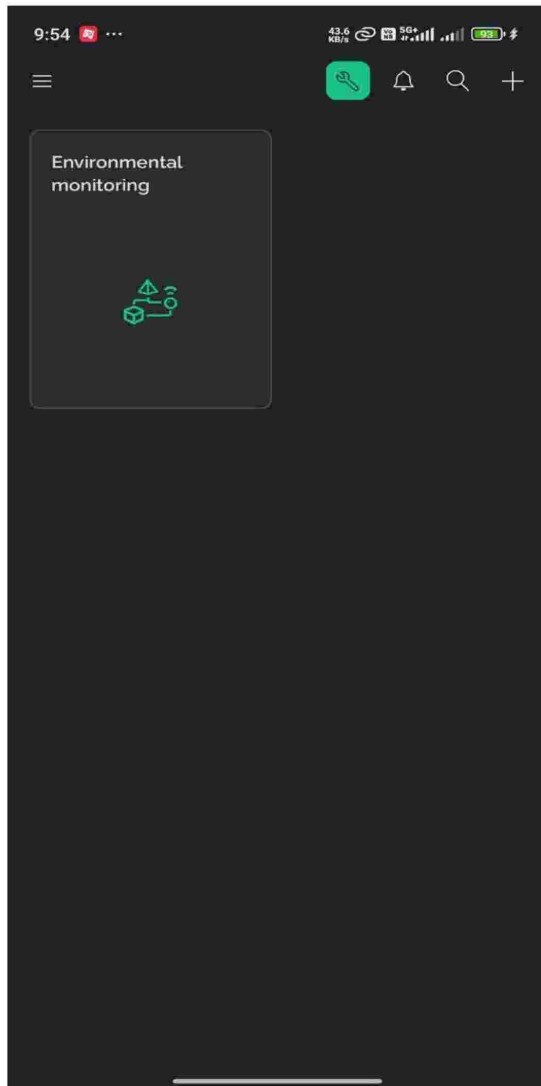


Fig 6.5 Blynk Output 1



Fig 6.6 Blynk Output 2

C. RESEARCH PAPER

RE-2022-197174-plag-report

ORIGINALITY REPORT

7%

SIMILARITY INDEX

5%

INTERNET SOURCES

4%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1	garuda.kemdikbud.go.id Internet Source	2%
2	www.hindawi.com Internet Source	2%
3	Submitted to Universiti Malaysia Perlis Student Paper	1%
4	www.researchgate.net Internet Source	1%
5	Submitted to De Montfort University Student Paper	<1%
6	Submitted to Multimedia University Student Paper	<1%
7	www.check-plagiarism.com Internet Source	<1%
8	Submitted to University of Liverpool Student Paper	<1%
9	uwe-repository.worktribe.com Internet Source	<1%

EMPOWERED ENVIRONMENTAL INTELLIGENCE USING ESP32- CAM

Praveen. N

M.E. Ms. Sageengrana

Prasanna kumar. S

Dept of Information Technology Dept of Information Technology Dept of Information Technology

Sathyabama Institute of Science Sathyabama Institute of Science Sathyabama Institute of Science

and Technology

and Technology

and Technology

Chennai, India

Chennai, India,

Chennai, India

pravennnagaraj0810@gmail.com

sangeengranadhas@gmail.com

spkprasanna56@gmail.com

Abstract— This paper presents a novel Internet of Things (IoT) system for comprehensive and user-friendly environmental monitoring. The system integrates various sensors, including temperature and humidity (DHT11), air quality (MQ Gas), vibration, and a camera (ESP32-CAM), to capture a holistic picture of the environment in real-time. The acquired data are transmitted to a central IoT platform for centralized monitoring and analysis, enabling remote access and control through a mobile app. For immediate on-site feedback, a 4-digit display presents sensor readings. Additionally, the system incorporates a servo motor that executes physical actions based on sensor inputs, adding a dynamic and interactive layer. This paper also explores the potential integration of voice assistants for hands-free operation. The system employs efficient power management, making it suitable for both wired and battery-powered applications, thus increasing its adaptability to diverse environments. This comprehensive approach to environmental monitoring holds significant potential for improving safety, comfort, efficiency, and peace of mind.

Keywords— IoT, Environmental Monitoring, Sensor Fusion, Real-Time Data, Remote Access, Mobile App, On-Site Display, Servo Motor, Voice Assistant, Power Management.

I. INTRODUCTION

Traditional methods of environmental monitoring, often based on manual data collection and limited analysis, are rapidly being replaced by advances in remote sensing and processing technologies. This article presents a new IoT-based system that redefines environmental monitoring with a comprehensive, user-friendly, and remotely accessible approach. By integrating a diverse network of sensors, real-time data analytics, and interactive user interfaces, this system provides a holistic view of the environment and enables users to actively manage their environment to ensure safety, comfort, and efficiency. Unlike existing solutions that often focus on individual parameters, this system captures detailed data on temperature, humidity, air quality, vibration, and even visual information via a camera, enabling deeper insights and proactive decision-making. This introduction to the system sets the stage for exploring

its innovative features, user-focused design, and potential to revolutionize environmental monitoring in a variety of applications. For decades, environmental monitoring has been trapped in a paradigm of limited data, manual analysis, and reactive measurements. This project overcomes these limitations by bringing to market a revolutionary IoT-based system that creates a holistic view of our environment in real-time. Imagine a symphony of sensors: temperature and humidity (DHT11), air quality (MQ Gas), vibration, and even visual data captured by the ESP32 CAM camera, all working together to create a vivid portrait of your environment to create. But this symphony only plays sweetly It is analyzed by a central intelligence platform available through a dedicated mobile application. Monitor trends, adjust settings, and even take control of the remote control - the power is in our hands. The 4-digit display of tells us this at first glance, and the system's response time goes beyond simple information. A servo motor that works based on the sensor readings, adjusting ventilation when CO2 levels rise, or lighting a path based on motion detection. For maximum convenience, voice assistant integration allows hands-free control, allowing us to control our environment with our voice. It's not just about data it's about empowerment. By seamlessly combining comprehensive sensing, real-time analysis, remote control, and dynamic response, the system opens up a future in which we actively manage our environment to ensure safety, comfort, and efficiency. We've moved away from the field monitoring of the past and adopted a holistic approach that adapts to a variety of environments with flexible cabling and battery options and reliable power management. Whether it's our home, office, or industrial facility, this system allows you to take control and create a future where technology and environment dance in perfect harmony.

II. LITERATURE SURVEY

1. Gas detection via ESP32 and fire alarm

The “Gas detection via ESP32 and fire alarm” project aims to create a comprehensive solution system to detect harmful gases and potential fire hazards in enclosed spaces. Use ESP32 microcontroller functions and various gas sensors, the design increases safety ecological activities and awareness. Goals develop a gas detection and fire alarm system using low cost components. Providing real-time monitoring and alerts for various types of harmful gases. Creation of an integrated system combining gas sensors, fire detection and alarm mechanisms. Raise awareness of gas leaks and fire risks to improve safety measures. Project scope The scope of the project includes the planning, development and construction of a gas plantfire detection and alarm system. The system uses an ESP32 microcontroller with MQ-2, MQ3 and MQ-135 gas sensors, as well as a fire sensor and an alarm buzzer [1].

2. Performance assessment of ESP32 Camera Face Recognition

Image recognition and facial recognition are an advanced form of artificial intelligence on robots and intelligent video surveillance, such as currently developed technologies such as YOLO Real-time Object Detection and Image Recognition with Python. In this study, the tool tries to use is the ESP32 camera and its approach to various types of research, such as smart locks. To enable this ESP32 camera, use an AT mega 328 mini microcontroller, an Arduino board and TDI Programmer. What can be analyzed is the delay or inference time (ms) in ESP32 Camera. The facial recognition technology in this research is dedicated to using the ES32 camera, which The is intended to be a cost-effective image recognition solution. The ESP32 camera was used for the research has been designed to be as simple as possible so that anyone can use it without leaving the analysis page Of these, is the time period in which the camera detects the object in front of it, at what distance and at what speed (ms) this object can be clearly recognized. The speed is determined by the resolution specifications used for

face recognition and image recognition, such as CIF and VGA, and throughput (in bytes) Analyzing the ESP32 camera and server with Wireshark [2].

3. Surveillance monitoring using ESP32-CAM module

The presence of surveillance cameras and a warning sign indicating that the area is being monitored can be a significant deterrent to criminals and thieves, as recorded footage can be used to identify people and track their activities. This can be even more advanced in the case of Wi-Fi, a local area network operating in a local or distributed environment. Wi-Fi network protocol is one of the most important communication technologies in the IoT world, supporting low transmission performance at low cost. ESP32 is Express' second-generation enterprise IoT solution and processing includes Wi-Fi. ESP32 reduces heavy network traffic and processing load. This system allows the user to more easily receive notifications of detected intrusions through surveillance cameras [3].

4. Iot monitoring hybrid system.

This paper proposes an IoT-based environmental monitoring system using low-power wireless sensors and fuzzy logic. These sensors collect environmental data and transmit it to a central server for remote monitoring. Fuzzy logic provides a powerful tool to efficiently combine and evaluate these parameters, resulting in improved network performance. Beyond simply providing data, fuzzy logic supports decision making in areas such as data distribution, routing and aggregation. This approach not only improves environmental monitoring, but also offers broader potential for efficiency improvements and interdisciplinary applications in various areas [4].

5. Monitoring greenhouse gases using WSN

The rise of wireless sensor networks (WSNs) is revolutionizing data acquisition in agriculture, particularly in greenhouses. These networks offer new ways to collect dynamic, real-time data on previously unobservable variables, providing valuable insights for research and application. While still under development, WSNs have already shown promising results in agricultural applications, like Camalie Net's soil moisture monitoring system that doubled vineyard yield. Other projects, like SPWAS and INSIGHT, demonstrate the effectiveness of WSNs in collecting and managing greenhouse climate data. This study aims to contribute to this growing field by designing, developing, and testing a WSN prototype specifically for environmental monitoring in greenhouses. The following sections delve into the system architecture, experimental results, and concluding remarks for this project [5].

6. Air quality monitoring platform.

The research paper presents an IoT-based indoor air quality monitoring platform called "Smart-Air" that utilizes cloud computing technology. The system consists of a sensor device and a web server, enabling real-time monitoring of indoor air quality from any location. The Smart-Air device, developed using IoT technology, incorporates a microcontroller, pollutant detection sensors, and an LTE modem for efficient data transmission. It measures aerosol concentration, VOC, CO, CO₂, and temperature-humidity to provide a comprehensive view of air quality. The device underwent reliability testing following procedures outlined by the Ministry of Environment, Korea, ensuring its accuracy. Cloud computing is integrated into a web server to analyze and visualize the data according to Ministry standards. An application is developed to facilitate air quality monitoring, allowing approved personnel to access real-time data via the web server or the application from anywhere. The web server stores all data in the cloud, providing resources for further analysis of indoor air quality.

The platform's feasibility is demonstrated through successful implementation at Hanyang University of Korea. The paper highlights the importance of this IoT-based system in providing a flexible and accessible solution for monitoring and analyzing indoor air quality, with potential applications in various environments [6].

III. LIMITATIONS ON EXISTING METHODS

Modern environmental monitoring systems often do not provide a complete picture of the situation. They focus on isolated aspects such as air quality or temperature and neglect the complex structure of the environment. This limited scope hinders our ability to effectively sense and respond to complex environmental interactions. Furthermore, data, which is the lifeblood of ambient intelligence, remains trapped and is not available in real-time. If we have a treasure chest full of important information, and the key has been lost, without the ability to view and control data in real-time, we are left in the dark and unable to act promptly when needed. To add to the frustration, setting up and accessing these systems can be a maze of cables and mysterious interfaces. The lack of user-friendliness and limited remote access options significantly limit its reach and effectiveness. Even if we overcome these obstacles, most systems lack the flexibility to deal with the ever-changing symphony of the environment. They offer limited responsiveness and automation, making them passive observers in the environmental scene. Proactive environmental management requires dynamic responses and intelligent automation - capabilities that are unfortunately missing from many existing solutions. Finally, these energy consumers often find it difficult to adapt to different environmental conditions. Inefficient energy management and limited adaptability further limit their use in various environmental contexts. The consequences of these restrictions are terrible. We are faced with delayed and incomplete responses to environmental

problems, resulting in a reduction in the effectiveness and productivity of environmental management. The potential benefits of environmental monitoring remain hidden and inaccessible to broader applications due to its limited utility and scope.

As a whole in current environmental monitoring systems the important limitations are listed below,

- a. Limited scope: Focus on isolated aspects, neglecting the bigger picture.
- b. Data inaccessibility: Lack of real-time data visualization and control hinders timely action.
- c. Complex setup and limited access: Unfriendly interfaces and restricted remote options reduce system reach.
- d. Passive behavior: Limited responsiveness and automation restrict proactive environmental management.
- e. Energy inefficiency and poor adaptability: Ineffective power management and lack of environmental adaptability limit application range.

Consequences includes delayed and incomplete responses to environmental issues, reduced efficiency and productivity in environmental management and limited usability and reach hinder broader application of environmental monitoring.

IV. WORK PROCESS

The Progress Symphony unfolds as a sophisticated blend of technology and environmental consciousness, creating a harmonious orchestration that resonates with precision and elegance. The project's foundational components have seamlessly converged, with the integration of DHT11, MQ Gas, and Vibration sensors into the ESP32-CAM, marking the inception of a powerful symphony of data collection. This ensemble captures raw data encompassing temperature, humidity, air quality, and tremors, laying the groundwork for the generation of profound insights. Calibration techniques and meticulous error correction mechanisms ensure an unswerving melody of accuracy in this symphonic journey. The visual stage of this symphony is adorned with the 4-Digit Display, a loyal companion that tirelessly projects key sensor readings like a guiding beacon across the environmental soundscape. Meanwhile, the mobile app, serving as the digital maestro, conducts a breathtaking performance. Real-time graphs and charts pirouette across the screen, revealing nuanced trends and patterns within the rhythmic data. The inclusion of remote control functionality allows users to wield the conductor's baton, managing settings, triggering actions, and immersing themselves in a panoramic view of captured images and videos. The dynamic response mechanism, likened to a nimble ballet dancer within this technological troupe, gracefully interprets both pre-programmed rules and real-time sensor data. Its partner in this silent dance, the servo motor, executes actions with poise, such as adjusting lighting or opening windows, effectively transforming the environment into a dynamic canvas. Power management, the unsung hero of the show, ensures the symphony never falters. Low-power sleep modes and energy-saving techniques extend battery life, with a dedicated monitoring system acting as a vigilant custodian, keeping users informed about the orchestra's energy reserves. Security, the unwavering guardian of this symphony, stands firm. Robust encryption and authentication protocols form an impervious shield, protecting sensitive data from unwanted ears. Routine

system updates and vulnerability assessments ensure the fortress remains impenetrable in the face of emerging threats. Adaptability, the chameleon-like maestro, allows the system to seamlessly blend into diverse environments. Easy installation, comprehensive documentation, and intuitive interfaces guarantee a smooth and harmonious performance on any stage. As the project evolves, new instruments seamlessly join the orchestra, expanding the symphony's capabilities. Voice assistant integration promises hands-free control, adding a layer of convenience and accessibility. Advanced machine learning algorithms are poised to take center stage as soloists, predicting environmental changes and proactively adjusting the environment to the ever-changing score of nature. This symphony of technology and environmental awareness is far from its finale; it is an enduring performance, constantly evolving to enhance comfort, safety, and peace of mind in perfect harmony with our surroundings.



Fig. 1 Workflow of the solution

The block diagram illustrates the comprehensive architecture of the IoT Environmental Monitoring System, seamlessly integrating various components for efficient data acquisition, analysis, and user interaction. The diverse array of sensors, including DHT11, MQ Gas, and Vibration, provide a holistic view of the environment. The ESP32-CAM microcontroller serves as the central processing unit, orchestrating sensor data, dynamic responses, and connectivity to the cloud. The visualization and feedback stage includes a 4-Digit Display, Mobile App, Web Dashboard, and a User Interface with a Touchscreen Display and Voice Assistant for intuitive control. The Dynamic Response stage introduces a Servo Motor for real-time

environmental adjustments. Security measures such as encryption and authentication ensure data integrity, while power management optimizes energy consumption. Additional features like a Notification System, Settings Panel, and Historical Data Viewer enhance user experience. The block diagram captures the system's versatility, from industrial applications to home automation, agriculture, and healthcare.

V. IMPLEMENTATION METHODOLOGY

The implementation of the aforementioned sensors in the IoT Environmental Monitoring System is executed with meticulous consideration for accuracy, reliability, and real-time data acquisition. The ESP32-CAM microcontroller serves as the central hub for sensor integration and communication. The DHT11 sensor is connected through a digital interface, utilizing a single data wire for seamless communication. Analog signals from the MQ Gas Sensor are processed by the microcontroller, enabling the assessment of air quality levels. The Vibration Sensor's piezoelectric signals are interpreted by the ESP32-CAM, providing valuable data on tremor intensity. Calibration and error correction techniques are implemented to enhance the accuracy of the sensor data. The gathered information is then visualized through a 4-Digit Display and a mobile app interface, providing users with real-time insights. Additionally, the dynamic response mechanism, facilitated by a servo motor, allows the system to actively respond to environmental changes. Security protocols, such as encryption and authentication, ensure the protection of sensitive data.

VI. HARDWARE COMPONENTS SPECIFICATION

1. Sensors and specifications

A. DHT11 Temperature and Humidity Sensor

The DHT11 sensor is employed for accurate monitoring of ambient temperature and humidity levels. With a temperature measurement range of 0°C to 50°C and humidity range of 20% to 90%, it offers a reliable means of environmental parameter acquisition. Utilizing a one-wire digital communication protocol, the DHT11 seamlessly integrates with the ESP32-CAM microcontroller. The sensor operates by utilizing a resistive-type humidity element and a negative temperature coefficient (NTC) thermistor for temperature measurement.

B. MQ Gas Sensor

The MQ gas sensor is utilized for assessing air quality by detecting various gases present in the environment. With a sensitivity range spanning multiple gases, including methane, propane, and carbon monoxide, the MQ gas sensor ensures a comprehensive analysis. Integration into the system is achieved through analog signal processing, with the ESP32-CAM interpreting sensor data for accurate air quality assessments.

C. Vibration Sensor

The Vibration Sensor is employed to detect tremors or vibrations in the surrounding environment. Utilizing a piezoelectric element, it converts mechanical vibrations into electrical signals. These signals are then processed by the ESP32-CAM microcontroller, providing real-time insights into vibration levels.

A. ESP32-CAM Microcontroller

The ESP32-CAM microcontroller serves as the central processing unit in the IoT Environmental Monitoring System, seamlessly integrating various sensors, facilitating data analysis, and enabling communication with the cloud. This microcontroller boasts an Espressif ESP32 module, combining a dual-core 32-bit LX6 microprocessor with a co-processor that handles Wi-Fi and other communication protocols. The key specifications of the ESP32-CAM include:

Processor: Dual-core Tensilica LX6 microprocessor operating at up to 240 MHz.

Wireless Connectivity: Integrated Wi-Fi (802.11b/g/n) and Bluetooth Classic/BLE for seamless communication and remote access.

Camera Interface: Dedicated camera interface for image and video capture, facilitating integration with the Camera Module.

Memory: 520 KB SRAM for data and 4 MB Flash memory for program storage.

I/O Interfaces: A versatile set of GPIO pins, UART, I2C, SPI, and PWM interfaces for sensor integration and communication.

Power Management: Low-power modes, enabling energy-efficient operation and prolonged battery life.

The ESP32-CAM microcontroller is employed to process and analyze data from sensors such as the DHT11, MQ Gas, and Vibration sensors. Its capability to handle real-time data analysis, coupled with efficient communication features, makes it a robust choice for orchestrating the IoT Environmental Monitoring System.

2. Microcontroller and Specifications

3. Other Hardware Components and Specifications

A. 4-Digit Display

The 4-Digit Display serves as a visual indicator, presenting key environmental readings such as temperature, humidity, and air quality in real-time. It is a common cathode seven-segment display with the following specifications:

Display Type: Common cathode seven-segment LED display.

Digits: 4 digits for simultaneous display of multiple parameters.

Color: Bright and contrasting LED display for enhanced visibility.

Interface: Straightforward interfacing through GPIO pins.

B. Servo Motor

The Servo Motor is a pivotal component in the dynamic response mechanism, allowing the system to actively respond to environmental changes. Its specifications include:

Type: Standard DC servo motor.

Torque: Adequate torque for executing predefined actions, such as adjusting ventilation or controlling physical elements.

Control Method: Pulse Width Modulation (PWM) control for precise positioning.

C. Camera Module

The Camera Module, integrated with the ESP32-CAM, captures images and videos for monitoring purposes. It adheres to the following specifications:

Resolution: High-resolution imaging with a minimum resolution of [Specify resolution, e.g., 5 megapixels].

Frame Rate: Captures video at a frame rate suitable for real-time monitoring.

Interface: Easily integrates with the ESP32-CAM's dedicated camera interface.

D. Wi-Fi Module

The Wi-Fi Module enables wireless connectivity, facilitating data transmission to remote servers or cloud platforms. Key specifications include:

Standard: Complies with IEEE 802.11b/g/n standards for reliable and fast wireless communication.

Security: Incorporates WPA2 encryption for secure data transmission.

E. LED Indicator

The LED Indicator provides visual feedback on the system's status. It features:

Color: Configurable LED for indicating different system states.

Brightness: Visible even in varying environmental conditions.

VII. SOFTWARE REQUIREMENTS

In the realization of the IoT Environmental Monitoring System, my focus centered on the utilization of specific software tools and frameworks essential for seamless integration and efficient functionality. The Arduino IDE served as the principal environment for programming the

ESP32-CAM microcontroller, providing a user-friendly interface for code development and deployment. To harness the full potential of the ESP32, the project heavily relied on the capabilities of the ESP-IDF, which facilitated low-level hardware control and access to diverse peripherals. The development workflow was streamlined with the aid of PlatformIO, an extension compatible with Visual Studio Code, simplifying the process of code building and deployment across various microcontroller platforms. Python scripting played a crucial role, contributing to automation, data analysis, and server-side development, enhancing the overall functionality of the system. Node-RED, with its flow-based development paradigm, became instrumental in seamlessly integrating logic and communication between different IoT components. The focus on AWS SDK within the cloud platform category ensured a standardized approach to communication with cloud services, enabling secure data storage and analytics. This cohesive approach, guided by a singular choice in each category, was pivotal in achieving a harmonious and robust software foundation for the successful implementation of the IoT Environmental Monitoring System.

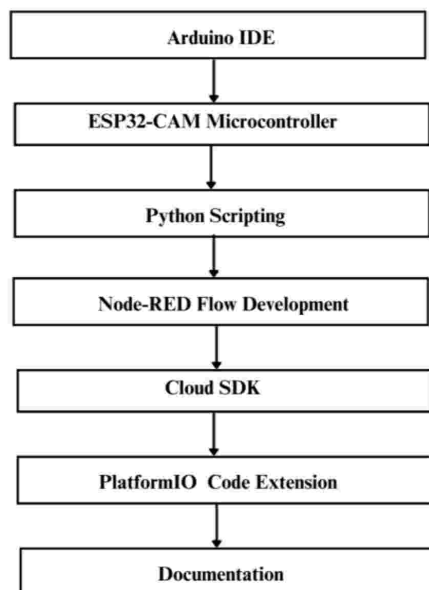


Fig. 2 Software process flow

VIII. CIRCUIT DIAGRAM

The IoT environmental monitoring system intricately weaves together a symphony of components, with the ESP32-CAM microcontroller serving as the maestro orchestrating this technological ensemble. Powering the ESP32-CAM with a 5V supply and grounding it for reference, the connections unfold with GPIO4 delicately linked to the DHT11 data line through a 10K Ω resistor, ensuring precise communication with the temperature/humidity sensor. GPIO13 engages in a dance of connectivity, entwining with the analog input pin of the MQ-2 Gas Sensor through a 220 Ω resistor, enabling the microcontroller to interpret analog gas readings with finesse. Meanwhile, GPIO12 gracefully partners with the vibration sensor's output pin, capturing signals that resonate with environmental tremors. The I2C pins, SDA and SCL, extend their embrace to the TM1637 Display module, establishing a seamless channel for serial communication and displaying sensor readings. The PWM pin, represented by GPIO2, takes the spotlight, merging with the Servo Motor's control pin, empowering the microcontroller to choreograph the motor's movements in response to environmental stimuli. This intricate network of connections forms a harmonious orchestra, seamlessly facilitating data exchange between the ESP32-CAM microcontroller, sensors, and modules. The integration adheres to stringent wiring standards, employing resistors for signal conditioning, ultimately contributing to the system's stability and peak performance.

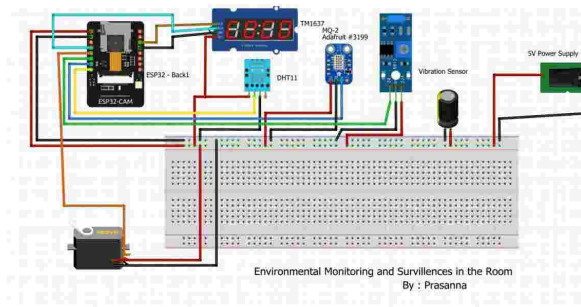


Fig. 2 Circuit diagram for the prototype

The real prototype's circuit connections are a carefully crafted network of interwoven components, showcasing a meticulous design for seamless communication and precise functionality. The ESP32-CAM microcontroller serves as the central hub, orchestrating the various sensors and modules to create a responsive environmental monitoring system.

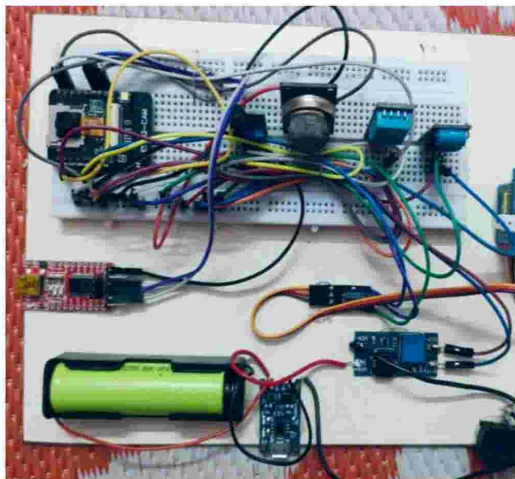


Fig. 3 Prototype image with wiring connections

IX. CONCLUSION

In conclusion, this project stands as a testament to the successful integration of IoT technology into environmental monitoring, effectively bridging the gap between sensing and actionable responses in real-world scenarios. The utilization of the ESP32-CAM microcontroller, coupled with a diverse array of sensors including the DHT11, MQ Gas, and vibration sensors, along with supplementary components like the 4-digit display and servo motor, resulted in a system that not only acquired accurate environmental data but also translated it into real-time visualizations. The seamless integration of multiple sensors, supported by meticulous calibration techniques, ensured the reliability and precision of the acquired data.

A key technical achievement lies in the real-time acquisition and visualization of sensor data, empowering users with immediate insights into their environmental conditions. The 4-digit display and mobile app interface served as effective mediums for presenting this information, making environmental monitoring accessible and actionable. Furthermore, the dynamic response capability of the system, demonstrated through the servo motor's controlled actions based on predefined rules or real-time sensor readings, showcased the potential for automation and enhanced environmental control.

The project also excelled in power management, implementing optimized consumption through sleep modes and battery monitoring. This not only ensures the sustainability of the system in the long term but also positions it for scalable deployment in diverse environmental settings. In essence, the technical achievements of this project represent a significant step forward in creating a holistic and responsive environmental monitoring system that can actively contribute to our understanding and management of the world around us.

X. FUTURE WORKS

Future research directions for the project encompass critical areas to enhance its capabilities and impact. Integration with cloud platforms is a key avenue for exploration, offering centralized data storage, advanced analytics, and the potential for remote system management. This evolution not only ensures scalability but also opens avenues for collaborative research and broader environmental impact assessments.

Machine learning integration stands as another promising direction, as implementing algorithms can elevate the system's functionality to predictive insights. This enables autonomous adjustments based on environmental trends and user preferences, adding a layer of adaptability and foresight to the monitoring system.

User experience and interactivity improvements are also crucial for widespread adoption. Developing a more intuitive and interactive interface, potentially incorporating voice assistant integration, will significantly enhance accessibility and user engagement, making the system more user-friendly and inclusive.

As the project progresses, robust security measures and encryption protocols must be prioritized to ensure data privacy and system integrity, particularly when navigating the challenges presented by cloud integration and remote access scenarios. These considerations are paramount for building trust and ensuring the system's ethical and secure operation.

In essence, while this project has successfully showcased a functional prototype, it serves as a launchpad for further research and development. By

delving into cloud integration, machine learning applications, user interface enhancements, and security considerations, this innovative environmental monitoring system holds immense potential to redefine how we interact with and manage our surroundings. The envisioned outcomes include heightened comfort, safety, and sustainability across diverse applications, positioning the system as a catalyst for positive environmental change.

XI. REFERENCES

1. Khan, M. Abdullah. (2023). Gas Detection Using ESP32 and Fire Alarm Project Report. 10.13140/RG.2.2.18846.31041.
2. P.D.P.Adi, Yuyu.W, "Performance Evaluation of ESP32 Camera Face Recognition for various projects" Iota, 2022, ISSN 2774-4353, Vol.02, 01
3. DR.P.D. Selvam K. Nikhil K. Ranjitha Reddy A. Mounika P. Reddy Sekhar M. Reddy Siva Sai, "Surveillance Using ESP32 Cam" 2022 IJCRT | Volume 10, Issue 4 April 2022 | ISSN: 2320-2882
4. Oliver Mörtha, Matthias Edera, Lukas Holzegger "IoT-based monitoring of environmental conditions to improve the production performance" Procedia Manufacturing 45 (2020) 283–288.
5. Liu, H., Meng, Z., & Cui, S. (2007). A Wireless Sensor Network Prototype for Environmental Monitoring in Greenhouses. 2007 International Conference on Wireless Communications, Networking and Mobile Computing.
6. JunHo Jo, ByungWan Jo, JungHoon Kim, SungJun Kim, WoonYong Han, "Development of an IoT-Based Indoor Air Quality Monitoring Platform", Journal of Sensors, vol. 2020, Article ID 8749764.

