

EDGE-INTELLIGENT DATA REDUCTION AND ADAPTIVE COMMUNICATION IN STM32-ESP32 BASED IOT GATEWAYS

Chandana T V
Department of Electronics and Communication,
JSSATE,
Bangalore, India
chandanatvchandu@gmail.com

Dr. Usha S M
Department of Electronics and Communication,
JSSATE,
Bangalore, India
chandanatvchandu@gmail.com

Mrs. Kavitha H S
Department of Electronics and Communication,
JSSATE,
Bangalore, India
chandanatvchandu@gmail.com

Ranjitha Bhat
Department of Electronics and Communication,
JSSATE,
Bangalore, India
chandanatvchandu@gmail.com

Abstract: The rapid growth of Internet of Things (IoT) systems has led to massive volumes of sensor data being generated continuously from remote and distributed sensing nodes. Transmitting all raw sensor data directly to the cloud results in excessive bandwidth usage, increased power consumption, higher operational costs, and network congestion. These problems arise especially in battery-powered IoT gateways using Wi-Fi-based communication. This project proposes an edge-intelligent IoT gateway using a dual-MCU architecture consisting of STM32 and ESP32 microcontrollers. The STM32 performs local data preprocessing, filtering, aggregation, and event detection, while the ESP32 is used only for uploading meaningful or exception-based data to the cloud. The system uses adaptive communication techniques such as time-based, threshold-based, and event-based data transmission to balance data accuracy and power efficiency. The objective is to significantly reduce transmitted data volume Without losing meaningful and actionable information. By moving intelligence to the edge, the system improves battery life, reduces network load, and enhances scalability. The proposed framework is suitable for industrial, environmental, agricultural, and smart city applications where long-term, low-power operation is required.

Keywords: IoT, Edge Computing, STM32, ESP32, Data Reduction, Adaptive Communication, Low Power.

I. INTRODUCTION

All Over the past decade, the internet of things (IOT) has steadily evolved and found applications across a wide range of domains, including industrial monitoring, smart environments, and healthcare systems. in such applications, multiple sensors are deployed to continuously observe parameters like temperature, humidity, pressure, vibration, light intensity, gas levels, and motion. conventionally, the data generated by these sensors is transmitted to cloud platforms, where it is stored and further processed for analysis and visualization. however, as the scale of iot deployments continues to grow, the volume of data being produced has increased significantly, resource management.

In many existing IoT architectures, data processing is largely centralized in the Blynk cloud. Edge devices or sensor nodes typically forward raw data streams directly to remote servers without any form of local filtering. while this simplifies the design of edge devices, it also introduces certain limitations. continuous transmission of data places a heavy demand on network bandwidth and leads to increased energy consumption, particularly when wireless communication technologies such as wi-fi or cellular networks are used. for battery-operated systems, this can reduce operational lifetime and require more frequent maintenance, which may not be feasible in remote or large-scale deployments.

It is also important to note that a considerable portion of the collected sensor data does not carry significant information. under normal operating conditions, many parameters tend to remain within a stable range for extended periods. meaningful insights are typically derived only when there are noticeable deviations or abnormal patterns. for example, in industrial environments, parameters such as temperature or vibration often remain steady, and only sudden fluctuations indicate potential faults or system anomalies. transmitting all data, including routine readings, can therefore result in inefficient utilization of both bandwidth and power.

In this approach, edge devices are capable of performing preliminary analysis, thereby reducing the need to transmit large volumes of raw data. in the present work, an edge-intelligent IoT gateway is implemented using STM32 and ESP32 microcontrollers. The STM32 is primarily responsible for low-power data handling and local decision-making, whereas the ESP32 is utilized for communication with the cloud when relevant information needs to be transmitted.

By applying local processing along with adaptive communication strategies, the proposed system aims to reduce unnecessary data transfer while maintaining the quality of information delivered to the cloud. This not only improves energy efficiency but also enhances overall system scalability and reliability. such an approach is particularly useful in applications where continuous monitoring is required but only selective data holds practical significance.

The main objective of this work is to design and implement an edge-enabled IoT gateway based on STM32 and ESP32 microcontrollers, with a focus on improving communication efficiency. rather than relying entirely on cloud-based processing, the proposed system introduces local data handling at the gateway level. this shift helps in reducing unnecessary data transmission, lowering energy consumption, and supporting more scalable IoT deployments.

A key objective is to enable data processing directly at the edge. by handling sensor data locally, the system can respond more quickly to changes without depending on continuous cloud interaction. this not only reduces latency but also makes the system more suitable for real-time monitoring applications.

Another important goal is to limit the amount of data sent over the network. instead of transmitting all raw sensor readings, the system is designed to identify and forward only relevant or significant information. This selective communication helps in minimizing network congestion and makes better use of available bandwidth.

The work also aims to incorporate flexible communication strategies. Different approaches, such as time-based updates, event-driven transmission, and threshold-based reporting, are considered to adapt data transfer according to the situation. this ensures that communication remains efficient under varying operating conditions.

Energy efficiency is also taken into account during the system design. by reducing unnecessary wireless transmissions and utilizing low-power features of the microcontrollers, the overall power consumption can be kept under control. this is particularly important for systems that operate on limited power sources.

Finally, the system is intended to be scalable and reliable. it should be capable of supporting multiple sensors while maintaining consistent performance and accuracy. as the number of connected devices increases, the gateway should continue to operate effectively without significant degradation in performance.

II. EXISTING METHODOLOGY

Recent growth in the Internet of Things (IoT) has led to a sharp rise in connected devices that continuously generate data. Earlier systems largely depended on centralized cloud platforms for processing and storage, which, although convenient, often resulted in higher latency, increased bandwidth usage, and greater energy consumption.

To address these concerns, researchers have increasingly explored edge computing, where data is processed closer to its source. This shift reduces the need for constant cloud communication and allows quicker system response. In many cases, simple operations such as filtering or aggregating data at the edge have been shown to lower the volume of transmitted data without losing essential information.

Various data reduction methods have also been investigated, including threshold-based filtering, compression, and event-driven transmission. These approaches are particularly useful when sensor values remain stable, as they limit communication to only meaningful changes. Alongside this, adaptive communication strategies—such as combining periodic, event-based, and threshold-based methods—help maintain a balance between efficiency and data accuracy.

Another area of interest is system architecture, especially the use of dual-controller designs. In such setups, one controller is typically assigned to low-power data handling, while the other manages communication tasks. This division helps improve energy efficiency and optimizes resource usage.

Overall, existing studies emphasize the benefits of combining edge processing, efficient communication, and data reduction techniques. However, integrating all these features into a single, scalable system remains an open challenge. This work aims to address that need by developing an edge-intelligent IoT gateway that improves data efficiency without compromising system performance.

Recent work by Spicher et al. [1] examines the use of edge computing within 5G networks for real-time ECG signal analysis. Their approach shifts data processing closer to the source, which helps in reducing latency and enables quicker detection of cardiac abnormalities. The study also points out that limiting the transmission of raw medical data can improve both bandwidth efficiency and data privacy.

In a related direction, Dallaf [2] discusses how integrating edge computing into IoT networks can address common performance limitations. The study highlights improvements in latency, communication efficiency, and system scalability, especially when handling large volumes of sensor data.

Similarly, Rahmati [3] introduces an energy-aware framework that combines federated learning with edge computing in 5G-enabled IoT systems. By allowing devices to collaborate without sharing raw data, the approach not only reduces communication overhead but also strengthens data security and privacy.

A broader perspective is provided in [4], where the authors present a detailed survey on reducing communication overhead across the IoT–edge–cloud continuum. The paper reviews various protocols and data reduction techniques, emphasizing that efficient data handling is essential for minimizing network congestion and energy consumption.

Security aspects in edge-based systems are further explored in [5], which focuses on different access control models used in IoT environments. The study highlights the importance of lightweight and adaptable security mechanisms to ensure safe data handling without adding excessive computational burden.

The integration of intelligent technologies is discussed in [6], where the convergence of embedded systems, edge computing, and machine learning is examined. The paper explains how combining these technologies can enhance system intelligence, enabling faster and more efficient decision-making at the edge.

In addition, [7] presents work from the IEEE World Forum on IoT that looks into optimizing communication algorithms and data encoding techniques. The study suggests that improved encoding and transmission strategies can significantly reduce data redundancy and enhance overall communication efficiency.

The role of artificial intelligence in edge-enabled IoT systems is highlighted by Wu et al. [8]. Their work demonstrates how AI techniques can support intelligent data processing, efficient resource allocation, and faster system response, particularly in real-time applications.

Finally, Balador et al. [9] explore the concept of distributed edge computing enhanced by artificial intelligence. Their study shows that distributing computational tasks across multiple edge nodes can improve system scalability and reduce dependence on centralized cloud infrastructure.

This work looks at how edge-based (MEC) platforms can better manage IoT data by considering the specific needs of applications. Instead of treating all data equally, the system prioritizes tasks based on their requirements, which helps reduce delays and improve resource usage. The study also shows that efficient data placement and sharing across edge nodes can significantly enhance overall system performance.[10]

The authors provide a detailed overview of how edge computing supports modern IoT systems. Their discussion highlights how moving computation closer to devices helps reduce latency and improves responsiveness. At the same time, they point out challenges such as limited resources and the need for efficient coordination between edge and cloud layers.[11]

This paper focuses on security aspects in edge-based IoT systems. It examines different access control approaches and explains how they can be applied at various stages of data handling. The study emphasizes that security mechanisms must be lightweight and adaptable to suit resource-constrained edge devices.[12]

This study explores how tasks can be intelligently distributed between cloud and edge layers. By coordinating both environments, the system is able to reduce unnecessary communication while maintaining computational efficiency. The results show that such orchestration improves system responsiveness and scalability.[13]

This work discusses how edge computing can contribute to energy-efficient and sustainable IoT systems. The authors highlight that reducing data transmission and processing it locally can lower energy consumption and support environmentally friendly system design.[14]

This recent work presents emerging applications of wireless edge computing across domains such as smart cities, healthcare, and industrial systems. It highlights how combining wireless technologies with edge intelligence enables faster data processing and supports large-scale IoT deployments.[15]

III. PROPOSED METHODOLOGY

The proposed system is designed with the objective of improving data handling efficiency in IoT environments by shifting key processing tasks closer to the data source. Instead of continuously forwarding raw sensor data to the cloud, the system performs preliminary analysis at the gateway level, allowing only relevant information to be transmitted. This approach not only reduces communication overhead but also enhances responsiveness.

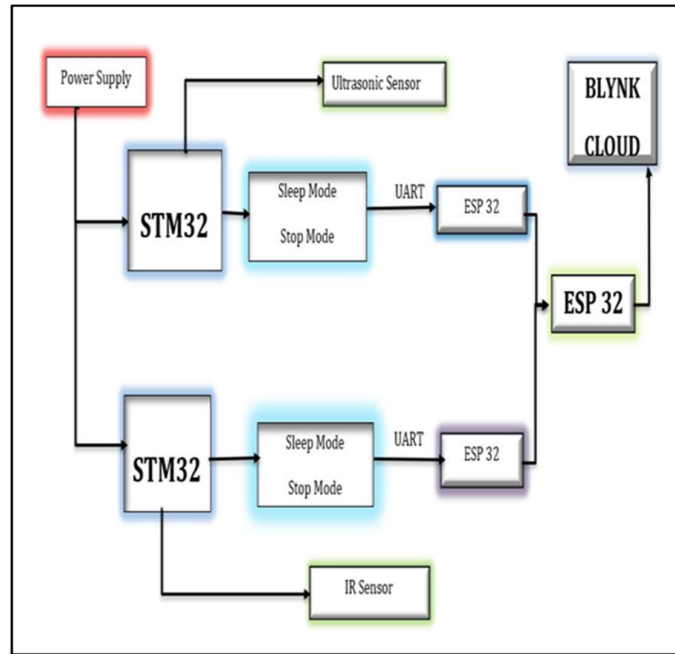


Fig III(A): Block Diagram of the Proposed Methodology

At the core of the system, a dual-controller architecture is adopted to distribute workload effectively. A low-power microcontroller is responsible for interfacing with sensors and carrying out basic data processing operations. This includes filtering noise, checking for predefined conditions, and identifying meaningful variations in the collected data. By handling these tasks locally, the system minimizes unnecessary data generation at the source itself.

The second controller is dedicated to communication functions. It remains inactive during normal conditions and is triggered only when the processed data meets certain criteria. This selective activation plays a key role in conserving energy, especially in applications where continuous transmission is not required. The communication module is used primarily for sending alerts, summarized data, or event-specific information to external platforms.

To further improve efficiency, the system incorporates multiple data transmission strategies. A time-based approach is used for periodic updates, ensuring that the system maintains a consistent flow of information when needed. In addition, threshold-based logic is applied so that data is transmitted only when sensor readings exceed predefined limits. Event-driven communication is also implemented to handle sudden or abnormal changes in the environment. By combining these methods, the system adapts dynamically to varying conditions without overloading the network.

Another important aspect of the methodology is energy management. Both controllers are configured to operate in low-power modes whenever possible. The system avoids continuous operation of high-energy components, activating them

only when required. This significantly reduces overall power consumption and makes the design suitable for battery-powered deployments.

From a structural perspective, the system is designed to be scalable. Additional sensors can be integrated without major modifications to the core architecture. The modular nature of the design allows flexibility in adapting the system for different applications, ranging from environmental monitoring to industrial automation.

Overall, the methodology focuses on balancing performance and efficiency by combining local intelligence, adaptive communication, and energy-aware operation. This integrated approach ensures that the system remains responsive and reliable while minimizing unnecessary resource usage.

A. Architecture of the System

The architecture of the proposed system is structured around a layered and modular design, where sensing, processing, and communication tasks are clearly separated to improve efficiency and flexibility. Instead of relying on a single processing unit, the system distributes responsibilities across multiple components to achieve better performance and reduced energy consumption.

At the input stage, a set of sensors is deployed to continuously monitor physical parameters such as distance, motion, or environmental conditions. These sensors generate raw data at regular intervals, which is then forwarded to the primary processing unit. Since raw sensor data often contains noise or redundant values, it is not transmitted directly but first undergoes local refinement.

The first level of processing is handled by a low-power microcontroller, which acts as the edge-processing unit. Its role is to interpret incoming sensor readings, remove unnecessary fluctuations, and evaluate whether the data represents a meaningful change.

Simple decision-making logic is applied at this stage, such as comparing values against predefined thresholds or identifying abnormal patterns. By performing these operations locally, the system avoids excessive data generation and reduces dependency on external systems.

Once the data has been processed, it is selectively passed to the communication layer. This layer is managed by a secondary controller equipped with wireless capabilities. Unlike conventional systems where communication modules remain active continuously, this controller operates in an on-demand manner. It is triggered only when the processed data satisfies specific conditions, such as exceeding a threshold or detecting an event. This selective activation significantly lowers power usage and minimizes unnecessary network traffic.

To support efficient operation, the architecture incorporates adaptive communication strategies. Depending on the situation, the system can switch between periodic updates, threshold-based reporting, and event-driven transmission. This flexibility allows the system to maintain a balance between data accuracy and communication efficiency.

Another important feature of the architecture is its scalability. The modular design allows additional sensors or functional units to be integrated without disrupting the existing workflow. Each component operates independently but contributes to the overall system objective, making the architecture suitable for a wide range of IoT applications.

In essence, the proposed architecture emphasizes local intelligence, selective communication, and energy-aware design. By combining these elements, the system achieves improved performance while maintaining efficient use of computational and network resource

B. Strategy for Operations

The overall operation of the system follows a structured yet adaptive sequence, where data is handled efficiently from the moment it is sensed to the point it is transmitted. The design focuses on minimizing unnecessary activity while still ensuring that important information is captured and communicated without delay.

At the beginning of each cycle, the sensors collect real-time data from the surrounding environment. These readings are continuously fed into the primary controller, which acts as the first level of decision-making. Instead of forwarding every data point, the controller evaluates the incoming values to determine their significance. Minor fluctuations or stable readings are treated as routine and are not immediately considered for transmission.

The system then applies a set of simple evaluation rules to identify meaningful changes. If the sensed data crosses predefined limits or shows a noticeable variation compared to previous readings, it is marked as relevant. This step helps in distinguishing between normal conditions and situations that require attention. By doing so, the system avoids generating redundant data while still maintaining awareness of critical events.

To maintain flexibility, the system does not rely on a single communication pattern. Instead, it operates using a combination of strategies. Periodic updates are used to provide occasional system status, ensuring that the remote end remains informed even during stable conditions. Alongside this, event-driven communication handles sudden or unexpected changes, while threshold-based transmission ensures that only values beyond defined limits are reported. The integration of these approaches allows the system to adapt to different operating conditions without compromising efficiency.

Another important aspect of the operation is energy management. Both controllers are designed to enter low-power modes when active processing or communication is not required. This controlled activation and deactivation cycle significantly reduces energy consumption, making the system suitable for long-term deployment.

Overall, the strategy of operation is centered on intelligent decision-making at the edge, selective communication, and efficient resource usage. By carefully controlling when and what data is transmitted, the system achieves a balance between responsiveness and conservation of power and bandwidth.

C. The Power Management Method

The energy optimization of this proposed system is achieved through several techniques:

Functional Decoupling: To stop high-power components from operating continuously, tasks are split among controllers.

Event-Based Execution: To prevent redundant processing, system actions are only initiated in certain circumstances.

Low-Power Modes: When not in use, the communication module uses deep sleep, and the sensing controller runs in energy-efficient states.

Controlled Duty Cycle: The wireless module's active time is decreased by limiting communication to brief intervals. Together, these strategies effectively uses of the power resources at hand.

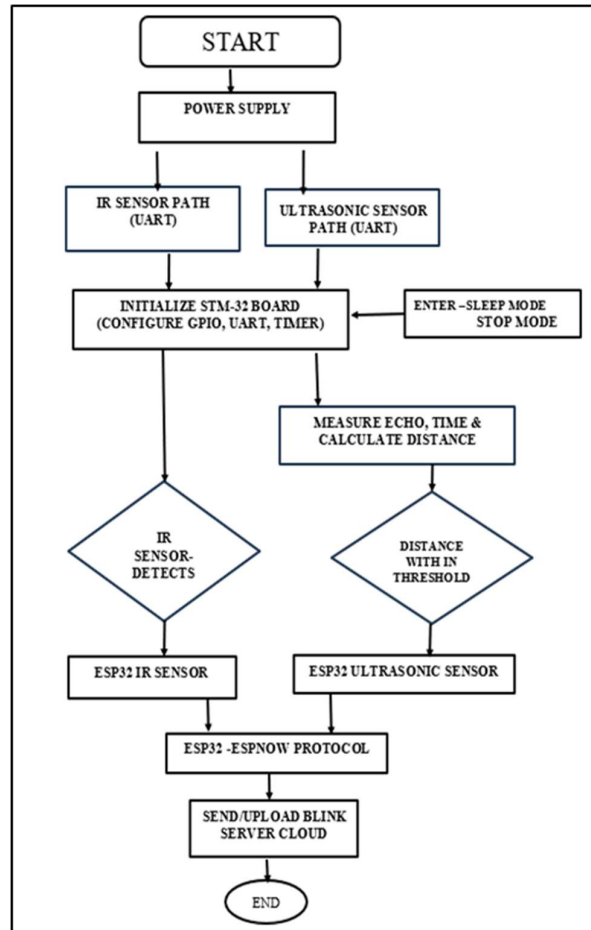


Fig III(B): Flow chart of the Proposed Methodology

D. Communication Between Controllers

For smooth operation, there must be dependable communication between the dual controllers. A serial communication protocol is used to exchange data, assuring precise transmission of control commands and sensor readings. The communication unit serves as a secondary node that performs transmission responsibilities on request, with the sensing controller serving as the major decision- maker. This hierarchical control structure decreases system overhead and facilitates coordination.

E. Considerations for Implementation

Embedded development platforms appropriate for each controller are used to implement the system. While the communication unit is set up for dependable network connectivity and data transmission, the sensing unit is programmed to facilitate effective data collecting and low-power operation. To guarantee reliable system behavior, particular emphasis is paid to synchronization, power switching, and transition delays between active and sleep modes. Future additions can easily incorporate more sensors or communication functions because the main design is kept modular.

IV. IMPLEMENTATION

The proposed system was implemented using a dual-controller setup to efficiently separate sensing, processing, and communication tasks. A low-power microcontroller was used to interface with the sensors and carry out initial data handling. This included reading sensor values, performing basic filtering, and applying simple decision rules to identify whether the data required further action.

The controller was programmed to operate in cycles, allowing it to periodically wake, process data, and return to a low-power state when idle. A secondary controller with built-in wireless capability was configured to manage data transmission. Instead of remaining active continuously, it was conditions, such as threshold violations or event detection. Communication between the two controllers was established through a lightweight serial interface, ensuring smooth data exchange without introducing significant overhead.

On the software side, the logic was designed to support multiple transmission modes, including periodic updates, threshold-based reporting, and event-driven alerts. These modes were integrated in a way that allowed the system to adapt its behavior depending on the nature of the sensed data. Care was taken to keep the algorithms simple and efficient so that they could run reliably on resource-constrained hardware. To improve energy efficiency, both controllers were configured to utilize power-saving features wherever possible. Unnecessary peripherals were disabled when not in use, and communication modules were activated only for short durations during data transfer. The implementation was tested under different operating conditions to verify that the system could maintain stable performance while reducing unnecessary data transmission.

Overall, the implementation reflects a practical realization of edge-based data handling, where local processing and selective communication work together to improve efficiency without increasing system complexity.

The suggested solution was put into practice as an embedded platform intended to verify how well the dual-controller architecture decreases the power consumption. Within a coordinated framework, the implementation incorporates cloud interaction, regulated communication, sensing, and local processing.

A. Hardware Configuration

The hardware setup of the proposed system is arranged to support efficient sensing, local processing, and controlled communication. The design makes use of two microcontrollers with distinct roles, allowing tasks to be distributed in a balanced manner. A low-power controller is used as the primary interface for

sensor data acquisition, while a wireless-enabled controller is responsible for handling external communication.

The sensor unit is connected directly to the primary controller through appropriate input pins, enabling continuous monitoring of physical parameters. Depending on the type of sensor, both analog and digital interfaces are utilized. Signal conditioning is considered wherever required to ensure stable and reliable readings. The controller operates at a suitable clock frequency to maintain a balance between processing speed and energy consumption.

Communication between the two controllers is established through a serial interface, which provides a simple and reliable method for data exchange. This connection allows the processed data to be forwarded only when necessary, avoiding continuous interaction between modules. The secondary controller, equipped with built-in Wi-Fi capability, is configured to transmit data to external platforms when triggered.

Power management is an important aspect of the hardware configuration. The system is designed to operate using a regulated power supply, ensuring consistent voltage levels for all components. Both controllers are configured to make use of low-power operating modes, reducing energy usage during idle periods. Peripheral components are also selectively enabled to further improve efficiency.

B. Design of Software

The software is designed to run on each controller as two separate yet synchronized programs. Data collection, condition assessment, and power management are all carried out in a continuous loop by the main controller. By using low-power states during idle times and only waking up for sampling or decision-making tasks, it functions in an energy-efficient manner. By default, the communication module is set up to be inactive. Only when the principal controller initiates it is it initialized. When it is activated, it sets up a network connection, gets the relevant data, and sends it to a cloud platform or distant server. It reverts to a low-power mode once the transmission is completed. To assure dependable data transfer, a straightforward communication protocol is put in place between the controllers. To avoid the loss of data during transmission, this involves fundamental synchronization and organized data packets.

C. Flow of Operations

The system operates according to a predefined set of steps. Initially, the main controller configures each peripheral and begins monitoring sensor inputs. The collected data is compared to preset standards, such as threshold constraints or time-based triggers. If no significant problem is discovered, the system stays in low-energy monitoring mode. In reaction to a trigger situation, the primary controller transmits a control signal to the communication module. The communication unit then switches on, establishes a connection, and transmits the data required. After the completion of the transmission, the communication module is either deactivated or put into deep sleep. The system then goes back to its monitoring phase. Repetition of this cycle ensures that communication occurs only when necessary and the power is used as per requirement.

TABLE I: Comparison on Single and Dual controller system

Parameter	Traditional Single-Controller System	Proposed Dual-Controller System
Power Consumption	Higher due to continuous operation	Reduced using task separation and selective activation
Task Management	All tasks handled by one controller	Task distribution between two controllers
System Efficiency	Moderate efficiency	Improved efficiency due to parallel processing
Communication Overhead	Frequent and sometimes unnecessary data transmission	Minimized communication with selective data transfer

D. Implementation of Power Optimization

During deployment, a number of doable strategies were used to lower energy consumption: During routine monitoring, the communication module is totally idle. Transmission of data is restricted to brief, event-driven intervals. When feasible, the principal controller runs in low-power modes. During idle times, peripherals that are not in use are turned off. Together, these actions lower the system's total energy consumption.

E. Results of Implementation

The implemented system was evaluated under different operating conditions to observe its behavior in terms of data handling, responsiveness, and energy usage. The results indicate that performing initial data processing at the gateway level significantly reduces the amount of information that needs to be transmitted. Instead of continuously sending raw sensor readings, the system forwards only filtered and relevant data, which helps in lowering communication overhead.

V. RESULT

The synthesis stage involves converting the developed design into a form that can be executed on the target hardware. In this project, both firmware development and system integration were carried out using appropriate development tools. The results of synthesis indicate that the proposed dual-controller architecture operates efficiently with optimized resource usage and stable performance.

Tools Used

The implementation and synthesis of the system were performed using the following tools:

Development Tools for Microcontrollers

- STM32 Development Environment used for writing the embedded C code, compiling, and debugging firmware for the STM32. It supports peripheral configuration, code generation, and real-time debugging.

- Arduino IDE / ESP-IDF used for programming the ESP32. It enables easy integration of Wi-Fi features and cloud communication.

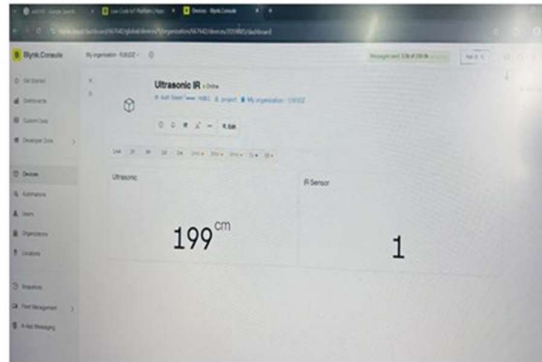


Fig V(A) Result-1

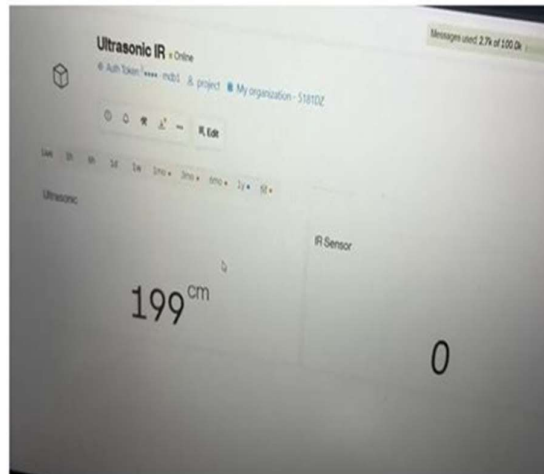


Fig V(B) Result-1

Programming and Debugging Tools

- Embedded C language is used for firmware development in project.
- Serial monitor for debugging UART communication its done serial.
- Flashing tools for uploading code into microcontrollers.

Cloud Platform

- Blynk Cloud was used for real-time data visualization and monitoring while the sensor data is continuous but we will get data in event based.
- It provides a user interface for displaying sensor data and alerts in the Blynk online server.

Synthesis Outcome

- After compiling and uploading the code:
- Both controllers operated as expected
- Sensor data was successfully processed and transmitted

- Communication between modules was stable the system was tested multiple times to ensure consistency and reliability.

Comparative Discussion

Compared to traditional single-controller systems:

- The proposed system reduces power consumption
- Improves efficiency by separating tasks
- Minimizes unnecessary communication

This shows that the dual-controller approach is more effective for IoT applications.

Key Advantages of Implementation

1. Edge intelligence, by processing and filtering data locally at ESP32 nodes
2. Reduced network congestion, by avoiding continuous raw data transmission
3. Low latency communication, using ESP-NOW protocol.
4. High reliability and modularity, due to distributed architecture
5. Scalability, allowing additional sensor nodes to be integrated easily

Limitations Observed

Although the system performed well, some limitations were noted:

- Slight delay during Wi-Fi initialization
- Dependence on network availability
- Increased hardware complexity due to dual controllers

These limitations can be improved in future enhancements. The results confirm that the proposed system meets its objectives of efficient sensing, reliable communication, and reduced power consumption. The combination of STM32 and ESP32 provides a balanced approach for IoT applications, making the system suitable for real-time monitoring with optimized energy usage.

VI. CONCLUSIONS AND FUTURE SCOPE

The project introduces an optimized method for designing IoT systems by dividing sensing and communication tasks between two controllers. In this project, the STM32 microcontroller is responsible for continuous data monitoring from sensors, while the ESP32 module maintains wireless data transmission through unique MAC address. This separation of responsibilities improves throughout system performance and helps reduce unnecessary power consumption in system.

The implemented system collects data from sensors and transmits it accurately to a cloud-based Blynk platform. By using an event-based communication programming, the wireless module is activated only when required, so that the system lowers energy usage. In addition, the use of sleep mode which already inbuilt and low-power operating modes further improves the system's efficiency.

While we are executing the code, the results indicate that the system operates reliably, responds fast to sensor inputs, and maintains stable communication without evident delays. The UART-based serial communication between the controllers was observed to be consistent and dependable. Moreover, integration with the cloud platform allows real-time monitoring and provides easy access for users.

In summary, the proposed system successfully achieves its goals of improving energy efficiency, ensuring reliable data transfer, reduces power consumption and maintaining effective system performance. This design is well-suited for IoT applications that needs continuous monitoring while operating under power constraints.

The system can be further improved by incorporating modern technologies to enhance its intelligence, scalability, and efficiency. The main key advancement is the use of machine learning techniques at the edge computing system. By analyzing patterns in sensor data, the system can make better decisions, identify abnormal conditions more accurately, and minimize wrong alerts. This approach allows the system to adapt more effectively to active environments.

Another possible improvement in the inclusion of additional sensing units. Sensors such as gas detectors, temperature and humidity modules, soil sensor, and vibration sensors can be integrated to extend the system's functionality. This usage of multiple sensors will improve the performance in the diverse applications including environmental observation, industrial monitoring, and smart farming.

The communication capability can also be improved by adopting long-range, low-power technologies like LoRa or NB-IoT. These communication methods are well-suited for applications that require wide coverage and can operate efficiently even in areas where conventional Wi-Fi is not available.

Improving energy efficiency is another important area for future development. Techniques such as energy harvesting, particularly through solar power, along with better power

management strategies like optimized sleep cycles, can greatly extend battery life. This would support continuous operation of the system over long durations without frequent maintenance.

During testing, it was observed that the use of threshold-based and event-driven transmission effectively limits unnecessary data flow, especially when the monitored parameters remain stable over time. At the same time, the system was able to respond promptly when abnormal conditions were introduced, ensuring that important events were not missed. This demonstrates that selective communication does not compromise system reliability.

The dual-controller arrangement also contributed to improved efficiency. By separating sensing and communication tasks, each controller was able to operate more effectively within its role. The communication module remained inactive for a significant portion of time and was triggered only when required, which resulted in noticeable energy savings compared to continuous transmission approaches.

In terms of overall performance, the system maintained consistent operation without delays or data loss during repeated trials. The integration of local processing and adaptive communication strategies provided a balanced outcome, where both responsiveness and resource optimization were achieved. These observations confirm that the proposed approach is suitable for IoT applications that require efficient data management and long-term operation.

REFERENCES

- [1]. M. Spicher, A. Klingenberg, V. Purrucker, and T. M. Deserno, "Edge computing in 5G cellular networks for real-time analysis of ECG signals," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 8, pp. 4123-4132, Aug. 2022.
- [2]. A. B. Dallaf, "Edge computing in IoT networks: Enhancing efficiency, reducing latency, and improving scalability," *International Journal of Advanced Network Monitoring and Controls*, vol. 10, no. 1, 2025.

- [3]. M.Rahmati, "Energy-aware federated learning for secure edge computing in 5G-enabled IoT networks," *Journal of Electrical Systems and Information Technology*, vol. 12, no. 1, 2025.
- [4]. "Reducing communication overhead in the IoT-edge-cloud continuum: A survey on protocols and data reduction strategies," *Internet of Things Journal*, vol. 31, 2025.
- [5]. "Edge computing for IoT: Novel insights from access control models," *Computer Networks*, vol. 270, 2025.
- [6]. "Internet of Intelligent Things: Convergence of embedded systems, edge computing and machine learning," *Internet of Things Journal*, 2024.
- [7]. IEEE World Forum on IoT, "Optimization of IoT communication algorithms and data encoding perception technology," *IEEE WF-IoT Conference*, 2025.
- [8]. H. Wu et al., "Artificial intelligence empowered communication and computation for edge-enabled IoT systems," *IEEE Internet of Things Magazine*, 2025.
- [9]. A. Balador, S. Sinaei, M. Pettersson, and I. Kaya, "Artificial intelligence enabled distributed edge computing for IoT," *IEEE IoT Newsletter*, 2022.
- [10]. S. Bolettieri, R. Bruno, and E. Mingozzi, "Application-aware resource allocation and data management for MEC-assisted IoT services," *IEEE Internet of Things Journal*, 2025.
- [11]. B. T. Hasan and A. K. Idrees, "Edge computing for IoT: Architecture, opportunities, and challenges," *IEEE Access*, 2024.
- [12]. T. Xue, Y. Zhang, Y. Wang, W. Wang, S. Li, and H. Zhang, "Edge computing for IoT: A comparative analysis of access control models," *IEEE Communications Surveys & Tutorials*, 2024.
- [13]. Y. Wu et al., "Cloud-edge orchestration for efficient IoT data processing," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12792–12805, 2021.
- [14]. J. Wu, X. Chen, and B. Gupta, "Internet of Things and edge computing for green and sustainable systems," *IEEE Internet of Things Magazine*, 2025.
- [15]. D. Mandal, M. N. Eshwarappa, and J. Frnda, "Wireless edge computing in Internet of Everything: Applications and challenges," *IEEE Conference Proceedings / IoT Systems*, 2026.