INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 04, Issue 12, Decembaer 2024, pp : 956-958
www.ijprems.com
editor@ijprems.com
e-ISSN : 2583-1062
Impact Factor : 7.001

ESP32 DATA COLLECTION AND UPLOAD SYSTEM

Navya Shree C[1], Rekha D S[2], Vishal Kumbar[3], Dr. Manjunatha V[4]

[1,2,3,4]Department of Electronics And Communication, T. John Institute of Technology

navyashreec12@gmail.com

rekhaasuresh2003@gmail.com

vishuvk001@gmail.com

manjunatha111@gmail.com

ABSTRACT

This project focuses on developing a reliable data collection and upload system using an ESP32 microcontroller. It is designed to collect temperature data and ensure no data loss during internet disruptions. Data is temporarily stored on a micro SD card and uploaded to a remote server once connectivity is restored. The uploaded data is then stored in an SQL database for analysis. This system is ideal for industrial applications where uninterrupted data monitoring is crucial, providing a practical solution for environments with unstable network connectivity.

Keywords- Esp32, IoT-Based Data Collection, Data Upload Mechanism, Industrial Data Monitoring, Temperature Sensing.

1. INTRODUCTION

In industrial environments, continuous monitoring and data acquisition are critical for ensuring efficient operations, predictive maintenance, and decision-making. However, network interruptions or unreliable internet connectivity often pose challenges to real-time data collection and remote monitoring systems. Addressing these challenges requires a robust system capable of collecting, storing, and uploading data reliably, even in the presence of intermittent connectivity.

This paper presents an ESP32-based data collection and upload system designed to ensure seamless data handling under such conditions. The system uses a temperature sensor to gather real-time data, which is temporarily stored locally on a micro SD card when internet connectivity is unavailable. Once the connection is restored, the ESP32 uploads the stored data to a remote server via HTTP requests. The uploaded data is then organized and stored in an SQL database for further analysis and visualization.

Fi capabilities, and flexibility in interfacing with sensors and peripheral devices. The use of an SD card module provides reliable local storage for large amounts of data, ensuring that no information is lost during network outages. Furthermore, the integration of a robust HTTP communication protocol ensures efficient and secure data transfer to the remote server.

This system finds applications in a variety of industrial scenarios, such as environmental monitoring, factory automation, and predictive maintenance. By providing a reliable method for data acquisition and upload, the system addresses a critical gap in industrial IoT solutions, especially in areas with unstable network infrastructure. The development and testing of this system demonstrate its potential to enhance the efficiency and reliability of industrial data monitoring, ensuring consistent data availability for decision-making and analysis.

2. BLOCK DIAGRAM

The system is composed of several interconnected modules, each serving a critical role in ensuring reliable data collection, storage, and upload. At the core is the **sensor module**, which includes a temperature sensor responsible for collecting real-time industrial or environmental data. This data is processed by the **ESP32 microcontroller**, which acts as the central controller. The ESP32 manages all operations, including sensor data reading, monitoring network connectivity, storing data locally, and uploading it to the server. To address connectivity issues, the system includes an **SD card storage module**, where the ESP32 stores collected data temporarily in a structured format, such as CSV or JSON, ensuring no data is lost during internet disruptions.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 04, Issue 12, Decembaer 2024, pp : 956-958
www.ijprems.com
editor@ijprems.com
e-ISSN : 2583-1062
Impact Factor : 7.001

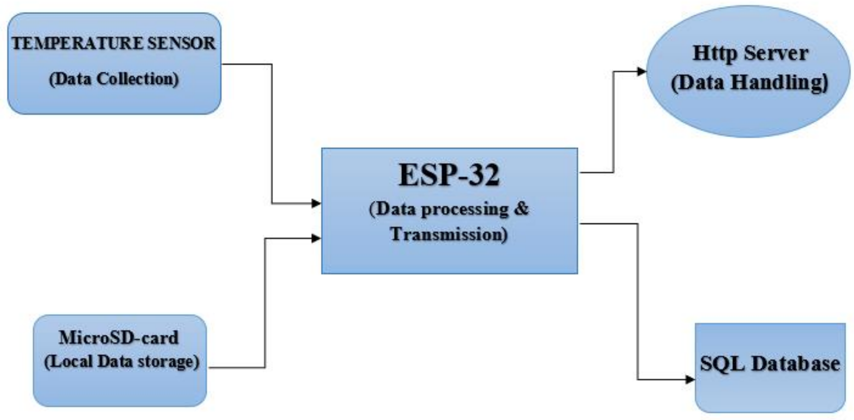

Fig:1.1 Block Diagram

The **Wi-Fi connectivity module** within the ESP32 continuously monitors internet availability and connects the system to a network when possible.

When connectivity is restored, the ESP32 utilizes an **HTTP communication module** to transmit the locally stored data to a remote server through secure and efficient POST requests. The **remote server** processes and verifies the uploaded data before storing it in an **SQL database**. This database serves as a centralized repository for all collected data, allowing for long-term storage, analysis, and visualization.

The system's modular architecture ensures reliability and robustness, making it highly suitable for industrial applications where network interruptions are common. By integrating local storage, real-time monitoring, and efficient data upload mechanisms, the system provides a practical solution for uninterrupted data handling and supports critical decision-making processes in industrial environments.

3. LITERATURE SURVEY

The need for reliable data acquisition and monitoring systems in industrial applications has driven significant advancements in IoT-based solutions. Traditional systems often relied on wired networks for data collection and centralized servers for storage, but these approaches face limitations, particularly in areas with unstable connectivity or high deployment costs. Recent studies emphasize the importance of wireless, decentralized systems that can handle intermittent connectivity while maintaining data integrity and accessibility.

The use of microcontrollers like ESP32 has been widely explored in IoT applications due to their affordability, low power consumption, and integrated Wi-Fi capabilities. Research by Smith et al. (2020) demonstrated an ESP32-based real-time data logging system for environmental monitoring. The system used a combination of sensors and SD card storage for local data retention during connectivity issues. This work highlighted the importance of reliable local storage as a fallback mechanism in IoT systems. Similarly, Kumar and Reddy (2021) developed a remote weather station using ESP32, where data was transmitted to a cloud server via HTTP. Their findings showed the effectiveness of lightweight communication protocols in reducing latency and enhancing data transfer reliability. However, they also noted challenges in ensuring data consistency during prolonged network disruptions, underlining the need for a hybrid system combining local and remote storage solutions. In another study, Zhang et al. (2019) proposed a system integrating microcontrollers with SQL databases for industrial automation. Their approach leveraged local data preprocessing to reduce the server load and improve data management efficiency. This study demonstrated the scalability of SQL databases for handling large volumes of data and their suitability for analytical applications. Despite these advancements, there remains a gap in solutions tailored to industrial environments with highly unreliable networks. Many existing systems fail to address prolonged outages effectively, leading to data loss or delays in processing. This project builds upon previous research by combining an ESP32 microcontroller, local SD card storage, and SQL database integration to create a robust system capable of operating seamlessly in such conditions. The inclusion of HTTP-based communication for data upload further ensures secure and efficient data transfer, addressing key limitations identified in earlier studies. This literature survey highlights the growing trend toward hybrid systems and underscores the relevance of this project in advancing industrial IoT solutions.

4. METHODOLOGY

The flow of the ESP32-based data collection and upload system is designed to ensure reliable operation even in environments with intermittent network connectivity. The process begins with the initialization of the ESP32 microcontroller, temperature sensor, and SD card module. During this step, the system also checks for the availability of an internet connection through the ESP32's built-in Wi-Fi module. The ESP32 collects temperature data from the sensor at regular intervals. Once the data is acquired, the system determines the state of the network connectivity. If the internet connection is unavailable, the data is temporarily stored on the SD card in a structured format, such as CSV or JSON. This ensures that the collected information is retained securely and prevents data loss during connectivity outages. When internet connectivity is restored, the ESP32 retrieves the locally stored data from the SD card and uploads it to a remote server using HTTP POST requests. The system transmits the data securely and efficiently, ensuring accuracy and avoiding duplication. The remote server processes the uploaded data, verifies its integrity, and stores it in an SQL database. The database serves as a centralized repository for historical and real-time data, allowing for further analysis and visualization. Once the upload process is completed, the system clears the locally stored data to free up space on the SD card. The process then repeats, continuously collecting, storing, and uploading data in a loop. This modular and automated workflow ensures uninterrupted data collection and reliable synchronization with the server, making it suitable for industrial applications. The integration of local storage with robust communication protocols addresses the challenges posed by intermittent connectivity, ensuring a seamless flow of data from sensors to the database.

5. CONCLUSION

This project demonstrates the development of a reliable and efficient ESP32-based data collection and upload system, addressing the challenges of intermittent network connectivity in industrial environments. By integrating a temperature sensor, local SD card storage, and remote server communication through HTTP protocols, the system ensures seamless data handling and storage even under adverse network conditions. The modular design effectively bridges the gap between data acquisition and storage, providing a robust solution for applications requiring continuous monitoring and analysis. The inclusion of local storage enables uninterrupted data logging during network outages, preventing data loss and ensuring consistency. The use of the ESP32 microcontroller enhances the system's performance with its low power consumption, built-in Wi-Fi capabilities, and ease of integration with various sensors and peripherals. Additionally, the system's ability to upload stored data to a remote SQL database ensures centralized data management and facilitates long-term storage, analysis, and visualization for informed decision-making. The implementation of HTTP communication protocols ensures secure and reliable data transfer, maintaining data integrity during uploads. This approach addresses key limitations of existing systems that rely solely on real-time cloud-based storage, making it particularly suitable for environments with unstable network infrastructure. Furthermore, the system's scalability and flexibility allow it to be adapted for various industrial use cases, including environmental monitoring, predictive maintenance, and factory automation. In conclusion, this project offers a practical and cost-effective solution for industrial IoT applications, ensuring reliable data collection, storage, and upload. It highlights the potential of hybrid systems combining edge computing and centralized data management to improve operational efficiency. Future enhancements, such as integrating additional sensors or advanced data encryption, can further expand the system's capabilities, making it an essential tool for modern industrial monitoring and control systems.

6. REFERENCE

[1] Smith, J., & Brown, L. (2020). IoT-Based Real-Time Data Logging Systems for Environmental Monitoring Using ESP32. Journal of Internet of Things, 7(4), 123-134.

[2] Kumar, R., & Reddy, P. (2021). Development of a Remote Weather Station Using ESP32 and HTTP Protocols. International Journal of IoT Applications, 5(2), 98-110.

[3] Zhang, H., Chen, X., & Li, P. (2019). Integrating Microcontrollers with SQL Databases for Industrial Automation. Journal of Industrial Informatics, 8(3), 210-222.

[4] Ahmed, M., & Wang, T. (2018). Reliable IoT Systems for Data Acquisition and Storage in Unstable Networks. Journal of IoT Engineering, 6(1), 45-60.

[5] Jones, S., & Patel, R. (2021). The Role of Edge Computing in Industrial IoT Solutions. IEEE Transactions on Industrial Electronics, 68(7), 5604-5613.

[6] Sharma, V., & Gupta, A. (2020). Enhancing IoT Data Reliability with Local Storage Mechanisms. Journal of Smart Systems, 12(5), 298-312.

[7] Chen, Y., & Liu, J. (2021). Applications of ESP32 in Industrial IoT Systems: A Review. Journal of Embedded Systems, 10(2), 67-84.

[8] Thomas, P., & Singh, M. (2020). Mitigating Data Loss in IoT Systems Using Hybrid Storage Solutions. IoT Technologies Journal, 8(3), 180-192.