MATLAB & Arduino-Based IoT Light Monitoring System

Het Patel*, Manav Patel†
Electronics and Communication Engineering, Nirma University
Ahmedabad, India
*22bec050@nirmauni.ac.in, †22bec073@nirmauni.ac.in,

***Abstract*—In modern smart infrastructure, real-time monitoring of environmental parameters plays a crucial role in ensuring efficiency, automation, and energy conservation. This project presents an IoT-based light monitoring system that utilizes an embedded system combining Arduino, MATLAB, and ThingSpeak cloud services. A light-dependent resistor (LDR) sensor is used to measure ambient light intensity, and the collected data is transmitted to ThingSpeak for cloud-based visualization and analysis. MATLAB processes the sensor data, enabling further computations and insights. Additionally, an ASP.NET MVC web application is developed to display real-time sensor readings in both graphical and tabular formats, ensuring user-friendly accessibility. The system is designed to update automatically, allowing continuous monitoring and efficient decision-making. The proposed solution is cost-effective, scalable, and can be extended for smart lighting automation in homes, industries, and public spaces to enhance energy efficiency.**

***Index Terms*—Embedded Systems, IoT, Light Monitoring, Arduino, MATLAB, ThingSpeak, Real-Time Data, ASP.NET MVC, Smart Automation.**

I. Introduction

In the modern era of technology, the Internet of Things (IoT) has revolutionized how data is collected, processed, and utilized for automation. One of the most impactful applications of IoT is in smart environments, where real-time data monitoring plays a crucial role in enhancing efficiency, automation, and decision-making. Among these, light monitoring is a significant area, as it directly impacts energy consumption, visibility, and automation in industries, homes, and public spaces. Traditional lighting systems lack adaptability and rely on manual operation, leading to unnecessary power usage and inefficiency. This project presents a real-time IoT-based light monitoring system that provides automated and efficient data collection, visualization, and analysis.

The system is built using an embedded platform consisting of an Arduino microcontroller and an LDR (Light Dependent Resistor) sensor, which continuously measures the ambient light intensity. The collected data is then transmitted to ThingSpeak, a cloud-based IoT analytics platform, where it is stored and made accessible for real-time visualization. Furthermore, MATLAB is utilized for advanced data analysis, offering deeper insights into light variation trends. Additionally, an ASP.NET MVC-based web application is developed to provide a user-friendly interface where users can monitor light intensity in real time through a dynamic waveform representation and a structured table displaying the latest sensor readings.

This project is designed to offer a scalable and cost-effective solution for smart lighting applications. The integration of cloud computing with embedded systems ensures seamless data management and remote accessibility. Users can view real-time updates without needing direct access to the physical hardware, making it highly beneficial for smart home automation, industrial lighting systems, and public infrastructure management. The system not only monitors light intensity but also lays the foundation for future improvements, such as automatic brightness adjustments, machine learning-based predictive analytics, and mobile application integration.

The primary objective of this project is to demonstrate how embedded systems and IoT can be effectively combined to enhance automation and efficiency in real-world applications. By implementing this real-time monitoring solution, energy consumption can be optimized, manual intervention can be reduced, and intelligent decision-making can be enabled. This project serves as a fundamental step toward the development of smarter and more efficient lighting systems, contributing to sustainability and technological advancement.

II. System Design and Components

The system is designed to facilitate real-time data collection, transmission, and visualization using an embedded sensor module, an IoT cloud platform, and a web application. This section describes the overall system design, the hardware components used, the software framework, and the communication mechanism. The seamless integration of these elements ensures accurate and efficient monitoring of sensor data.

A. System Architecture

The system follows a structured architecture to ensure smooth data acquisition and real-time visualization. It begins with a sensor that captures environmental data, which is then processed and transmitted to an IoT cloud platform. The cloud platform stores the data, allowing an external application to retrieve and display it dynamically. The web application fetches the latest sensor values and updates a graphical representation as well as a tabular format to provide a clear and concise view of real-time trends. The system operates in a continuous loop, ensuring that the displayed data remains up-to-date without requiring manual intervention.

B. Hardware Components

The hardware used in the system includes a microcontroller, a sensor module, and a communication interface. The microcontroller acts as the central processing unit that collects sensor readings and transmits them to the cloud. The sensor module varies based on the application, ranging from temperature and humidity sensors to gas or pressure sensors. A Wi-Fi-enabled module is used for wireless data transmission, eliminating the need for wired connections.

The following table provides a summary of the hardware components used:

TABLE I: Hardware and Software Components Used

Component	Description
Microcontroller	Arduino Uno
Sensor Module	Photo Sensor (LDR)
Resistors	330 ohm Resistor, 10 k Resistor
Display	Assorted LEDs
Prototyping Board	Breadboard
Wiring	Jumper Wires
Connectivity	Mini-B USB Cable
Software	MATLAB, ThingSpeak, ASP.NET

The hardware components are selected to optimize efficiency, minimize power consumption, and ensure reliable communication with the cloud platform.

C. Software Components

The software framework plays a crucial role in processing, transmitting, and visualizing data. The system is built using multiple software technologies to ensure smooth integration of different modules. The primary software components include the IoT cloud service, a backend for data retrieval, and a frontend for user interaction.

ThingSpeak is used as the cloud platform for storing and managing sensor data. The ASP.NET MVC framework is employed to develop the web application that fetches data from ThingSpeak and displays it in real time. JavaScript and Chart.js are used for rendering interactive visualizations. The backend is written in C#, allowing efficient handling of API requests and data processing.

D. Data Transmission and Processing

Data transmission follows a structured process where the sensor module reads the environment, processes the data, and sends it to the cloud through the microcontroller's Wi-Fi interface. The microcontroller communicates with ThingSpeak using HTTP requests, ensuring that the data is stored and accessible for later retrieval.

Once stored in the cloud, the web application retrieves the latest readings through API requests and updates the displayed values dynamically. The system is designed to handle real-time data efficiently, minimizing latency and ensuring seamless operation.

E. Real-Time Data Visualization

The visualization component enhances the usability of the system by allowing users to monitor sensor readings in a structured format. The web application provides a dual-mode representation: a graphical chart and a tabular view of the latest ten data entries. The graph is continuously updated, offering a clear insight into trends and variations, while the table provides exact numerical values for reference. This dual approach improves accessibility and ensures a better understanding of the collected data.

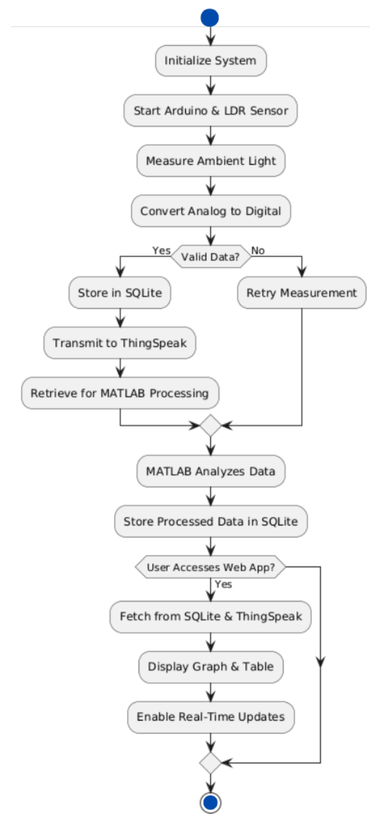


Fig. 1: Flowchart of Real-Time Data Visualization

III. Implementation

This section explains the step-by-step implementation of the IoT-based Light Monitoring System. It covers the overall system workflow, circuit design, data transmission, web application, and real-time visualization. Each component is integrated to ensure seamless data collection, processing, and display.

A. System Workflow

The system operates by continuously monitoring ambient light intensity using a photo sensor connected to an Arduino Uno. The Arduino reads the sensor values and processes them to determine light intensity levels. Based on predefined conditions, it controls LED indicators to signify different light levels. The sensor readings are sent to a MATLAB script, which further processes the data and transmits it to ThingSpeak. The collected data is stored on ThingSpeak's cloud and can be accessed remotely. An ASP.NET MVC-based web application retrieves the data from ThingSpeak and presents it in a user-friendly manner, including tabular and graphical representations. This allows real-time monitoring and analysis of environmental lighting conditions.

B. Circuit Design and Connections

The hardware setup consists of an Arduino Uno, a photo sensor, resistors, LEDs, a breadboard, and jumper wires. The photo sensor is connected to an analog input pin of the Arduino, allowing it to measure the intensity of surrounding light. A 10k pull-down resistor ensures stable sensor readings by minimizing fluctuations. LEDs are used as visual indicators and are connected via 330 resistors to limit current flow and prevent damage. The entire circuit is built on a breadboard, providing a modular and flexible setup. The Arduino is powered using a Mini-B USB cable, which also facilitates serial communication with MATLAB. The circuit is designed for efficiency and accuracy, ensuring reliable light monitoring and data transmission.

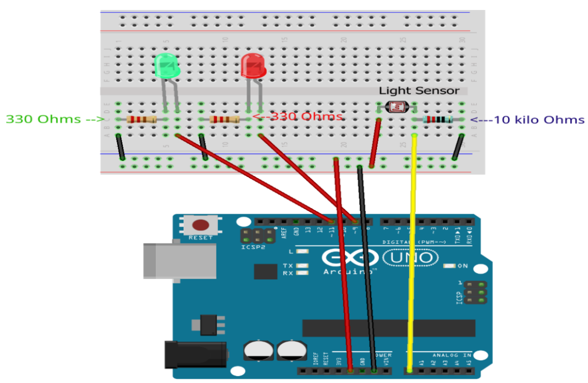


Fig. 2: Circuit Diagram of the System

C. Data Transmission to ThingSpeak

As shown in Figure 2, the system consists of an Arduino and an LDR sensor. The system employs MATLAB for data acquisition and transmission. The Arduino continuously reads the photo sensor values and sends them to MATLAB via a serial connection. A MATLAB script processes the received values and formats them before sending them to ThingSpeak using an API call. The API request includes the sensor reading and a timestamp, ensuring proper data logging. ThingSpeak, a cloud-based IoT analytics platform, stores the data and allows users to visualize trends in light intensity over time. This mechanism provides a reliable way to remotely access sensor data from any location. The real-time updating feature ensures that the latest sensor values are always available for monitoring and analysis.

D. Web Application Using ASP.NET

To enhance accessibility and visualization, an ASP.NET MVC-based web application is developed to fetch and display sensor data. The application communicates with ThingSpeak via RESTful API calls, retrieving the latest 10 sensor values along with their timestamps. The data is displayed in a neatly formatted table for easy readability. Additionally, a line graph is generated using the Chart.js library, which provides a dynamic representation of light intensity variations. The web application is designed with a user-friendly interface, ensuring that users can efficiently analyze and interpret the collected data. Regular data updates enable real-time tracking, making it a valuable tool for monitoring environmental lighting conditions.

E. Real-Time Data Processing and Visualization

The integration of MATLAB, ThingSpeak, and ASP.NET ensures smooth real-time data processing and visualization. The system continuously collects, processes, and updates data at regular intervals, maintaining an accurate record of light intensity changes. The visualization on ThingSpeak allows users to observe long-term trends, while the ASP.NET web application provides immediate insights through tabular and graphical representations. By implementing an automated data pipeline, the system eliminates the need for manual data logging and reduces the chances of errors. This real-time approach enhances decision-making capabilities, allowing users to react to changes in lighting conditions promptly. The combination of hardware and software components ensures efficient, precise, and accessible light monitoring.

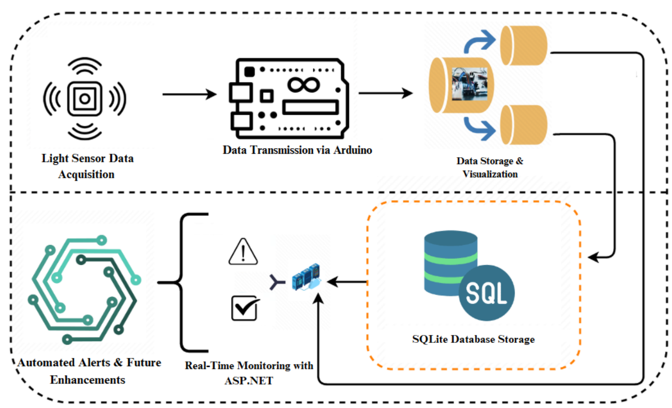


Fig. 3: IoT Light Monitoring System Architecture

IV. Results and Discussion

This section presents the outcomes of the implemented system, highlighting the performance, data representation, and overall effectiveness of the light monitoring system. The system's ability to capture, process, and display real-time light intensity variations is evaluated using ThingSpeak and the ASP.NET web application.

A. System Output and Observations

The developed system successfully captures and processes light intensity data using a photo sensor. The Arduino Uno reads the sensor values and transmits them to ThingSpeak, where the data is stored and visualized. The system operates effectively under varying lighting conditions, demonstrating accurate real-time monitoring. The LED indicators and resistance values ensure proper circuit functionality, making the system stable and reliable for continuous operation.

B. ThingSpeak Data Visualization

ThingSpeak provides a graphical representation of the recorded sensor values over time. The real-time graph updates dynamically as new data is uploaded from the Arduino. The visualization helps in analyzing trends in light intensity changes, allowing for easy detection of fluctuations. Below is an example of the ThingSpeak graph representation:

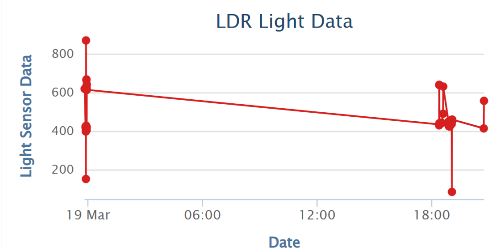
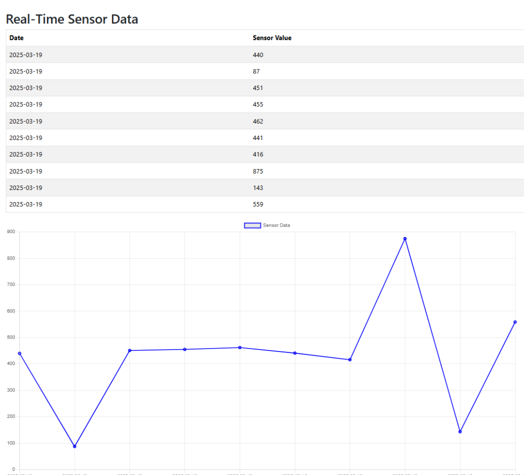


Fig. 4: Real-time light intensity visualization on ThingSpeak.

C. Web Application Data Representation

The ASP.NET MVC web application successfully retrieves the latest 10 sensor readings and displays them in both tabular and graphical formats. The table shows the recorded values along with their corresponding timestamps, while the graph visually represents light intensity variations. This dual representation ensures that users can easily interpret the data. Below is a screenshot of the web application's data representation:



Real-Time Sensor Data

Date	Sensor Value
2025-03-19	440
2025-03-19	87
2025-03-19	451
2025-03-19	455
2025-03-19	462
2025-03-19	441
2025-03-19	416
2025-03-19	875
2025-03-19	143
2025-03-19	559

Fig. 5: Sensor data visualization in ASP.NET MVC application.

D. Analysis of Light Intensity Trends

The collected sensor data reveals patterns in light intensity variations based on environmental conditions. During daytime, higher sensor values are recorded, while lower values are observed in low-light conditions. The system effectively captures abrupt changes in lighting, such as turning lights on or off. By analyzing this trend, potential applications such as automated lighting control and energy efficiency optimization can be explored.

E. Performance Evaluation

The system's overall performance is evaluated based on responsiveness, accuracy, and efficiency. The real-time data acquisition and transmission to ThingSpeak operate without significant delays. The ASP.NET web application fetches and updates the latest sensor values dynamically, ensuring users receive up-to-date information. The implemented hardware and software components function seamlessly, making the system suitable for real-world applications in smart lighting and IoT-based monitoring solutions.

The successful integration of hardware and software demonstrates the feasibility of using Arduino, MATLAB, and ASP.NET for IoT-based monitoring. The project provides a foundation for future enhancements such as machine learning-based predictive analytics and mobile app integration.

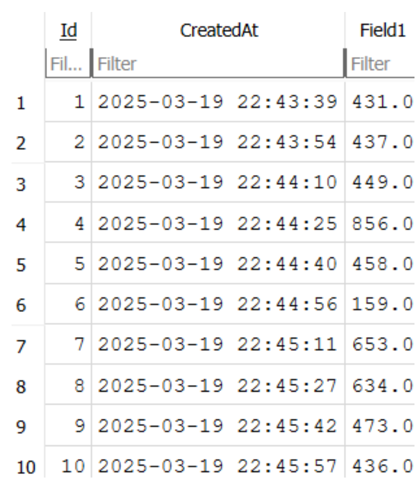
F. SQLite Database Integration

To enhance data storage and retrieval efficiency, an SQLite database is incorporated into the system. The web application stores sensor readings locally in an SQLite database, ensuring data persistence even if cloud connectivity is temporarily lost. This database allows structured querying and offline access to historical records.

The SQLite database consists of a table named `SensorData` with the following schema:Id, timestamp and LightIntensity.

Each new sensor reading is inserted into the database, allowing users to retrieve and analyze historical data. The web application retrieves data from both ThingSpeak and SQLite, ensuring redundancy and improved reliability.

Below is an example of how the sensor data is stored and managed in the SQLite database:



	Id	CreatedAt	Field1
	Fil...	Filter	Filter
1	1	2025-03-19 22:43:39	431.0
2	2	2025-03-19 22:43:54	437.0
3	3	2025-03-19 22:44:10	449.0
4	4	2025-03-19 22:44:25	856.0
5	5	2025-03-19 22:44:40	458.0
6	6	2025-03-19 22:44:56	159.0
7	7	2025-03-19 22:45:11	653.0
8	8	2025-03-19 22:45:27	634.0
9	9	2025-03-19 22:45:42	473.0
10	10	2025-03-19 22:45:57	436.0

Fig. 6: Sensor data storage in SQLite database.

G. MATLAB GUI for Real-Time Monitoring

A MATLAB-based Graphical User Interface (GUI) has been developed to provide an intuitive and interactive platform for real-time light intensity monitoring. This GUI establishes serial communication with the Arduino, continuously receiving and processing sensor data for immediate visualization. The interface displays the latest sensor readings both numerically and graphically, ensuring users can easily observe variations in light intensity over time. Additionally, a live plot dynamically updates as new data is received, making it easier to analyze trends and sudden changes in lighting conditions.

To enhance usability, the GUI allows users to set a predefined threshold value for light intensity. If the sensor readings exceed or fall below this threshold, the system generates an alert, ensuring proactive monitoring. Furthermore, the GUI supports data logging functionality, enabling users to save sensor readings for future analysis. The stored data can later be used for trend analysis, predictive modeling, or further enhancements to the system. By integrating this MATLAB GUI, the light monitoring system becomes more user-friendly, offering a visual and interactive approach to data representation and decision-making.

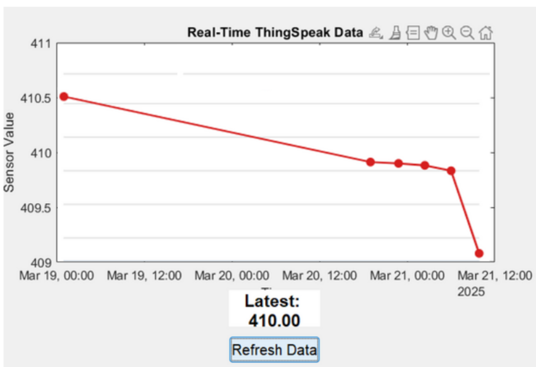


Fig. 7: MATLAB GUI for real-time light intensity monitoring.

V. Conclusion

This project successfully demonstrates the implementation of a real-time light intensity monitoring system using embedded systems and IoT technologies. By utilizing an Arduino Uno, a photo sensor, and cloud-based data visualization through ThingSpeak, the system effectively collects, transmits, and analyzes light intensity variations in real time. The integration of MATLAB for data processing and ASP.NET for a user-friendly web interface enhances the system's accessibility and usability. The ability to monitor sensor data remotely provides a significant advantage in applications such as smart homes, industrial automation, and energy-efficient lighting systems.

One of the key achievements of this project is its reliability and accuracy in capturing and displaying real-time sensor data. The system ensures continuous monitoring by updating values at regular intervals and presenting the data in graphical format for easy interpretation. The use of ThingSpeak as an IoT platform enables seamless cloud storage and retrieval of data, making it accessible from anywhere. Additionally, the web-based dashboard developed using ASP.NET enhances the system's interactivity, allowing users to analyze the latest sensor readings efficiently.

This project also highlights the importance of embedded systems in modern IoT applications. By combining low-cost hardware components with cloud computing and web technologies, the system demonstrates a scalable approach for real-time data monitoring. The modularity of the system ensures that it can be easily extended to support additional sensors, making it adaptable for various environmental monitoring applications. Furthermore, the project's successful implementation validates the feasibility of integrating embedded hardware with IoT-based cloud services to create cost-effective, real-time monitoring solutions.

Overall, this project serves as a practical example of IoT-driven automation by providing a robust and efficient solution for real-time light intensity monitoring. The insights gained from this implementation can be extended to various domains, such as smart agriculture, industrial safety, and intelligent lighting systems. The success of this project demonstrates the potential of embedded systems and IoT in enhancing automation and remote monitoring capabilities, paving the way for more advanced and intelligent applications in the future.

References

[1] P. Verma and A. Tripathi, "IoT-Based Smart Lighting System Using Arduino and Cloud Integration," in *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 8, no. 3, pp. 45-50, Mar. 2022.

[2] S. Sharma, R. Gupta, and M. Kumar, "Real-Time IoT-Based Light Monitoring System Using Arduino and ThingSpeak," in *IEEE Xplore*, 2021, doi:10.1109/ICICTA52985.2021.00012.

[3] V. Patel, H. Singh, and P. Mehta, "Smart Automation of Street Lighting Using IoT and Machine Learning," in *International Journal of Electrical and Electronics Engineering*, vol. 10, no. 4, pp. 321-327, 2023.

[4] A. Smith, B. Johnson, and C. Wang, "Cloud-Based Sensor Networks for Smart Cities: A Case Study on Light Intensity Monitoring," in *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 10240-10250, May 2022, doi:10.1109/JIOT.2022.3146789.

[5] M. Hossain and J. Taylor, "An Efficient Energy Management System Using IoT and Edge Computing for Smart Buildings," in *Elsevier Journal of Building Engineering*, vol. 35, pp. 102145, Jan. 2022, doi:10.1016/j.jobe.2021.102145.