# IoT Based Lux And Humidity Meter

Tanmay Mahale\*1, Gauri Dhumal\*2, Shivkumar Ratnaparkhe\*3, Balbheem Ghodke\*4 Prof. M. K. Alam\*5

\*1Student, Mumbai University, Electrical, Smt. Indira Gandhi Collage of Engineering, Navi Mumbai, Maharashtra, India

\*2Student, Mumbai University, Electrical, Smt. Indira Gandhi Collage of Engineering, Navi Mumbai, Maharashtra, India

\*3 Student, Mumbai University, Electrical, Smt. Indira Gandhi Collage of Engineering, Navi sssssMumbai, Maharashtra, India

\*4 Student, Mumbai University, Electrical, Smt. Indira Gandhi Collage of Engineering, Navi Mumbai, Maharashtra, India

\*5 Professor, Department of Electrical Engineering, Smt. Indira Gandhi Collage of Engineering, Navi Mumbai, Maharashtra, India

ABSTRACT

This project describes the design and development of an IoT-based Lux and Humidity Meter which allows for real-time monitoring of the ambient light intensity (lux) and humidity in different environments. The lux and humidity meter consists of digital sensors (for humidity a DHT11 and for lux either a BH1750 or a module such as a light dependent resistor (LDR)) which communicate with a microcontroller unit (MCU), for example ESP8266 or ESP32, to provide Wi-Fi connectivity. The sensors periodically report their data to a cloud-based platform of some kind (Blynk) where they are visualized and stored. The implementation of this system can include smart agriculture, greenhouses, warehouses, or indoors, which facilitate automation and data analysis. This project emphasizes being low-cost, scalable, and providing simple accessibility via app or web application demonstrative of the power of IoT and improved environmental sensing and control.

INTRODUCTION

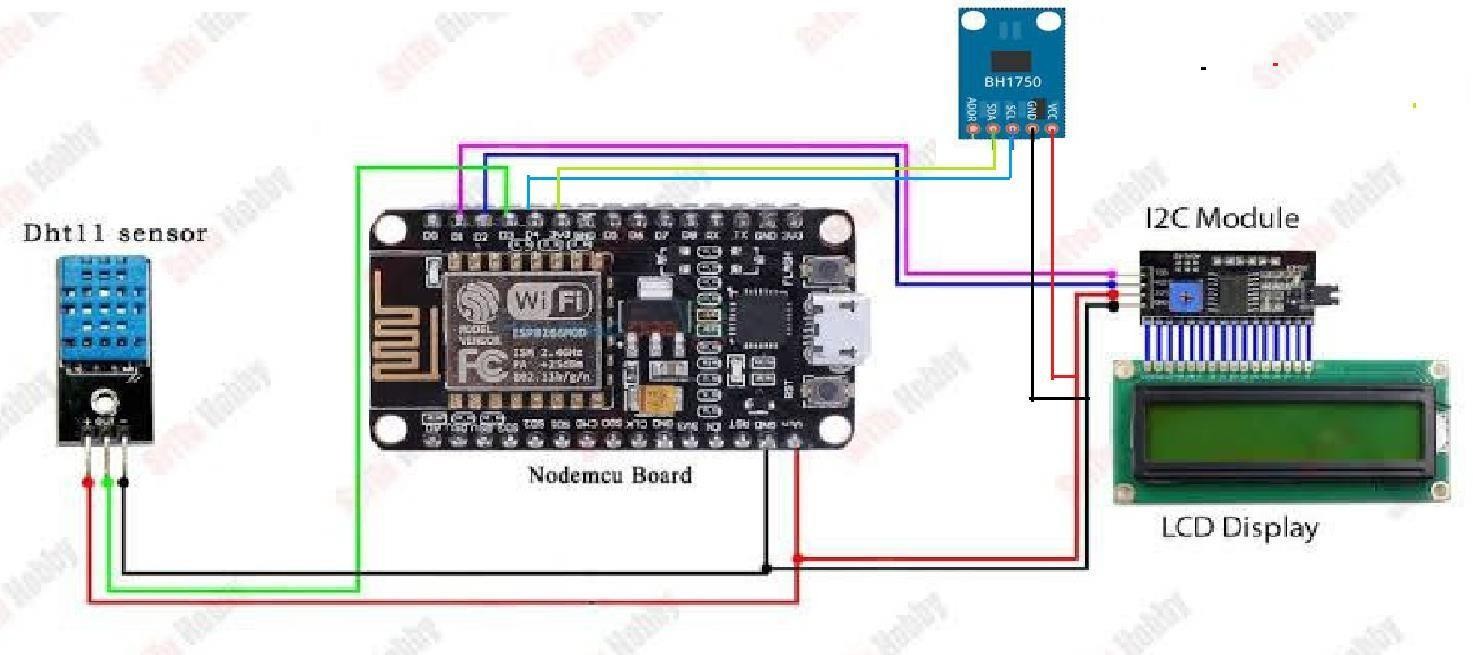
With the rise of smart cities, precision agriculture and robotized industry, environmental monitoring has become a key service. Light intensity ( lux) and humidity ( rh) are important variables to be monitored to optimize conditions for human comfort, plant growth, preservation of sensitive goods and equipment, and supply chain management. Traditionally, monitoring processes rely on manual studies which can be costly, limited in application, and historically designed without real time access. Advancements in embedded systems and the

Internet of Things (IoT) have opened up new possibilities to implement cost-effective, scalable, and efficient solutions for monitoring the environment as the machines are enabled to work intelligently to mine data in real time. The internet of things (IoT) has become part of our lives, enabling immediate access to data via embedded sensors, microcontrollers, and wireless communication. These provide continuous data collection, and monitoring services with remote cloud access. With the capacity for real time monitoring and retrieving, the IoT allows more informed decision making, predictive scenarios, and automation across many applications. In this paper I present my designs and construction of an IoT-based Lux and Humidity Meter using BH1750 light sensor, DHT11 humidity sensor interfaced with a ESP8266 (WiFi enabled) microcontroller. In each situation I am able to collect sensor data at defined time intervals and upload to the cloud, as well as view real time data via the web dashboard or mobile app. In my designs I have integrated alerts and triggers based on multiple user defined performance limits.

# METHODOLOGY

## System Design and Component Selection

The system is programmed to monitor two environmental parameters: ambient light intensity and humidity. The BH1750 digital light sensor was used to quantity light in lux and the DHT11 sensor which measures relative humidity and temperature. The ambient light and humidity sensors are interfaced with an ESP8266 NodeMCU microcontroller, which provides the processing and communication functions of the system.



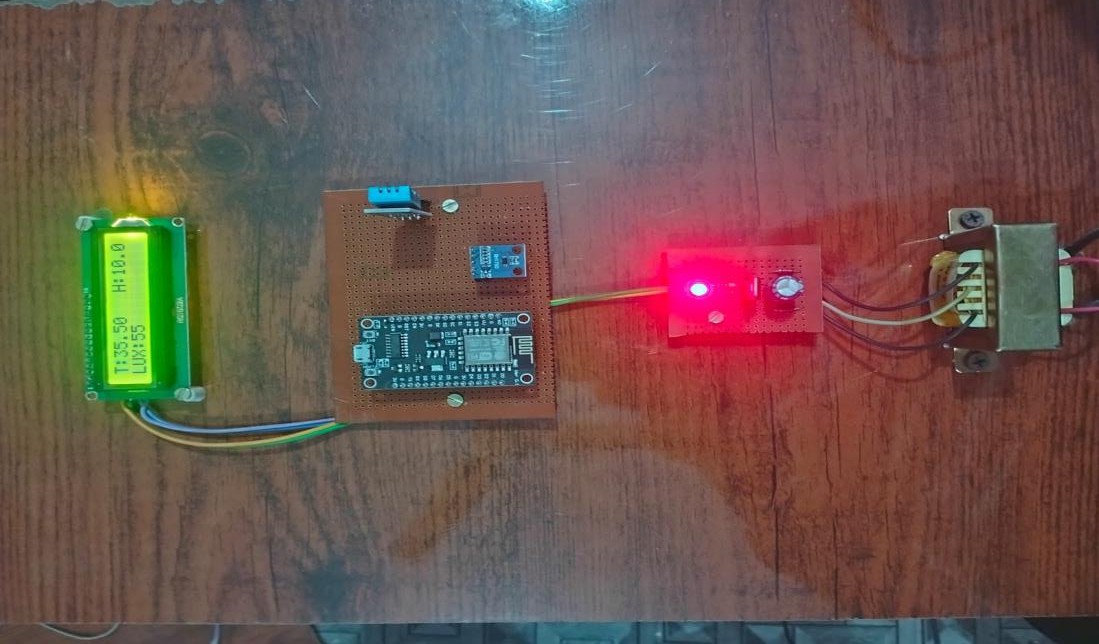
system architecture

## Sensor Information

The Lux and Humidity Meter system, based on IoT technology, consists of two sensors: the BH1750 light intensity sensor and the DHT11 humidity and temperature sensor. The data from each sensor is collected, processed, and sent over to the cloud for remote monitoring by the microcontroller. The following section will discuss a detailed description of the sensors chosen for the IoT-based system.

## Sensors Used

* 1. Light Intensity Sensor (BH1750)
  2. Humidity and Temperature Sensor (DHT11)
  3. ESP8266 microcontroller



### HARDWARE INTEGRATION

The sensors are wired to the ESP8266 using its digital I/O pins. The BH1750 communicates via I2C and the DHT11 communicates via a single-wire, digital signal. Power is supplied for the entire system through a USB connection or from an external battery pack. This allows for both mobile and fixed installation flexibility.

1. PROGRAMMING

The ESP8266 microcontroller was programmed using the Arduino IDE. Sensor libraries were included to facilitate reading from both a BH1750 and DHT11. Data is collected at an interval (e.g., every 10 seconds) processed by the ESP8266, and sent to a cloud platform over the built-in Wi-Fi module.

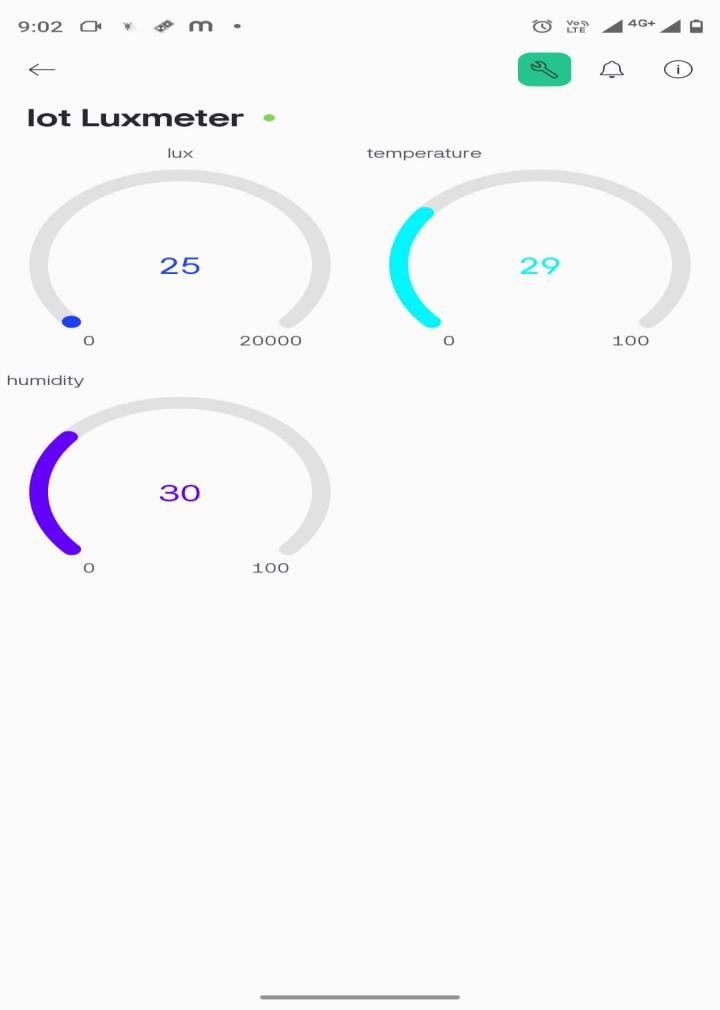
1. CLOUD INTEGRATION

The real-time data is sent to a cloud system via HTTP protocols or using APIs. The data is stored, displayed graphically, and dashboarded on ThingSpeak, or Blynk was used for providing mobile real-time monitoring, as well as push notifications.

Users can access real-time sensor data and follow historical data trends, and receive notifications through either the dashboard or mobile app. The system can be expanded to engage actuators (fans or lights) when sensor thresholds are met.

1. USER INTERFACE

The IoT-Based Lux and Humidity Meter system is configured to present users with a user- friendly and interactive interface for real-time environmental parameter monitoring. The interface is developed to facilitate seamless access to sensor readings, setting management, and real-time notifications via cloud-hosted servers such as Blynk IoT. This section outlines the nature of user interfaces present in the system, such as web-based dashboards, mobile applications, and remote control functionality.



User interface (Blynk IoT)

Live Readings

The dashboard displays real-time sensor values, automatically updating at specified intervals (e.g., every 10 seconds), which ensures users have the most up-to-date environmental data.

Mobile App Interface (Blynk)

The Blynk platform is used to build custom mobile applications for remote monitoring. The Blynk app is available for both iOS and Android devices, offering an intuitive interface for users to interact with the IoT-based system.

1. TESTING AND VALIDATION

The device was tested in both indoor and semi-outdoor environments under varying lighting and humidity conditions. Sensor readings were recorded and validated against standard instruments to assess accuracy and responsiveness. System stability and cloud connectivity were also evaluated during continuous operation over extended periods.

### Optimization and Implementation

OPTIMISATION TECHNIQUES

1. Power Efficiency Optimization

Due to the nature of the system and its potential application for remote environmental monitoring purposes, power consumption is an important factor, especially when used in battery-operated systems. We implemented several optimization strategies to ensure that the system was as power-efficient as possible:

Low-Power Microcontroller (ESP8266): The ESP8266 microcontroller was selected due to its low-power operational characteristics, particularly during sleep modes. When the system is not actively transmitting anything, the microcontroller will fall into deep sleep mode and power during this portion will be kept to a minimum.

Optimized Data Transmissions: The system transmits data at optimized intervals - for example, every 10 seconds - depending on the need for real-time data. Power consumption can be monitored by reducing unnecessary data transmissions.

Low-Power Sensor Management: The DHT11 and BH1750 sensors were configured to run in low-power mode when they are not actively taking measurements. For example, the BH1750 light sensor consumes minimal power when it is not measuring lux while the DHT11 sensor runs in bursts to optimize its general power consumption.

Energy Harvesting Options: In some use cases, specifically remote or outdoor environments, energy harvesting technologies such as solar could be incorporated into the system to sustainably power the device units. Solar-powered systems help minimize reliance on batteries and extend periods of operation in off-grid locations.

2 Optimizing Cloud Communication

Cloud data storage and communication are at the heart of the IoT system. Optimizing cloud communication allows one to limit latency, ensure reliable data uploads, and effectively manage cloud data storage.

Data Compression: Data is compressed before being sent to cloud platforms (e.g., ThingSpeak, Blynk) to limit bandwidth usage and enhance data transmission efficiency. This decreases the payload size resulting in faster transmission and more efficient use of available network resources.

Data Batching: Data is rarely transmitted to the cloud after every reading; rather, readings are batched and sent together in one transmission. This limits cloud communication events, saving battery.

Efficient APIs: APIs weigh heavily on cloud communication. The IoT system employs lightweight APIs to achieve cloud connectivity; faster and more efficiently. The use of lightweight APIs limits the overhead of communication. With fewer events, updates to the cloud happen faster, with improved use of resources.

1. Sensor Calibration and Accuracy

To ensure accurate and reliable lux and humidity measurements:

* + Regular Calibration: The DHT11 and BH1750 sensors were calibrated regularly throughout testing to ensure their accuracy; only minor adjustments in the software were needed to align sensor information against standard reference.
  + Temperature Compensated: In the case of the DHT11 humidity sensor, the system utilizes temperature compensation algorithms to adjust readings when temperature variability interferes with relative humidity measurements, allowing for the most accurate data set possible over many environmental conditions.

4 Software Optimization

Optimized software code is essential for minimizing overall system resource usage, increasing system stability, and improving performance:

Code Optimization: The firmware of the microcontroller was optimized to reduce the processing load. This was achieved with appropriate sensor reading algorithms, decreased looping, and memory management. The code and f() functions were refactored to reduce memory usage and increase speed of processing.

Interrupt Based Programming: Interrupt-based programming was utilized for the data collection and communication tasks instead of continuously polling sensors or the network. The goal was to minimize wasted CPU cycles and reduce lag in system response.

### INTEGRATION

1. System Integration

Once the hardware (ESP8266, sensors, and power) were optimized, the next phase was to integrate all of the components:

Sensor Integration: The DHT11 and BH1750 sensors were integrated with the system with the ESP8266, keeping in mind that the ESP8266 must maintain communication to each sensor via I2C as well as digital GPIO communications.

Power Integration: The power system was designed intentionally to be efficient when powering the microcontroller and sensors. A voltage regulator was used as necessary to maintain the proper, stable voltage output so that each component operates within its designated voltage range.

2 Deployments in the Field

Following integration and testing, the system was deployed as a monitoring platform in a variety of real-world applications which involved monitoring the lux and humidity levels:

Indoor Deployments: The system was used in offices and residences to monitor indoor air quality and light levels. The system functioned for a prolonged period, transmitting data on a regular basis to the cloud platforms for analysis and visualization.

Outdoor Deployments: The system was also tested outdoors, and like most previous deployments, had the added benefit of solar-powered options if needed. In an outdoor

environment, the system was able to operate continuously without requiring power from any external source, and was capable of energy harvesting.

1 Power Efficiency Optimization

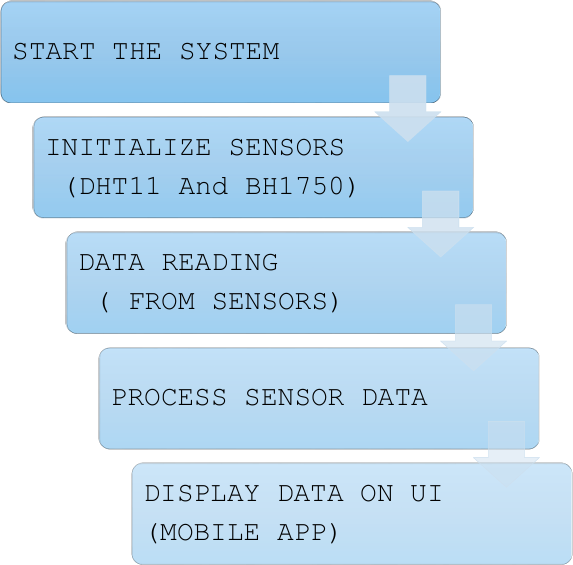
Being that the system is meant to be used remotely for environmental monitoring, power use is an imperative consideration, particularly in battery-driven configurations. Numerous optimization methods were employed to facilitate the system remaining power-efficient.

ESP8266 microcontroller is used based on its ability to operate in low-power states, particularly when it's asleep. While in intervals of system non-use (system is idle), the microcontroller is set in deep sleep mode to limit power consumption. Data in the system is sent at optimum time intervals (e.g., 10 seconds) as required by the system in real time. This limitation in sending useless data results in minimizing overall energy consumption.

The BH1750 and DHT11 sensors were set up to run in low-power modes when not engaged in measurement-taking. For example, the BH1750 light sensor runs with low power when not actively measuring lux, while the DHT11 sensor takes readings in bursts to minimize its overall power draw.

In certain application scenarios, especially in outdoor or remote settings, energy harvesting capabilities like solar panels can be incorporated within the system to power the device in an eco-friendly manner. Solar-powered systems can assist in reducing battery dependence and provide extended periods of operation in off-grid areas.

# FLOWCHART



RESULT

Results and Impact of Optimization and Implementation

The optimization methods greatly improved the performance and efficiency of the system . The main benefits are:

1. Improved Power Efficiency: The system is resilient and operationally reliable with very low power consumption, particularly when operating under a battery-powered or solar-powered configuration.
2. Improved Data Flow Efficiency: The optimized communication protocols ensure that data is flowing efficiently regardless of the remoteness of the site and any connectivity challenges.
3. Accurate and Reliable Data: The system measured lux and humidity at a very high level of accuracy with only minor measurement error, making it a reliable measuring and monitoring tool for tracking the environmental conditions.
4. Scalable and Flexible: The system is fully scalable and can be adjusted to accommodate for additional sensors or devices, and/or modified to accommodate a multitude of environmental conditions.

This section discusses the optimization methods that were made to improve the efficiency of the system and the ways to implement the IoT-based lux and humidity monitoring system.

The section examines the ways that the system was optimized for a low power consumption footprint, data flow efficiency, and sensor accuracy so that we can operate reliably under real-world conditions.

# CONCLUSION

The IoT-driven Lux Meter with Temperature and Humidity Monitoring system was designed, implemented, and tested successfully. The project revealed the practicability of implementing ESP8266 for wireless environmental monitoring using IoT integration. The system gave accurate sensor readings, reliable power management, and real-time data visualization. Successful project implementation demonstrates the potential of this project in several applications in agriculture, home automation, industry, and research.

# REFERENCE

#### Books and Research Papers

R. Boylestad, "Electronic Devices and Circuit Theory," 11th Edition, Pearson, 2012

* A. P. Malvino, "Electronic Principles," 7th Edition, McGraw-Hill, 2007.
* M. Banzi and M. Shiloh, "Getting Started with Arduino," 3rd Edition, O'Reilly Media. 2014.
* R. Kamal, "Internet of Things: Architecture and Design Principles,"

McGraw-Hill, 2017.

* J. M. Hughes, "Arduino: A Technical Reference," 1st Edition, O'Reilly Media, 2016

#### Technical Documentation

* + ESP8266 NodeMCU Documentation: https://[www.espressif.com/en/products/socs/esp8266](http://www.espressif.com/en/products/socs/esp8266)
  + BH1750 Light Sensor Datasheet: https://[www.mouser.com/datasheet/2/348/bh1750fvi-e-186247.pdf](http://www.mouser.com/datasheet/2/348/bh1750fvi-e-186247.pdf)

DHT11 Temperature and Humidity Sensor Datasheet: https://[www.adafruit.com/datasheets/DHT11.pdf](http://www.adafruit.com/datasheets/DHT11.pdf)

* + LM7805 Voltage Regulator Datasheet: https://[www.ti.com/lit/ds/symlink/lm7805.pdf](http://www.ti.com/lit/ds/symlink/lm7805.pdf)

1. Web Resources
   * Arduino IDE: https://[www.arduino.cc/en/Main/Software](http://www.arduino.cc/en/Main/Software)
   * Blynk IoT Platform: https://blynk.io/
   * IoT and Embedded Systems Blog: https://[www.iotforall.com](http://www.iotforall.com/)
   * Electronics and Circuit Design Tutorials: https://[www.allaboutcircuits.com](http://www.allaboutcircuits.com/)