ISSN: 2395-1680 (ONLINE)
DOI: 10.21917/ijme.2025.0330

AN IOT-BASED SOLUTION FOR SATELLITE POWER MANAGEMENT AND ENVIRONMENTAL DATA COLLECTION

Allacheruvu Brahmaiah[1], Ramagalla Srinivas[2] and Savitha Sunkari[3]
[1]*Department of Computer Science, RK College of Engineering, India*
[2,3]*National Remote Sensing Centre, Indian Space Research Organisation, Hyderabad, India*

Abstract
This study presents an innovative method to develop an Internet of Things (IoT) system. The prototype IoT system described herein is designed to remotely control demo-satellite power-enabled technologies while capturing real-time environmental parameters such as air temperature, humidity, and atmospheric carbon dioxide (CO2) levels. These data are then stored in a dedicated server database along with timestamps for efficient data management and analysis. The system can be interfaced with the Blynk IoT application for seamless remote-control access of the demo satellite and monitoring environmental parameters. Moreover, it provides an easy way to visualise stored data through graphical representations, enabling users to track trends and make informed decisions. The system can be expanded to accommodate additional sensors and parameters, further enhancing its functionality for diverse applications in environmental monitoring and remote-control systems, it supports the up to 50m range.

Keywords:
Internet of Things, Demo-Satellite, Temperature, Humidity Atmospheric Carbon Dioxide

1. INTRODUCTION

Internet of Things (IoT) technology's explosive expansion has created new avenues for innovation in a wide range of sectors, especially in remote control and environmental monitoring applications. To control power-enabled gadgets in a demo satellite, this study offers a novel method for developing an Internet of Things system. The system can operate remotely and record real-time data on important environmental parameters including air temperature, humidity, and carbon dioxide (CO2) levels thanks to its interaction with the Blynk IoT application [1]. To provide structured data management and enable accurate monitoring and analysis of environmental trends, the collected data is kept in a server-based database [2].

This IoT prototype offers easy-to-use data visualization tools in addition to remote control and data-collecting functions. These technologies help users make better judgments by making it simpler to monitor environmental trends through graphical representations [3]. Additionally, the system may be easily expanded to accommodate a wide range of use cases due to its modular architecture, which makes it possible to add new sensors to monitor other environmental data. Due to its adaptability, the system may be used in a variety of fields, including space technology and environmental studies [4].

The integration of IoT technologies greatly simplifies the remote control of the demo satellite, enhancing both scalability and flexibility. Utilizing the Blynk IoT platform, users can manage the satellite's power systems and monitor environmental conditions from virtually any location, providing ease of access and operational efficiency. The system's ability to process and store large volumes of data contributes to effective real-time monitoring and control. As IoT continues to develop, this prototype offers a foundation for more advanced applications in areas such as agriculture, climate monitoring, and space exploration, demonstrating its potential for widespread use and future expansion.

2. DESIGN AND IMPLEMENTATION

We outline the design methodology for integrating IoT into control demo satellite power technology, establishing a database on NodeMCU for real-time sensor data storage, and utilising predictive analysis techniques to forecast future predictions. The methodology encompasses various stages, including system architecture design, IoT device selection and integration, database design and implementation, predictive analysis framework, testing and validation, and further prediction monitoring [5].

2.1 DEVICE SELECTION AND INTEGRATION

Selecting IoT devices compatible with the Blynk IoT platform is crucial. The Fig.1 shows the integration of connection devices such as MQ-135, humidity/temperature sensor, and 4 relays connected to the database server are installed in NodeMUC, and thermostats are chosen based on compatibility and functionality. They are then integrated into the IoT system and configured to communicate with the central control hub via Wi-Fi and Bluetooth or other supported protocols of the Blynk IoT application [6]. The Fig.1, The diagram illustrates a smart sensor system controlled by a NodeMCU. The process begins with the power supply activation, which powers the NodeMCU [5]. The NodeMCU is connected to two types of sensors: the DHT-11[7], which measures temperature and humidity, and the MQ-135[8], which measures CO2 levels. The NodeMCU communicates data from these sensors via a Wi-Fi network to a cloud-based database Xammp server database [9], utilizing an IPv4 address for network communication. Additionally, the system can control multiple relays, which could be used to manage other devices based on sensor data. The system is integrated with the Blynk 2.0 platform for monitoring and control [6], [10]

The Fig.2 highlights the connections between different parts and displays the integrated system's overall design. Nodemcu receives sensor data for processing and archiving, and IoT devices use the control interface to instruct actuators on how to run Blynk IoT applications remotely over Wi-Fi. This architecture's ability to provide data in temperature, humidity, and carbon dioxide to store table format based on table features, as well as remote monitoring and control via Blynk IoT operating buttons, improves the efficiency and dependability of the demo satellite movements.[6].

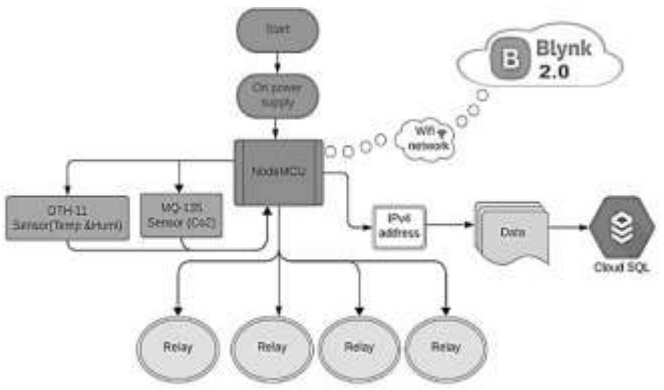

Fig.2. Device Selection and Integration of Components

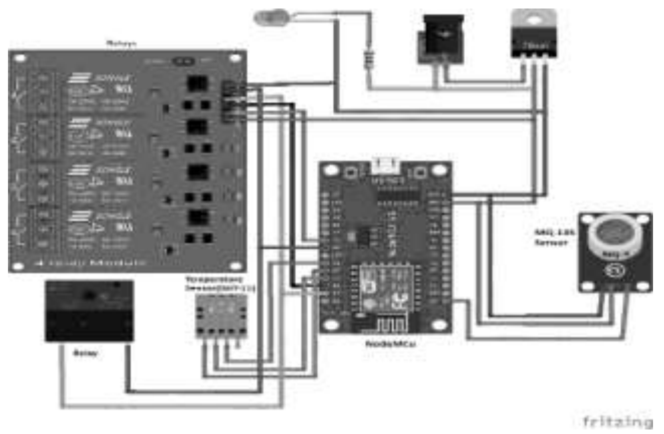

Fig.3. Inner connection of IoT System

2.1.1 Components Working:

NodeMCU is an open-source firmware and development kit that helps you prototype IoT (Internet of Things) products. It's based on the WI-FI module (ESP8266) and integrates the Blynk IoT application. NodeMCU, you can utilize these storage options based on your specific application requirements, balancing factors like data size, read/write speed, and ease of access, Thermal sensors connected to NodeMCU, also known as temperature sensors are devices designed to measure temperature & humidity levels and variations in their surrounding environment.[2], [6], [11]. In Table.1, these sensors detect and quantify thermal energy (heat) by converting temperature changes into electrical signals to stored Temperature data with timestamps that can be interpreted and processed by electronic devices. An air pollution sensor is a device designed to measure the quality of the air in an atmosphere environment. This sensor detects the atmosphere's Carbon dioxide (CO2). The below table1 can expresses some properties of Sensors.[6], [11]. The table provides specifications for two sensors: the DHT-11 and the MQ-135. The DHT-11 measures temperature and humidity, with a temperature range of 0°C to 50°C and an accuracy error percentage of ±2°C, and a humidity range of 20% to 80% with an accuracy error percentage of ±5%. It operates within a voltage range of 3.3V to 5V. The MQ-135 sensor measures CO_2 levels from 100 to 5000 ppm, with accuracy dependent on climate conditions, and requires a 5V power supply.

Table.1. Properties of IoT System Sensor

Sensor	Parameter	Range	Accuracy	Voltage
DHT-11	Temperature	0°C to 50°C	+-20C	3.3to 5V
DHT-11	Humidity	20% to 80%	+-5%	3.3 to 5V
MQ-135	Co_2	100-5000 ppm	Based on Climate	5V

2.2 INSTALLATION AND PROCESS OF NODEMCU

To install the NodeMCU board for programming, Fig.3 is shows the representation of the install process [1], [11].

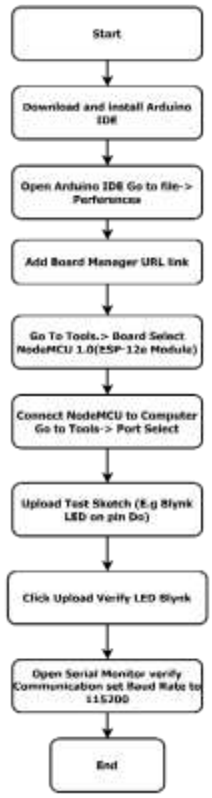

Fig.3. The installation process of NodeMCU

The typically follow these steps:

- **Download and Install Arduino IDE**: If you haven't already, download and install the Arduino IDE (Integrated Development Environment) from the official Arduino website.
- **Install ESP8266 Board Manager Package:** Open the Arduino IDE and go to File -> Preferences. In the "Additional Boards Manager URLs" field, add the following URL: http://arduino.esp8266.com/stable/package_esp8266com_index.json. Then, go to Tools -> Board -> Boards Manager, search for "ESP8266", and install the package.
- **Select NodeMCU Board:** After installing the ESP8266 board package, you can now select the NodeMCU board. Go to Tools -> Board, and under "ESP8266 Boards", choose "NodeMCU 1.0 (ESP-12E Module)".

- **Choose Port**: Connect your NodeMCU board to your computer via a USB cable. Then, go to Tools -> Port and select the port corresponding to your NodeMCU board.
- **Upload a Test Sketch:** To verify that everything is working correctly, you can upload a simple test sketch to your NodeMCU board. For example, you can try uploading the "Blynk" sketch, which Blynks an LED connected to pin D0 (GPIO 16) on the NodeMCU board.
- **Upload the Sketch:** Click on the "Upload" button (right arrow icon) in the Arduino IDE to compile and upload the sketch to your NodeMCU board. You should see the LED Blynk on the NodeMCU board if upload is successful [5].

2.3 CREATE THE DATABASE SENSOR DB IN XAMPP

A database named "Sensor dB" includes columns for Timestamp, Temperature, Humidity, and Air Carbon Dioxide (CO_2) in XAMPP, you would typically use a tool like phpMyAdmin, which is included in XAMPP. Here are the steps[9]:

- **Start XAMPP:** Start the XAMPP control panel and make sure Apache and MySQL services are running.
- **Access phpMyAdmin:** Open your web browser and navigate to http://localhost/PhpMyAdmin/ to access the phpMyAdmin interface.
- **Login:** Enter your MySQL username and password. If you haven't set up any custom credentials, the default username is usually "root" and there is no password.
- **Create Database:** Once logged in, click on the "Databases" tab and enter "Sensor dB" in the "Create database" field. Then click on the "Create" button to create the database.
- **Create Table:** After creating the database, select it from the left sidebar. Then, click on the "SQL" tab and enter the following SQL query to create a table with the specified columns:

```
CREATE TABLE IF NOT EXISTS Sensor_data (
Timestamp TIMESTAMP DEFAULT
CURRENT_TIMESTAMP,
Temperature FLOAT,
Humidity FLOAT,
CO2 FLOAT);
```

- **Linking Database to NodeMCU Microcontroller**: To link this database to your NodeMCU microcontroller, you would typically write a program in the programming language supported by the NodeMCU (usually Lua or Arduino C++). This program would collect sensor data (temperature, humidity, CO_2) and send it to a server endpoint running on your computer where XAMPP is installed.
- **Setting up Server-side Code:** On the server side (your computer with XAMPP), you would need to set up a web server (e.g., using Apache) and a server-side scripting language (e.g., PHP) to handle incoming requests from the NodeMCU, parse the sensor data, and insert it into the database.
- **Securing Your System**: Ensure that your system is properly secured, especially if it's exposed to the internet. Use proper authentication mechanisms, input validation, and secure communication protocols to prevent unauthorized access and data. Breaches localhost is 127.0.0.1 and the Server version is 10.4.32-MariaDB [9].

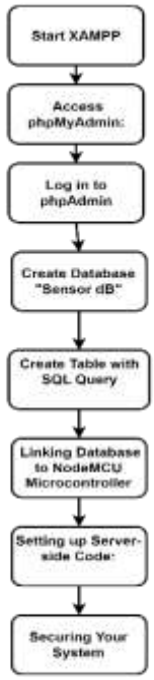

Fig.4. Create a Database in XAMPP to connect of IoT system

2.4 BLYNK IOT INSTALLATION & AND ITS SETUP

Below are the general steps to set up your project[10] and Fig.5. Shows the Blynk IoT application setup:

- **Install Blynk App:** Download and install the Blynk app from the App Store (for iOS) or Google Play Store (for Android).
- **Create a New Project:** Open the Blynk app and create a new project.

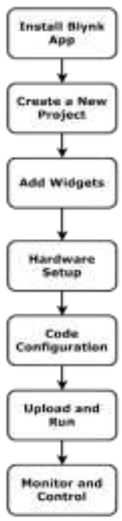

Fig.5. Blynk IoT Application Setup

- **Add Widgets:** In your Blynk project, tap on the "+" icon to add widgets. For buttons: Add a Button widget for each button you want to control. Configure each button with the desired properties (e.g., pin, mode). For temperature and Humidity, the Air quality gauge Add a Value Display widget. Configure it to display temperature readings from your temperature sensor and air pollution sensor.
- **Hardware Setup:** Connect your microcontroller to your computer via USB. Connect your temperature sensor to the appropriate pins on your microcontroller. Upload the Blynk example sketch corresponding to your microcontroller model (e.g., ESP8266 for NodeMCU, Arduino for Arduino boards). You can find these examples in the Blynk library or documentation.
- **Code Configuration:** Open the Blynk example sketch in your Arduino IDE. Replace the placeholder authentication token with the token you received via email. Modify the sketch to read temperature sensor data and update the Blynk widgets accordingly.
- **Upload and Run:** Upload the modified sketch to your microcontroller. Once uploaded, power on your microcontroller.
- **Monitor and Control:** Open the Blynk app on your outreach facility campus phone. You should see your widgets (buttons and temperature gauge) in your project. Press the buttons to control your hardware (e.g., turn on/off LEDs, motors) [2], [6].

Table.2. Range of Blynk IoT via Wi-fi to controlled by the IoT System to Power technologies of the Demo satellite.

Range distance	Command	Status
0.5 m	ON/OFF	ON
1 m	ON/OFF	ON
10 m	ON/OFF	ON
20 m	ON/OFF	ON
50 m	ON/OFF	ON
100 m	ON/OFF	OFF

2.5 ABOUT DEMO SATELLITE

Demo Satellite is an innovative spacecraft equipped with cutting-edge features tailored for monitoring and detection tasks. It utilizes advanced technologies like OpenCV for computer vision and an array of sensors for environmental surveillance. Key functionalities include motion detection for identifying movement, object detection for recognizing objects, and gesture detection for interaction with the Blynk IoT application to Control the power technologies of sensors.

3. RESULTS

The IoT System can capture the Environment parameters to store the dedicated database, its server is 127.0.0.1, and the Database is Sensor_db table name is Sensor_data. The Fig.6. Is shows the storing of the parameters in the database.

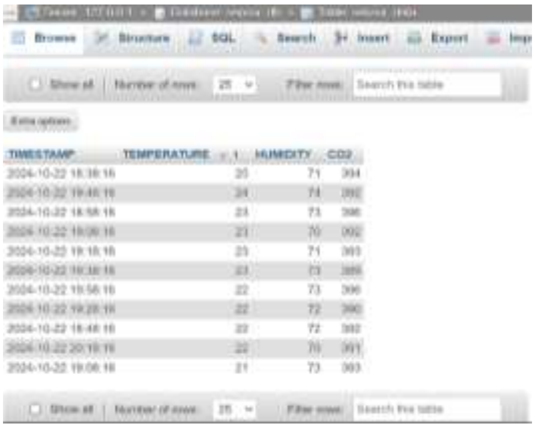

Fig.6. Capturing Data Store in XAMMP Server database

Xammp Server makes it very easy to export the data in multiple formats like CSV, SQL, PDF etc. The Fig.7. Shows the types of data exporting formats.

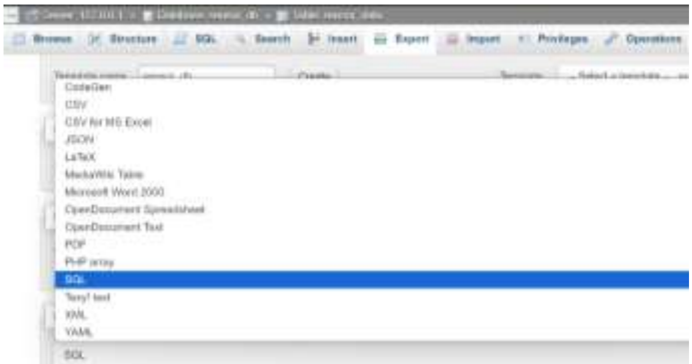

Fig.7. Data Exporting Formats

Below the data can be easily. Visualised in the graphical formats, in Fig.8. Capturing data of Temperature a long with Timestamp and its Range.

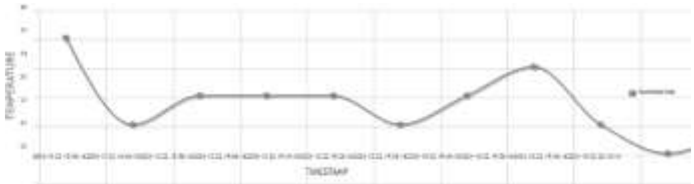

Fig.8. Capturing data of Temperature

The Fig.9 visualised the capturing data of Humidity along with timestamp and its ranges.

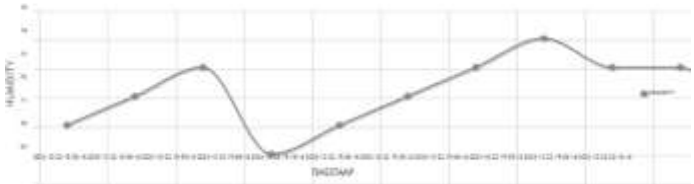

Fig.9. Capturing data of Humidity

The Fig.10 visualised the capturing data of AtmosCO_2 along with a timestamp and its ranges.

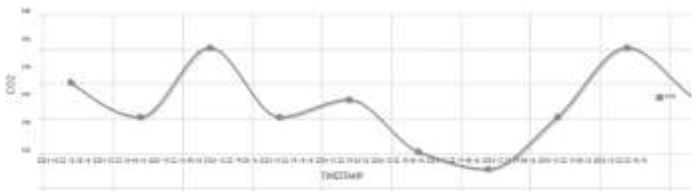

Fig.10. Capturing data of Atmos Co_2

4. CONCLUSION

This study presents a successful approach to developing a robust IoT system tailored for remote control and environmental monitoring applications. The prototype demonstrated effective control over demo-satellite power systems, while seamlessly capturing, storing, and visualizing real-time environmental data such as temperature, humidity, and CO_2 levels. By integrating with the Blynk IoT platform, the system enables users to remotely monitor and manage environmental parameters, supporting informed decision-making through easy-to-understand graphical data representations. The system's expandable design allows for the addition of more sensors, broadening its potential applications in various monitoring and control scenarios. This research underscores the versatility and scalability of IoT systems in enhancing remote environmental monitoring and automation capabilities.

REFERENCES

[1] A.J. Islam and Sadman Shahriar Alam, "Design and Development of a Low-Cost IoT based Environmental Pollution Monitoring System", *Proceedings of International Conference on Electrical Engineering and Information and Communication Technology*, pp. 1-6, 2019.

[2] Jasmeet Chhabra and Punit Gupta, "IoT based Smart Home Design using Power and Security Management", *Proceedings of International Conference on Innovation and Challenges in Cyber Security*, pp. 1-6, 2016.

[3] Genady Grabarnik, "Addressing Prerequisites for STEM Classes using an Example of Linear Algebra for a Course in Machine Learning", *Proceedings of International Conference on Mobile, Hybrid and On-line Learning*, pp. 57-62, 2020.

[4] George Mois, S. Folea and T. Sanislav, "Analysis of Three IoT-Based Wireless Sensors for Environmental Monitoring", *IEEE Transactions on Instrumentation and Measurement*, Vol. 66, No. 8, pp. 2056-2064, 2017.

[5] Datasheets, "NodeMCU ESP8266", Available at https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet, Accessed in 2020.

[6] Peerasak Serikul and Nuttapun Nakpong, "Smart Farm Monitoring via the Blynk IoT Platform", *Proceedings of International Conference on ICT and Knowledge Engineering*, pp. 1-6, 2019.

[7] Datasheets, "DHT11-Temperature and Humidity Sensor", Available at https://components101.com/sensors/dht11-temperature-sensor, Accessed in 2021.

[8] Datasheets, "Air Quality Gas Sensor", Available at https://www.winsen-sensor.com/d/files/PDF/Semiconductor%20Gas%20Sensor/MQ135%20(Ver1.4)%20-%20Manual.pdf, Accessed in 2015.

[9] IONOS, "XAMPP Tutorial", Available at https://www.ionos.com/digitalguide/server/tools/xampp-tutorial-create-your-own-local-test-server/, Accessed in 2023.

[10] No Author, "Blynk IoT Documentation", Available at https://docs.blynk.io/en, Accessed in 2015.

[11] S. Shinde, A.H. Karode and D.S.R. Suralkar, "Review on IOT based Environment Monitoring System", Available at https://iaeme.com/MasterAdmin/Journal_uploads/IJECET/VOLUME_8_ISSUE_2/IJECET_08_02_014.pdf, Accessed in 2017.