Journal of IoT in Social, Mobile, Analytics, and Cloud (ISSN: 2582-1369)
www.irojournals.com/iroismac/

# Centralised Chaining for Wireless Personal Area Network

**Halesh M R[1], Chethan G Kottari[2], Jamaluddin Ahamed[3], P Kedareshwar Rao[4], Vinayak Y Sandimani[5]**

Electronics and Communication Engineering, JSS Science and Technology University, Mysuru, India

**E-mail:** [1]haleshmr@sjce.ac.in, [2]chethankottari789@gmail.com, [3]jamaluddinahamed51@gmail.com, [4]puttigekedareshwar@gmail.com, [5]vinayakys784@gmail.com

## Abstract

Centralised chaining represents a method of organizing and coordinating the communication and interaction among the nodes within the PAN. It aligns with the objective of developing a PAN protocol with high interference immunity and low radio noise. The proposed idea aims to optimize the interconnectivity of the PAN nodes, enabling seamless communication, sensing, actuation, and data transmission within the network. The need for high-quality data, that uses minimal power (1mW to 100mW) and long distances is increasing every day. Since ninety percent of devices are used internally, more bandwidth, low latency, high spectral efficiency, and high smooth output are needed. Additionally, as the user base expands, researchers face greater difficulty in creating a dependable and quick wireless communication system using these current gadgets. The proposed new design of the PAN is expected to overcome these issues with the help of advanced algorithms and protocols that improve the interconnectivity of the devices participating in the PAN.

**Keywords:** Personal Area Networks (PAN), IoT, Cloud, Communication Protocols.

## 1. Introduction

Increasing internet connectivity to the point where the whole world uses it is giving birth to many new protocols, especially in the domain of personal area networks (PAN) [5]. Most networks that are found today are in some way a PAN, as the current trend in automation requires creating a network of sensors, actuators, and cloud computers. Networks, from a township to modern office spaces, are all PANs, or variations of it depending on the data rates. These networks are preferred over LAN or WLAN (which serve as the backbones of the


*DOI: https://doi.org/10.36548/jismac.2023.4.004*
*Received: 12.11.2023, received in revised form: 08.12.2023, accepted: 21.12.2023, published: 04.01.2024*

internet protocol) due to their scalability, ability to add more features into existing protocol stack, security, ease of implementation for small projects, and, if necessary, can be connected to the internet for external data communication [6-7].

One of the key features of a PAN is interconnectivity of the devices participating in the network (nodes) and how this interconnectivity is handled using algorithms and protocols (a software defined network, - SDN), which are flexible and ensure interoperability between nodes [3]. The research aims to build hardware nodes with gateway hardware and develop a chaining mechanism between the hardware nodes themselves and the gateway hardware, to form a personal area network protocol which is developer friendly, open sourced, has security, and troubleshooting capabilities. The hardware nodes developed operate in license free frequency bands thereby making it suitable for small to enterprise level sensor-actuator networks, monitoring networks and data sharing networks [9-15].

### 1.1 Motivation

In the proposed work, the research aims to cut down the problems and issues with PAN as well as design and develop PAN architecture. This architecture is easy to use and scale for sensor-actuator networks. This design and the software system design of the application program interface (API) will be open sourced so that the user can make any required modifications to the applications that are being developed. This is an effort to make this research more developer friendly and reusable. The hardware nodes are designed to have networking, sensing, and actuating capabilities. This PAN architecture can be connected to a cloud for computing or storage and provides additional flexibility to this architecture. With the requirement of networks for almost everything we do today, PANs are becoming very common, and there are very few PAN architectures that are available to the users. Even with the existing technologies there are drawbacks and discomfort, which include the inability to scale the network due to cost, greater power consumption, and multiple inter-networking devices.

### 1.2 Problem Statement

Automation technologies presently in industrial and agricultural domain make use of wireless communication technologies like Wi-Fi protocol IEEE 802.11, ZigBee, LoRa, and complex computing and networking devices. If the bandwidth of the data to be transmitted is very low in networks such as sensor-actuator networks or data collection from a simple sensor, the use of complex computing and networking devices is not feasible; hence, PANs that have

customized protocols with simple and low-cost hardware nodes are better-suited for sensor-actuator networks.

### 1.3 Objectives

i. To develop a low cost, compact hardware, radio friendly node with actuation and sensing abilities, this can communicate to the cloud.

ii. To reduce the requirement of inter-networking devices, mainly routers in a PAN.

iii. To reduce the internet bandwidth required in a PAN.

iv. To design and develop a network architecture/model for sensor area networks.

v. To develop a PAN protocol having high interference immunity and low radio noise.

## 2. Literature Survey

The concept of the paper [1] is the comparison of different wireless communication protocols, focusing on the parameters like clock synchronization, MAC protocol, and data non-interference algorithms among the commonly used protocols like IEEE 802.15.4, Fire-Fly, RT-Link, A-LNT, and A-Stack. Less scope is given for secured communication as most of the protocols lack encryption of data before sending. The transmission of non-encrypted data can be a security threat as the information can be leaked out even if a single node of the network is infected.

[2]. This research addresses the problems with IEEE 802.15.4 Low-Rate Wireless Personal Area Network Coexistence. A proposed standard called IEEE 802.15.4 aims at facilitating wireless sensor networks while fulfilling the demands of low-rate wireless personal area networks, or LR-WPAN. The importance of configuring end nodes to form a cluster consisting of star and peer-to-peer connections is discussed. Hence, the custom PAN protocol proposed will follow a similar network topology, and secured communication will be enabled by use of RSA based private key encryption.

[4]. Analysing the relative performance of wireless personal area networks using short-range wireless protocols. The study aims to perform a quantitative analysis of the parameters like encoding efficiency, bit rate, bit error rate, and short-range characteristics and report these as comparison. Bluetooth, Zigbee, WiFi, and UWB were compared. This research points out areas for improvement in these technologies. Because of the 5GHz band's higher power

consumption and poor spectral efficiency for the growing traffic and connection demand, the IEEE 802.11 Wi-Fi protocol is not as suitable for building PANs. This provides an opportunity to develop a solution and create a more favourable architecture for implementing PAN.

[8]. Multimedia application protocols and architecture for wireless personal area networks (WPAN). This paper discusses the architecture of two WPANs, namely: "IEEE 802.15.3 for High-Rate WPAN (HR-WPAN)" and "IEEE 802.15.4 for Low- Rate WPAN (LR-WPAN)". These protocols define the architecture for wireless multimedia PANs. This literature is about how nodes participate in a PAN and discusses the function of a standard IEEE network coordinator. The idea around a PAN that can be a personal operating space ensuring data security and data transfer convenience is highlighted here. This gives a detailed description of the network protocol stack of the above- mentioned WPANs and pushes towards the opportunities in WPAN like need for low-cost, and flexible architectures.

### 2.1 Summary of Literature Survey

There are several PAN architectures that are available on the market, but these technologies have disadvantages of their own, and these mainly affect their implementation efficiency in sensor and actuator networks for monitoring and control, like power consumption, cost of implementation, or cloud access ability. The proposed PAN architecture solves these issues and provides a cloud connection for any computation or storage.

The proposed architecture dictates the design and development of a gateway device that acts as the entry point for the IP network (external network) and a network node that can sense, actuate, and form connections with the neighbouring nodes to create a personal area network. The development is focused on making the network consume less power and be the most cost-effective architecture for deploying PANs.

## 3. Related Work: Design and Implementation

### 3.1 Hardware Requirements

Since there are two types of nodes being developed on the hardware side, i.e., for the gateway as well as the end node, each have different specifications and hence require different types of ICs as well as power supply:

### A. ESP 32

The ESP32 is a series of low-cost, low-power chip microcontrollers with built-in dual-mode Bluetooth and Wi-Fi. It utilizes the features of the Tensilica Xtensa LX6 dual-core and single-core CPU, as well as the Xtensa LX7 dual-core and single-core RISC-V microprocessor. It additionally has an integrated antenna switch, an RF balun, a power amplifier, an acquisition amplifier, filters, and low noise control modules. The ESP8266 micro controller. This is used for the design and development of gateway device that interfaces with the other hardware nodes for command and control, and also connects to the cloud for monitoring and meta data management. The ESP 32 is shown in Figure.1 below.

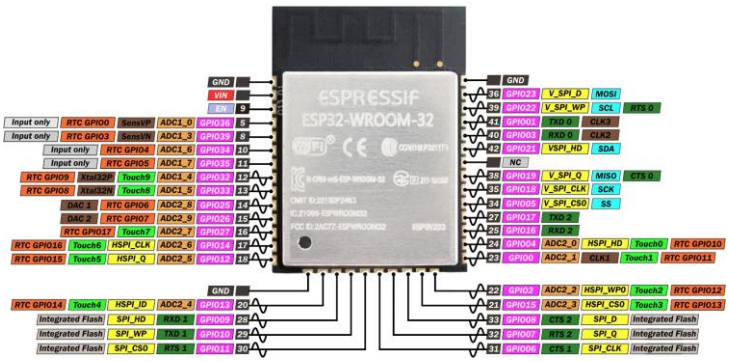



**Figure 1.** ESP 32

### B. ATmega328p

The Atmel 8-bit microcontroller based on AVR enhanced RISC architecture includes flash memory 32 KB ISP and reading capacity, 1 KB EEP- ROM, 2 KB SRAM, 23 lines for standard I/O, 32 standard performance registers, three flexible timer / counter with similar modes, internal and external interference, USART configuration series, byte-oriented two-wire serial interface, SPI serial port, six-channel 10- bit A / D converter, standard watchdog timer with internal oscillator, and five software saving options. The device operates between 1.8 and 5.5 volts. The device has access to up to 1 MIP / MHz usage. This chip is used to logically code the intelligence into the nodes that perform actuator and sensor functions and also communicates with gateway device. The pin details of the ATmega328p are shown in Figure.2 below.

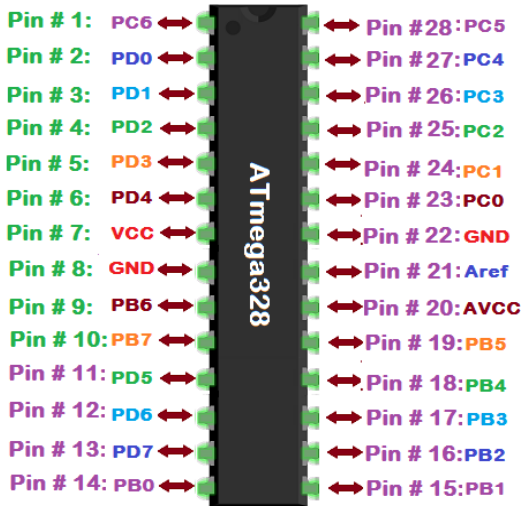



**Figure 2.** ATmega 328p C. nRF2401

The nRF24L01 is a low-power wireless application with 2.4GHz single-chip transceiver that has embedded baseband protocol. The nRF24L01 is intended to function between 2.400 and 2.4835 GHz on the worldwide ISM frequency range. nRF24L01 uses a Serial Peripheral Interface for configuration and operation. Every configuration register of nRF24L01 is included in the register map, which is available to all chip applications. Data transfer between the system's MCU and radio end is ensured by internal FIFOs. This controls every high-speed link layer function, hence lowering system expenses. A GFSK switch is used on the radio's front. The frequency channel, output power, and air data rate are among the user-adjustable factors. It is possible to set the air data rate that nRF24L01 supports to 2Mbps. nRF24L01 is very appropriate for very low power projects because of its high level of air data and two energy-saving techniques. High Power Supply Rejection Ratio (PSRR) and a broad range of power sources are guaranteed by internal power controllers. Data communication within networks is accomplished with this. The Figure.3 shows the details of the nRF2401.

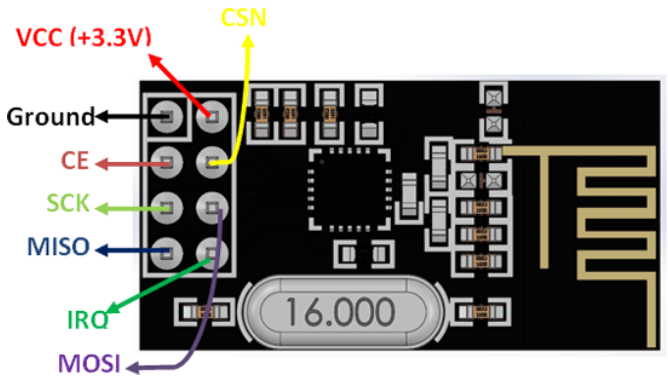



**Figure 3.** nRF2401

### 3.2 Software Requirements

Since the software was being developed for both the user- end as well as the cloud end, there were multiple skill as well as tools required to create and configure the custom networking protocol, and some level of basic embedded C was also used:

#### A. AWS Cloud Computer

These AWS cloud computing offers dispersed software tools and computer processing throughout AWS server farms. Among these services is Amazon Elastic Compute Cloud (EC2), which enables users to have an online virtual computer collection that is always available. Several real computer features are simulated by AWS virtual computers, such as pre-loaded application software, network, memory / RAM memory, hard disk / SSD storage, applications selection, and hardware for central processing units (CPUs) and graphics processing units (GPUs) for processing, sites, and CRM software.

#### B. Apache2

This research uses Apache2 for maintaining the network traffic so as there is not a huge load of traffic on the server.

#### C. Angular JS

AngularJS is used as the front of the MEAN stack, which includes MongoDB database, Express.js web server framework, AngularJS itself (or Angular), and Node.js server.

#### D. Python Flask

Flask supports extensions that can add app features as if they were done on Flask itself. There are extensions for object-related maps, form verification, download management, various open authentication technologies, and several tools related to the standard framework

### 3.3 Design and Implementation

The complete block diagram of the proposed implementation of centralised chaining for wireless personal area network is shown below in Figure 4. The diagram can be classified into three different sections (User-Left, Cloud-Centre and Hardware-Right). Here we can see the interconnection between the three sections and the internal architecture of each of the sections. Two different users with their PANs are considered here, the chaining among the nodes can be seen in PAN1 and PAN2.

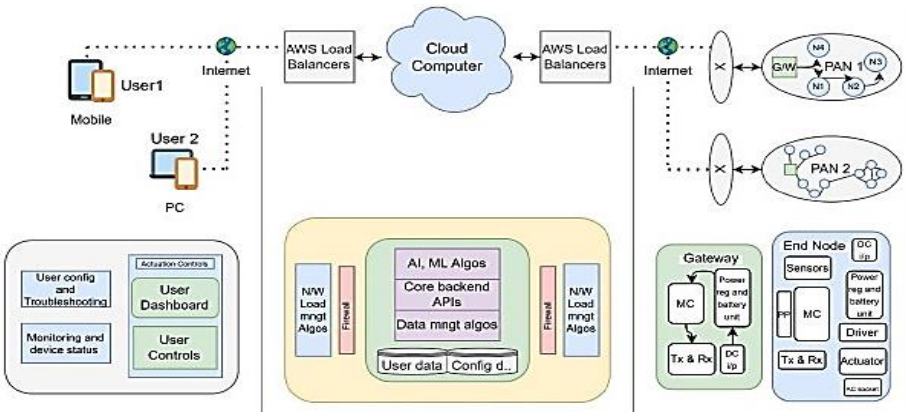
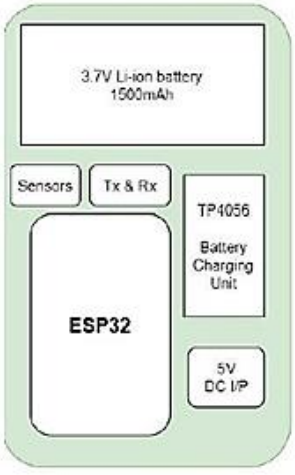



**Figure 4.** Block Diagram of Proposed Centralised Chaining for Wireless PAN

The design of a PAN with hardware nodes which can be controlled to operate a sensor-actuator can be categorized to three sub-divisions namely: control and monitoring (user side), computing (cloud functions) and the PAN (network with nodes). The user side will have a graphical user interface that represents data from the PAN after being processed in the cloud computer. This interface can be used to control the nodes of PAN and perform troubleshooting. The Cloud functions process, store, and create AI/ML models on the data that is acquired from the PAN, these functions regulate the network and provide security to the whole architecture. The network with nodes will have hardware nodes that perform sensor-actuator functions and communicate with each other and the gateway for optimal data transfer with efficient end device control.

### A. Hardware Implementation

The below figures show the different components present in the gateway and node device and the tentative positions can also be observed.



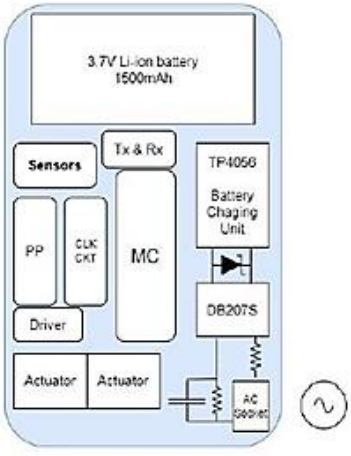




**Figure 5.** Gateway Node **Figure 6.** End Node

### B. Software Implementation

The software side of the project is designed for two parts of the project namely; the cloud and the user end part. At the user end a front-end application is designed for both Personal Computers as well as Mobile Phones. For this the AngularJS is used for interactive and ease of rendering. These are fed in by the APIs, which are created and documented and supplied by the cloud with information of each device in the mesh. This is a two-way communication. The user may interact with the mesh device remotely if he chooses to. These communications are done using the cloud for which we have used Amazon Web Services. They act as the middleman where all the data is stored and any communication to and from the user to the Personal Area Network take place. The backend of the app is developed using Java programming and the APIs are developed using FLASK API. The information passed between each node in the mesh is in Hypertext Transfer Protocol (HTTP). This is to decrease the payload size so that it takes minimum amount of bandwidth to process or transfer the data to each other.

## 4. Results and Discussion

### A. Hardware Output

The gateway device receives the information from the server and writes the same onto the mesh network, gateway device image and serial monitor output images are shown in Figures 7 and 8. The serial monitor output corresponding to four types of end nodes are discussed below.

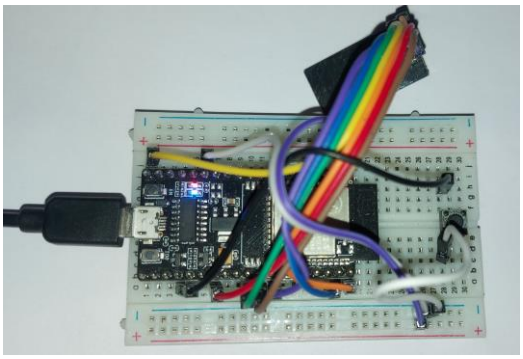


**Figure 7.** End Node for Agricultural Automation

Figure 8 shows the existing Wi-Fi connectivity details. Once existing details are read from internal memory, the ESP32 is connected to a specified Wi-Fi and connected to cloud server. Upon successful connection, it will receive the API details to which it will connect to receive node details. It will also receive the API details to which the sensor data has to be submitted to the server where it is saved and processed.

```
CCPAN


Gateway Device Started
WiFi Router SSID Stored in EEPROM:
Funking7

WiFi Router Password Stored in EEPROM:
12345678

Device ID Stored in EEPROM:
10001

Device API_KEY Stored in EEPROM:
GRNWNUI

WIFI Details Read
Connecting
.....HTTP Response code: 200

HTTP Response code: 200
http://fypiotsjce.ddns.net:5010/device_api/getdata?did=10001&dapi=GRNWNUI

Connected to WiFi network with IP Address: 192.168.137.117
SETUP OK

Creating Gateway MESH Network.....
0
```

**Figure 8.** Serial Monitor Readings Showing Initial Configurations

The gateway synchronously receives updates from the cloud server and will write the same over the entire mesh network on RF protocol. The nodes are actively listening to the gateway subscription to sense the change in the GPIO state and operate the relay to achieve actuation operation. The end node device receives the information from gateway by reading the information written over the mesh network. These devices are designed for four applications; the detailed description of different node devices with applications of each of them is discussed in upcoming subsections.

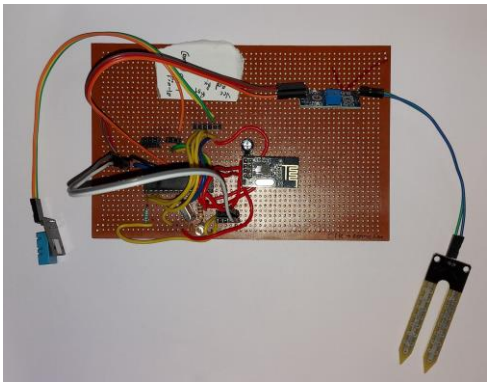


**Figure 9.** End Node for Industrial Automation

Figure 9 represents an ATmega328p chip with nRF24L01 connected to proper GPIOs. The Figure 10 represents a serial output of one of the end node devices, which shows information interception from the mesh and processing of the same. The sensor nodes collect sensor data and send back the same; the sent acknowledgement is being shown on the serial monitor. The actuator node receives the GPIO state and carries out the actuation.

```
Received packet from #0 say 0
gpio 7 state 1
Sending packet from node : 2 response : 1 ultra : 87 ir : 800 temp : 27 humidity : 80 moisture : 1023
Send OK
Sending packet from node : 2 response : 1 ultra : 87 ir : 800 temp : 27 humidity : 80 moisture : 1023
Send OK
```

**Figure 10.** Serial Monitor Output Showing Mesh Read and Write of Sensor Data

### B. Software Output

The user interface developed with Angular JS connects to the cloud over https and performs user controls over the mesh network, this control is facilitated by interactive UI and secured by authentication. The gateway device has an internal UI server deployed in order to get Wi-Fi connectivity details from the user. Users can access the Gateway UI by connecting to gateway access point through a specific URL.

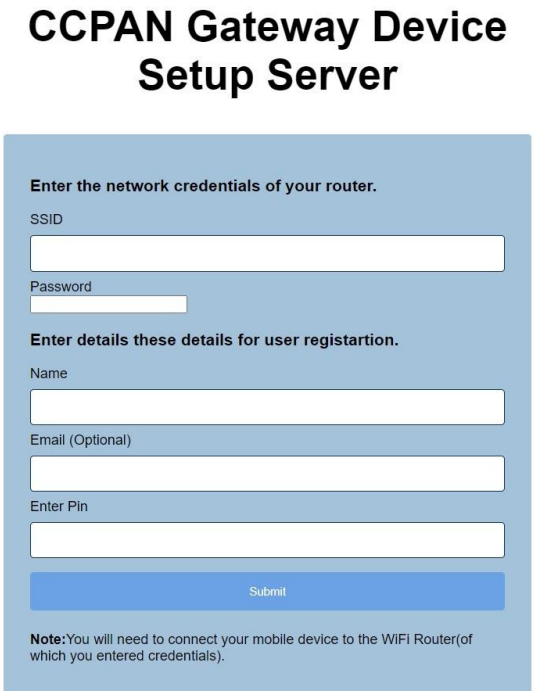



**Figure 11.** Gateway Internal UI for Network Registration.

Figure 11 shows the UI deployed inside the gateway device, which helps in configuring the Wi-Fi credentials. Figure 12 shows user authentication screen where the user enters the login credentials received during gateway registration. Upon successful authentication, the user will be redirected to the dashboard, where the user can operate the node actuators and monitor the sensor values of individual sensor nodes.

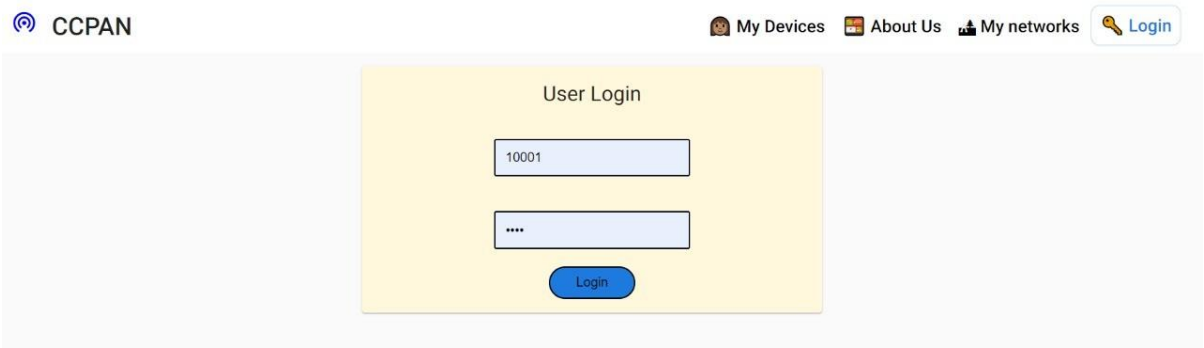



**Figure 12.** UI User Authentication Screen.

The screen in Figure 12 redirects to the dashboard screen. The dashboard in Figure 13 has buttons for actuation operations on individual nodes. The sensor data collected by the gateway from each of the sensor nodes is stored and processed in the cloud server and the same is retrieved by UI as well as displayed in a modal upon click of a button. The same can be seen in the UI snapshot, as shown in below Figure 14.


**Figure 13.** User Dashboard

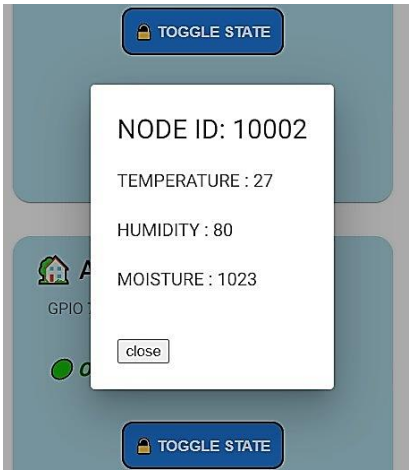



**Figure 14.** Sensor Data for a Specific Node

```
(venv) ubuntu@ip-172-31-13-183:~/fp/fpserver/device_api$ python3 app.py

       Application Started...
Opened database successfully
Closed database successfully
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on https://172.31.13.183:5001/ (Press CTRL+C to quit)
10001 GRNWNUI
Opened database successfully
GRNWNUI
API matched
[(10000, 2, 1), (10001, 6, 1), (10002, 7, 1), (10003, 6, 1), (10004, 7, 1), (10005, 6, 0)]
[[10000, 2, 1], [10001, 6, 1], [10002, 7, 1], [10003, 6, 1], [10004, 7, 1], [10005, 6, 0]]
150.129.63.34 - - [11/Jul/2022 05:16:40] "GET /device_api/getdata?did=10001&dapi=GRNWNUI HTTP/1.1" 200 -
10001 GRNWNUI
Opened database successfully
GRNWNUI
API matched
[(10000, 2, 1), (10001, 6, 1), (10002, 7, 1), (10003, 6, 1), (10004, 7, 1), (10005, 6, 0)]
[[10000, 2, 1], [10001, 6, 1], [10002, 7, 1], [10003, 6, 1], [10004, 7, 1], [10005, 6, 0]]
150.129.63.34 - - [11/Jul/2022 05:16:43] "GET /device_api/getdata?did=10001&dapi=GRNWNUI HTTP/1.1" 200 -
```

**Figure 15.** GPIO Data Fetch from API

The APIs for backend interactivity are deployed in Ubuntu 18.04 virtual machine staged in AWS EC2 server, The APIs are used to authenticate users and gateway devices. Upon successful authentication the users and gateway devices will be redirected to data fetching APIs from which user interface dashboard is populated and the gateway device fetches the updated GPIO state. The APIs are provided for gateway to upload the sensor data for the cloud server to store and process the same. The Figure 15 shows the console API which responds with the GPIO data for gateway.

The console output of authentication API describing the process of authentication is shown in Figure 16, the API sends back a https response upon successful authentication, in

case of gateway device the API send back the GPIO states and for the user interface it sends back a response containing Boolean which describes if the authentication was successful or the authentication has failed. Upon reception of this response the UI will redirect the user to dashboard and the gateway will write the updated GPIO states over the mesh network. The figure 16 shows the console output of the API discussed.

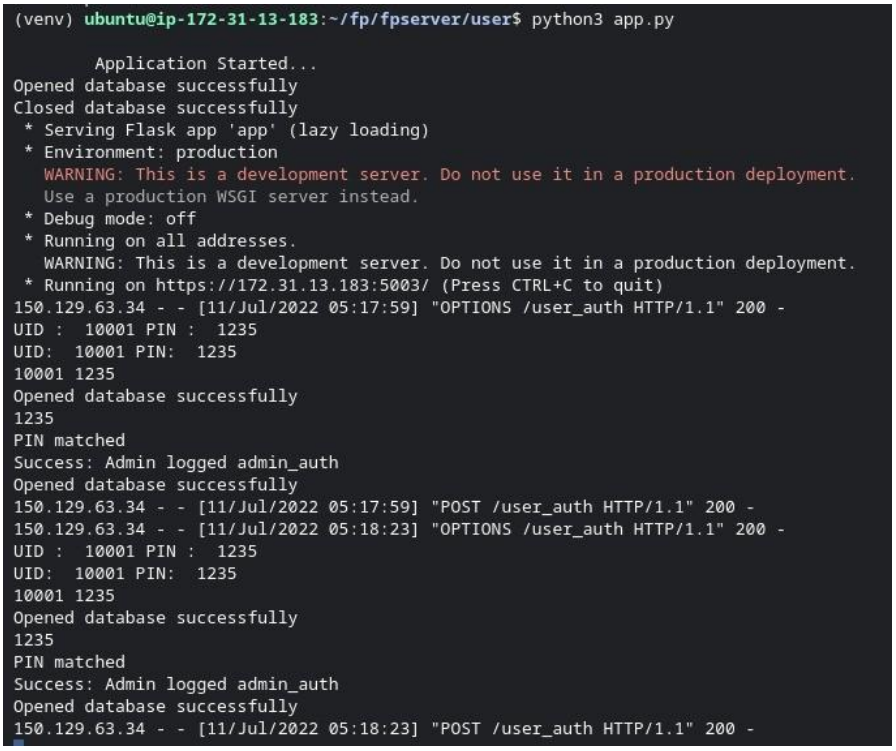



**Figure 16.** Authentication at Cloud End

## 5. Conclusion

Centralised chaining represents the centralized organization and management of interconnections within the PAN, serving as a fundamental component of the proposed PAN architecture and protocol. PAN is developed with a hardware demonstration that communicates with each other using a custom protocol and hence can be controlled using a single gateway node which is in-turn controlled by a single user owner. It has a merely good range thanks to the NRF frequency band that has a very light weight communication protocol, hence making the bandwidth spectrum used significantly less. An interactive UI for the user to control and monitor PAN is developed. All these information is stored in a database in which the data is secure and not vulnerable to any third-party clients.

## References

[1] Swaraj, C. M., and K. M. Sowmyashree. "IOT based smart agriculture monitoring and irrigation system." International Journal of Engineering Research & Technology (IJERT) 8, no. 14 (2020): 245-249.

[2] Kim, Beom-Su, HoSung Park, Kyong Hoon Kim, Daniel Godfrey, and Ki-Il Kim. "A survey on real-time communications in wireless sensor networks." Wireless communications and mobile computing 2017 (2017).

[3] Purwantana, Bambang, Fajar Siti Muzdrikah, M. Shohibun Nuha, and Muhammad Rivai. "Design of Wireless Sensor Network (WSN) with RF Module for Smart Irrigation System in Large." In 2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM), pp. 181-185. IEEE, 2018.

[4] M. Bagaa, T. Taleb, J. B. Bernabe and A. Skarmeta, “A Machine Learning Security Framework for Iot Systems,” in IEEE Access, vol. 8, pp. 114066-114077, 2020, doi: 10.1109/ACCESS.2020.2996214.

[5] Mubashar, Rehman, Muhammad Abu Bakar Siddique, Ateeq Ur Rehman, Adeel Asad, and Asad Rasool. "Comparative performance analysis of short-range wireless protocols for wireless personal area network." *Iran Journal of Computer Science* 4 (2021): 201-210.

[6] Z. Ma and X. Pan,” Agricultural environment information collection system based on wireless sensor network,” 2012 IEEE Global High Tech Congress on Electronics, 2012, pp. 24-28, doi: 10.1109/GHTCE.2012.6490118.

[7] I. Howitt and J. A. Gutierrez,” IEEE 802.15.4 low rate - wireless personal area network coexistence issues,” 2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003., 2003, pp. 1481-1486 vol.3, doi: 10.1109/WCNC.2003.1200605.

[8] E. Sakai, N. Ikeuchi and H. Suzuki, “A Proposal of Virtual Personal Area Network System Which Enables Direct Communication with PAN Devices in Remote Locations,” 2020 IEEE 9th Global Conference on Con- sumer Electronics (GCCE), 2020, pp. 682-683, doi: 10.1109/GCCE50665.2020.9291924.

[9] Khaled A. Ali, Hussein T. Mouftah, “Wireless per- sonal area networks architecture and protocols for multimedia applications”, Ad Hoc Networks, Volume 9, Issue 4, 2011, Pages 675-686, ISSN 1570-8705, https://doi.org/10.1016/j.adhoc.2010.09.006.

[10] Yaghoubi, Mohammad, Khandakar Ahmed, and Yuan Miao. "Wireless body area network (WBAN): A survey on architecture, technologies, energy consumption, and security challenges." Journal of Sensor and Actuator Networks 11, no. 4 (2022): 67.

[11] Paparao Nalajala, R. V. Krishnaiah, B Annapurna, Bhavana Godavarthi, “Security for Wireless Local Area Network with Pre-Share Key Authentication Using Wi-Fi Protected Access”, International Journal of Civil Engineering and Technology (IJCIET) Volume 8, Issue 8, August 2017, pp. 841–851.

[12] F. Sheldon, J. Weber, S. Yoo, W. Pan, The Insecurity of Wireless Networks, IEEE Computer Society, 10(4), July/August, 2012, pp. 54-61.

[13] S. Deepthi G Mary Swarnalatha, Paparao Nalajala, Wireless Local Area Network Security Using Wpa2-Psk, International journal of advanced trends in computer science and engineering, Volume V, Issue I, Jan 2016, pp. 41-45.

[14] L. Wang, B. Srinivasan, N. Bhattacharjee, Security Analysis and Improvements on WLANs, Journal of Networks, 6(3), March 2011, pp.470-481.

[15] P. Feng, Wireless LAN Security Issues and Solutions, IEEE Symposium on Robotics and Applications, Kuala Lumpur, Malaysia,3-5 June, 2012, pp.921-924.