ESP IJCEET
ESP International Journal of Communication Engineering & Electronics Technology
Volume 1 / Issue 1 Paper Id: IJCEET-V1I1P102 / Page: 9-13

Original Article

Establishing Wireless Mesh Network for Controlling Corridor Lights

G Siva Sai Teja[1], G Ashish Chandra[2], R P S Sunny[3], M Ravi Kumar[4]

Department of Electronics and Communication Engineering, KoneruLakshmaiah Education Foundation, Vaddeswaram, Guntur(Dist), Andhra Pradesh, India-522302.

Received Date: 04 March2023 Revised Date: 20 March 2023 Accepted Date: 05 April 2023

Abstract: *Here, an IoT network was combined with a wireless mesh network topology to establish communication between ESP-32 radio nodes without the use of wires and any intermediate router. Implementation was done using a hardware experimental setup that was created. Three NodeMCU ESP-32 boards coupled in a mesh network topology made up the system. The goal was to send the data to a third NodeMCU ESP-32 board without connecting to the internet or setting up a router in between. There was no parent node because every node in the network was able to communicate with every other node as long as a reliable path was provided. The experimental findings demonstrated that the mesh network topology-connected system has the innate ability to automatically establish communication between each node and would be able to self-heal in the event that any route becomes disabled.*

Keywords: *IoT, WiFi, Mesh Network, ESP-32 Modules(Node MCU), Relays, Resistors, PushButtons.*

I. INTRODUCTION

The growth of the Internet of Things (IoT) creates a wealth of possibilities for new computing paradigms and is driving a significant shift in how the Internet is evolving. The Internet of Things (IoT) has recently advanced, making it possible to build control interfaces using multisensor networks and other IoT devices and collect data from various contexts[1]. IoT network adoption continues to be a challenge despite the rapid growth of low-power communication technology confronts many challenges. Although wireless mesh networks have been discussed for many years, they haven't been widely implemented. This wireless mesh networking can significantly impact the current IoT era and effective and efficient networking solutions. In order to possibly make a difference in the new era, this work provided a quick explanation and demonstration of how these technologies may combine. It also showed how to incorporate the mesh network into current IoT networks. Through the linkage of data processing and transmission, we ere in this study suggest a methodology to draw from this resource pool. The suggested approach can lower the delay in transmitting processed information by taking use of the localization principle present in many IoT applications. In order to join billions of units and provide quick coverage at the lowest possible network cost, Mesh client interaction with the IoT has increased major relevance. To show how the system functions, a hardware prototype made up of active and passive nodes is created. In order to test Mesh networks and the connectivity of Wi-Fi between different nodes using the same Wi-Fi password for data exchange without the use of the internet or a router, the inexpensive WiFi microprocessor ESP-32 was used. There are many wireless communication technologies to control the corridor lights like Zigbee which is a low-powered, low-data-rate communications radio based on IEEE 802.15.4 that is intended for use in personal networking[2,3]. Unfortunately, compared to wireless communication radios like radio frequency, the cost of ZigBee on the market is higher. In this paper, the design of a wifi-based system for controlling corridor lighting is described. Internet protocol (IP)- based WiFi is commonly utilized for a variety of devices, including Laptops, mobile phones, etc. Price-wise, the wifi module is less expensive than the ZigBee module.

II. PROBLEM FORMULATION

- With the purpose of implementing the suggested work to select and learn more relating to the computing field with a built-in WiFiSoC.
- In order due to the system's distantly deployed nodes to be able to communicate with one another without the usage of the internet or a router, it is important to understand the key components of wireless mesh network topology and how they are implemented[4].
- To create a system architecture with all the components organized properly.
- study the NodeMCU, relays, and push buttons that have been installed here hardware setup for the demonstration's experimental purposes.

- To create the hardware for each of the three nodes, using ESP-32 boards serving as the hardware's central processing unit and input/output components.
- For each node, use the Arduino IDE to create an algorithm and build firmware.
- To adjust the system for the desired outcomes, test and debug the source code repeatedly, then upload it to each node to complete the calibration.
- To track how the experimental setup responds and to confirm the research using the desired results from an experiment.

III. PROPOSED WORK

Here, a system that uses IoT revolves Around the ESP-32 NodeMCU boards and three distinct Wi-Fi-enabled nodes was proposed. In order for the three nodes to communicate with one another, they had to interface with peripherals for input and output. Mesh network architecture had to be connected using this Internet of Things network in order to establish communication between these nodes. The nodes have to join together within a wireless network. In order to develop such a wireless network, each user necessary to select their personal Wi-Fi name and password in order to create an ad hoc meshwork. It was also important to remember that all nodes needed to share the same Wi-Fi credentials in order for communication to be feasible between them[5,6].

IV. ARCHITECTURE OF SYSTEM

As indicated in the diagram below, three distinct nodes were created for this system design and were built around three ESP32 NodeMCU Modules. These three nodes of the Internet of Things were connected through a network and to a few trigger devices as shown in Figure 1.

- The NodeMCU 1 module is connected to two tactile push buttons.
- The NodeMCU 2 and NodeMCU 3 modules are connected to Relay Switch Module and Relay Switch Module to the lamp.

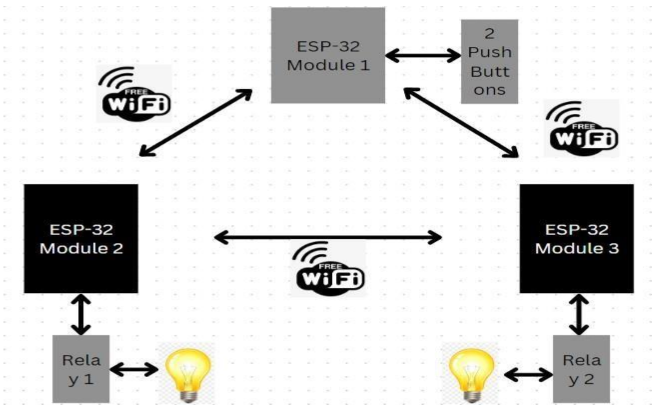

Figure 1: Structure of the System

These nodes all formed a mesh network topology and connected wirelessly in order to communicate with one another using the Wi-Fi protocol. Without using the internet or a router in between, each node in this network was sending and receiving data from the other two nodes[7].

V. PROCEDURE

This work used an experimental setup to conduct research and was intended to show how three hardware modules may interact locally over a WiFi network. These modules were designed to transmit data through a mesh network remotely for the purposes of monitoring and control[8,9]. In general, it's an Internet of Things (IoT) system with certain actuators, and computing boards powered by Wi-Fi to execute a networking strategy that doesn't need a router or the internet in between[10,11]. A Wi-Fi-enabled NodeMCU board with good performance and inexpensive cost was the hardware used in this project. digitalInput/Output pins are provided for connecting actuators. Three NodeMCU boards with the ESP-32 WiFi SOC, a single channel relay module, a pair of tactile push button modules, a pair of lamps, and a pair of +5V DC power source were used in this project to execute the idea. The work should be validated by the experimental findings using this hardware experimental set-up. Sincere efforts had to be made in order to acquire information from the datasheets of the various hardware components as part of the study project. The information acquired from these sources was utilized to create the general process for designing, developing, and putting the system into operation. The final report had the appropriate tables, flowcharts, and figures in addition to images of a functioning prototype.

Table 1: Components List

S.No.	Component	Qty.
1	NodeMCU Module	3
2	SPDT Relay	2
3	Resistor	2
4	Push Buttons	2
5	Lamps	2
6	Jumper Wires	10

VI. IMPLEMENTED SYSTEM

Node 1: As could be seen, this system contained three distinct devices/nodes. The result was the creation of three unique hardware components based on NodeMCU modules.. A NodeMCU module was connected to two push buttons in Fig. 2 below. Push buttons served as system inputs. Push buttons and the NodeMCU all operated on a 3.3V DC power supply[12]. Power pins were the remaining two pins. Two pull-up resistors were used to connect two push buttons to the NodeMCU's two digital I/O pins.

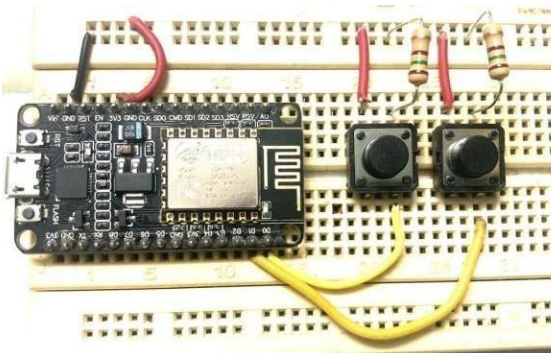

Figure 2: Architecture of Node1

Node 2: A NodeMCU module was also created around the Node-2 architecture, as seen in Fig. 3 below. Here, the NodeMCU was connected to Relay Switch Module and Relay Switch Module to the lamp.

Figure 3:Relay Circuit of Node 2

Node 3: A second NodeMCU module was created around the Node-3 architecture, as seen in Fig. 4 below. Here, the NodeMCU was interfaced with Relay Switch Module and Relay Switch Module to the lamp.

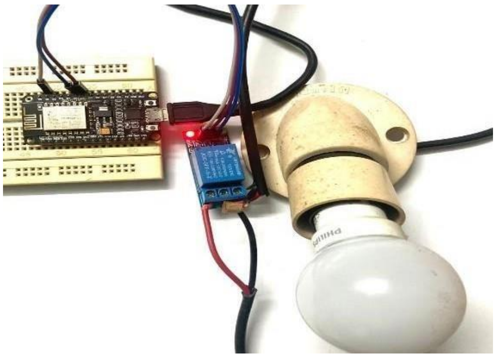

Figure 4:Relay Circuit of Node 3

Here, the two push button values served as an input device and was designed to run on 3.3V dc. One of the digital I/O pins of the NodeMCU was linked to another SPDT relay on the other side.

VII. EXPERIMENT RESULTS

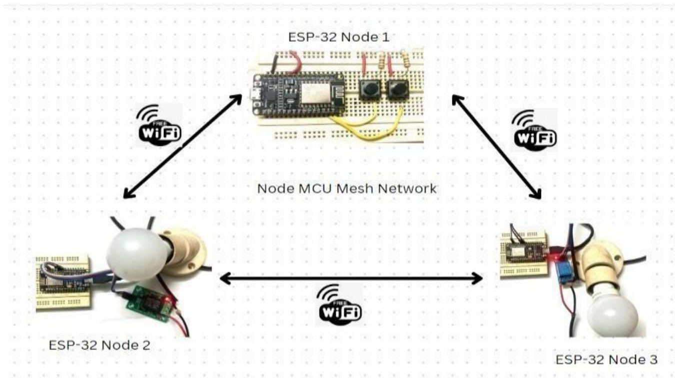

Figure 6: Experiment

Here, the hardware prototype's output response was checked to make sure it matched the desired output for each input. The finished product should verify the research. Obtaining the results of every node connected to this network using a serial port monitor was another method of verifying the outcomes[13]. The Arduino IDE includes a built-in serial port monitor. But PuTTY was employed here. Each NodeMCU board was connected to the system via cables connected to the laptop USB COM ports in order to power up the system and serially monitor these nodes. Although the COM port addresses for each serial monitor window varied, all nodes used the same baud rate of 115200 bps[14]. when we push button1 the lamp glow which is connected to node 1 and when we push the second button the lamp glow which is connected to node 2. This process is done wirelessly through the mesh network.

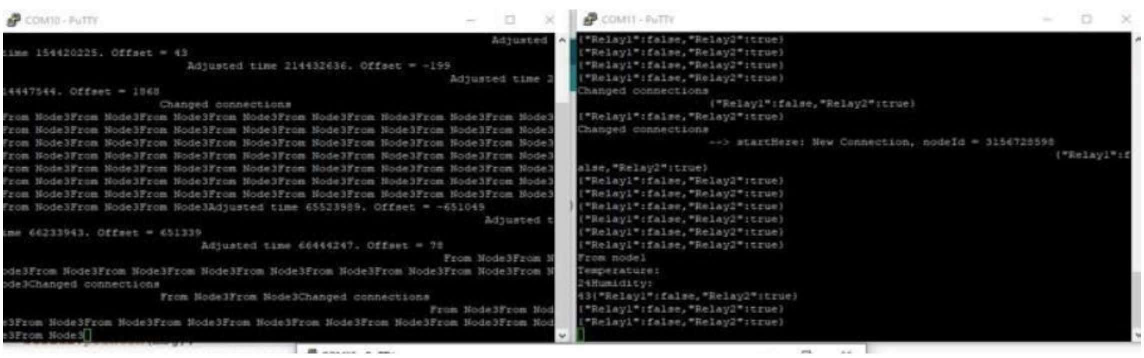

Figure 6: Results from Nodel as Seen on a SerialFigure 7: Results from Nodes 2 and 3 as Seen onSerial Monitor

Now, since the serial monitor in the Arduino IDE can only display one port's worth of data at once, you must use Putty or another serial program to view the data flowing from all of the nodes[15,16]. Select the serial option in putty and type each NodeMCU's com port, with 115200 as the speed. May each node's data can be viewed on the corresponding terminal as seen in the above Fig 5 which is for Node 1 showing that mesh is established and Fig 6 is for Node 2&3 which shows the output when we press the push buttons whether the relay is activated or not if you open terminals.

VIII. CONCLUSION

In this study, the wireless mesh network topology was combined with an Internet of Things (IoT) network. A group of nodes was made to exchange data with one another that they had collected from other input devices in order to carry out corresponding actions over the output devices without the use of the internet or a router in the middle. The NodeMCU computing platform, which combines ESP-32 Wi-Fi SoC with high performance, low power, and cheap cost, served as the foundation for the Internet of Things system. In this IoT network, input and output devices were the "things" utilized. Push buttons were used as input devices. A relay switch module and lights are served as output devices for displaying output results. Although the source code for each node was different, the Wi-Fi credentials entered in each source code were the same for all the nodes. Only three nodes were used for this presentation. With this mesh network structure, The relaying of messages between nodes is a shared responsibility, resulting in a substantially greater coverage area, allowing for a huge rise in the number of nodes[17]. The mesh network's capacity for self-healing was also noted. The work was validated by the hardware experimental setup that was constructed and presented.

IX. REFERENCE

[1] R. Mahendra and Y. H. Arzi, "Development of street lights controller using wifi mesh network," 2017 International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), Yogyakarta, Indonesia, 2017, pp. 105-109, doi: 10.1109/ICON-SONICS.2017.8267830.

[2] G. Shahzad, H. Yang, A. W. Ahmad, and C. Lee,"Energy-Efficient Intelligent Street Lighting System Using Traffic- Adaptive Control," IEEE Sens. J., vol. 16, no. 13, 2016.

[3] Kashyap; Mohamed Azma ;Jithu G. Panicker, "Ubiquitous Mesh: A Wireless Mesh Network for IoT Systems in Smart Homes and Smart Cities", IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2019, Publisher: IEEE

[4] Y. K. Tan, T. P. Huynh, and Z. Wang, "Smart personal sensor network control for energysaving in DC grid powered LED lighting system," IEEE Trans. Smart Grid, vol. 4, no. 2, 2013.

[5] Ms. Manvi1, Mr. Sunil Kr. Maakar 2 " Implementing Wireless Mesh Network Topology between Multiple Wi-Fi Powered Nodes for IoT Systems" Volume: 07 Issue: 10 | Oct 2020

[6] Yoppy , R Harry Arjadi , EndahSetyaningsih , PriyoWibowo , M I Sudrajat "Performance Evaluation of ESP8266 Mesh Networks" IOP Conf. Series: Journal of Physics: Conf. Series 1230 (2019) 012023.

[7] M. S. K. M. Ms.Manvi. Implementing Wireless Mesh Network Topology between Multiple Wi-Fi Powered Nodes for IoT Systems.International Research Journal of Engineering and Technology. 2020;7(10):1242–1244.

[8] D. Caicedo and A.Pandharipande, "Distributed illumination control with local sensing and actuation in networked lighting system," IEEE Sensor J., vol. 13, no. 3, pp 1092-1104, March. 2013.

[9] T. Wang, B. Zheng, Z-L Liang, " The design and implementation of wireless intelligent light link control system based on Zigbee light link," in Proc. ICCWAMTIP, Chengdu, 2013, pp. 122-125.

[10] Muhendra, R.; Rinaldi, A.; Budiman, M. Development of WiFi Mesh Infrastructure for Internet of Things Applications. Procedia Eng. 2017, 170, 332–337.

[11] Cilfone, A.; Davoli, L.; Belli, L.; Ferrari, G. Wireless Mesh Networking: An IoT-Oriented Perspective Survey on Relevant Technologies. Future Internet 2019, 11, 99.

[12] I. L'hadi, M. Rifai, Y. Salih Ali "An energy-efficient WSN-based traffic safety system," in Proc. ICICS, Irbid, 2014, pp. 1-6.

[13] T. Hoi Yan, et al., "ZigBee sensor network for advanced metering infrastructure," in Proc. ICCE, Las Vegas, NV, 2010, pp. 95-96.

[14] Senthil Kumar R, Ambika.M, Boomika P, Haritha N, Pavithra R, 2022. "Iot Based Smart Irrigation and Animal Prevention System Using Arduino" ESP Journal of Engineering & Technology Advancements 2(4): 1-4.

[15] D. Benyamina, A. Hafid and M. Gendreau, "Wireless Mesh Networks Design — A Survey," in IEEE Communications Surveys & Tutorials, vol. 14, no. 2, pp. 299-310, Second Quarter 2012, doi: 10.1109/SURV.2011.042711.00007.

[16] G. Suprianto and Wirawan, "Implementation of Distributed Consensus Algorithms for Wireless Sensor Network Using NodeMCU ESP8266," 2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS), Batu, Indonesia, 2018, pp. 192-196, doi: 10.1109/EECCIS.2018.8692952.

[17] Santos, L.; Costa, T.; Caldeira, J.M.L.P.; Soares, V.N.G.J. Performance Assessment of ESP8266 Wireless Mesh Networks. Information 2022, 13, 210.

[18] NurAzzurinAfifie, Adam Wong Yoon Khang, AbdShukur Bin Ja'afar, Ahmad Fairuz Bin Muhammad Amin "Evaluation Method of Mesh Protocol over ESP32 and ESP8266", 2021-12-15.