VISVESVARAYA TECHNOLOGICAL UNIVERSITY,
JNANASANGAMA, BELAGAVI - 590018

BLDEA's V.P. Dr. P.G. HALAKATTI COLLEGE OF ENGINEERINGAND TECHNOLOGY, VIJAYAPUR

DEPARTMENT OF
ELECTRONICS AND COMMUNICATIONENGINEERING

A Mini Project report on

"IOT-BASED COMMUNICATION SYSTEM FOR COLLEGE DEPARTMENT"

Submitted in partial fulfillment for the award of degree of Bachelor of Engineeringin Electronics and Communication Engineering

Submitted by

SAIRAJ INGALE	**2BL22EC410**
GOVIND HADPAD	**2BL22EC404**
BIRADAR VISHWANATH	**2BL22EC401**
SUPRITA I MATHAPATI	**2BL21EC102**

Under the Guidance of

Prof. B. K. Gudur

2023-24

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI

B.L.D.E. Association's

V.P Dr. P.G HALAKATTI COLLEGE OF ENGINEERING AND TECHNOLOGY, VIJAYAPUR

DEPARTMENT OF

ELECTRONICS ANDCOMMUNICATION ENGINEERING

CERTIFICATE

This is Certified that the Mini project work entitled **"IoT-Based Communication System For College Department"** carried out by **Sairaj Ingale, Govind Hadapad, Biradar Vishwanath, Suprita I Mathapati,** bonafide students of **V.P. Dr P.G Halakatti College of Engineering and Technology, Vijayapura** in partial fulfillment for the award of **Bachelor of Engineering in Electronics and Communication Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year 2023-2024. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The Mini project report has been approved as it satisfies the academic requirement in respect of Mini project work prescribed for the said degree.

GUIDE	H.O.D	PRINCIPAL
Prof. B. K. GUDUR	Dr. UMESH DIXIT	Dr. V. G. SANGAM

B.L.D.E. Association's

VACHANA PITAMAHA Dr. P.G. HALAKATTI COLLEGE OF ENGINEERING & TECHNOLOGY, VIJAYAPURA

DEPARTMENT OF ELECTRONICS & COMMINUCATION ENGINEERING

DECLARATION

We, students of Sixth semester B.E, at the department of Electronics & Communication Engineering, hereby declare that, the Mini Project entitled " **IoT-Based Communication System For College Department** ",embodies the report of our mini project work, carried out by us under the guidanceof **Prof. B. K. Gudur,** We also declare that, to the best of our knowledge and belief, the work reported here in does not form part of any other report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this by any student.

Place: - Vijayapura Date:-

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose consistent guidance and encouragement crowned our efforts with success. We consider it as our privilege to express the gratitude toall those who guided in the completion of our Mini Project.

First and foremost, we wish to express our profound gratitude to our respected Principal **Dr. V.G. Sangam, B.L.D.E. Association's VACHANA PITAMAHA Dr. P.G. HALAKATTI COLLEGE OF ENGINEERING & TECHNOLOGY, Vijayapura**, for providing us with a congenial environment towork in.

We would like to express our sincere thanks to **Dr. U D Dixit**, the HOD of **Electronics and Communication Engineering**, **B.L.D.E.Association's VACHANA PITAMAHA Dr. P.G. HALAKATTI COLLEGE OF ENGINEERING & TECHNOLOGY, Vijayapura**, for his continuous support and encouragement.

We are greatly indebted to our guide **Prof. Guide name**, Department of **Electronics and Communication Engineering**, **B.L.D.E. Association's VACHANA PITAMAHA Dr. P.G. HALAKATTI COLLEGE OF ENGINEERING & TECHNOLOGY, Vijayapura**, who took great interest in our work. He motivated us and guided us throughout the accomplishment of this goal. We expressour profound thanks for his meticulous guidance.

ABSTRACT

This project proposes the development and implementation of an IoT-based communicationsystem designed to streamline and enhance communication within college departments. The system enables quick and efficient summoning of key personnel, such as Heads of Departments (HODs) and teachers, through strategically placed IoT-enabled buttons or panels. The IoT communication system consists of several core components, including IoT buttons/panels, microcontrollers, communication modules, and a backend server. The buttons or panels, installed at key locations within the department, are labeled with the names or roles of the designated individuals. When a button is pressed, a signal is sent to the backend server via a communication module.

The server processes this signal and sends a notification to the mobile device of the designated person, providing details about the caller's location and any additional message if input capability is available. This project aims to improve communication efficiency, reduce the need for physical movement, and streamline the operational workflow within the college department. The system is designed to be user-friendly, allowing students and staff to quickly and easily summon faculty members when needed. Additionally, the system includes features for response tracking andstatus updates, ensuring accountability and transparency.

MQTT (Message Queuing Telemetry Transport) is a lightweight, publish/subscribe messaging protocol designed for low-bandwidth, high-latency, or unreliable networks. It is widely used in IoT applications to facilitate communication between devices. MQTT employs a central broker to manage message distribution, enhancing scalability and reliability. Its low overhead makes it ideal for resource-constrained devices, while features like Quality of Service (QoS) levels and support for TLS/SSL encryption ensure reliable and secure message delivery. Common applications include home automation, healthcare monitoring, and industrial control systems.

Chapter 1

INTRODUCTION

Internet of Things (IOT) is a concept where each device is assigned to an IP address and through that IP address anyone makes that device identifiable on internet. The mechanical and digital machines are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. Basically, it started as the "Internet of Computers." Research studies have forecast an explosive growth in the number of "things" or devices that will be connected to the Internet. The resulting network is called the "Internet of Things" (IOT).

The recent developments in technology which permit the use of wireless controlling environments like, Bluetooth and Wi-Fi that have enabled different devices to have capabilities of connecting with each other. Using a WIFI shield to act as a Micro web server for the Arduino which eliminates the need for wired connections between the Arduino board and computer which reduces cost and enables it to work as a standalone device. The Wi-Fi shield needs connection to the internet from a wireless router or wireless hotspot and this would act as the gateway for the Arduino to communicate with the internet.

MQTT is a standards-based messaging protocol, or set of rules, used for machine-to-machine communication. Smart sensors, wearables, and other Internet of Things (IoT) devices typically have to transmit and receive data over a resource-constrained network with limited bandwidth. These IoT devices use MQTT for data transmission, as it is easy to implement and can communicate IoT data efficiently. MQTT supports messaging between devices to the cloud and the cloud to the device.

An embedded system is a specialized computing system designed to perform dedicated functions or tasks within a larger mechanical or electrical system. Unlike general-purpose computers, embedded systems are tailored for specific applications, often operating in real-time environments where reliability and efficiency are crucial. These systems integrate both hardware and software components, with the hardware typically comprising microcontrollers or microprocessors, sensors, and actuators, while the software is custom-built to execute precise functions.

Embedded systems are ubiquitous in modern technology, found in everyday devices such as smartphones, home appliances, automotive control systems, medical equipment, and industrial machines. They are essential for enabling automation, improving functionality, and enhancing the performance of various devices, making them integral to the advancement of technology and the optimization of numerous industrial and consumer applications.

NodeMCU is an open-source IoT development platform that integrates the ESP8266 Wi-Fi module, providing a robust and versatile solution for building connected devices. Known for its ease of use and flexibility, NodeMCU combines both hardware and firmware components to facilitate rapid development and deployment of IoT applications. The hardware is based on the ESP-12 module, featuring a powerful 32-bit microcontroller with built-in Wi-Fi connectivity, digital I/O pins, and an analog input. The firmware, which can be programmed using the Lua scripting language or the Arduino IDE, offers a user-friendly environment for developers of all skill levels. NodeMCU's capabilities make it ideal for a wide range of applications, from home automation and remote monitoring to prototyping and educational projects. Its affordability, extensive community support, and comprehensive feature set make NodeMCU a popular choice for both hobbyists and professionals in the IoT space.

Chapter 2

LITERATURE REVIEW

"Performances analysis of microcontroller used in IOT Technology", by K. Swathi, T. Uday Sandeep, A. Roja Ramani [1].

Internet of Things, a revolutionary invention, is always transforming in to some new kind of Hardware and Software making it unavoidable for anyone. An IOT microcontroller or development board is a prototyping solution that features low-power processors which support various programming environments, collect sensor data using firmware and transfer it to an on-premises or cloud-based server. Entering the era of Internet of Things, the use of small, cheap and flexible computer hardware that allow end-user programming become present. In this paper we provide an overview of the state-of-the-art hardware available and explores performance of different microcontrollers, they are Arduino, node MCU, Raspberry Pi. We present IOT device characteristics, features, applications and comparisons between different boards. The users access the website on the server address obtained from Raspberry Pi.

"A Hybrid Transmission Technique for IOT Networking Environment, by Hangki Joh, Inhwan Yang, Intae Ryoo. [2]

Recently, there comes quite a many common Internet of Things (IOT) devices that are using Bluetooth Low Energy (BLE) module for saving energies. General example includes smart watch and location-based service providing devices that are connecting to external or remote devices by using BLE. This paper proposes an original scheme that immediately connect various kinds of IOT devices and at the same time guarantee data transmission speed for them by efficiently utilizing the characteristics of Bluetooth and Wi-Fi technologies.

"A Wi-Fi P2P Communication Platform between Wireless LAN Memory Cards: Prototype Implementation and Performance Evaluation. [3]

Wi-Fi Peer to Peer (P2P), commercially known as Wi-Fi Direct, is a recent industry standard that allows user devices to communicate with each other without requiring a Wi-Fi access point or Internet connectivity. Offering promising solutions for secure and high-

throughput device to device communication over moderately high range, Wi-FiP2P can potentially revolutionize M2M communication and Internet of Things (IOT) towards an all-connected wireless ecosystem. Secure Digital (SD) memory cards with built-in processing unit and Wireless LAN chipset can serve as a small standalone communication terminal which on installation of a communication protocol stack and a data-sharing application may enable connection establishment and data communication between memory cards. Such architecture may find interesting applications with manifold advantages for Device-to-device data sharing in public gatherings without requiring cellular network or WLAN access points. In this paper, we develop architecture to establish Wi-Fi P2P connection between SD memory cards and enable automatic data sharing by proximity based device-to-device communication between mobile user equipments. The prototype does not use any of the resources of host device barring power. The key contribution of this paper is the development of a standalone prototype of our proposition on off-the-shelf WLAN SD memory cards and its experimental performance evaluation.

“MQTT protocol specification. [Online] Available:"http://public.dhe. ibm.com/software/dw/web services/ws-mqtt/mqtt-v3r1.html".

RA Atmoko, R Riantini, and MK Hasin. (2017) "IOT real time data acquisition using MQTT protocol" Journal of. Physics.: 853

ASIP programming model." computer communication: 1-15 [5] El Kam chou chi H. and El Shafee [4]

A. (2012) "Design and prototype implementation of SMS based home automation system." International Internet of Things (IOT) allow connection among devices using internet with the ability to gather and exchange data. These devices are usually attached with micro-controllers like Arduino, sensors, actuators and internet connectivity. In this context, Message Queuing Telemetry Transport protocol (MQTT) plays an important role to exchange the data or information between the devices in IOT without knowing the identities of each other. This paper presents different service models for communication in Internet of Things (IOT). Model A present’s use of serial USB as transmission medium while Model B uses the Message Queuing Telemetry Transport protocol (MQTT) which deploy a Wi-Fi module (ESP8266-12) to connect the system to internet. For communication, concept of publisher and subscriber is used. Messages are published or subscribed with the help of a

broker or server. This agent is in charge of dispersing messages to intent clients depending on the choice of the topic of a message. Broker in MQTT is also called server. Some brokers used in MQTT are: - Mosquitto, Adafruit, hiveMQ.

"Internet of Things for Smart Campus Management". [5]

- Authors: M. M. Hassan, Y. S. Cho

- Journal: IEEE Access

- Summary: This paper discusses the application of IoT technologies for managing campus facilities, including energy management, security, and smart classrooms.

"IoT-Based Communication Systems for Educational Institutions: A Review". [6]

- Authors: J. Zhang, L. Zhao

- Journal: International Journal of Information Management

- Summary: A comprehensive review of IoT-based communication systems, highlighting their benefits and challenges in educational settings.

"Smart Campus: An IoT-Enabled Model for Enhancing Communication and Learning". [7]

- Authors: A. S. Patel, R. Kumar

- Journal: Sensors

- Summary: This paper explores a model for integrating IoT with campus communication systems to improve learning outcomes and administrative efficiency.

"Security and Privacy Issues in IoT-Based Campus Environments".[8]

- Authors: C. Li, X. Chen

- Journal: Computers & Security

- Summary: Analyzes the security and privacy concerns associated with IoT deployments on campus and proposes mitigation strategies.

"Applications of IoT in Education: A Review and Future Directions". [9]

- Authors: F. Alshamrani, H. Al-Ghamdi

- Journal: Educational Technology Research and Development

- Summary: Reviews various applications of IoT in education, including communication systems and classroom management.

"Energy-Efficient IoT Systems for Smart Campus Management". [10]

- Authors: N. G. Chen, H. Xie

- Journal: Energy Reports

- Summary: Discusses energy-efficient solutions for IoT systems in campus settings, focusing on reducing operational costs and environmental impact.

Chapter 3

IOT AND MQTT

3.1 IOT

The Internet of Things (IoT) is a network of physical objects that are embedded with sensors and other technologies to connect and exchange data with other devices and systems over the internet. IoT devices can include anything with a sensor that is assigned a unique identifier (UID), such as household objects, industrial tools, vehicles, and even living beings. The primary goal of the IoT is to create self-reporting devices that can communicate with each other and users in real time.

3.1.1 What is IOT (Internet of Thing)?

IOT (Internet of things) IOT as a term has evolved long way because of convergence of multiple technologies, machine learning, embedded systems, and commodity sensors. IOT is a system of interconnected devices assigned a UIDS, enabling data transfer and control of devices over a network. It reduced the necessity of actual interaction to control a device. IOT is an advanced automation and analytics system which exploits networking, sensing, big data, and artificial intelligence technology to deliver complete systems for a product or service. These systems allow greater transparency, control, and performance when applied to any industry or system.

3.1.2 Features of IOT

Intelligence

IOT comes with the combination of algorithms and computation, software & hardware that makes it smart. Ambient intelligence in IOT enhances its capabilities which facilitate the things to respond in an intelligent way to a particular situation and supports them in carrying out specific tasks. Despite all the popularity of smart technologies, intelligence in IOT is only concerned as a means of interaction between devices, while user and device interaction are achieved by standard input methods and graphical user interface.

Connectivity

empowers the Internet of Things by bringing together everyday objects. Connectivity of these objects is pivotal because simple object level interactions contribute towards collective intelligence in the IOT network. It enables network accessibility and compatibility in the things. With this connectivity, new market opportunities for the Internet of things can be created by the networking of smart things and applications.

Dynamic

Nature The primary activity of Internet of Things is to collect data from its environment; this is achieved with the dynamic changes that take place around the devices. The state of these devices changes dynamically, example sleeping and waking up, connected and/or disconnected as well as the context of devices including temperature, location and speed. In addition to the state of the device, the number of devices also changes dynamically with a person, place and time.

Enormous Scale

The number of devices that need to be managed and that communicate with each other will be much larger than the devices connected to the current Internet. The management of data generated from these devices and their interpretation for application purposes becomes more critical. Gartner (2015) confirms the enormous scale of IOT in the estimated report where it stated that 5.5 million new things will get connected every day and 6.4 billion connected things will be in use worldwide in 2016, which is up by 30 percent from 2015. The report also forecasts that the number of connected devices will reach 20.8 billion by 2020.

Sensing

IOT wouldn’t be possible without sensors that will detect or measure any changes in the environment to generate data that can report on their status or even interact with the environment. Sensing technologies provide the means to create capabilities that reflect a true awareness of the physical world and the people in it. The sensing information is simply the analog input from the physical world, but it can provide a rich understanding of our complex world.

Heterogeneity

Heterogeneity in Internet of Things as one of the key characteristics. Devices in IOT are based on different hardware platforms and networks and can interact with other devices or service platforms through different networks. IOT architecture should support direct network connectivity between heterogeneous networks. The key design requirements for heterogeneous things and their environments in IOT are scalabilities, modularity, extensibility, and interoperability.

Security

IOT devices are naturally vulnerable to security threats. As we gain efficiencies, novel experiences, and other benefits from the IOT, it would be a mistake to forget about security concerns associated with it. There is a high level of transparency and privacy issues with IOT. It is important to secure the endpoints, the networks, and the data that is transferred across all of it means creating a security paradigm.

3.1.3 Advantages of IOT

Communication

IOT encourages the communication between devices, also famously known as Machine-to-Machine (M2M) communication. Because of this, the physical devices are able to stay connected and hence the total transparency is available with lesser inefficiencies and greater quality.

Automation and Control

Due to physical objects getting connected and controlled digitally and centrally with wireless infrastructure, there is a large amount of automation and control in the workings. Without human intervention, the machines are able to communicate with each other leading to faster and timely output.

Information

It is obvious that having more information helps making better decisions. Whether it is mundane decisions as needing to know what to buy at the grocery store or if your company has enough widgets and supplies, knowledge is power and more knowledge is better.

Monitor

The second most obvious advantage of IOT is monitoring. Knowing the exact quantity of supplies or the air quality in your home, can further provide more information that could not have previously been collected easily. For instance, knowing that you are low on milk or printer ink could save you another trip to the store soon. Furthermore, monitoring the expiration of products can and will improve safety.

Time

As hinted in the previous examples, the amount of time saved because of IOT could be quite large. And in today's modern life, we all could use more time.

Money

The biggest advantage of IOT is saving money. If the price of the tagging and monitoring equipment is less than the amount of money saved, then the Internet of Things will be very widely adopted. IOT fundamentally proves to be very helpful to people in their daily routines by making the appliances communicate to each other in an effective manner thereby saving and conserving energy and cost. Allowing the data to be communicated and shared between devices and then translating it into our required way, it makes our systems efficient.

Efficient and Saves Time

The machine-to-machine interaction provides better efficiency, hence; accurate results can be obtained fast. This results in saving valuable time. Instead of repeating the same tasks every day, it enables people to do other creative jobs.

Better Quality of Life

All the applications of this technology culminate in increased comfort, convenience, and better management, thereby improving the quality of life.

3.1.4 Disadvantages of IOT

Compatibility

Currently, there is no international standard of compatibility for the tagging and monitoring equipment. I believe this disadvantage is the easiest to overcome. The manufacturing

companies of this equipment just need to agree to a standard, such as Bluetooth, USB, etc. This is nothing new or innovative needed.

Complexity

As with all complex systems, there are more opportunities of failure. With the Internet of Things, failures could skyrocket. For instance, let's say that both you and your spouse each get a message saying that your milk has expired, and both of you stop at a store on your way home, and you both purchase milk. As a result, you and your spouse have purchased twice the amount that you both need. Or maybe a bug in the software ends up automatically ordering a new ink cartridge for your printer every hour for a few days, or at least after each power failure, when you only need a single replacement.

Privacy/Security

With all this IOT data being transmitted, the risk of losing privacy increases. For instance, how well encrypted will the data be kept and transmitted with? Do you want your neighbors or employers to know what medications that you are taking or your financial situation?

Safety

Imagine if a notorious hacker changes your prescription. Or if a store automatically ships you an equivalent product that you are allergic to, or a flavor that you do not like, or a product that is already expired. As a result, safety is ultimately in the hands of the consumer to verify any and all automation. As all the household appliances, industrial machinery, public sector services like water supply and transport, and many other devices all are connected to the Internet, a lot of information is available on it. This information is prone to attack by hackers. It would be very disastrous if private and confidential information is accessed by unauthorized intruders.

3.2 MQTT

3.2.1 What is MQTT?

MQTT (originally an initialism of MQ Telemetry Transport) is a lightweight, publish-subscribe, machine to machine network protocol for message queue/message queuing service. It is designed for connections with remote locations that have devices with resource

constraints or limited network bandwidth, such as in the Internet of Things (IoT). It must run over a transport protocol that provides ordered, lossless, bi-directional connections—typically, TCP/IP. It is an open OASIS standard and an ISO recommendation (ISO/IEC 20922).

Any device that uses the TCP/IP network protocol and has software that implements MQTT client functionality can be called an MQTT client. MQTT is designed to work on top of the TCP/IP protocol, so any device that speaks TCP/IP and implements the MQTT protocol can be an MQTT client. The client implementation of the MQTT protocol is straightforward and streamlined, making it ideally suited for small devices.

3.2.2 Working of MQTT

MQTT (Message Queuing Telemetry Transport) is a lightweight, publish-subscribe messaging protocol optimized for constrained devices and low-bandwidth networks, making it a cornerstone in the Internet of Things (IoT) landscape. As depicted in Figure 3.1, the MQTT ecosystem centres around clients and a broker. Clients, acting as both publishers and subscribers, initiate communication by establishing a connection with the broker. Once connected, a client can disseminate data to a specific topic, akin to broadcasting a message on a particular channel. Other clients can subscribe to these topics, acting as receivers of relevant information. The broker, a central hub, efficiently manages the distribution of published messages to all interested subscribers. This decentralized publish-subscribe paradigm, coupled with MQTT's inherent efficiency, reliability, and scalability, positions it as a robust solution for IoT deployments demanding seamless and dependable communication among a multitude of devices.

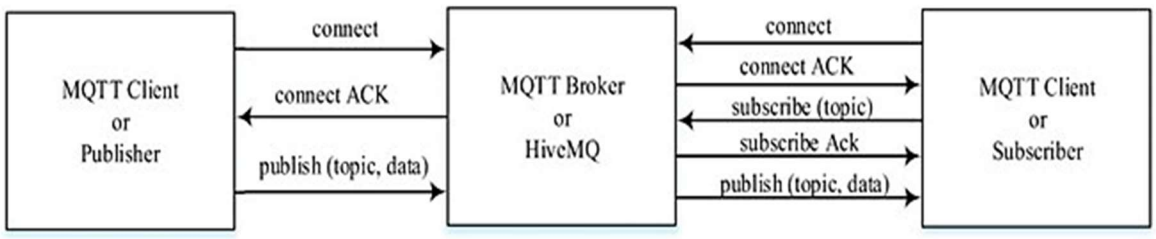

Figure 3.1: Working of MQTT

MQTT's core strength lies in its ability to decouple publishers and subscribers, facilitating flexible and scalable communication architectures. This decoupling enables devices and applications to independently join and leave the network without affecting overall system operation. Moreover, MQTT's publish-subscribe model ensures efficient message delivery by targeting only interested subscribers, minimizing network traffic and optimizing resource utilization.

Beyond its core functionalities, MQTT's flexibility is evident in its ability to support various Quality of Service (QoS) levels, ensuring different degrees of message reliability to accommodate diverse application requirements. For instance, in critical IoT applications where data loss is unacceptable, a higher QoS level can be employed to guarantee message delivery. Conversely, in scenarios where occasional data loss is tolerable, a lower QoS level can be selected to optimize network efficiency. MQTT's adaptability to varying QoS needs underscores its versatility and suitability for a wide range of IoT use cases.

Furthermore, MQTT's lightweight nature and minimal overhead render it ideal for battery-powered devices and environments with intermittent connectivity, such as those encountered in remote IoT deployments. MQTT's efficient message format and reduced protocol overhead minimize power consumption and data transfer, extending battery life and enhancing the overall performance of IoT devices operating in challenging conditions.

Chapter 4

COMPONENTS AND HARDWARE REQUIREMENT

4.1 NODEMCU

NodeMCU (Node Microcontroller Unit) is a low-cost open source IOT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module. Later, support for the ESP32 32-bit MCU was added.

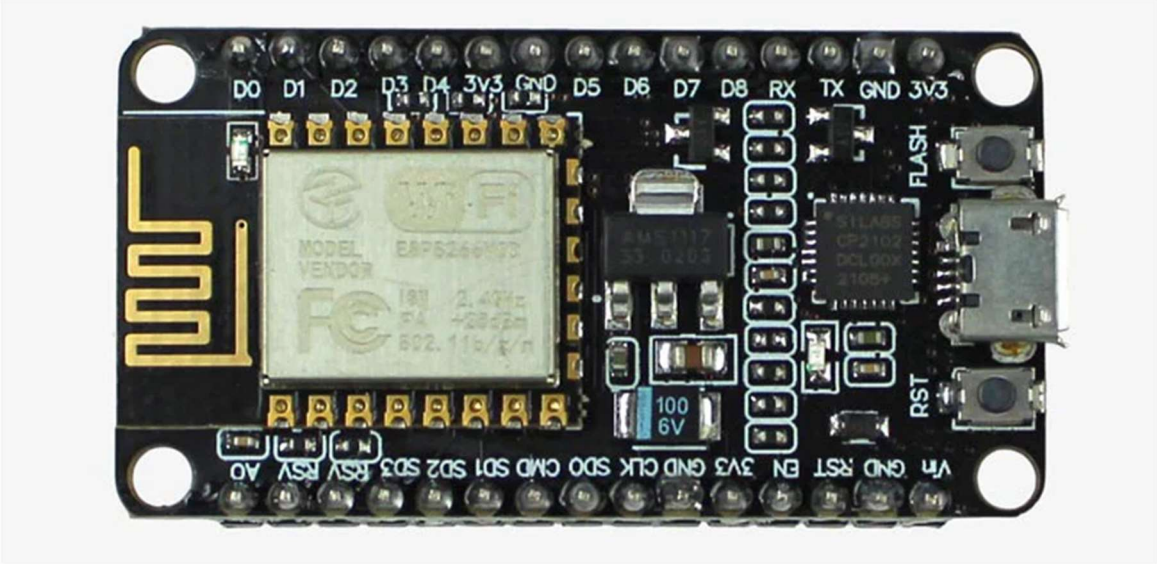

Figure 4.1: NodeMCU ESP8266 Board

Node MCU is an open-source firmware for which open-source prototyping board designs are available. The name “Node MCU” combines “node” and “MCU” (micro-controller unit). The term “Node MCU” strictly speaking refers to the firmware rather than the associated development kits. Both the firmware and prototyping board designs are open source. The firmware uses the Lua scripting language. The firmware is based on the e Lua project and built on the Es press if Non-OS SDK for ESP8266.

It uses many open-source projects, such as luacjson and SPIFFS. Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented.

The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards.

The design was initially based on the ESP-12 module of the ESP8266, which is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IOT applications.

4.1.2 Pin Configuration of Node MCU Development Board

This module provides an access to the GPIO subsystem. All the access is based on I/O index number of Node MCU kits, not the internal GPIO pins.

Table 4.1: NodeMCU index – GPIO mapping

PIN NAME ON NODE MCU DEVELOPMENT KIT	ESP8266 INTERNAL GPIO PIN NUMBER	PIN NAME ON NODE MCU DEVELOPMENT KIT	ESP8266 INTERNAL GPIO PIN NUMBER
0 [*]	GPIO16	**7**	GPIO13
1	GPIO5	**8**	GPIO15
2	GPIO4	**9**	GPIO3
3	GPIO0	**10**	GPIO1
4	GPIO2	**11**	GPIO9
5	GPIO14	**12**	GPIO10
6	GPIO12		

The ESP8266 Node MCU has total 30 pins that interface it to the outside world. The pins are grouped by their functionality as:

Power pins: There are four power pins viz. one VIN pin & three 3.3V pins. The VIN pin can be used to directly supply the ESP8266 and its peripherals if you have a regulated 5V voltage source. The 3.3V pins are the output of an on-board voltage regulator. These pins can be used to supply power to external components.

GND: is a ground pin of ESP8266 Node MCU development board.

12 IC Pins: are used to hook up all sorts of I2C sensors and peripherals in your project. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

GPIO Pins: ESP8266 Node MCU has 17 GPIO pins which can be assigned to various functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light, and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.

ADC Channel: The Node MCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC viz. Testing power supply voltage of VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.

UART Pins: ESP8266 Node MCU has 2 UART interfaces, i.e. UART0 and UART1, which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. It supports fluid control. However, UART1 (TXD1 pin) features only data transmit signal so, `it is usually used for printing log.

SPI Pins: ESP8266 features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the following general-purposes features 4 timing modes of the SPI format transfer, Up to 80 MHz and the divided clocks of 80 MHz, Up to 64-Byte FIFO

SDIO Pins: ESP8266 features Secure Digital Input/output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported. PWM Pins: The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 up to 10000 us, i.e., between 100 Hz and 1 kHz.

Control Pins: are used to control ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin

1. **EN pin –** The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
2. **RST pin –** RST pin is used to reset the ESP8266 chip.
3. **WAKE pin –** Wake pin is used to wake the chip from deep-sleep.

4.1.3 Installation of Node MCU

Mostly these days devices download and install drivers on their own, automatically. Windows doesn't know how to talk to the USB driver on the Node MCU so it can't figure out that the board is a Node MCU and proceed normally. Node MCU Amica is an ESP8266 Wi-Fi module-based development board. It has got Micro USB slot that can directly be connected to the computer or other USB host devices. Ti has got 15X2 header pins and a Micro USB slot, the headers can be mounted on a breadboard and Micro USB slot is to establish connection to USB host device. It has CP2120 USB to serial converter. To install CP2120 (USB to serial converter), user is needed to download the driver for the same. Once user downloads drivers as per its respective operating system, the system establishes connection to Node MCU. The user needs to node down the COM post allotted to newly connected USB device (Node MCU) from device manager of the system. This com port number will be required while using Node MCU Amica. As the CP2120 driver has been installed, the Node MCU can be programmed using Arduino IDE software by coding in embedded C. this requires ESP8266 board installation in Arduino IDE from board manager and assigning communication port.

4.1.4 Advantages of Node MCU

1. Low cost, the Node MCU is less costly compared to any other IOT based device.
2. Node MCU has Arduino Like hardware I/O. It is becoming very popular in these days that Arduino IDE has extended their software to work in the field of ESP 8266 Field module version.
3. Node MCU has easily configurable network API.
4. Integrated support for Wi-Fi network: ESP 8266 is incorporated in Node MCU, which is an easily accessible Wi-Fi module.
5. Reduced size of board.
6. Low power consumption.

4.1.5 Disadvantages of Node MCU

1. The operation of the circuit depends on the working internet connection. If the working internet connection is not available, then it will not run.
2. Node MCU also depends on the free server provided by the third party, if the free server is not working then it will not run.

4.2 Small Piezoelectric Buzzer - 5V Passive Buzzer

This is Small PCB Mountable 5V Passive Buzzer. It is great to add Audio Alert to your electronic designs. It operates on 5V supply, uses a coil element to generate an audible tone. This is a Passive buzzer and not an Active Buzzer, meaning you need external circuits to make the buzzer beep. Connecting it to power directly will not work. You can check out Active buzzer for direct beeping.

Figure 4.2: Small Piezoelectric Buzzer

It operates on 5V supply, uses a coil element to generate an audible tone. Important: This is a Passive buzzer and not an Active Buzzer, meaning you need external circuits to make the buzzer beep. Connecting it to power directly will not work. You can check out Active buzzer for direct beeping.

Features:

1. Input Voltage (Max.): 5V
2. Resistance: 42 Ω
3. Resonance Frequency: 2048 Hz
4. Body Size: 12 x 8mm
5. Pin Pitch: 6mm
6. External Material: Plastic & Black color

4.3 The 4 x 4 Matrix Keypad

The 4x4 matrix keypad is a simple mechanism that resembles the numeric input on your computer keyboard, except that it has an additional '*,' '#' and 4 other auxiliary buttons that can be used for various functions in the application. The keypad is usually made of plastic materials and is relatively cheap compared to touch screen displays.

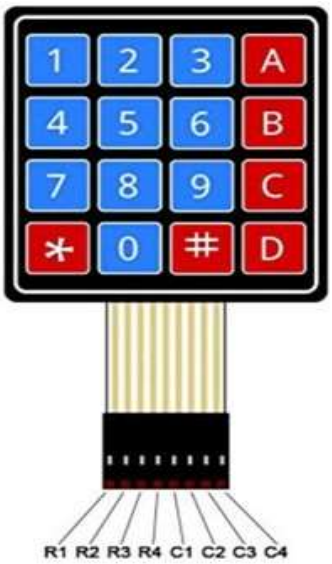

Figure 4.3: 4x4 Matrix Keypad

A 4x4 matrix keypad can be implemented separately or within the physical product itself, such as a security access controller, where it is used for PIN identifications. Either way, the mechanism of the mechanical keypad remains the same when hardware and firmware designers are concerned.

If you've never designed with a 4x4 mechanical keypad, the best way to visualize the internal mechanism is a matrix of push-button switches. A 4x4 keypad has a total of 8 connections, where 4 of them are connected to the column and the remaining rows of the matrix of switches.

When an individual button is pressed, a connection is established between one of the rows and columns. The microcontroller then deciphers the physical button based on the index of the row and column that is activated.

4.4 Jumper Wires

Generally, jumpers are tiny metal connectors used to close or open a circuit part. They have two or more connection points, which regulate an electrical circuit board. Their function is to configure the settings for computer peripherals, like the motherboard. Suppose your motherboard supported intrusion detection. A jumper can be set to enable or disable it.

Jumper wires are electrical wires with connector pins at each end. They are used to connect two points in a circuit without soldering. You can use jumper wires to modify a circuit or diagnose problems in a circuit. Further, they are best used to bypass a part of the circuit that does not contain a resistor and is suspected to be bad.

This includes a stretch of wire or a switch. Suppose all the fuses are good and the component is not receiving power; find the circuit switch. Then, bypass the switch with the jumper wire.

4.4.1 Types of Jumper Wires

Jumper wires come in three versions as shown in figure 4.5

1. Male-to-male jumper
2. Male-to-female jumper
3. Female to Female jumper

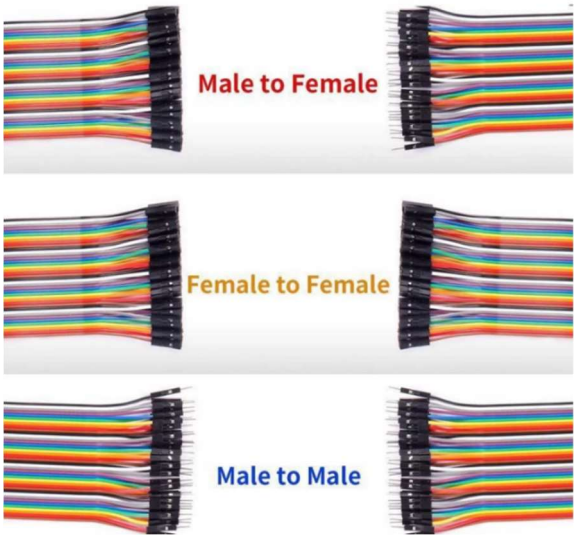

Figure 4.4: Jumper Wires

4.5 Light-Emitting Diode (LED)

A light-emitting diode (LED) is a semiconductor device that emits light when an electric current flows through it. When current passes through an LED, the electrons recombine with holes emitting light in the process. LEDs allow the current to flow in the forward direction and blocks the current in the reverse direction.

Light-emitting diodes are heavily doped p-n junctions. Based on the semiconductor material used and the amount of doping, an LED will emit coloured light at a particular spectral wavelength when forward biased. As shown in the figure, an LED is encapsulated with a transparent cover so that emitted light can come out.

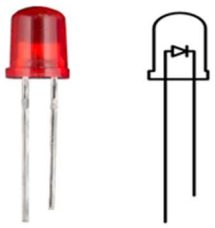

Figure 4.5: LED

4.5.1 Uses of LED

LEDs find applications in various fields, including optical communication, alarm and security systems, remote-controlled operations, robotics, etc. It finds usage in many areas because of its long-lasting capability, low power requirements, swift response time, and fast switching capabilities. Below are a few standards LED uses:

- Used for TV back-lighting.
- Used in displays.
- Used in Automotives.
- LEDs used in the dimming of lights.

4.5.2 Types of LED

Below is the list of different types of LED that are designed using semiconductors:

- Miniature LEDs.
- High-Power LEDs.
- Flash LED and Bi and Tri-Color.

Chapter 5

METHODOLOGY

5.1MQTT Broker Server

The MQTT broker is a server that receives messages from publishers and delivers them to subscribers based on their topic subscriptions. It manages client connections, handles subscriptions and un-subscriptions and ensures message delivery according to the specified Quality of Service (QoS) levels.

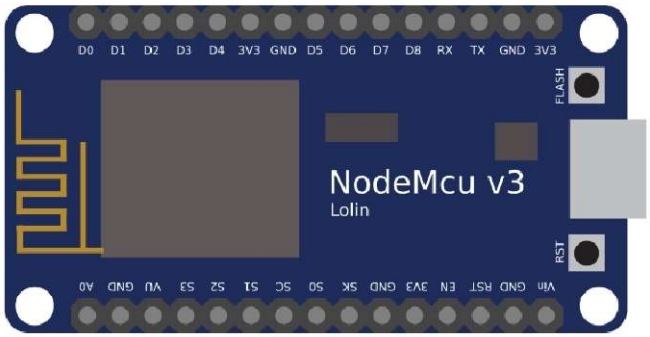

Figure 5.1: MQTT Broker Server

Role: The MQTT broker is the central hub through which all messages are sent. It receives messages from publishers and routes them to the appropriate subscribers.

Communication: Publishers send messages on specific topics, and subscribers receive messages from the topics they are subscribed to.

Setting up an MQTT broker server involves choosing the right broker, configuring it for your specific needs, ensuring security, and monitoring its performance.

5.2 MQTT Clients

MQTT clients can be publishers, subscribers, or both. Publishers send messages to the MQTT broker, while subscribers receive messages from the broker. Clients can be any device or application that can establish a connection to the MQTT broker using the MQTT protocol, such as IoT devices, mobile applications, or other servers.

An MQTT client can be any device, ranging from a tiny microcontroller to a gigantic server, that runs an MQTT library and connects to an MQTT broker over a network. An MQTT

client library is a software module or package that implements the MQTT protocol and provides an interface for devices or applications to communicate using MQTT. These libraries make it easier to add MQTT support to applications or devices without implementing the protocol from scratch.

5.3 MQTT Publisher

The MQTT Publisher destination publishes messages to a topic on an MQTT broker. The destination functions as an MQTT client that publishes messages, writing each record as a message. For information about supported versions, see Supported Systems and Versions.

When you configure the destination, you specify the information needed to connect to the MQTT broker. You must define connection credentials when the MQTT broker requires a user name and password.

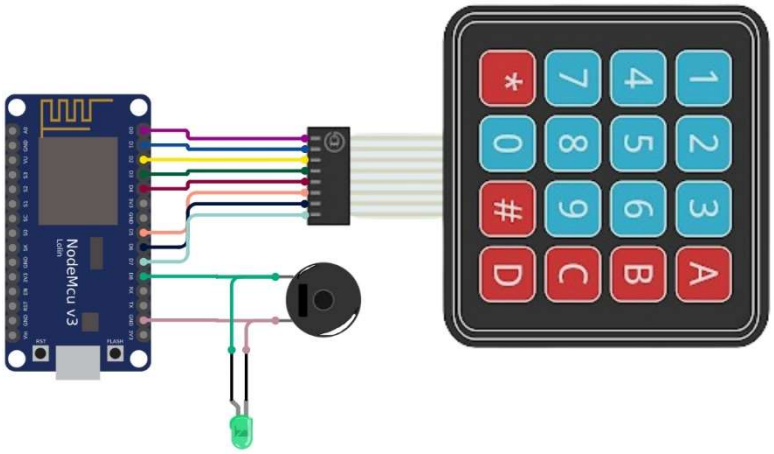

Figure 5.2: MQTT Publisher

You can also configure SSL/TLS properties, including default transport protocols and cipher suites.

The stage supports high availability MQTT clusters. For a cluster without a load balancer, you configure a list of brokers in the cluster. After losing a connection to a broker, the stage connects to the next available broker in the list.

You specify the topic on the MQTT broker that the destination delivers the message to. You also configure the quality-of-service level and the persistence mechanism that the destination uses to enable reliable messaging.

5.4 MQTT Subscriber

The MQTT Subscriber origin subscribes to topics on an MQTT broker to read messages from the broker. The origin functions as an MQTT client that receives messages, generating a record for each message. For information about supported versions, see Supported Systems and Versions.

When you configure the origin, you specify the information needed to connect to the MQTT broker. You must define connection credentials when the MQTT broker requires a user name and password. You can also use a connection to configure the origin.

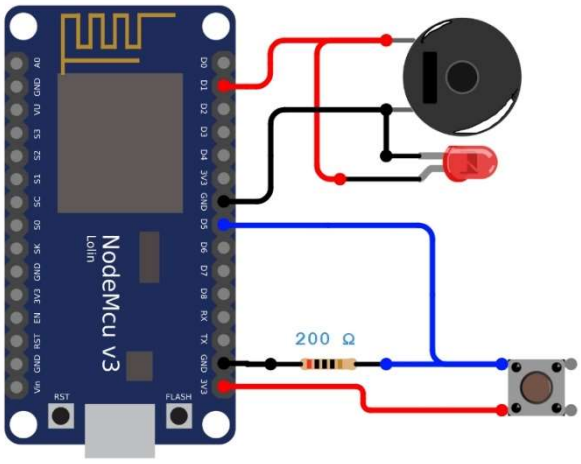

Figure 5.3: MQTT Subscriber

You can also configure SSL/TLS properties, including default transport protocols and cipher suites.

The stage supports high availability MQTT clusters. For a cluster without a load balancer, you configure a list of brokers in the cluster. After losing a connection to a broker, the stage connects to the next available broker in the list.

You specify one or more topics on the MQTT broker that the origin subscribes to. The origin includes the name of the originating topic for each record in a record header attribute.

You also configure the quality-of-service level and the persistence mechanism that the origin uses to enable reliable messaging. The MQTT Subscriber origin subscribes to topics on an MQTT broker to read messages from the broker. The origin functions as an MQTT

client that receives messages, generating a record for each message. For information about supported versions, see Supported Systems and Versions.

When you configure the origin, you specify the information needed to connect to the MQTT broker. You must define connection credentials when the MQTT broker requires a user name and password. You can also use a connection to configure the origin.

You can also configure SSL/TLS properties, including default transport protocols and cipher suites.

The stage supports high availability MQTT clusters. For a cluster without a load balancer, you configure a list of brokers in the cluster. After losing a connection to a broker, the stage connects to the next available broker in the list.

You specify one or more topics on the MQTT broker that the origin subscribes to. The origin includes the name of the originating topic for each record in a record header attribute.

You also configure the quality-of-service level and the persistence mechanism that the origin uses to enable reliable messaging.

Chapter 6

RESULTS AND DISCUSSION

The experimental model was made according to the circuit diagram and the results were as expected. The IoT-based communication system significantly improved communication efficiency within the college department, reducing delays and physical movement, and enhancing workflow. Users reported high satisfaction due to its ease of use, reliability, and real-time notifications. The system also increased accountability and transparency through request tracking. It proved scalable and adaptable, with successful deployment indicating potential for broader use. Cost savings were realized through reduced inefficiencies and lower maintenance costs. Enhanced collaboration and secure, energy-efficient design further supported sustainability and long-term viability. Overall, the system greatly improved departmental communication and operations.

Advantages:

- Enhanced Communication Efficiency.
- Improved Workflow.
- User-Friendly Interface.
- Cost-Effective
- Scalability
- Reliability
- Security
- Reduced Physical Movement
- Sustainability
- Enhanced Collaboration.
- Simple interface.

Applications:

- Emergency Alerts
- Classroom Management
- Administrative Coordination

- Facility Management
- Student Services
- Library Services
- Laboratory Management
- Campus Security
- Feedback and Suggestions
- Meeting Coordination
- Research Collaboration.

Significant reduction in delays and physical movement, enhancing overall workflow. High levels of satisfaction due to ease of use, reliability, and real-time notifications. Increased through effective request tracking. Secure, energy-efficient design supporting long-term viability. Increased through effective request tracking.

Describe the core components used in the system (sensors, controllers, communication modules, etc.). Detail the network design and protocols used for communication. Specific feedback from users about their experience with the system. Information on training provided to users for effective utilization. Information on training provided to users for effective utilization. Metrics indicating the reliability of the system (e.g., uptime, error rates).

Chapter 7

FUTURE SCOPE AND CONCLUSION

7.1 CONCLUSION

It is evident from this project work that an individual control classroom automation system can be cheaply made from low-cost locally available components and can be used to control multifarious classroom appliances ranging from the security lamps, the television to the air conditioning system and even the entire classroom lighting system and better still, the components required are so small and few that they can be packaged into a small inconspicuous container.

The designed classroom automation system was tested a few times and certified to control different home appliances used in the lighting system, air conditioning system, projectors, computer systems and many more. Hence, this system is scalable and flexible.

7.2 FUTURE SCOPE

The next phase for the classroom automation market will occur based on a few key improvements in the technology available in automation, such as improvements in wireless automation solutions as well as lowering of price points as the market begins to accept home automaton usage in larger volumes. Some trends that we foresee for this phase of the industry are big companies like Philips, Siemens & Schneider will eventually bring out mass-market automation products with appealing user interface but at a lower price point today, and more people will be able to afford the products.

Future Prospects:

- **Potential Expansions**: Ideas for expanding the system to other departments or institutions.
- **Planned Upgrades**: Any planned upgrades or additional features to enhance the system further.

If you provide more specific details or areas you want to focus on, I can tailor the information to better suit your needs.

REFERENCES

[1] “Performances analysis of microcontroller used in IOT Technology”, by K. Swathi, T. Uday Sandeep, A. Roja Ramani.

[2] “A Hybrid Transmission Technique for IOT Networking Environment, by Hangki Joh, Inhwan Yang, Intae Ryoo.

[3] “A Wi-Fi P2P Communication Platform between Wireless LAN Memory Cards: Prototype Implementation and Performance Evaluation.

“MQTT protocol specification. [Online] Available:"http://public.dhe. ibm.com/software/dw/web services/ws-mqtt/mqtt-v3r1.html".

RA Atmoko, R Riantini, and MK Hasin. (2017) "IOT real time data acquisition using MQTT protocol" Journal of. Physics.: 853

[4] “ASIP programming model” computer communication: 1-15 [5] El Kam chou chi H. and El Shafee.

[5] "Internet of Things for Smart Campus Management". Authors: M. M. Hassan, Y. S. Cho . Journal: IEEE Access.

[6] "IoT-Based Communication Systems for Educational Institutions: A Review". Authors: J. Zhang, L. Zhao. Journal: International Journal of Information Management.

[7] "Smart Campus: An IoT-Enabled Model for Enhancing Communication and Learning". Authors: A. S. Patel, R. Kumar. Journal: Sensors.

[8] "Applications of IoT in Education: A Review and Future Directions". Authors: F. Alshamrani, H. Al-Ghamdi. Journal: Educational Technology Research and Development.

[9] "Energy-Efficient IoT Systems for Smart Campus Management". Authors: N. G. Chen, H. Xie. Journal: Energy Reports.

[10] “Performances analysis of microcontroller used in IOT Technology”, by K. Swathi, T. Uday Sandeep, A. Roja Ramani.

****