**HOME AUTOMATION USING ESP8266**

**Abhijit Dutta**

**Department of Computer Science,**

**Kanchrapara College, Kanchrapara,**

**West Bengal ,India**

[**ad3052056@gmail.com**](mailto:ad3052056@gmail.com)

**Rohit Ghosh**

**Department of Computer Science,**

**Kanchrapara College, Kanchrapara,**

**West Bengal ,India**

[**rohitghosh9290@gmail.com**](mailto:rohitghosh9290@gmail.com)

**ABSTRACT**

Internet of Things is composed of things that have unique identities and are connected to each other over internet. It is simply connecting and monitoring various devices and sensors through Internet. This paved the way for home automation and monitoring which makes human life more comfortable and secured. This paper describes the overall notion of the IOT based sensing systems and monitoring systems for implementing an automated home. The proposed prototype uses Node MCU board with internet being remotely controlled by Android/iOS OS smart phone. Node MCU is the heart of this system and it can perform as a micro web server and it acts as an interface for the wide range of hardware modules. To control lights, fans and other home appliances which are connected to the relay system, the system offers switching functionalities. As technology is advancing so houses are also getting smarter. Modern houses are gradually shifting from conventional switches to centralized control system, involving remote controlled switches. Presently, conventional wall switches located in different parts of the house makes it difficult for the user to go near them to operate. Even more it becomes more difficult for the elderly or physically handicapped people to do so. Remote controlled home automation system provides a most modern solution with smart phones. In order to achieve this, a microcontroller board with built in wifi module at the receiver end while on the transmitter end, a GUI application on the cell phone sends ON/OFF commands to the receiver where loads are connected. By touching the specified location on the GUI, the loads can be turned ON/OFF remotely through this technology.

**Keyword**

1. ESP8266
2. Home Automation
3. Relay Module
4. Smart Light Control
5. Smart Fan Control
6. Voice Controlled Home Automation
7. Google Assistant Integration
8. IOT (Internet of Things)
9. Jumper Wires
10. Node MCU
11. Wi-Fi Controlled Devices
12. Voice Command
13. Smart Home System
14. Low-Cost Automation
15. Arduino IDE
16. DIY Home Automation
17. Wireless Control
18. Google Voice API
19. Home Appliances Automation
20. Home Automation using ESP8266

**CONTENTS**

Abstract………………………..……………………………………………………………………………………………………………….2

Keyword ………………………………………………………………………………………………………………3

Introduction ………………………………………………………………………………………………………………6

* 1. About Internet Of Things…………………………………………………………………………………………………...6
  2. EARLIER HOME AUTOMATION SYSTEM………………………………………………………………………………6
  3. EMBEDDED SYSTEMS………………………………………………………………………………………………………….7

1.3.1 NODE MCU Board…………………………………………………………………………………………8

1.3.1.1 Node MCU ESP8266 SPECIFICATIONS & FEATURES……………………………………….9

1.3.1.2 Applications of Node MCU…………………………………………………………………………….9

1.3.2 SOFTWARE USED ………………………………………………………………………………………………………9

1.3.2.1 Arduino IDE …………………………………………………………………………………….……….9-10

1.3.2.2 Programming Node MCU ESP8266 with Arduino IDE

2 AIM AND SCOPE ………………………………………………………………………………………………………………………11

2.1 PROJECT AIM……………………………………………………………………………………………………………………..11

2.2 PROJECT SCOPE …………………………………………………………………………………………………………………11

2.2.1 SMART HOMES……………………………………………………………………………………………………… 11

2.2.2 SMART CITIES…………………………………………………………………………………………………………11

2.2.3 AGRICULTURE…………………………………………………………………………………………………………11

2.3 PROJECT OBJECTIVE…………………………………………………………………………………………………………11

3METHODS AND MATERIAL USED ………………………………………………………………………………………………12

3.1 HARDWARE COMPONENTS USED AND SPECIFICATION ……………………………………………………..12

3.1.1 NODE MCU (ESP8266- 12E) ……………………………………………………………………………………….12

3.1.1.1 FEATURES ………………………………………………………………………………………………13

3.1.1.2 Node MCU Development Board Pinout Configuration ……………………………13

3.1.1.3 BRIEF About NODE MCU……………………………………………………………………13-14

3.1.2 4- CHANNEL RELAY MODULE ………………………………………………………………14-15

3.1.2.1 Working Principles of RELAY……………………………………………………………….16

3.1.2.2 CONSTRUCTION Of RELAY……………………………………………………………………16

3.1.2.3 Connectivity Of RELAY…………………………………………………………………………17

3.1.4 JUMPER WIRES…………………………………………………………………………………………………….17

3.1.4.1 TYPES…………………………………………………………………………………………………18

3.1.4.2 VERSIONS OF SOLID TIPS JUMPER WIRES……………………………………………18

3.1.5 5V 1 AMPS AC To DC ADAPTER ……………………………………………………………………………20

3.1.5.1 SPECIFICATIONS ………………………………………………………………………………….20

3.2 SOFTWARE REQUIREMENTS ……………………………………………………………………………………………..21

3.2.1 ARDUINO IDE ………………………………………………………………………………………………………21

3.2.1.1 NODE MCU CODE VIA ARDUINO IDE …………………………………………………..21

3.3 Sinric Pro, Google Home, and ESP Mail integration ……………………22

RESULTS AND DISCUSSION………………………………………………………………………….31

4.1 Light Control Test Results ……………………………………………………………………………………..31

4.2 SYSTEM ANALYSIS …………………………………………………………………………………………………32

4.3 The Final Hardware Circuit Connections ………………………………………………………………..32

CONCLUSION……………………………………………………………………………………………..33

REFERENCES ..…………………………………………………………………………………………….34

**LIST OF ABBREVIATIONS**

**ABBREVATIONS EXPANSION**

**LCD - Liquid Crystal Display**

**IDE - Integrated Devlopment Environment**

**PIR - Passive Infrared Sensor**

**ADC - Analog to Digital Converter**

**SPI - Serial Peripheral Interface**

**GUI - Graphical User Interface**

**IOT - Internet Of Things**

**Chapter 1**

**INTRODUCTION**

**1.1 About Internet Of Things**

Internet of Things depicts a general concept for the power of network devices to sense and collect data from the planet around us, then share that data across the Internet where it can be processed and applicable for various interesting purposes. Internet of Things is extremely quickly becoming a reality. We can see the results of it around us. Our devices are becoming smarter every day from smart phones to smart TV to smart car to Smart kitchen. Everything is now getting connected to Internet. Internet of Things (IoT) describes a network of physical objects that hook up with one another through the web. Objects or ‘things’ can transfer data wirelessly without requiring human interaction. A ‘thing’ are often any object which will be assigned an IP address and given the power to transfer data over a network. A Thing, within the Internet of Things (IOT), are often an individual with a cardiac monitor implant, a livestock with a biochip transponder, an automobile that has built-in sensors to alert the driving force when tire pressure is low -- or the other natural or man-made object which will be assigned an IP address and given the power to transfer data over a network. These devices collect useful data with the assistance of varied existing technologies then autonomously flow the information between other devices. Current market examples which includes smart thermostat systems and washer/dryers that employ Wi-Fi for remote monitoring. Internet of Things (IoT) is an architecture that comprises of especial hardware boards, Software systems, web APIs, and protocols which together creates a seamless environment that allows smart embedded devices which is to be connected to the web such that sensory data can be accessed and control system can be activated over internet. Also such devices could be connected to internet using various means like WiFi, Ethernet and so on. Furthermore devices might not needed to be connected to internet independently .

**1.2 EARLIER HOME AUTOMATION SYSTEM**

A home automation system generally connects controlled devices to a main server or "interface". The interface for control of the system uses either wall-mounted terminals, tablet or desktop computers, a mobile application, or an internet interface which will even be accessible off-site through the Internet. While there are many competing sellers, there are gradually increasing efforts towards open source systems (OSS). However, there are issues with the present state of home automation including a lack of authorized security measures and deprecation of older devices without backwards compatibility. Home automation has high potential for sharing data between relations or trusted individuals for private security and will cause energy saving measures with a positive environmental impact in the future. Early home automation began with labor-saving machines. Self-contained electric or gas powered home appliances became viable within the 1900s with the introduction of electrical power transmission and led to the introduction of washing machines , water heaters , refrigerators, sewing machines, dishwashers, and garments dryers. In 1975, the primary general purpose home automation network technology, X10, was developed.It is a communication protocol for electronic devices. It primarily uses electrical power transmission wiring for signalling and control, where the signals involves brief frequency bursts of digital data, and remains the foremost widely available.

**1.3 EMBEDDED SYSTEMS**

In these present scenario, the world is becoming more and more digital and using advanced technologies as like connected and automated. Embedded system is one of the advanced technology. This system is the one which has computer hardware with software embedded in it. The embedded system is designed to perform a specific task/function. But the task assigned has to be completed in a given time interval and it is an arrangement in which all its units, assembled work together according to set of info provided. When the system performs the given tasks at high speed, then it is used for real-time applications. A set of information or code embedded into the micro controller.

The characteristics of the Embedded Systems are:-

* Specific-Functioned
* Tightly constrained
* Reactive and Real time
* Microprocessors based
* Memory
* HW-SW systems
* Accuracy, reliability and adaptability.

**Advantages**

* Easily Customizable
* Less power consumption
* Cheap
* Enhanced performance

**Disadvantages**

* High development effort
* Larger time to market

**1.3.1 NODE MCU Board**

Node MCU is an open-source Lua based firmware and development board specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The Node MCU ESP8266 development board comes with the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor.

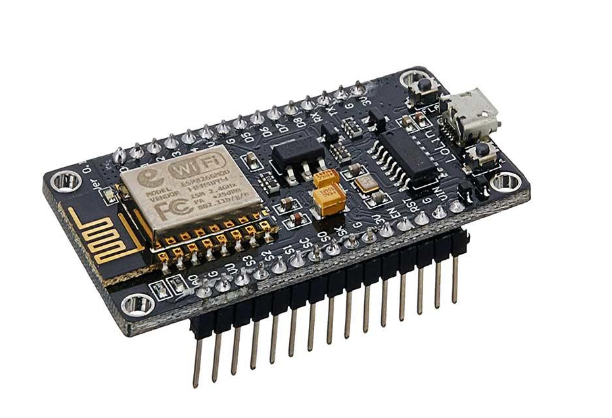
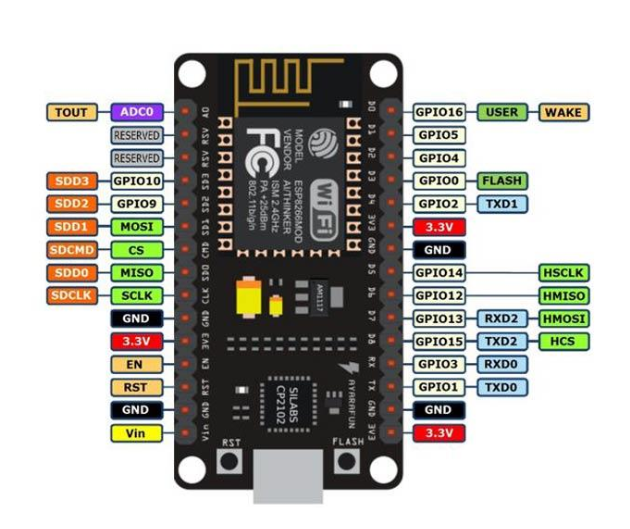


Figure 1.3.1 ESP8266 Development Board

This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. Node MCU has 128 KB RAM and 4MB of non-volatile storage to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects. Node MCU are often powered using Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.



**Figure 1.3.2 Node MCU ESP8266 Pinout**

**1.3.1.1 Node MCU ESP8266 SPECIFICATIONS & FEATURES**

* Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
* Operating Voltage: 3.3V
* Input Voltage: 7-12V
* Digital I/O Pins (DIO): 16
* Analog Input Pins (ADC): 1
* UARTs: 1
* SPIs: 1
* I2Cs: 1
* Flash Memory: 4 MB
* SRAM: 64 KB
* Clock Speed: 80 MHz
* USB-TTL supported CP2102 is included onboard, Enabling Plug n Play
* PCB Antenna
* Small Sized module to suit smartly inside your IoT projects.

**1.3.1.2 Applications of Node MCU**

* Prototyping of IoT devices
* Low power battery operated applications
* Network projects
* Projects requiring multiple I/O interfaces with Wi-Fi and Bluetooth functionalities.

**1.3.2 SOFTWARE USED**

**1.3.2.1 Arduino IDE**

The Arduino IDE (integrated development environment) is a cross platform application which is written in the functions from C, C++ and JAVA. The Arduino IDE is also a derivative of Processing IDE. The Arduino IDE is used for easy to write and upload programs in Arduino boards by using a cable that is connected between board and IDE. The operating system for Arduino software can be Windows, Mac Os and Linux depending upon the user. The IDE has a software library from the wiring projects and to provide a common input and output procedures.

* **setup():** a function that runs once at the start of a program and that can declare and initialize settings.
* **loop():** a function that is called repeatedly until the board powers off.

Figure 1.3.2.1 Arduino IDE Setup

**1.3.2.2 Programming Node MCU ESP8266 with Arduino IDE**

The Node MCU Development Board can be easily programmed with Arduino IDE since it has an easy access. Programming Node MCU with the Arduino IDE may hardly take 5-10 minutes or depending upon the task that is to be performed. All you need is the Arduino IDE, a USB cable and the Node MCU board.

**ADVANTAGES**

* Huge documentation and support
* Large library collection
* Open source software
* Easily portable
* Less power consumption
* Highly customizable.

**APPLICATIONS**

* Home Automations
* ISP Programming
* Entry level circuit designing

**CHAPTER 2**

**AIM AND SCOPE**

**2.1 PROJECT AIM**

The aim of the project is to design and construct a home automation system which will remotely switch on or off any household appliances connected to it, using a microcontroller with a built-in WiFi for WIFI based Web Application.

**2.2 PROJECT SCOPE**

**2.2.1 SMART HOMES**

In the coming years, fully automated smart homes will surely become a reality because the home automation is growing rapidly in this tech world.Due to good user convenience and accessibility, smart homes are appealing for a wide range of people all over the world. The User can check for the electricity usage, the condition of their devices and receive notifications accordingly.

**2.2.2 SMART CITIES**

With increasing automation and IOT, devices can share info with each other.This will help in building modern and hi-tech cities. Cities that might be free form all types of pollutions, traffic accidents, etc. problems.

**2.2.3 AGRICULTURE**

The proposed system might be used in Agricultural as well. The various devices utilized in fields are often operated from any remote location.

**2.3 PROJECT OBJECTIVE**

The objective of this project is to implement a very low cost, reliable and scalable home automation system which will be used for remotely switching on or off any home appliance, using a microcontroller to achieve hardware simplicity low cost built-in wifi module at the receiver end while on the transmitter end, a GUI application on the cell phone sends ON/OFF commands at the receiver end where loads are connected.

As technology is advancing so houses also are getting smarter. Modern houses are gradually shifting from conventional switches to centralized system, involving remote controlled switches. Presently, conventional wall switches located in several parts of the house makes it difficult for the user to travel near them to work. Even more it becomes harder for the elderly or physically handicapped people to try to do so. Remote controlled home automation system provides a latest solution with smart phones.

**CHAPTER 3**

**METHODS AND MATERIAL USED**

**3.1 HARDWARE COMPONENTS USED AND SPECIFICATION**

In this project some hardware components have been used to control the home appliances. Such as:

* Node MCU (ESP8266-12E)
* 4-Channel Relay Module
* Breadboard
* Jumper Wire
* 5 V 1Amp AC To DC Adapter

**3.1.1 NODE MCU (ESP8266- 12E)**

The heart of project is the WiFi enabled board that needs no introduction; the ESP8266 based Node MCU development board. Node MCU is an open-source Lua based software and development board specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is predicated on the ESP-12 module.Node MCU was born out of the desire to overcome the limitations associated with the first versions of the ESP8266 module which was not compatible with the breadboards, it was difficult to power and had more difficulty in programming. The Node MCU board is easy to use at a very low cost and that quickly endeared it to the heart of makers and it is one of the most popular boards today. Both the firmware and prototyping board designs are open source.The firmware uses the Lua scripting language. The firmware is predicated on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, like lua-cjson and SPIFFS. Thanks to resource constraints, users got 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 generally 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 allowed a easier way of prototyping on breadboards. The design was initially supported on the ESP-12 module of the ESP8266, which may be a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, that is widely utilized in IoT applications. The sole reason for using the Node MCU over Raspberry Pi is that the Node MCU has inbuilt Wi-Fi. This reduces the cost and hence the Node MCU is cheaper than the other devices available in the market. Inbuilt Wi-Fi helps in remote access. The system is accessible from any remote location round the world provided an online connection. Once given an input, the device will still operate albeit there’s no internet access. The device can be physically handled as well.

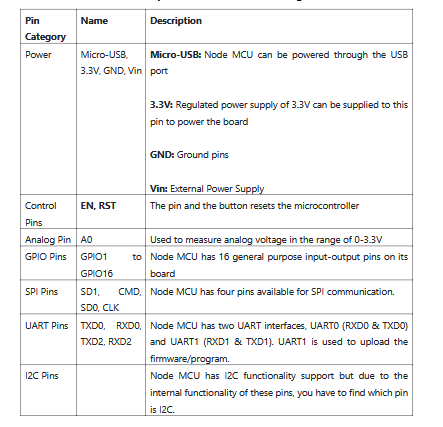


**Fig. 4.1.1 Node MCU**

**3.1.1.1 FEATURES**

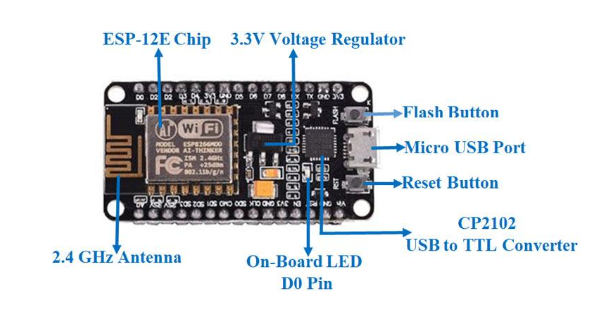
* Open-source
* Interactive
* Programmable
* Less cost
* Simple
* Smart
* WI-FI enabled
* Arduino-like hardware IO
* Advanced API for hardware IO, which can dramatically reduce the repeated work for configuring and manipulating hardware. Code like arduino, but interactively in the Lua script.
* Nodejs style network API

**3.1.1.2 Node MCU Development Board Pinout Configuration**

****

**3.1.1.3 BRIEF About NODE MCU**

The Node MCU ESP8266 development board comes with the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and function at 80MHz to 160 MHz adjustable clock frequency. Node MCU has 128 KB RAM and 4MB of non- volatile storage to store data and programs. Its higher processing power with the in-built Wi-Fi / Bluetooth and Deep Sleep Operating features makes it best for IoT projects. Node MCU are often powered using the Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.



**Figure 4.1.1.3 Briefing Node MCU**

**3.1.2 4- CHANNEL RELAY MODULE**

The relay is that device that open or closes the contacts to cause the operation of the opposite electric control. It detects the intolerable or unwanted condition with an assigned area and gives the commands to the circuit breaker to disconnect the affected area. Thus isolates the system from damage.The 4 Channel Relay Module may be a convenient board which may be used to control high voltage, high current load like motor, solenoid valves, lamps and AC load. It is designed to transmit and receive data with microcontroller such as Arduino, PIC and etc.

The relay ’s terminals (COM, NO and NC) are being brought out with screw terminal. It also comes with a LED indicator to indicate the status of relay. A relay module is an electrical switch that is operated by an electromagnet. The electromagnet gets activated by a separate low-power signal from a micro controller.When activated, the electromagnet pulls to either open or close an circuit. A simple relay consists of wire coil wrapped around a soft iron core, or solenoid, an iron yoke that delivers a less reluctance path for the magnetic lines of forces, a movable iron armature and one or more sets of contacts.

The movable armature is then hinged to the yoke and linked to one or more set of the moving contacts. Held in situ by a spring, the armature leaves a gap within the magnetic circuit when the relay is de-energized.

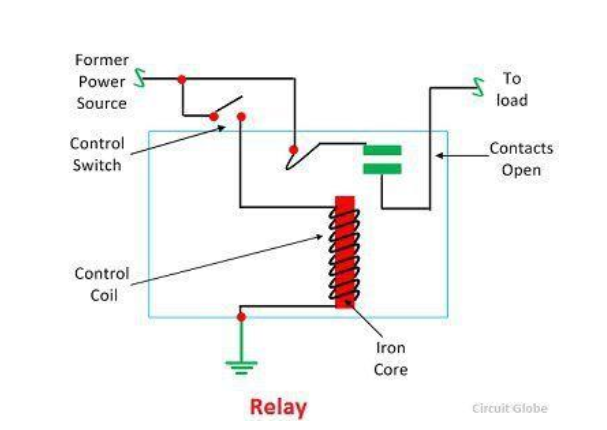
While in this position, one of the two sets of contacts is closed while the other set remains open.When electrical current is passed through a coil, it generates a magnetic flux that successively activates the armature. This movement of the movable contacts makes or breaks a reference to the fixed contact. When the relay is de-energized, the sets of contacts that were closed, open and breaks the connection and vice versa if the contacts were open. When switching off the current to the coil, the armature is returned, by force, to its relaxed position .

****

**Fig. 4.1.2 Relay Module**

**3.1.2.1 Working Principles of RELAY**

It works on the principle of an electromagnetic attraction. When the circuit of the relay senses the fault current, it energies the electromagnetic field which produces the temporary magnetic field. This magnetic field moves the relay armature for opening or closing the connections. The small power relay has only one contacts, and the high power relay has two contacts for opening the switch.The inner section of the relay is shown in the figure below. It has an iron core which is wound by a control coil. The power supply is given to the coil through the contacts of the load and the control switch. The current flows through the coil produces the magnetic field around it. Due to this magnetic field, the upper arm of the magnet attracts the lower arm.Hence close the circuit, which makes the current flow through the load. If the contact is already closed, then it moves oppositely and hence open the contacts.



**Figure 4.1.2.1 Working of Relay**

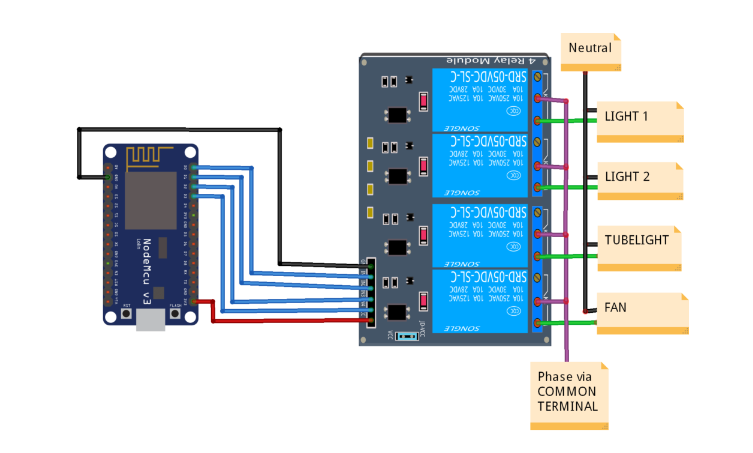
**3.1.2.2 CONSTRUCTION Of RELAY**

The relay operates both electrically and mechanically. It consists electromagnetic and sets of contacts which perform the operation of the switching. The construction of relay is mainly classified into four groups. They are the contacts, bearings, electromechanical design, terminations and housing.

* Contacts – The contacts are the most important part of the relay that affects the reliability. The good contact gives limited contact resistance and reduced contact wear. The selection of the contact material depends upon the several factors like nature of the current to be interrupted, the magnitude of the current to be interrupted, frequency and voltage of operation.
* Bearing – The bearing may be a single ball, multi-ball, pivot-ball and jewel bearing. The single ball bearing is used for high sensitivity and low friction.The multi-ball bearing provides low friction and greater resistance to shock.
* Electromechanical design – The electromechanical design includes the design of the magnetic circuit and the mechanical attachment of core, yoke and armature. The reluctance of the magnetic path is kept minimum for making the circuit more efficient. The electromagnet is made up of soft iron, and the coil current is usually restricted to 5A and the coil voltage to 220V.
* Terminations and Housing – The assembly of an armature with the magnet and the base is made with the help of spring. The spring is insulated from the armature by moulded blocks which provide dimensional stability. The fixed contacts are usually spot welded on the terminal link.

**3.1.2.3 Connectivity Of RELAY**

Relay module is being connected directly to digital circuits including microcontroller kits easily to control big loads by a microcontroller. The inputs IN1, IN2, IN3 and IN4 operate four relays with voltage between 3-5 volts DC. Input and output circuits are separated by Optocouplers to protect digital circuits in case connection mistakes happened or short circuits.



**Figure 4.1.2.3 Connectivity of Relay**

**3.1.4 JUMPER WIRES**

A jump wire (also known as jumper, jumper wire, jumper cable, DuPont wire or cable) is

an electrical wire, or group of them in a cable, with a connector or pin at each end (or

sometimes without them – simply "tinned"), which is normally used to interconnect the

components of a breadboard or other prototype or test circuit, internally or with other

equipment or components, without soldering.



**3.1.4.1 TYPES**

There are different types of jumper wires. Some have the same type of electrical connector at both ends, while others have different connectors. Some common connectors are:

* **Solid tips** – are used to connect on/with a breadboard or female header connector. The arrangement of the elements and ease of insertion on a breadboard allows increasing the mounting density of both components and jump wires without fear of short-circuits. The jump wires vary in size and colour to distinguish the different working signals.
* **Crocodile clips** – are used, among other applications, to temporarily bridge sensors, buttons and other elements of prototypes with components o equipment that have arbitrary connectors, wires, screw terminals, etc.
* Banana connectors – are commonly used on test equipment for DC and low-frequency AC signals.
* **Registered jack (RJnn) –** are commonly used in telephone (RJ11) and computer networking (RJ45).
* **RCA connectors** – are often used for audio, low-resolution composite video signals, or other low-frequency applications requiring a shielded cable.
* **RF connectors –** are used to carry radio frequency signals between circuits, test equipment, and antennas.
* **RF jumper cables -** Jumper cables is a smaller and more bendable corrugated cable which is used to connect antennas and other components to network cabling. Jumpers are also used in base stations to connect antennas to radio units. Usually the most bendable jumper cable diameter is 1/2".

**3.1.4.2 VERSIONS OF SOLID TIPS JUMPER WIRES**

**Jumper wires typically come in three versions:** male-to-male, male-to-female and female-to-female. The difference between each is in the end point of the wire. Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into. Male-to-male jumper wires are the most common and what you likely will use most often. When connecting two ports on a breadboard, a male-to-male wire is what you’ll need.

**MALE TO FEMALE JUMPER WIRES ,MALE TO MALE JUMPER WIRES ,FEMALE TO FEMALE JUMPER WIRES**

**3.1.5 5V 1 AMPS AC To DC ADAPTER**

This AC to DC power supply will do 5V at 1A! They're switch mode power supplies which means the output is regulated to 5V (no more 14V outputs!). These have a standard USB 'A' connector for the output so you can power your Arduino, Raspberry Pi, etc. through a USB cable.

**3.1.5.1 SPECIFICATIONS**

* Input Voltage (V): 100 ~ 280 VAC @50 ~ 60Hz.
* Input current (mA): 100.
* Output Power: 5V 1A.
* Input Plug: 2-Pin EU type.
* Output Plug: 5.5mm DC plug.

**3.2 SOFTWARE REQUIREMENTS**

**3.2.1 ARDUINO IDE**

The Arduino IDE (integrated development environment) is an cross platform application that is written in the functions from C, C++ and JAVA. The Arduino IDE is an derivative of Processing IDE. The Arduino IDE is used for easy to write and upload programs in Arduino boards by using an cable that was connected between board and IDE. The operating system of Arduino software is Windows, Mac Os and Linux. The IDE supplies a software library from the wiring projects and to provide a common input and output procedures.

**• setup():** a function that runs once at the start of a program and that can initialize settings.

**• loop():** a function called repeatedly until the board powers off.

The system is controlled using a Web Page which is based on IoT platform. The intended device can be powered ON or OFF using the above application. Also, it provides the facility to set the intensities of different appliances. The system becomes platform independent due to the use of a web application. It can be operated from any location by just opening the IoT platform web application. The web application also serves as a platform for managing the devices and the data.

**3.2.1.1 NODE MCU CODE VIA ARDUINO IDE**

To code Node MCU via Arduino IDE, the Node MCU needs to be added to Arduino IDE library first by adding this address to Arduino IDE preferences. After this reference is added to Arduino IDE, download node MCU to boards manager and then select Node MCU 1.0 (ESP12E Module). After node MCU is added to Arduino IDE library, upload this code with changing hotspot name and password.

The code includes the hotspot name and password match with the android. The code does not need to identify the relay input, as it is included in [Arduino IDE]. When access point is given by the microcontroller sent as password and SSID is the name of smart phone hotspot.

**3.3 Sinric Pro, Google Home, and ESP Mail integration**

**Sinric Pro & Google Home Integration:** For smart and wireless control over household appliances such as lights and fans, we integrated our ESP8266-based system with Sinric Pro and Google Home. Sinric Pro is an IoT platform that enables developers to remotely control devices through smart assistants like Amazon Alexa and Google Assistant. For this project, we utilized Google Home for hassle-free voice control functionality.

**Working Process**: Devices are registered on Sinric Pro dashboard with individual Device IDs. The ESP8266 creates a WebSocket connection with Sinric Pro through its API key and secret key.Voice commands are issued to Google Assistant, tied to Sinric Pro. Sinric Pro broadcasts the command to ESP8266 through WebSocket. ESP8266 operates the associated relays to switch lights/fans ON/OFF. Real-time device status is updated on the Sinric Pro dashboard.

**Benefits:** Easy Google Assistant integration ,Real-time intelligent control , External server hosting is not required , Cross-platform compatibility (iOS, Android).

**ESP Mail Notification System**

In order to enhance user awareness and security, we included an Email Notification System utilizing the ESP Mail Client library.An email notification is sent to the user whenever a device (e.g., fan/light) is turned OFF or ON.

**Setup Overview**:

ESP8266 is set up with SMTP settings (e.g., Gmail). An email is triggered using the ESP Mail Client library based on device state. Email includes timestamp and action taken (e.g., "Light Turned ON"). Example Use Case: "Living Room Light has been switched ON at 9:30 PM."

**Features:**

Logs appliance usage Notifies users of unauthorized access or suspicious activity , Improves system monitoring and accountability.

**3.4 EXSTING SYSTEM**

A home automation system is an automating the bulk of electronic and electrical tasks within a home. It uses a combination of hardware and software to enable control and management over appliances and devices within a home. Home automation not only refers to reduce human efforts but also energy efficiency and time saving. Here home automation is done by using ESP8266 – WiFi SoC and Blynk Play store App.

**3.4.1 WHAT THEY HAVE FOLLOWED IN EXSTING SYSTEMS**

In this existing system, home automation is done by using the Blynk app which sends the command by it’s Blynk Server. Home Automation mean controlling lighting, climate, entertainment systems, and appliances without a manual switch. Blynk is a mobile application which has its own server to process user requests. It is an open source application and anybody can use it in their Home Automation to control devices, monitor sensor data and get a notification by some trigger actions. Blynk app has been used to read data from sensors located in home environment and user controls home appliances based on these data. Being busy in hectic schedule of daily life user may not be able to read sensor data continuously to take some action through app. Blynk has a nice GUI but one may sometimes face a problem due to it’s busy server and get a late response or zero response.

**3.5 PROPOSED SYSTEM**

This project uses the Cloud and a Web Browser to control the manually operated switches. A cloud server is created for the environment where the switches are mounted. The switches are interfaced with Node MCU which has an inbuilt Wi-Fi. It can use this to enable or disable the switches. The user communicates with the processor through the Web Browser. The processor then controls the switches based on the commands received from the user and also updates the user about the status of the switches after the control operation is performed to the cloud.

**Hardware Requirements**

• Node MCU (ESP8266-12E) DEVELOPMENT BOARD

• 4-Channel Relay Module

• Breadboard

• Jumper Wire

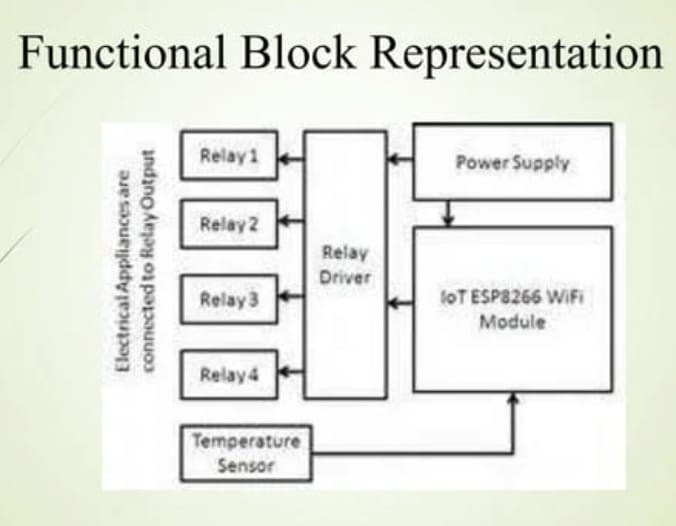
• 5 V 1Amp AC To DC Adapter

**Software Requirements**

Arduino IDE – Compiler

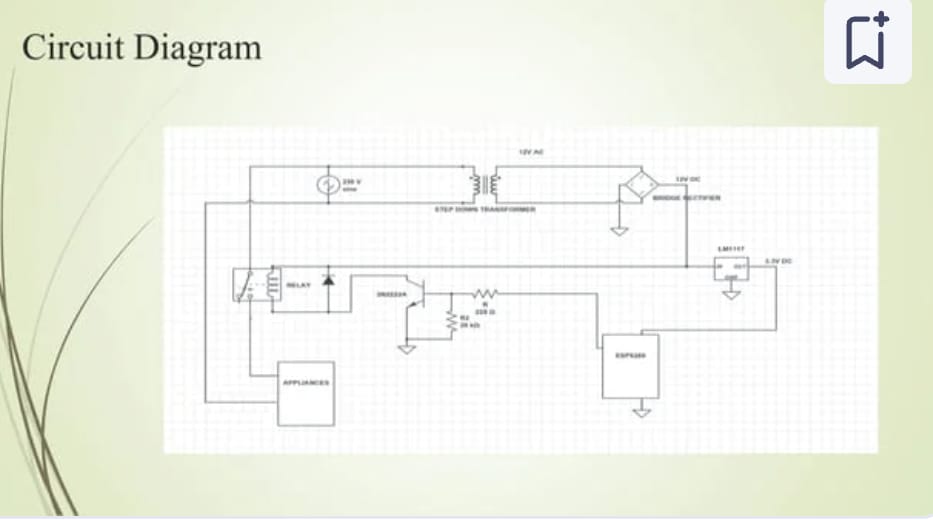
Language – Embedded C

**3.5.1 BLOCK DIAGRAM**

****

**Figure 4.4.1 Block diagram for proposed system.**

3.5.2 CIRCUIT DIAGRAM



**Figure 4.4.2 Circuit diagram for proposed system.**

**3.5.3 Working process**

The heart of this project is the WiFi enabled board that needs no introduction; the ESP8266 based Node MCU development board. It is an open source platform for developing WiFi based embedded systems and it is based on the popular ESP8266 WiFi Module, running the Lua based Node MCU firmware. Node MCU was born out of the desire to overcome the limitations associated with the first versions of the ESP8266 module which was not compatible with breadboards, it was difficult to power and even more difficult to program. The Node MCU board is easy to use, low cost and that quickly endeared it to the heart of makers and it is one of the most popular boards today.

For this project, we will add a 4-channel relay module to the ESP8266 board. The project flow involves the control of Node MCU’s GPIOs from a webpage on any device connected on the same network as the board. The status of the GPIOs control the coils of the relays and that causes the relay to alternate between normally open (NO) and normally closed (NC) condition depending on the state of the GPIO, thus, effectively turning the connected appliance “ON” or “OFF”.

**3.5.3.1 CONNECTIONS**

To make the connections easy to follow, here is a pin map of the connection between the NODE MCU and the Relay Module:

**Node MCU – Relay Module**

3.3V - VCC

GND - GND

D1 - IN1

D2 - IN2

D5 - IN3

D6 - IN4

**3.5.3.2 CODE**

One of the easiest way to program Node MCU is via the Arduino IDE. This, however, requires setting up the Arduino IDE by installing the board support file for Node MCU. The code is based on the ESP8266WiFi.h library which allows the easy use of WiFi functionalities of the board. It contains all we need to create or join a WiFi access point and also create a server and client which are all important for this project. The library comes attached with the Node MCU board files for the Arduino, so there is no need to install it once the board files have been installed.

The code for this project will enable us to control appliances connected to the GPIOs (via relays) of the Node MCU board remotely. To start with, we include the library that we will use for the project, which in this case, is the **ESP8266WiFi.h** library.

Next, we add the credentials of the WiFi access point to which the Node MCU will be connected. Ensuring that the username and password are between the double quotes. We also specify the port through which the system will communicate and create a variable to hold requests.

As shown in the below fig.

Figure 4.4.3.2 Code 1

Next, is the void setup() function. We start by initializing the serial monitor (as it will be used for debugging later on) and setting the pin Modes of the pins to which the relays are connected as output. We then set the pins “LOW” to ensure the system starts at OFF state.

Now, we connect to the access point using the credentials provided as arguments to the WiFi.begin() function and we use the WiFi.status() function to check if connection was successful. The system will keep trying until the connection is successful.

As shown in the below Figure.

Figure 4.4.3.3 Code 2

If the connection is successful, a text is printed on the serial monitor to indicate this, along with the IP address of the Node MCU. This IP address becomes the web address for the server and should be entered on any web browser on the same network as the server so we are able to access it.

Upload the code to the Node MCU. Ensure everything is connected as described under the schematics section. After uploading the code, the IP address of your web server displayed in the serial monitor as shown below.

**PRECAUTIONS**

It’s important to note that, connecting the appliances to the relay involves interaction with AC voltages which could be dangerous. Ensuring to have experience interacting with AC voltages and do so in a safe manner.

**CHAPTER 4**

**RESULTS AND DISCUSSION**

**4.1 Light Control Test Results**

The Light Control Test is done by pressing the ON / OFF button widget on the Web application on the respective Android smart phone/Pc for lights and fans. This is done after the system is turned on and connected to a Wi-Fi internet connection. If at any time the internet connection is lost or bad signal, then it also affects system performance.



Figure 5.1.1 Web App Output 1

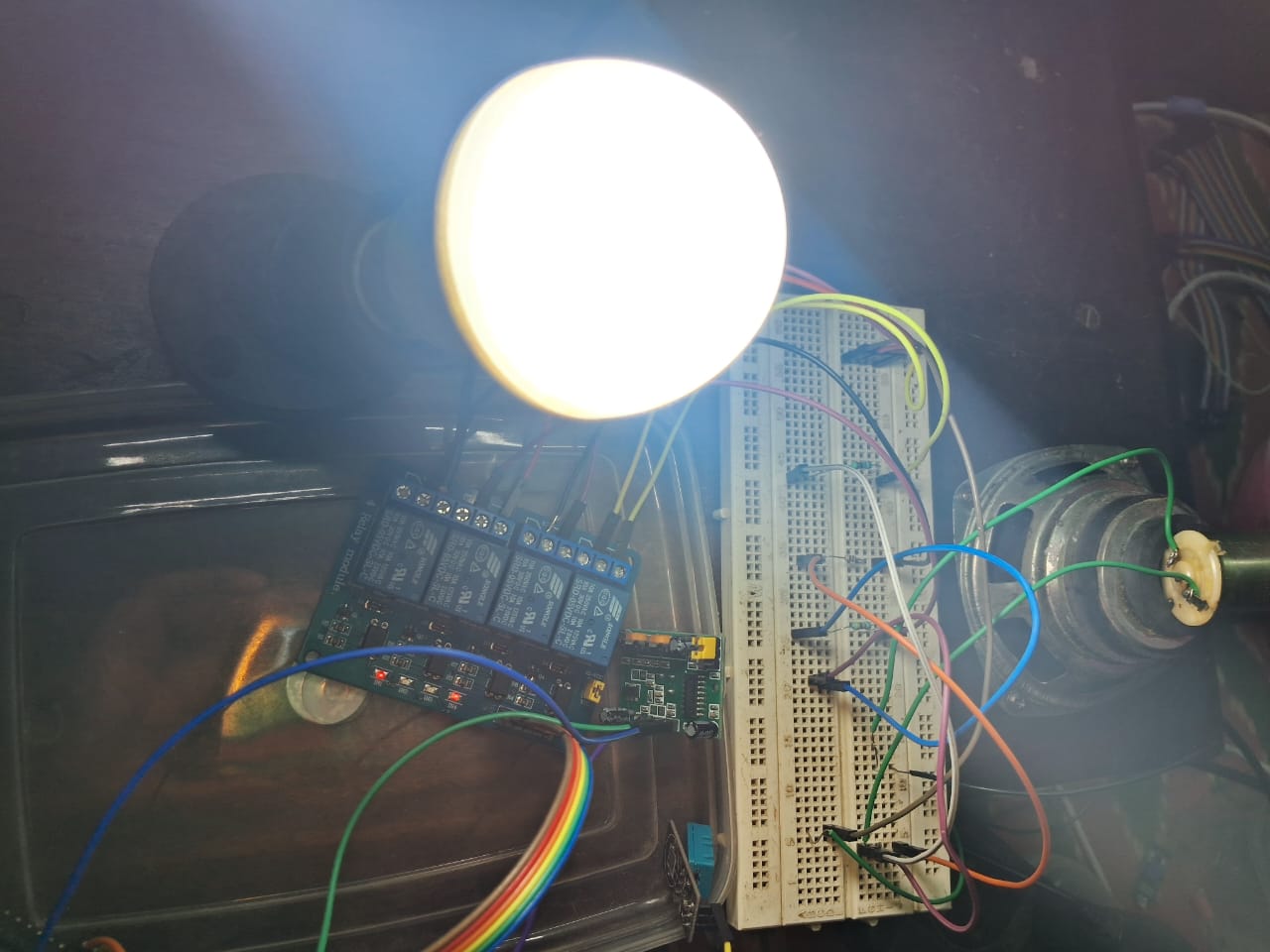


Figure 5.1.2 Web App Output 2

**4.2 SYSTEM ANALYSIS**

From testing the entire system above, the smart home works according to the purpose of this project. Comparison of this research with previous studies, namely this study uses control buttons, thus increasing the diversity of the smart home system itself. Also, used a microcontroller that is different from previous studies that is the Node MCU ESP8266 module which has advantages compared to other microcontrollers. The smart home has been successfully built with hardware arranged in such a way that it can achieve results that are as expected. In this case the hardware plays a very important role as the main device is the Node MCU ESP8266 module. The advantages of using the Node MCU ESP8266 are more practical than buying various components and then assembling them by yourself.





**CHAPTER 5**

**CONCLUSION**

Based on the results of analysis of all data obtained by testing the smart home with the Internet of Things based Node MCU ESP6288 module, the following conclusions can be drawn:

1) Smart Home with Internet of Things (IoT) based Node MCU ESP8266 Module can be designed with various components hardware and software support so that it can be arranged into a smart home system that is controlled with the Web application according to what is intended.

2) The Smart Home with this Internet of Things (IoT) based Node MCU ESP8266 Module can be implemented to control some of the home electronics performance including lighting controls, fan control, temperature monitoring, early warning systems and etc.

3) Main purpose of home automation system is to provide ease to people to control differnt home appliances with the help of the web application using their mobile phones or desktop and to save time and money.

**REFERENCES**

1)BOHORA, Bharat; MAHARJAN, Sunil; SHRESTHA, Bibek Raj. IoT Based Smart Home Using Blynk Framework. Zerone Scholar, [S.l.], v. 1, n. 1, p. 26-30, dec. 2016. ISSN 25422774. google scholar.

2) DC-DC Step Down Converter Power Supply Provides Regulated 5VDC Output with Range Input of 10-32VDC, Model GTD21088L-1505-T2.

3)Home Automation Using Internet of Thing 2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) Published: 2016. Google Scholar.

4)Internet of Things in Home Automation and Energy Efficient Smart Home Technologies Simon G. M. Koo Department of Computer Engineering, Santa Clara University, CA 95053, USA

5)Low Cost Implementation of Smart Home Automation Ravi Kishore Kodali Department of Electronics and Communication Engineering National Institute of Technology, Warangal , 506004 India

6)Mobile based home automation using Internet of Things (IoT) 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT) Published: 2015

7)NodeMCU Features and Pinout. A Brief Tutorial on the Introduction to NodeMCU V3.

8) Yoyosteven in Circuits Microcontrollers. NODEMCU 1.0 (ESP8266) CONTROLLED RELAY USING BLYNK (OVER THE WEB).

9)5V 4-Channel Relay Interface Board, Standard Interface that can be Controlled Directly by Microcontroller.

10) 15-17 March 2018 U. Venkanna IoT Based Smart Home Automation System Using Sensor Node. Google Scholar.