JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Implementation of IoT Based Home Automation

[1]Abhishek Gupta, [2]Bhavya Deep Sharma, [3]Bhavyanshu Shrimali, [4]Deeksha Choudhary, [5]Deekshant Tak
Department of Electrical Engineering, Swami Keshvanand Institute of Technology, Jaipur

Abstract: The growth of automation in this era is addressing the needs of the elderly and disabled, especially those who are living alone. It is essential that the automation systems in the home can meet the needs of the elderly and disabled. In the current home automation system (HAS), IoT technology is not available, or user interface is not friendly, and the range of wireless transmission is limited. The aim of this paper is to design a home automation system using the Blynk application and an ESP8266 microcontroller. Connecting blynk application and ESP8266 microcontroller through Wi-Fi we can control devices and home appliances with the mobile using blynk application. A Wi-Fi connection is required for this system to be controlled and monitored via the Blynk application, making it an IoT system (Internet of Things). The system is programmed in the Arduino Integrated Development Environment using libraries available for the sensors and for the Blynk application.

Keywords **- Internet of Thing (IoT), Home Automation System (HAS), Blynk application, ESP8266, Arduino Integrated Development Environment.**

I. INTRODUCTION

Internet of Things is an impression where each device is assigned together through that IP addressanyone makes that device individual on internet. Basically, it started as the "Internet of Computers". Research studies have forecast a volatile growth in the number of "things" or devicesthat will be connected to the Internet. The resulting network is called the "Internet of Things" (IoT) [1].

ESP8266 is an inexpensive System-on-a-Chip (SoC) based on an open-source software and hardware environment known as NodeMCU (Node Microcontroller Unit). Espress if Systems designed and manufactured the ESP8266, which includes computer elements such as CPU, RAM, connectivity (Wi-Fi), and even an operating system. As a result, it makes an excellent choice for any kind of Internet of Things (IoT) project [2].

The current developments in technology have permitted the use of Wi-Fi. Wi-Fi has enabled different devices to have capabilities of connecting with each other like one to one and one to multi- devices. By Using a Wi-Fi shield to act as a Micro web server for the Node MCU ESP8266 which eliminates the need for wired connections between the Node-MCU Esp8266 board and computer which reduces cost level, and your smartphone will send commands wirelessly via the internet to an ESP8266 module. A certain application must be running to encrypt the commands on a smartphone and deliver them to the ESP8266. There are other programs available, but we're going to select the greatest and most accessible one, "Blynk application ".

Home automation is an extremely attractive field of IoT. Through different connection modes, these areas represent how sensors and actuators are configured in a building, how they are connected automatically and how they provide remote control. IoT device manufacturers offer such connection modes, but sometimes the internet connection function is missing, which means that an auxiliary hardware device connected via IPv4 protocol to an internet connection is required to enable the internet device connection. Using a suitable software application, these devices can be controlled remotely using PCs and mobile devices using a directinternet connection or auxiliary hardware [3].

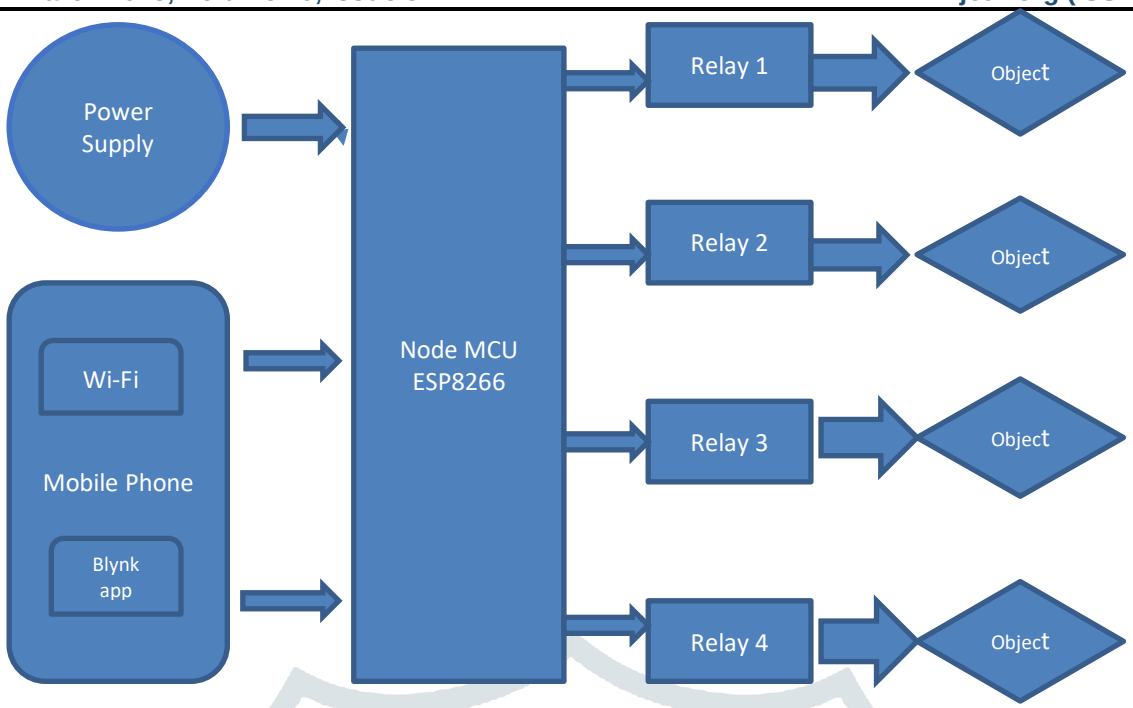

Fig: - 1 Home Automation Block Diagram

II. HARDWARE DESCRIPTION

2.1 Node-MCU ESP8266

A typical circuit board incorporates the ESP8266 chip. A built-in USB port that is already wiredup in the chip is included on the board. Wi-Fi antenna, LED lights and pins of the GPIO (General Purpose Input Output) standard size that may be plugged into a bread board. Its processor, the L106 32-bit RISC microprocessor core is based on the Ten-Silica Xtensa DiamondStandard 106 Micro, and it operates at 80 MHz Its memory includes 32 kb of instruction RAM,32 kb of instruction cache RAM, 80 kb of user RAM and 16 KB of system RAM [4].

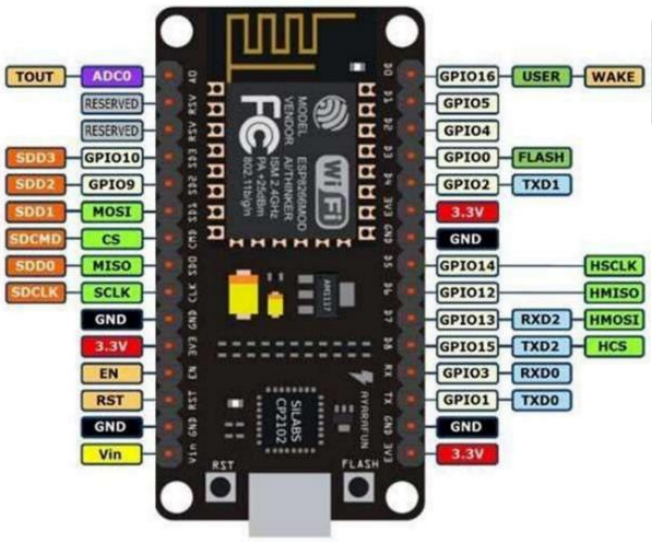

Fig.2 Pin Diagram of ESP8266

2.2 Relay

Four 5V relays and the related switching and isolating components are included in the four- channel relay module, which simplifies interface with a microcontroller or sensor and requires the fewest parts and connections. Two relays share each of the two terminal blocks, which have a total of six terminals each. The screw form of the terminals allows for simple and flexible connection to mains wiring [5, 12].

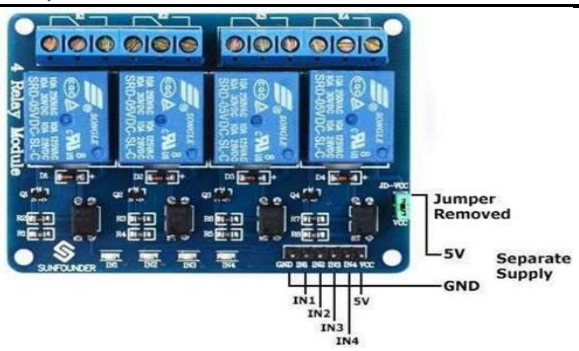

Fig.3 4-channel relay module

III. BLYNK APPLICATION

For use with the Internet of Things, Blynk was created. It has several amazing features, like remote hardware control, sensor data display, data storage, data visualization and many more.

- Blynk was created with the Internet of Things in mind. It is capable of remote hardware control.
- Using the many widgets we offer, the Blynk App enables you to create stunning interfaces for your projects.
- Blynk Server is in charge of overseeing all communications between the hardware and the smartphone. You can run your locally or use our Blynk Cloud. The open- source software may even be run on a Raspberry Pi and is readily capable of switching thousands of devices at once.
- Blynk Libraries enable communication with the server and process all incoming and outgoing commands for all popular hardware platforms [6].

This application make the project more accurate instead using AVR microcontroller of 8051 family. This is how all the shortcomings going to be minimized using Blynk Application instead of AVR technology [13].

IV. IMPLEMENTATION

The home automation circuit is built around ESP8266, Blynk Android App and a 4 - channel relay board. The hardware set up should be according to the circuit diagram. AC mains appliances (Bulbs) will be connected to relays which are controlled by the ESP8266. Once Arduino IDE is installed in the computer, connect the board with the computer using the USB cable. Now open the Arduino IDE and choose the correct board by selecting Tools>Boards>Node- MCU1.0 (ESP-12E Module) and choose the correct Port by selecting Tools>Port. To get it started with the Node-MCU board and blynk the built-in LED, load the example code by selecting Files>Examples>Basics>Blynk. Once the example code is loaded into your IDE, click on the "upload" button given on the top bar. Once the upload is finished, you should see the built-in LED of the board blinking. The user has to install and configure the Blynk App as per the above instructions. Node-MCU to 4- Channel Relay Board Connect D0 pin of Node-MCU to D1 pin of 4-Channel Relay board, Connect D1 pin of Node-MCU to D2 pin of 4-Channel Relay board, Connect D2 pin of Node-MCU to D3 pin of 4-Channel Relay board, Connect D3 pin of Node-MCU to D4 pin of 4-Channel Relay board, connect 3.3V of Node-MCU to Vcc pin of 4-Channel Relay board, Connect GND pin of Node-MCU to GND pin of 4-Channel Relay board. We are including ESP8266 Wi-Fi library which provides ESP8266 specific Wi-Fi routines, and we are calling it to connect to the network. BlynkSimpleEsp8266 library establishes the communication between Blynk App and ESP8266 [7].

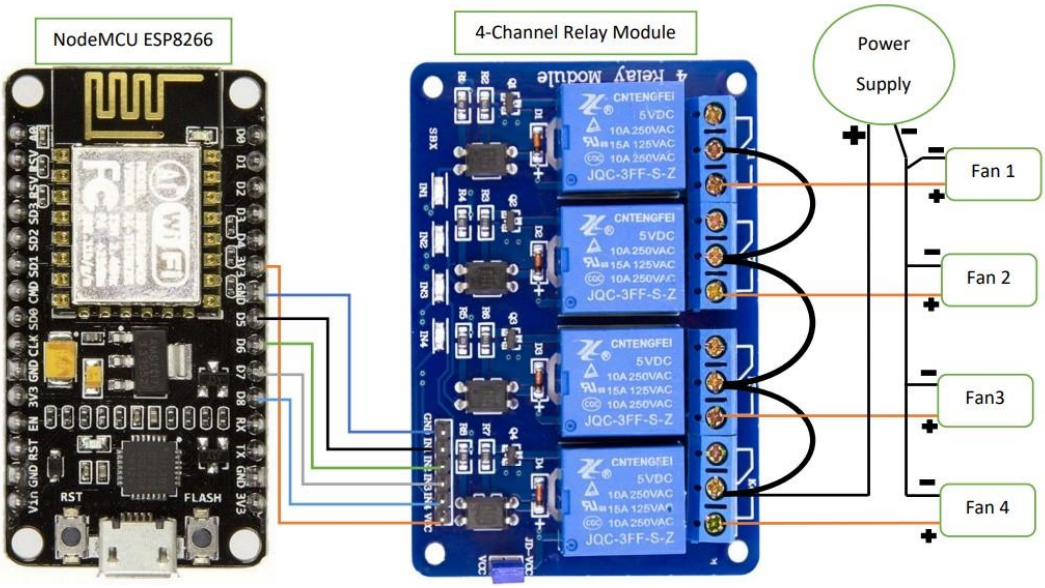

Fig: - 4 Circuit Diagram of Home Automation circuit

V. CODE

Code is written in Arduino Integrated Development Environment.

```
#define BLYNK_TEMPLATE_ID "TMPLztqLK6Hx"
#define BLYNK_DEVICE_NAME "Home automation"
 #define BLYNK_AUTH_TOKEN "Token"


#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth [] = BLYNK_AUTH_TOKEN;
const char *ssid = "Wi-Fi name"; // name of your Wi-Fi
 const char *pass = "password"; // password of your Wi-Fi


BLYNK_WRITE(V0)
{
int value = param.asInt();
Serial.println(value);
if (value == 1)
{
digital Write (D5, LOW);
 Serial.println("LED 1 ON");
}
if (value == 0)
{
digital Write (D5, HIGH);
Serial.println("LED 1 OFF");
}
}

BLYNK_WRITE(V1)
{
int value = param.asInt();
 Serial.println(value);
if (value == 1)
{
Digital Write (D6, LOW);
 Serial.println("LED 2 ON");
}
if (value == 0)
{
digital Write (D6, HIGH);
Serial.println("LED 2 OFF");
}
}

BLYNK_WRITE(V2)
{
int value = param.asInt();
 Serial.println(value);
 if (value == 1)
{
digital Write (D7, LOW);
```

```
Serial.println("LED 3 ON");
}
if (value == 0)
{
digital Write (D7, HIGH);
 Serial.println("LED3 OFF");
}
}

BLYNK_WRITE(V3)
{
int value = param.asInt();
 Serial.println(value);
 if (value == 1)
{
digital Write (D8, LOW);
 Serial.println("LED 4 ON");
}
if (value == 0)
{

digital Write (D8, HIGH);
Serial.println("LED 4 OFF");
}
}


void setup ()
{
Serial.begin(115200);
Blynk.begin(auth, ssid, pass);
pin Mode (D5, OUTPUT);
pin Mode (D6, OUTPUT);
pin Mode (D7, OUTPUT);
pin Mode (D8, OUTPUT);
}

void loop ()
{
Blynk.run();
}
```

VI. RESULT

Our project aimed to automate devices and appliances using ESP8266. It is easy to use, inexpensive, and competitive to choose either option [8]. A very simple API is also used in the blynk IOT network design to make it more flexible. blynk IOT is a complete smart home prototype with numerous features - automation, control, monitoring, and security - and can be continually enhanced and developed in the future [9]. Essentially, the paper serves as a survey of recent (in the last 10 years) published research, commercial solutions, and open- source home automation systems. The proposed work differs from previous works in the literature by detailing both the hardware and software implementations. Most of the smart home systems presented in the literature [10]. There are a limited number of devices that blynk IOT works with, and it assumes common APIs, firmware, etc. The firmware we provide is open source, which means there are no hacks and no third-party hubs or clouds needed; all devices are controlled the same way via the same API. The installation procedure is well documented and supported devices are thoroughly tested. The blynk IOT platform can also be customized for different peripherals, networks, and technologies with add-ons. Due to its master-slave architecture, blynk IOT is highly scalable in terms of number of devices [11].

After successful testing and experiments with remotely controlled home appliances, safety systems and security systems, results are attained. Smart home users that successfully utilize the Blynk App to remotely manage their home appliances may turn on and off lights, fans, and other equipment. Fans and lights in a smart house will turn on when the user presses the button using the Blynk app, and vice versa when he presses the button to turn them off.

VII. CONCLUSION

It is clear from this project's work that a home automation system for individual control can be easily constructed from inexpensive locally available components and used to control a variety of home appliances, including security lamps, televisions, air conditioners, and even the entire home lighting system. Better yet, the few numbers of components needed allow them to be packaged into a discrete and small container. The proposed home automation system has undergone numerous tests that allow it to operate a variety of household equipment used in lighting systems, air conditioning systems, home entertainment systems, and many more. The estimated expense of the project is 1560rs/. Consequently, this system is adaptable and scalable.

VIII. REFERENCES

[1] D. Norris, Smart Home Automation Based on IOT and Android Technology M. Abivandhana1, K. Divya2, D. Gayathri3, R. RuhinKouser4 Student1, 2, 3, Assistant Professor4 Department of CSE Kingston Engineering College, Katpadi, Vellore, India.
[2] Brian Benchoff (25 October 2014). "An SDK for the ESP8266 Wi-Fi chip". Hackaday. Retrieved 2 April 2015.
[3] V Measurement & Control using Smartphone & Tablet, Hans J. Berndt, independently published, 4 Oct. 2017, ISBN-10: 1549604325, ISBN-13: 978- 1549604324.
[4] Brian Benchoff (2 January 2015). "A DEV BOARD FOR THE ESP LUAINTERPRETER". Hackaday. Retrieved 2 April 2015. Jump up to: "IBM Developer". Mpx. "Lua CJSON is a fast JSON encoding/parsing module for Lua", Pellepl. "Wear-leveled SPI flash file system for embedded devices". GitHub. Retrieved 2 April 2015. "Node- MCU - A Perfect Board for IoT". circuito.io blog. 2018- 11-21. Retrieved 2021-05-27.
[5] "Coordinated Power Systems Protection". Department of the Army Technical Manual. United States Department of the Army (811–814):3-1-1991.
[6] Martin Bates (2006). Interfacing PIC Microcontrollers Embedded Design by Interactive Simulation. Newness, London, Bray, Hiawatha (21 December 2017), published, Mike Prospero (2022-05-24).
[7] International Journal for Modern Trends in Science and Technology, 7(03): 151-153, 2021 Copyright © 2021 International Journal for Modern Trends in Science and Technology ISSN: 2455-3778 online.
[8] Baraka K., Ghobril M., Malek S., Kanj R., Kayssi A. Low-Cost Arduino/Android- Based Energy-Efficient Home Automation System with Smart Task Scheduling; Proceedings of the 2013 5th International Conference on Computational Intelligence, Communication Systems and Networks; Madrid, Spain. 5–7 June 2013; pp. 296–301. S [Google Scholar]
[9] Gunputh S., Murdan A.P., Oree V. Design and implementation of a low-cost Arduino- based smart home system; Proceedings of the 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN); Guangzhou, China. 6–8 May 2017; pp. 1491–1495. [Google Scholar]
[10] Bhatt A., Patoliya J. Cost effective digitization of home appliances for home automation with low-power WiFi devices; Proceedings of the 2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bioinformatics (AEEICB); Chennai, India. 27–28 February 2016; pp. 643–648. [Google Scholar]
[11] Shafana A.R.F., Aridharshan A. Android based automation and security system for smart homes. Int. J. Comput. Sci. Inf. Technol. 2017; 5:26–30. [Google Scholar]
[12] Gupta, Abhishek, Mohit Kothari, Prabhakar Kalani, Prakhar Goyal, Prateek Kambar, and Shurveer Singh. "Automatic Transformer Distribution and Load Sharing Using Microcontroller."
[13] Gupta, Abhishek, Siddharth Khinchi, Sidharth Jindal, Tejpal Singh, and Sneha Sharma. "Induction motor speed control using android application." IJ of Electrical and Electronic Research 4, no. 2 (2000).