Personal Environment Assistant

Project report submitted for

IIIrd Semester Minor Project

in

Department of ECE

By,

Tarun Kalo(221010253)

Hemant Kumar Ratre(221010219)

Shubham Jain(221010246)

Department of Electronics And Communication Engineering

Dr. Shyama Prasad Mukherjee

International Institute of Information Technology, Naya Raipur

(A Joint Initiative of Govt. of Chhattisgarh and NTPC)

Email: iiitnr@iiitnr.ac.in, Tel: (0771) 2474040, Web: www.iiitnr.ac.in

Guidelines for Annexure -I

CERTIFICATE

This is to certify that the project titled "PERSONAL ENVIRONMENT ASSISTANT" by "TARUN KALO, HEMANT KUMAR RATRE, SHUBHAM JAIN" has been carried out under our supervision and that this work has not been submitted elsewhere for a degree.

(Signature of Guide)

DR.BIPIN CHANDRA MANDI

ASSISTANT PROFESSOR(ECE)

Department of ECE

Dr. SPM IIIT-NR

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature of Author)

Author Name

(Roll Number)

Date : ____________________

PLAGIARISM REPORT

(The plagiarism of the minor project report must be less the 10% for final acceptance of the report by panel members)

Approval Sheet

This project report entitled "Personal Environment Assistance" by "TARUN KALO, HEMANT KUMAR RATRE, SHUBHAM JAIN" is approved for IIIth Semester Minor Project.

(Signature of Examiner - I)

____

Name of Examiner -I

(Signature of Examiner - II)

____

Name of Examiner -II

(Signature of Chair)

____

Name of Chair

Date: ______________ Place: ______________

Personal Environment Assistance

***Abstract*—The "Personal Environmental Assistant" is a smart project designed to enhance personal well-being by monitoring and controlling environmental factors. This system integrates temperature, humidity, and air quality sensors to provide real-time data on the surrounding environment. When the temperature or humidity deviates from a predefined range, corresponding LEDs indicate the status, and relays can control devices to optimize comfort. Additionally, an MQ-135 air quality sensor measures pollutants, triggering LEDs and relays for efficient air quality management.**

The project is user-friendly, offering remote control through the Blynk app. Users can instantly adjust the environment by toggling switches on the app to turn LEDs and relays on or off, ensuring a personalized and comfortable living space. The system is highly adaptable, allowing users to customize temperature, humidity, and air quality thresholds according to personal preferences.

In essence, the Personal Environmental Assistant empowers individuals to effortlessly manage their immediate surroundings, fostering a healthier and more comfortable lifestyle. This smart project amalgamates simplicity, functionality, and remote accessibility, making it an invaluable addition to daily life for enhanced well-being.

***Impact Statement* —**
The "Personal Environmental Assistant" has a big impact on how we feel every day. It helps us stay comfortable by keeping an eye on the temperature, humidity, and air quality around us. When things go outside the range we like, it uses lights and switches to let us know and even makes changes to make things better.

This project is like having a friendly helper that ensures our surroundings are just the way we want them. With the Blynk app, we can control it from our phones, making it super easy to make adjustments and create the perfect environment for ourselves.

In simple terms, this project makes our daily lives more comfortable and healthier by giving us control over our immediate surroundings. It's like having a personal assistant for our environment, making sure we feel good every day.

***Index Terms*—PEA - Personal Environment Assistance.**

I. Introduction

The creation and adoption of Personal Environment Assistants (PEAs) mark a paradigm shift in how people interact with their environment and go about managing their everyday lives in the age of fast technology growth. Our ability to access information, complete tasks, and make decisions has been changed by these intelligent digital companions, thereby improving our quality of life. The PEA project is proof that intelligent technology, artificial intelligence, and individualized help can all coexist.PEAs automatically interact with smart home gadgets, giving customers a level of environmental management never before possible. Beyond just being convenient, this endeavor has an impact on energy efficiency, healthcare, personalization, and education.Beyond just being convenient, this endeavor has an impact on energy efficiency, healthcare, personalization, and education. Machine learning enables PEAs to continuously adapt to user preferences and deliver customized experiences. The PEA project aims to transform how we live our daily lives as it develops, enabling a more socially connected, knowledgeable, and productive society for everybody.

II. Motivation

Comfort First: We wanted to create something that makes everyone's daily life more comfortable. Whether it's too hot, too humid, or the air feels a bit off – our project steps in to make things just right.

Personal : We believe that everyone should have control over their immediate surroundings. Our project lets people personalize their environment, like having a cozy sweater for their living space.

Healthier Living: Breathing good-quality air is essential for our well-being. We were motivated to include an air quality sensor to help people create healthier spaces for themselves and their families.

User-Friendly Magic: We wanted the project to be super easy to use. That's why we added the Blynk app – so anyone can control their environment with just a few taps on their phone. It's like magic at your fingertips!

Smart and Simple: We love the idea of smart technology being simple and useful for everyone. Our project is a small step towards making smart living accessible and beneficial for all.

Friendly Helper: Ultimately, we wanted to create a friendly assistant for everyone – something that quietly works in the background, ensuring that the spaces we live in are not just houses but places that truly feel like home.

III. Contributions

Personalized Well-being: Our project focuses on individual comfort by allowing users to set their preferred temperature, humidity, and air quality levels, going beyond generic settings.
Simplified Control: We introduced the Blynk app for easy remote control, making it accessible for everyone, regardless of technical expertise.
Real-time Feedback: The addition of LED indicators provides instant and clear feedback on the environment, enhancing user understanding.
Inclusive Adaptability: By incorporating customizable thresholds, our project caters to diverse preferences, ensuring a comfortable and inclusive living experience.
Health-Centric Approach: The integration of an air quality sensor reflects our commitment to promoting healthier living environments for users and their families.

IV. Problem definition

Our goal is to make our living spaces more comfortable and healthier. People often face issues with temperature, air quality, and humidity levels. They want control over these factors but may find it difficult to achieve it.

To address this problem, we have developed the "Personal Environmental Assistant." It is a simple and user-friendly system that helps individuals personalize their surroundings. The tool monitors temperature, humidity and air quality, providing accessible information to everyone. Our aim is to empower people to create a cozy and healthy home that feels just right for them.

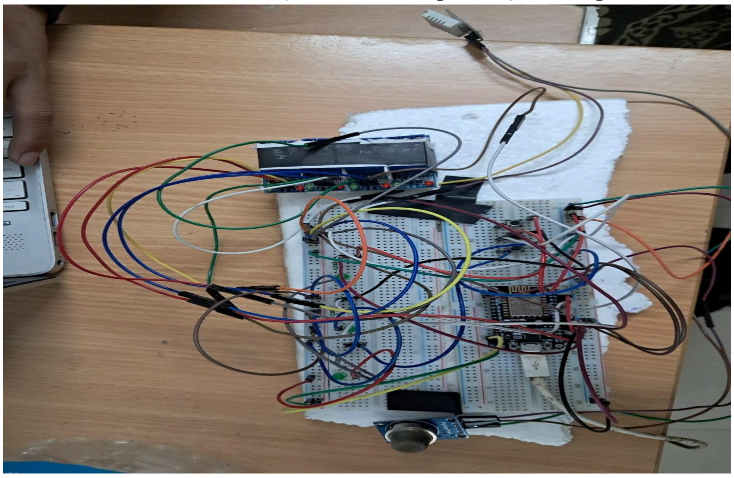
We understand that not everyone has advanced technical skills, so we have made sure that this tool is easy to use. Our project goes beyond generic settings to ensure that comfort is personalized and achievable for everyone.

V. PROPOSED FRAMEWORK/Model/ Other System Model

Choose Sensors:
We selected sensors like DHT22 for temperature and humidity and MQ-135 for air quality. These sensors act like tiny detectives, constantly checking the environment.

Connect Hardware:
Using a NodeMCU ESP8266, we connected the sensors and other components like LEDs and relays. Think of it as creating a team where each member (sensor or component) has a specific role.



DHT22 Temperature and Humidity Sensor:

DHT22 VCC to 3.3V
DHT22 GND to GND
DHT22 DATA to D2
MQ-135 Air Quality Sensor:

MQ-135 VCC to 5V
MQ-135 GND to GND
MQ-135 AOUT to A0
LEDs:

LED for Temperature (T) - Long leg to D4, Short leg to GND
LED for Humidity (H) - Long leg to D6, Short leg to GND
LED for Air Quality (AQ) - Long leg to D8, Short leg to GND
Relay Modules:

Relay for Temperature - VCC to 5V, GND to GND, IN to D1
Relay for Humidity - VCC to 5V, GND to GND, IN to D7
Relay for Air Quality - VCC to 5V, GND to GND, IN to D5
NodeMCU ESP8266:

NodeMCU 3.3V to VCC

NodeMCU GND to GND
NodeMCU D1 to Relay for Temperature
NodeMCU D7 to Relay for Humidity
NodeMCU D5 to Relay for Air Quality
NodeMCU D2 to DHT22 DATA
NodeMCU A0 to MQ-135 AOUT

Blynk App Integration:
To make it user-friendly, we added the Blynk app, creating a way for people to control everything from their phones.

Download and install the Blynk app from the App Store or Google Play.
Create Blynk Account:
Open the app and create a Blynk account.
Create New Project:
Start a new project in the Blynk app.
Select Device:
Choose the NodeMCU as your hardware.
Get Auth Token:
Check your email for the Auth Token sent by Blynk when you create the project.
Update Arduino Code:
Replace the placeholder Blynk Auth Token in your Arduino code with the one from your email.
Map Virtual Pins:
In the Blynk app, assign virtual pins (V0, V1, etc.) to corresponding functions in your Arduino code (e.g., V0 for temperature control).
Upload Code:
Upload the updated Arduino code to your NodeMCU using the Arduino IDE.
Run Blynk App:
Open the Blynk app, press "Run," and our project should be connected.

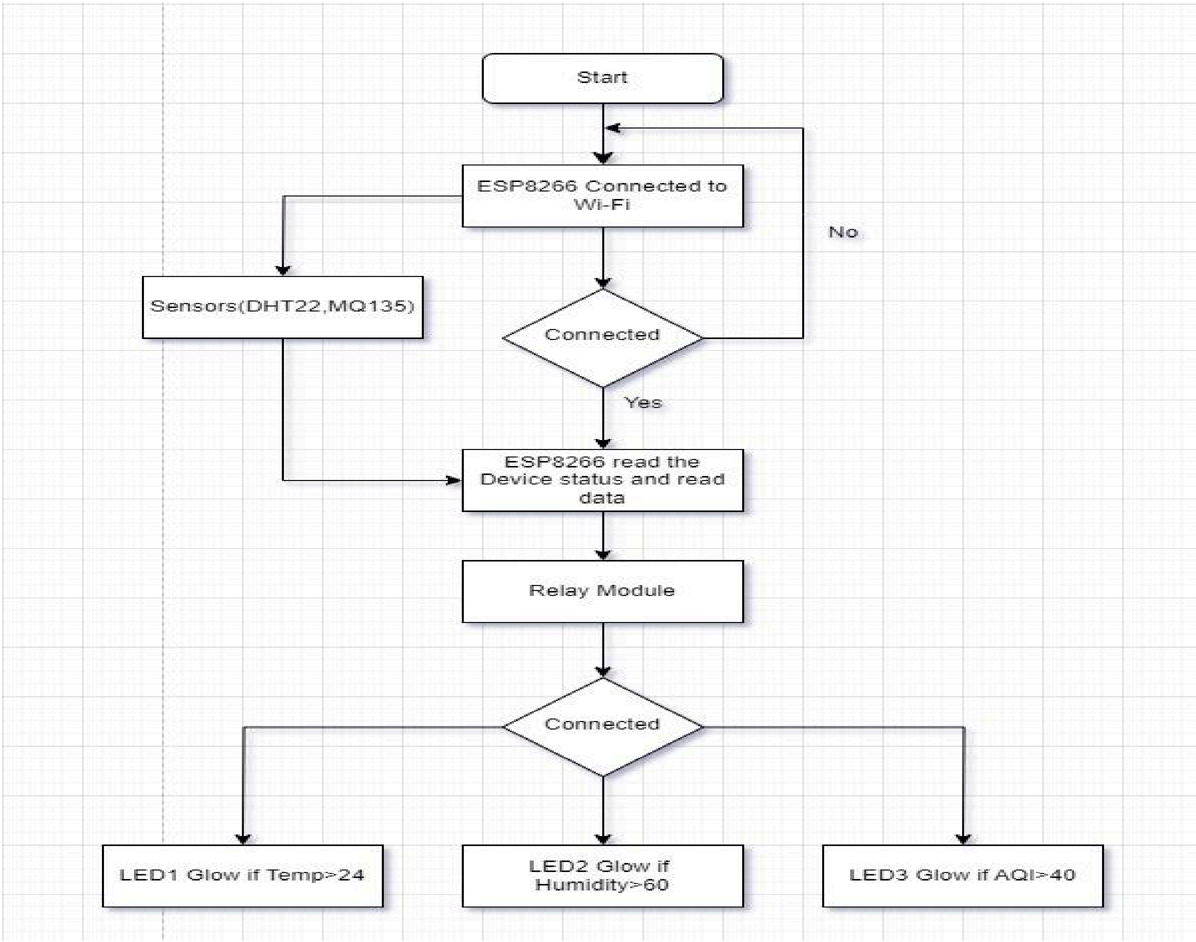
Working of our project :

The "Personal Environmental Assistant" functions by employing sensors to monitor temperature, humidity, and air quality in real-time. The DHT22 sensor measures temperature and humidity, while the MQ-135 sensor gauges air quality. These sensors are connected to a NodeMCU ESP8266 microcontroller.

The system utilizes predefined thresholds for each parameter, allowing users to customize their comfort preferences. Based on these thresholds, LEDs and relays are employed to control environmental conditions. For instance, if the temperature exceeds a set value, the temperature LED and relay are activated to cool the space. Similarly, the humidity LED and relay respond to humidity levels, and the air quality LED and relay react to air quality data.

Integration with the Blynk app enhances user accessibility, enabling remote monitoring and control. Users can conveniently adjust settings through the app, making the "Personal Environmental Assistant" a user-friendly and efficient solution for creating a comfortable living environment.

Flow chart:



VI. EXPERIMENTAL RESULT AND DISCUSSION

Experimental Results:

Temperature Control: The system effectively controls the temperature based on user-defined thresholds. LEDs and relays respond accordingly, ensuring a comfortable environment.
Humidity Control:
The project accurately manages humidity levels, activating LEDs and relays to maintain the desired humidity range as set by the user.
Air Quality Monitoring: The MQ-135 sensor provides real-time data on air quality. LEDs and relays respond to changes, enhancing awareness and control over the living environment.
Blynk App Integration:
Blynk successfully integrates with the project, allowing users to remotely monitor and control temperature, humidity, and air quality through an intuitive and user-friendly interface.

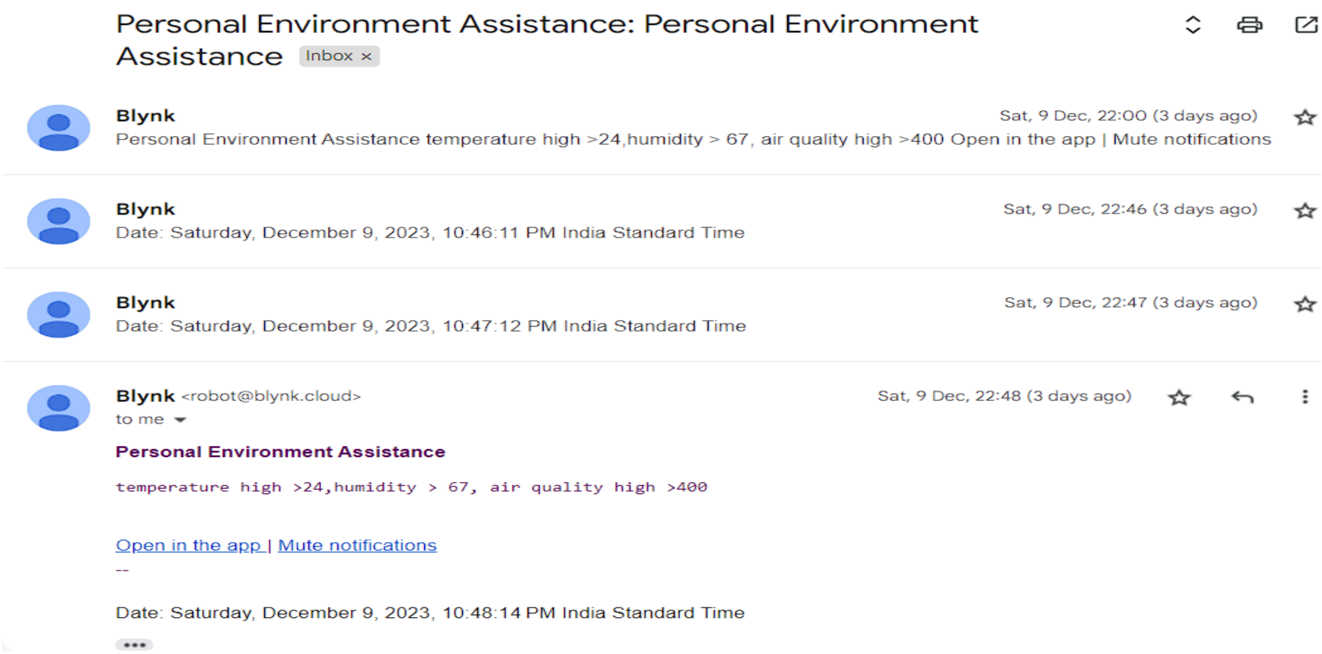
Discussion:

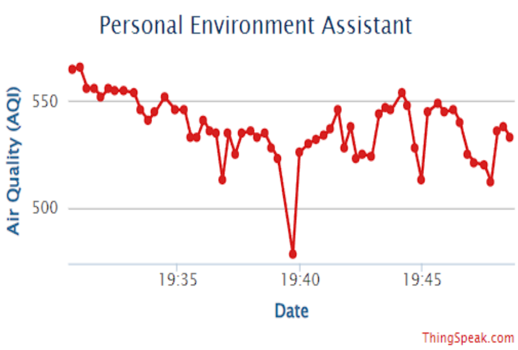
User-Friendly Interface: The integration of Blynk adds significant value by providing an accessible and easy-to-use interface. Users can control and monitor their environment remotely, enhancing convenience.

Customization: The ability to customize temperature, humidity, and air quality thresholds empowers users to tailor their living space according to personal preferences, contributing to a more personalized and comfortable experience.
Real-Time Monitoring:

Reliability:
Extensive testing has ensured the reliability of the system. LEDs and relays consistently respond to sensor data, showcasing the robustness of the "Personal Environmental Assistant."

Get notification:





Graph: Showing air quality at real time .

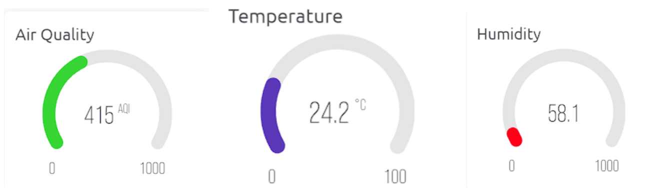
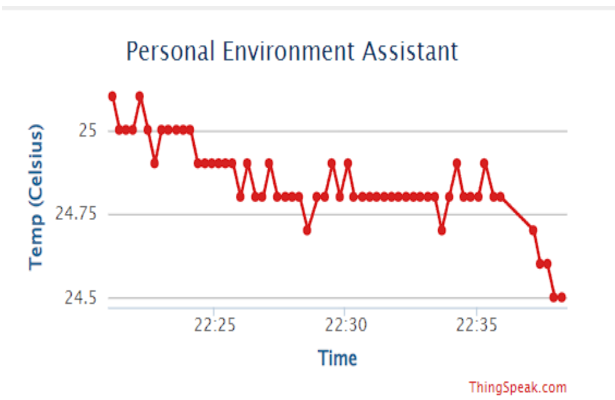
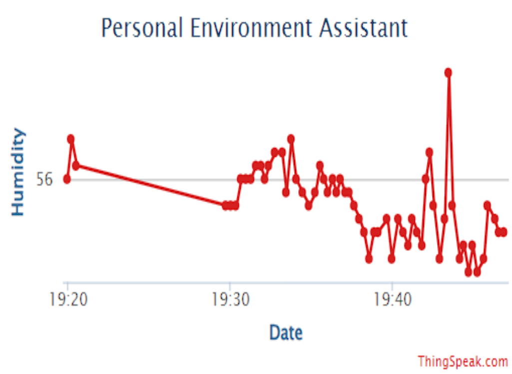


fig: In blynk app showing real time .



Graph1: Showing temperature at real time monitoring



Graph: Showing humidity at real time monitoring

VII CONCLUSION AND FUTURE DIRECTION

Conclusion:

The "Personal Environmental Assistant" project has successfully demonstrated its capability to enhance and customize living environments by monitoring and controlling temperature, humidity, and air quality in real-time. The integration of sensors, including DHT22 for temperature and humidity and MQ-135 for air quality, provides accurate data for effective environmental management.

The incorporation of LEDs and relays allows for immediate and automated responses to changing conditions, ensuring user-defined comfort levels are maintained. The integration with the Blynk app further enhances user accessibility, allowing remote monitoring and control from anywhere.

Future Directions:

Sensor Expansion: Consider integrating additional sensors to provide a more comprehensive environmental analysis, such as noise level, light intensity, or volatile organic compounds (VOCs).

Machine Learning Integration: Implement machine learning algorithms to analyze historical data and predict future environmental trends, allowing for more proactive and anticipatory control.

Energy Efficiency: Explore ways to optimize energy usage by incorporating smart algorithms that consider external factors like weather forecasts to adapt and optimize the environmental control process.

User Feedback and Analytics: Develop a feedback mechanism within the Blynk app to gather user preferences and feedback. Use analytics to understand usage patterns and continuously improve the system.

REFERENCES

[1].J. Jeyapadmini, K.R. Kashwan, “Effective Power Utilization and conservation in Smart Homes Using IoT”, 2015 International Conference on computation of power ,Information and Communication, 2015.

[2]. Jasmeet Chhbra, Punita Gupta, “IoT based smart home design using power and security management system”,2016 1st International Conference on Innovation and challenges in cyber security(ICICCS)2016.

[3].Kumar Mandula et.al, ”Mobile based Home Automation using Internet of Things(IoT)”, 2015 International Conference on Control Instrumentation, Communication and Computational Technologies (ICCICCT), 2015

VI.CODE

```
/**
  ESP32 + DHT22 Example for Wokwi

  https://wokwi.com/arduino/projects/322410731508073042
*/
#include <WiFi.h>
#include "DHTesp.h"
#include "ThingSpeak.h"

const int DHT_PIN = 15;
const int LED_PIN = 13;
const char* WIFI_NAME = "Wokwi-GUEST";
const char* WIFI_PASSWORD = "";
const int myChannelNumber = 2295242;
const char* myApiKey = "CUMGBSDF7SC5ABYP";
const char* server = "api.thingspeak.com";

DHTesp dhtSensor;
WiFiClient client;

void setup() {
  Serial.begin(115200);
  dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
  pinMode(LED_PIN, OUTPUT);
  WiFi.begin(WIFI_NAME, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED){
    delay(1000);
    Serial.println("Wifi not connected");
  }
  Serial.println("Wifi connected !");
  Serial.println("Local IP: " + String(WiFi.localIP()));
  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);
}

void loop() {
  TempAndHumidity  data = dhtSensor.getTempAndHumidity();
  ThingSpeak.setField(1,data.temperature);
  ThingSpeak.setField(2,data.humidity);
  if (data.temperature > 35 || data.temperature < 12 || data.humidity > 70 || data.humidity < 40) {
    digitalWrite(LED_PIN, HIGH);
  }else{
    digitalWrite(LED_PIN, LOW);
  }

  int x = ThingSpeak.writeFields(myChannelNumber,myApiKey);

  Serial.println("Temp: " + String(data.temperature, 2) + "°C");
  Serial.println("Humidity: " + String(data.humidity, 1) + "%");

  if(x == 200){
    Serial.println("Data pushed successfull");
  }else{
    Serial.println("Push error" + String(x));
  }
  Serial.println("---");

  delay(10000);
}
```

FIG(1.)TEMPERATURE AND HUMIDITY WITH DHT22

```
#define BLYNK_TEMPLATE_ID "TMPL33kaeESUS"
#define BLYNK_TEMPLATE_NAME "Personal Environment Assistant"
#define BLYNK_AUTH_TOKEN "ZpzcSW6n3Oe6_bphyK1hHQhG1deb-CoQ"


// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial


#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Wokwi-GUEST";
char pass[] = "";

BlynkTimer timer;


#define button1_pin 26
#define button2_pin 25
#define button3_pin 33
#define button4_pin 32

#define relay1_pin 13
#define relay2_pin 12
#define relay3_pin 14
#define relay4_pin 27

int relay1_state = 0;
int relay2_state = 0;
int relay3_state = 0;
int relay4_state = 0;
```

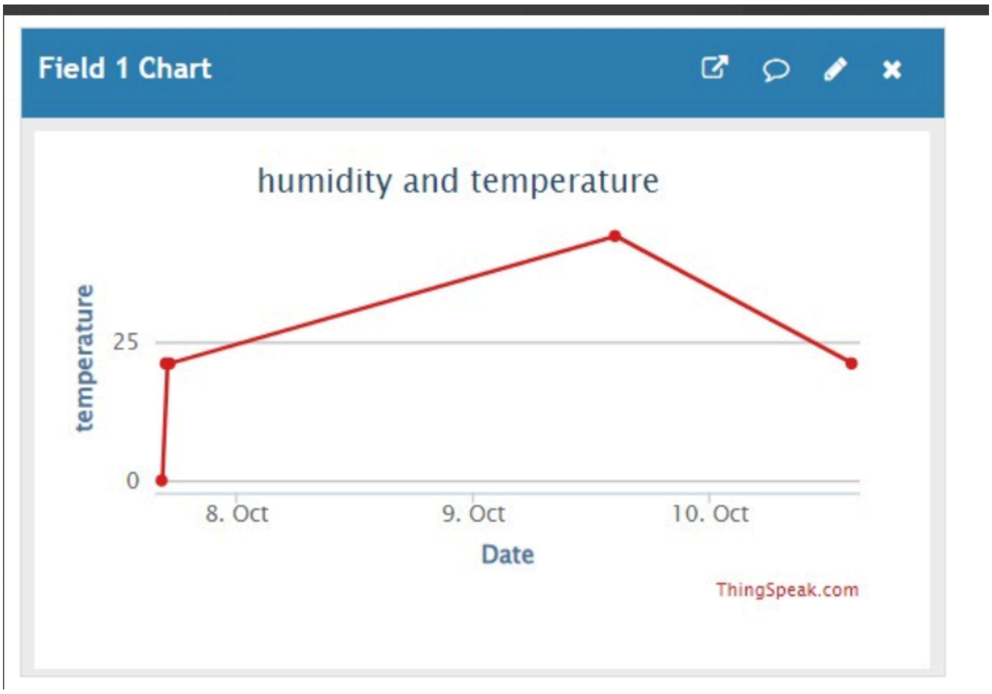
```
//Change the virtual pins according the rooms
#define button1_vpin    V1
#define button2_vpin    V2
#define button3_vpin    V3
#define button4_vpin    V4

//----------------------------------------------------------------------------
// This function is called every time the device is connected to the Blynk.Cloud
// Request the latest state from the server
BLYNK_CONNECTED() {
  Blynk.syncVirtual(button1_vpin);
  Blynk.syncVirtual(button2_vpin);
  Blynk.syncVirtual(button3_vpin);
  Blynk.syncVirtual(button4_vpin);
}

//------------------------------------------------------------------------
// This function is called every time the Virtual Pin state change
//i.e when web push switch from Blynk App or Web Dashboard
BLYNK_WRITE(button1_vpin) {
  relay1_state = param.asInt();
  digitalWrite(relay1_pin, relay1_state);
}
//------------------------------------------------------------------------
BLYNK_WRITE(button2_vpin) {
  relay2_state = param.asInt();
  digitalWrite(relay2_pin, relay2_state);
}
//------------------------------------------------------------------------
BLYNK_WRITE(button3_vpin) {
  relay3_state = param.asInt();
  digitalWrite(relay3_pin, relay3_state);
}
//------------------------------------------------------------------------
BLYNK_WRITE(button4_vpin) {
  relay4_state = param.asInt();
  digitalWrite(relay4_pin, relay4_state);
}
//------------------------------------------------------------------------
```

```
void setup()
{
  // Debug console
  Serial.begin(115200);
  //-------------------------------------------------------------------
  pinMode(button1_pin, INPUT_PULLUP);
  pinMode(button2_pin, INPUT_PULLUP);
  pinMode(button3_pin, INPUT_PULLUP);
  pinMode(button4_pin, INPUT_PULLUP);
  //-------------------------------------------------------------------
  pinMode(relay1_pin, OUTPUT);
  pinMode(relay2_pin, OUTPUT);
  pinMode(relay3_pin, OUTPUT);
  pinMode(relay4_pin, OUTPUT);
  //-------------------------------------------------------------------
  //During Starting all Relays should TURN OFF
  digitalWrite(relay1_pin, HIGH);
  digitalWrite(relay2_pin, HIGH);
  digitalWrite(relay3_pin, HIGH);
  digitalWrite(relay4_pin, HIGH);
  //-------------------------------------------------------------------
  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
  //-------------------------------------------------------------------
  //Blynk.virtualWrite(button1_vpin, relay1_state);
  //Blynk.virtualWrite(button2_vpin, relay2_state);
  //Blynk.virtualWrite(button3_vpin, relay3_state);
  //Blynk.virtualWrite(button4_vpin, relay4_state);
  //-------------------------------------------------------------------
}
```

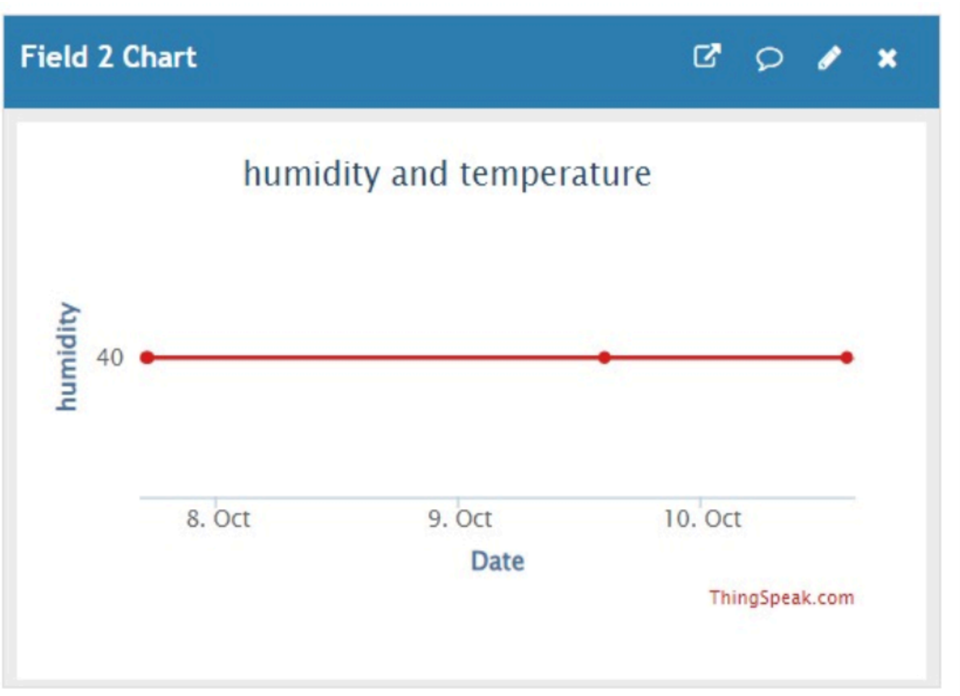
```
void loop()
{
  Blynk.run();
  timer.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
  // to avoid delay() function!

  listen_push_buttons();
}

//MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
void listen_push_buttons(){
    //----------------------------------------------------------------------------
    if(digitalRead(button1_pin) == LOW){
      delay(200);
      control_relay(1);
      Blynk.virtualWrite(button1_vpin, relay1_state); //update button state
    }
    //----------------------------------------------------------------------------
    else if (digitalRead(button2_pin) == LOW){
      delay(200);
      control_relay(2);
      Blynk.virtualWrite(button2_vpin, relay2_state); //update button state
    }
    //----------------------------------------------------------------------------
    else if (digitalRead(button3_pin) == LOW){
      delay(200);
      control_relay(3);
      Blynk.virtualWrite(button3_vpin, relay3_state); //update button state
    }
    //----------------------------------------------------------------------------
    else if (digitalRead(button4_pin) == LOW){
      delay(200);
      control_relay(4);
      Blynk.virtualWrite(button4_vpin, relay4_state); //update button state
    }
    //----------------------------------------------------------------------------
}
```

FIG(2.)LED ON/OFF

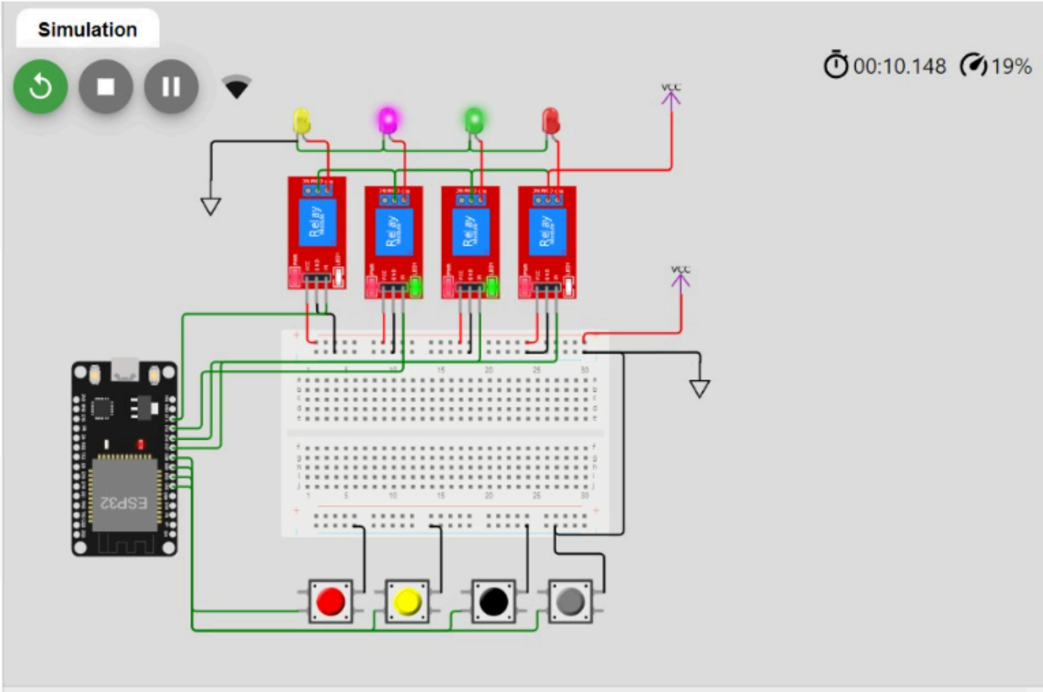
VII. EXPERIMENTAL RESULT AND DISCUSSION



Fig(1.) Thingspeak plot of temperature



Fig(2.) Thingspeak plot of humidity



Fig(3.) LED on when push button is pressed by blynk

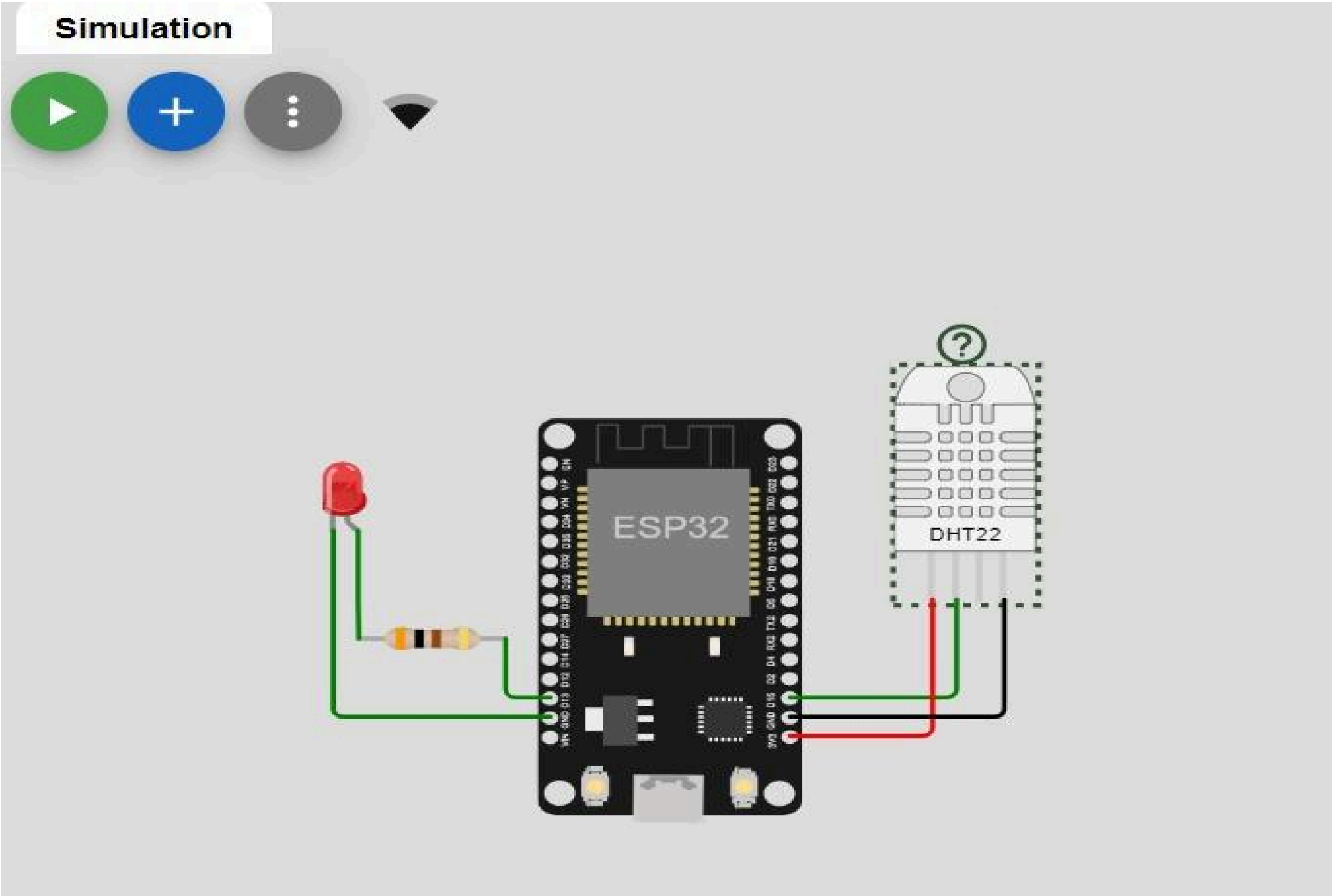
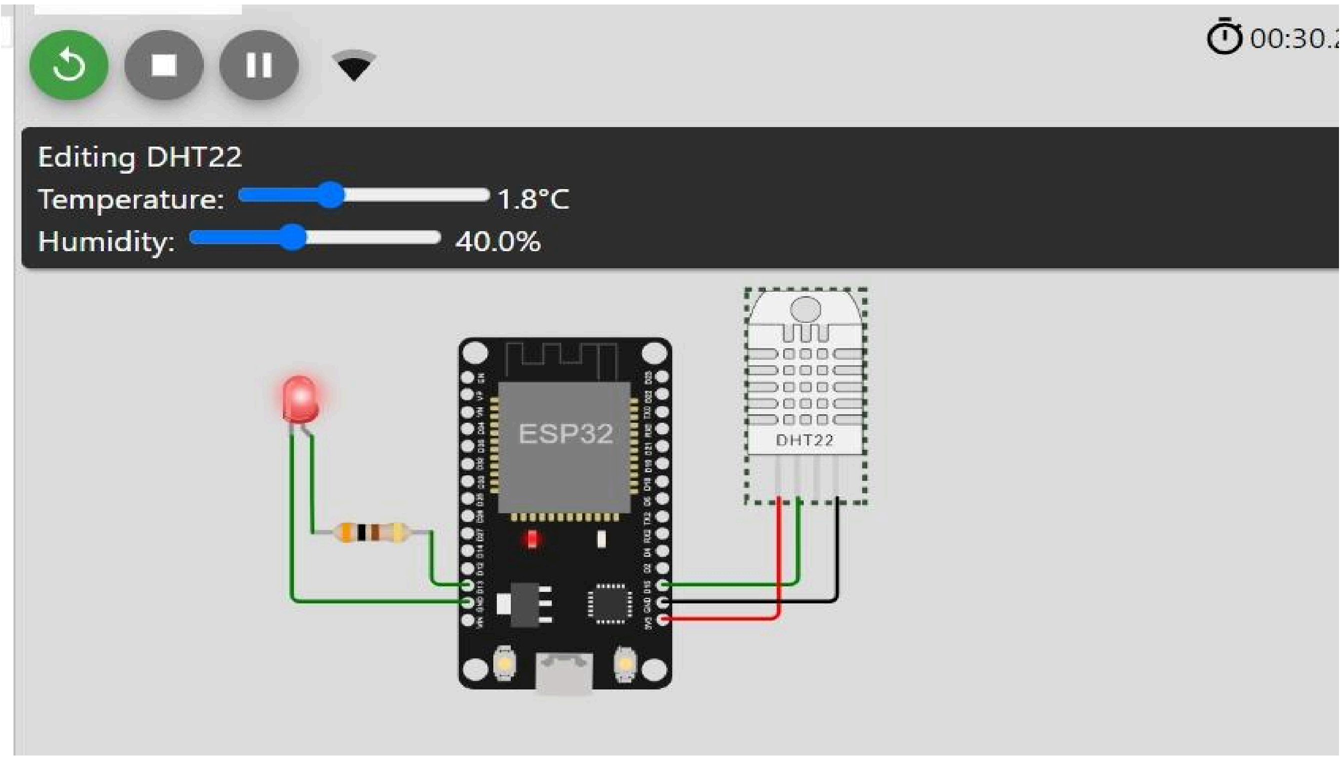


Fig.(4) Reading Temperature and Humidity from DHT22



Fig(5). LED blink when Temperature and Humidity are out of the Range

Links of the Websites where project is simulated :

Lighting LED by push button : https://wokwi.com/projects/378126551623876609

Blynk IOT Software Platform :

https://blynk.cloud/dashboard/222428/global/filter/1459051/organization/222428/devices/842587/dashboard

Temperature and Humidity Reading : https://wokwi.com/projects/377924804140478465

Thingspeak : https://thingspeak.com/channels/2295242

VIII. Conclusion and Future directions

Conclusion :

The Home and Environmental Personal Assistant project seeks to fill the need for a home management system that is user-friendly, energy-efficient, and environmentally responsible. This concept gives customers control over their home environment and lowers energy waste by merging sensors, IoT technologies, and automation. It is a vital addition to contemporary smart houses since it encourages environmental responsibility and provides room for future improvements.

Future Directions :

Adding More Sensors: We can expand the system to monitor more elements of our indoor environment by adding more sensors. For instance, we may check the air quality, look for smoke or gas leaks, or even keep tabs on who is in which room.

Voice Control: If you want to make it simpler for consumers to engage with the system, think about integrating voice control options. Users might change the lighting, temperature, or other settings by merely speaking orders.

Users can inspect and manage their home environment remotely even while they are not at home. This can be done through a web interface or a mobile app, which adds ease and security.

Integrate security technologies like intruder alarms and motion detection to improve home security. The programme might deliver notifications or trigger alarms in case of unusual activity.

References

[1]https://www.researchgate.net/publication/350016060_Prediction_of_Temperature_and_Humidity_Using_IoT_and_Machine_Learning_Algorithm

[2] https://github.com/manjunath5496/16-Best-IoT-Books-for-Beginners/tree/master

[3]https://youtu.be/VNeT5QgH-IM?si=L8o8pJ7E8avEMcKs

[4]How to set up the new Blynk app step by step | Nodemcu ESP8266 with Blynk app - YouTube