PROJECT REPORT
HOME AUTOMATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF

THE DEGREE OF

BACHELOR OF TECHNOLOGY

(Electronics and Communication Engineering)

JAN-JUNE 2023

SUBMITTED BY:

DHRUV SHARMA(12002020)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PUNJABI UNIVERSITY, PATIALA

INDEX

CHAPTER – 1

HOME AUTOMATION

1.1 INTRODUCTION

Smart homes are becoming increasingly popular with the advancement in technology. A smart home is a home equipped with technology that enables remote monitoring and control of various household appliances and systems. In this project, we aim to design and develop a smart home system that allows users to control various household appliances and systems using their smartphones.

The proposed smart home system consists of various components, including sensors, controllers, and communication devices. We will use various sensors such as temperature sensors, humidity sensors and motion sensors to collect data about the environment in the house. The collected data will be transmitted to a controller that will analyse it and make decisions based on the data. The controller will then send commands to various appliances and systems to control them.

The communication between the controller and the appliances will be through a wireless network. We will use Wi-Fi or Bluetooth technology for this purpose. The users will be able to access the system using their smartphones, where they will have a user interface to control various appliances and systems.

For the implementation of the smart home system, we will use a microcontroller board such as the Arduino or Raspberry Pi. We will connect various sensors to the microcontroller board and write code to collect data from the sensors. We will also write code to analyse the collected data and make decisions based on it.

For controlling various appliances and systems, we will use relays. The relays will be connected to the microcontroller board, and we will write code to control the relays based on the decisions made by the microcontroller board. The relays will be used to turn appliances on and off, adjust their settings, and control other systems such as lighting and security.

To allow users to access the system using their smartphones, we will develop a mobile application. The mobile application will have a user interface where users can control various appliances and systems in their homes.

1.1.1 OBJECTIVE

Home automation refers to the integration of technology with the everyday appliances and systems in your home to provide improved convenience, comfort, and security. The objective of home automation is to make your life easier and more efficient by automating routine tasks and providing remote access and control over your home systems.

The following are some key objectives of home automation:

Convenience and Comfort: One of the main objectives of home automation is to make your life more convenient and comfortable by automating routine tasks. With home automation, you can control your lighting, temperature, entertainment systems, and other appliances with just a touch of a button or voice command, making your home more comfortable and easier to manage.

Energy Efficiency: Another objective of home automation is to reduce energy consumption and lower your energy bills. With the help of smart thermostats, lighting controls, and energy-efficient appliances, you can monitor and manage your energy usage and reduce waste.

Security and Safety: Home automation can also improve the security and safety of your home. With the help of smart locks, cameras, and motion sensors, you can monitor your home remotely and receive alerts in case of any suspicious activity or emergencies.

Remote Access: With the help of home automation, you can access and control your home systems from anywhere in the world using your smartphone or tablet. This means you can turn on/off lights, adjust the temperature, and monitor your security cameras even when you're away from home.

Personalization: Home automation can also provide a personalized experience tailored to your individual preferences. With the help of machine learning and AI, your home can learn your habits and adjust settings accordingly, making your life more convenient and efficient.

In conclusion, the objective of home automation is to make your life easier and more efficient by automating routine tasks, improving energy efficiency, enhancing security and safety, providing remote access and personalization. By achieving these objectives, home automation can make your home a more comfortable, convenient, and secure place to live.

1.1.2 COMPONENTS USED

1 ESP32 MODULE- The ESP32 is a popular microcontroller module that features an integrated Wi-Fi and Bluetooth connectivity, as well as a dual-core processor, making it a powerful platform for a variety of IoT projects. Here are some key features of the ESP32 module:

i. Dual-core processor: The ESP32 module is powered by a dual-core Tensilica LX6 microprocessor, which enables it to handle multiple tasks simultaneously.

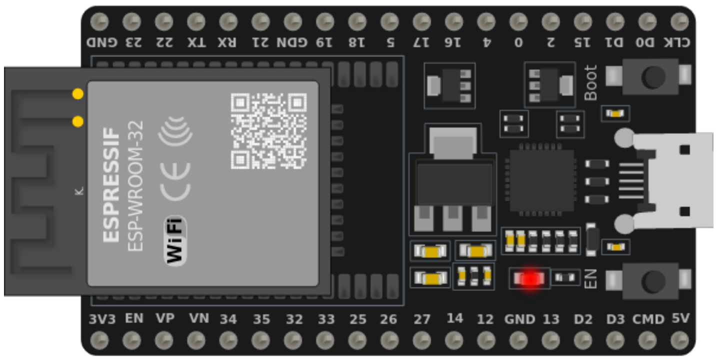
ii. Wi-Fi and Bluetooth connectivity: The module features integrated Wi-Fi and Bluetooth connectivity, making it easy to connect to the internet or other devices.

iii. Low power consumption: The ESP32 module has a low-power consumption mode, which makes it ideal for battery-powered IoT projects.

iv. Analog-to-Digital Converter (ADC): The module has a 12-bit ADC, which enables it to read analog signals from sensors.

v. GPIO pins: The ESP32 module has a large number of General Purpose Input/Output (GPIO) pins, which can be used to connect to a variety of sensors, actuators, and other devices.

vi. Programming: The ESP32 can be programmed using the Arduino IDE, as well as other programming languages such as Python and MicroPython.


GND 23 22 TX RX 21 GND 19 18 5 17 16 4 0 2 15 D1 D0 CLK
ESPRESSIF
ESP-WROOM-32
WiFi
Boot
EN
3V3 EN VP VN 34 35 32 33 25 26 27 14 12 GND 13 D2 D3 CMD 5V

2. RELAY MODULE - A relay module is an electronic device that is used to control electrical circuits by using low-power signals. It consists of a relay, which is a switch that is operated by an electromagnet, and a module that provides the necessary components to control the relay. Here are some key components of a typical relay module:

i. Relay: The relay is the main component of the module and consists of a coil, an armature, and a set of contacts. When the coil is energized, the armature moves and closes the contacts, allowing current to flow through the circuit.

ii. Driver circuit: The driver circuit provides the necessary current and voltage to control the relay. It typically consists of a transistor or other switching device and a diode to protect the transistor from voltage spikes.

iii. Input pins: The input pins of the module are used to connect to a microcontroller or other control device. When the input pin is set to high or low, the driver circuit is activated and the relay is switched on or off.

iv. Output pins: The output pins of the module are connected to the contacts of the relay. When the relay is switched on, the output pin is connected to the circuit being controlled, allowing current to flow through the circuit.

Relay modules are commonly used in a variety of electronic and electrical applications, such as home automation, robotics, and industrial control systems. They can be controlled by a wide range of devices, including microcontrollers, sensors, and switches, and can switch high currents and voltages with relatively low-power control signals.

3. BME 280 - The BME280 is a digital sensor that is commonly used to measure temperature, pressure, and humidity in various applications. It is manufactured by Bosch Sensortech and is based on microelectromechanical system (MEMS) technology.

The BME280 sensor integrates three sensors - a temperature sensor, a pressure sensor, and a humidity sensor - into a single package. It communicates with the host system using the I2C or SPI interface, and provides calibrated and temperature-compensated digital output for each of the measured parameters.

Some of the common applications of the BME280 sensor include:

i. Weather monitoring: to measure temperature, pressure, and humidity for weather forecasting and analysis.

ii. Indoor air quality monitoring: to measure temperature and humidity levels in buildings for comfort and energy efficiency purposes.

ii. Altitude measurement: to measure changes in atmospheric pressure and calculate altitude changes for navigation and aviation applications.

iv. Internet of Things (IoT) devices: to monitor environmental conditions in smart homes, smart cities, and other IoT applications.

The BME280 sensor is a popular choice for many electronic projects due to its compact size, low power consumption, and high accuracy.

4. IR SENSOR - An IR (Infrared) sensor is a type of electronic device that can detect and measure infrared radiation in its surrounding environment. Infrared radiation is a form of electromagnetic radiation with longer wavelengths than those of visible light, and it is emitted by all objects that have a temperature above absolute zero.

IR sensors are commonly used in a wide range of applications, including:

i. Proximity sensors: to detect the presence of an object by measuring the amount of infrared radiation reflected back from it.

ii. Temperature sensors: to measure the temperature of an object or a surrounding environment based on the amount of infrared radiation emitted by it.

iii. Motion sensors: to detect movement by measuring changes in the amount of infrared radiation emitted by objects in the sensor's field of view.

iv. Security systems: to detect the presence of intruders by monitoring changes in infrared radiation levels in a room or area.

IR sensors can be found in many consumer electronics devices, such as smartphones, remote controls, and smart home devices.

5. LDR MODULE - An LDR (Light Dependent Resistor) module is an electronic component that is used to detect the intensity of light in its surrounding environment. It is also known as a photoresistor module.

The module consists of an LDR sensor and a variable resistor that is used to adjust the sensitivity of the sensor. The LDR sensor is made of a special type of semiconductor material that changes its resistance based on the amount of light that falls on it. When the light intensity increases, the resistance of the LDR decreases, and when the light intensity decreases, the resistance of the LDR increases.

The LDR module is commonly used in various applications, including:

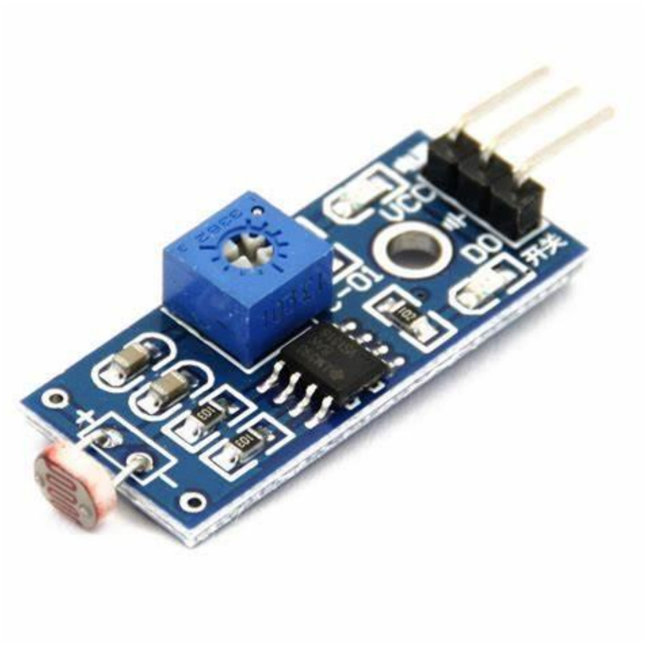
i. Streetlights: to turn on and off streetlights automatically based on the ambient light levels.

ii. Camera exposure control: to control the amount of light entering the camera lens to adjust the exposure settings.

iii. Automatic plant watering systems: to detect the amount of light received by plants and water them accordingly.

iv. Burglar alarms: to detect the presence of intruders by monitoring changes in ambient light levels.

The LDR module is a simple and cost-effective solution for light sensing applications and is widely used in many electronic projects.



6. SERVO MOTOR - A servo motor is a type of motor that is designed to provide precise control of its rotation or position. It is a rotary actuator that allows for precise control of angular position, velocity, and acceleration.

Servo motors typically have a small range of motion, typically less than 360 degrees, and are often used in robotics, automation, and other applications where precise control of movement is required. They work by using feedback from an encoder or other sensor to adjust the position of the motor shaft in response to commands from a control system.

Servo motors can be either AC or DC, and are typically characterized by their torque, speed, and power ratings. They can also be equipped with a variety of control options, such as pulse width modulation (PWM) or analog voltage control, to allow for precise control of their motion.

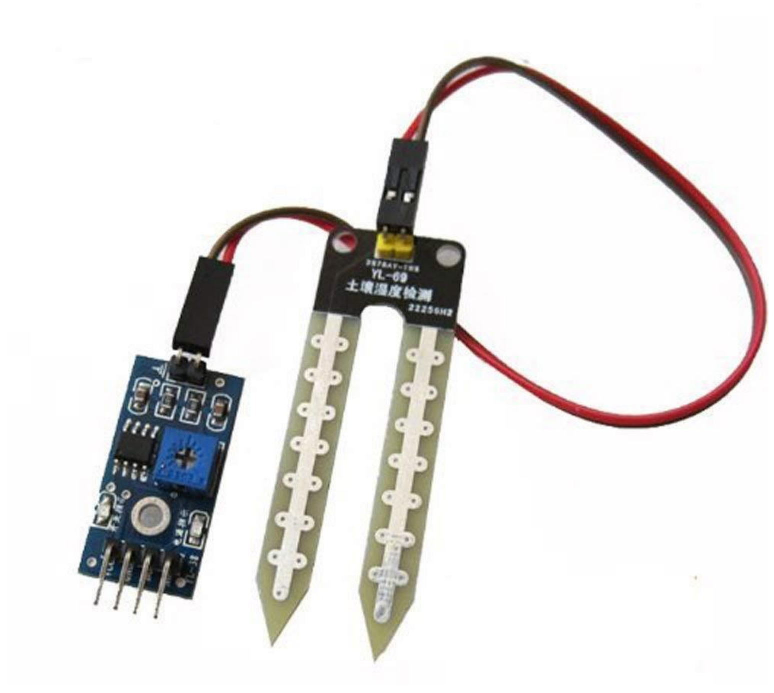
Overall, servo motors are useful for a wide range of applications that require precision and accuracy in motion control, and are commonly used in robotics, manufacturing, aerospace, and other industries

7. SOIL SENSOR - A servo motor is a type of motor that is designed to provide precise control of its rotation or position. It is a rotary actuator that allows for precise control of angular position, velocity, and acceleration.

Servo motors typically have a small range of motion, typically less than 360 degrees, and are often used in robotics, automation, and other applications where precise control of movement is required. They work by using feedback from an encoder or other sensor to adjust the position of the motor shaft in response to commands from a control system.

Servo motors can be either AC or DC, and are typically characterized by their torque, speed, and power ratings. They can also be equipped with a variety of control options, such as pulse width modulation (PWM) or analog voltage control, to allow for precise control of their motion.

Overall, servo motors are useful for a wide range of applications that require precision and accuracy in motion control, and are commonly used in robotics, manufacturing, aerospace, and other industries.



8. C++ - C++ is a popular programming language that can be used for developing software for smart home systems. Some of the key features of C++ that make it well-suited for smart home applications include:

i. Efficiency: C++ is a low-level language that allows for efficient use of system resources, which is important for smart home devices that often have limited processing power and memory.

ii. Object-oriented programming: C++ supports object-oriented programming (OOP) concepts, which can be useful for developing complex smart home systems that involve multiple interconnected devices and sensors.

iii. Libraries and frameworks: C++ has a rich set of libraries and frameworks that can be used for smart home development, such as Boost, Poco, and Qt.

iv. Cross-platform compatibility: C++ is a cross-platform language, which means that code written in C++ can be compiled and run on a variety of different hardware and operating systems.

Some examples of smart home applications that can be developed using C++ include:

i. Home automation systems that control lighting, heating, and other appliances.

ii. Security systems that monitor doors and windows, and alert homeowners to potential threats.

iii. Energy management systems that optimize energy usage and reduce energy costs.

iv. Smart appliances that can be controlled using a mobile app or voice commands.

Overall, C++ is a powerful and flexible language that can be used for developing a wide range of smart home applications. However, it requires a certain level of expertise and experience to develop complex systems in C++, so it may not be the best choice for beginners or hobbyists experience to develop complex systems in C++, so it may not be the best choice for beginners or hobbyists.

9. Arduino IDE - Arduino IDE (Integrated Development Environment) is a software platform that allows users to write, compile and upload code to the Arduino board. It was originally developed by Arduino LLC in 2003, and later on, it became an open-source platform.

The Arduino IDE is a simple yet powerful software tool for programming Arduino boards. It provides a user-friendly interface to write and upload code, making it accessible to beginners and professionals alike. The software supports multiple operating systems, including Windows, macOS, and Linux.

The Arduino IDE comes with a code editor, a compiler, and a bootloader. The code editor is used to write the code, while the compiler converts the code into machine language that can be executed by the board. The bootloader is a small program that allows the board to receive new code over the serial port.

The Arduino IDE uses a simplified version of C++ programming language, which is easy to learn and use. It comes with a set of libraries that can be used to control various components, such as sensors, motors, and displays, making it easier to work with different hardware components.

The Arduino IDE has a built-in serial monitor, which allows users to communicate with the board over the serial port. This feature is particularly useful for debugging and testing purposes.

The Arduino IDE also has a large community of developers and enthusiasts who share their knowledge and experience through forums and online tutorials. This community-driven approach makes it easier for beginners to learn and get started with Arduino.

In addition to the Arduino boards, the IDE supports various other microcontrollers, such as ESP8266 and ESP32, which can be programmed using the Arduino language and libraries.

Overall, the Arduino IDE is a powerful and flexible tool for programming and controlling microcontrollers. Its simplicity and user-friendly interface make it accessible to beginners, while its advanced features and libraries make it suitable for professionals. With the Arduino IDE, users can create a wide range of projects, from simple LED blinking to complex robotics applications.

1.2 BLOCK DIAGRAM

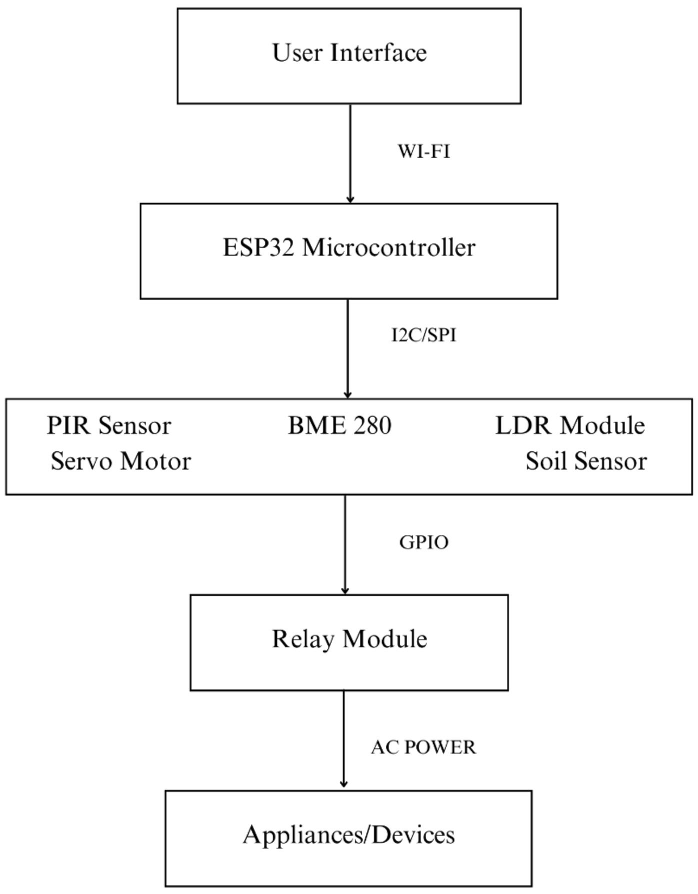


Fig 1.2.1

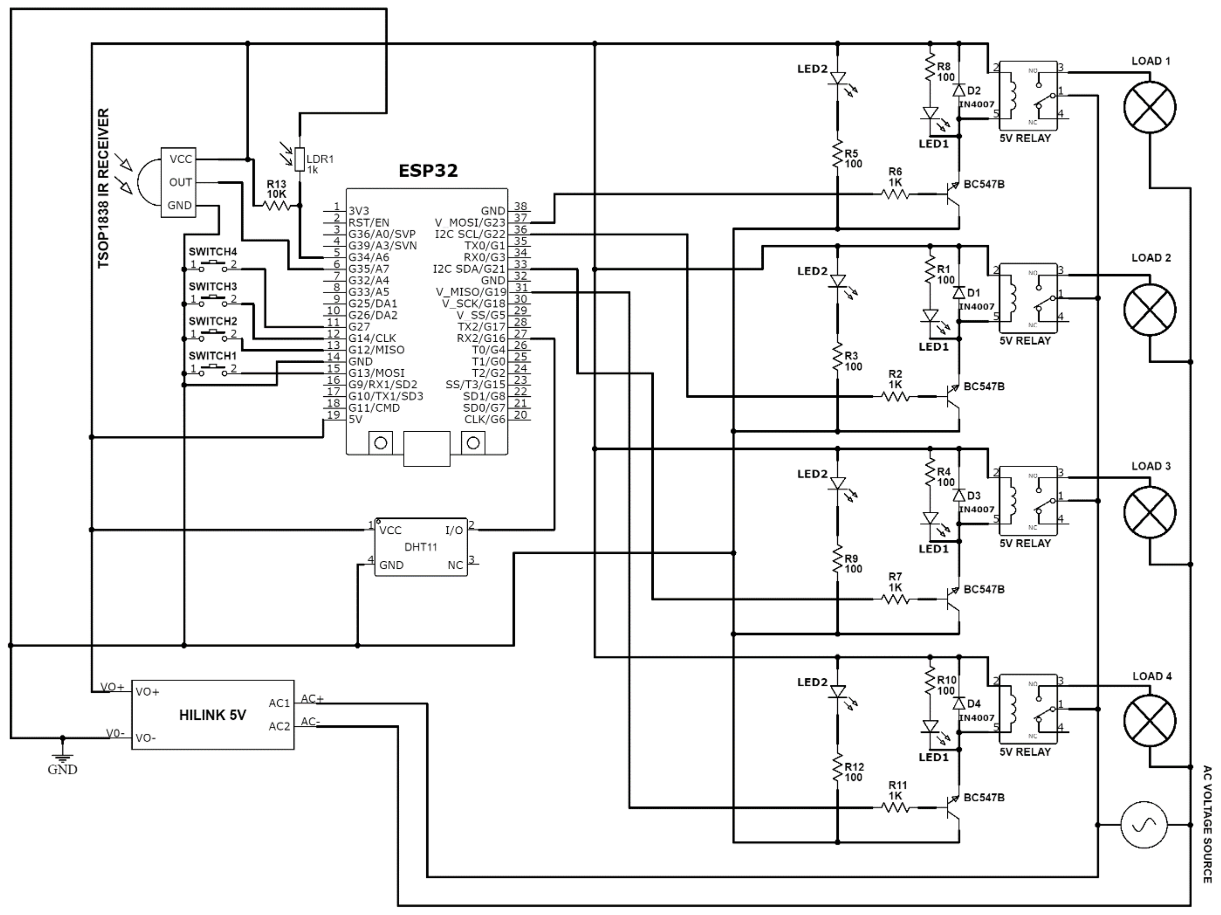
The user interacts with the system through a user interface, which may be a mobile app or a web-based interface. The user interface communicates with the ESP32 microcontroller through Wi-Fi.

The ESP32 microcontroller reads sensor data from the PIR sensor, LDR module, and BME 280 module through I2C or SPI communication protocols. It processes the sensor data and sends commands to the relay module through GPIO pins to switch on/off the connected appliances or devices.

The relay module controls the AC power to the connected appliances or devices. The appliances or devices may include lights, fans, air conditioning, or other home appliances.

1.3 CIRCUIT DIAGRAM AND CODE

1.3.1 CIRCUIT DIAGRAM



1.3.2 CODE

```
#include <WiFi.h>
#include <ESP32Servo.h>
#include <BlynkSimpleEsp32.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
//#include <HCSR04.h>
#define BLYNK_TEMPLATE_ID "TMPL3HN3UJmj-"
#define BLYNK_TEMPLATE_NAME "home automation"
#define BLYNK_AUTH_TOKEN "ur0Celf6s58cw-fvy_eR7ANGcTlWo311"
#define SEALEVELPRESSURE_HPA (1013.25)// variable saves the pressure at the sea level
in hectopascal.This variable is used to estimate the altitude for a given pressure by comparing
it with the sea level pressure.
#define irPin 14
#define pump3 15
#define fan 27
#define relayOut 25
#define relayOut4 2
#define soilsensor 33

Adafruit_BME280 bme; // Creating an object of adafruitBME_280, so that we can access
functions related to it //uses I2C communication protocol by default
Servo gateServo;


unsigned long delayTime;

char auth[] = "ur0Celf6s58cw-fvy_eR7ANGcTlWo311";
char ssid[] = "Ekam";
char pass[] = "88888888";
float humid = 0;
float temp = 0;
float altitude1 = 0;
float pressure1 = 0;
int moisturepercentage;
int analogoutput;
int IR;

BLYNK_WRITE(V4)
{
  int pValue1 = param.asInt(); // assigning incoming value from pin V1 to a variable
if(pValue1 == 0)
    {
      digitalWrite(25,HIGH);
    }
    else{
      digitalWrite(25,LOW);
    }
  // process received value
}
BLYNK_WRITE(V8)
{
  int pValue1 = param.asInt(); // assigning incoming value from pin V1 to a variable
if(pValue1 == 0)
    {
      digitalWrite(2,HIGH);
    }
    else{
      digitalWrite(2,LOW);
    }
  // process received value
}


void setup() {
  Serial.begin(115200);
  Blynk.begin( auth, ssid , pass );
  //  pinMode(12,OUTPUT);

  // digitalWrite(12,HIGH);
  pinMode(soilsensor, INPUT);

  digitalWrite(relayOut,HIGH);
  pinMode(relayOut, OUTPUT);


  digitalWrite(relayOut4,HIGH);
  pinMode(relayOut4, OUTPUT);

   digitalWrite(fan,HIGH);
   pinMode(fan, OUTPUT);

   pinMode(irPin, INPUT_PULLUP);
   gateServo.attach(5);

    digitalWrite(pump3,HIGH);
    pinMode(pump3, OUTPUT);
  // Serial.begin(115200)
  bool status;

  status = bme.begin(0x76);// address 0*76 passed to bme.begin method to start
communication with bme sensor //  This function initializes I2C interface with given I2C
Address and checks if the chip ID is correct. It then resets the chip using soft-reset & waits
for the sensor for calibration after wake-up.
  if (!status) {
    Serial.println("Could not find a valid BME280 sensor, check wiring!");
    while (1);
  }
  Serial.println("-- Default Test --");
  delayTime = 1000;

  Serial.println();
}



void loop() {





  Blynk.run();
  // Wait a few seconds between measurements.

   if(digitalRead(irPin) == LOW) {
    gateServo.write(90);
    delay(2000);
    gateServo.write(0);
   }

  analogoutput = analogRead(soilsensor);
  moisturepercentage = (100-((analogoutput/4095.00)*100));

   Serial.print(analogoutput);
  Serial.println(" *C");



  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  humid = bme.readHumidity();
  // Read temperature as Celsius (the default)
  temp = bme.readTemperature();
  pressure1 = bme.readPressure();
  altitude1 = bme.readAltitude(SEALEVELPRESSURE_HPA);
 IR=digitalRead(irPin);  /*digital read function to check IR pin status*/



  if(moisturepercentage<40){
  digitalWrite(pump3,LOW);
}

else {
  digitalWrite(pump3,HIGH);  /*if no reflection detected LED will remain OFF*/
}
  // Check if any reads failed and exit early (to try again).
  if (isnan(humid) || isnan(temp))
  {
    Serial.println("Failed to read from bme sensor!");
    return;
  }

  if (temp > 35)
  {
    digitalWrite(fan, LOW);

}
  else {
    digitalWrite(fan, HIGH);



  }

Puneet Uni, [30-04-2023 16:47]
// Compare Threshold value from Blynk and DHT Temperature value.

  Blynk.virtualWrite(V0, temp);
  Blynk.virtualWrite(V1, humid);
  Blynk.virtualWrite(V2, altitude1);
  Blynk.virtualWrite(V3, pressure1);
   Blynk.virtualWrite(V7,moisturepercentage);
   delay(2000);

}
void printValues() {
  Serial.print("Temperature = ");
  Serial.print(bme.readTemperature());
  Serial.println(" *C");



  Serial.print("Pressure = ");
  Serial.print(bme.readPressure() / 100.0F);
  Serial.println(" hPa");

  Serial.print("Approx. Altitude = ");
  Serial.print(bme.readAltitude(SEALEVELPRESSURE_HPA));
  Serial.println(" m");

  Serial.print("Humidity = ");
  Serial.print(bme.readHumidity());
  Serial.println(" %");

  Serial.println();
}
```

1.4 WORKING OF CIRCUIT AND CODE

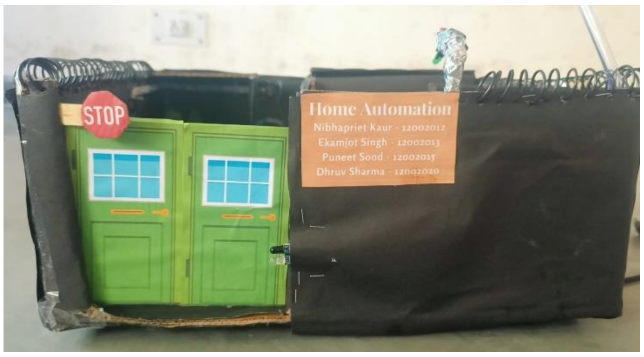


FIG 1.4.1 FRONT SIDE

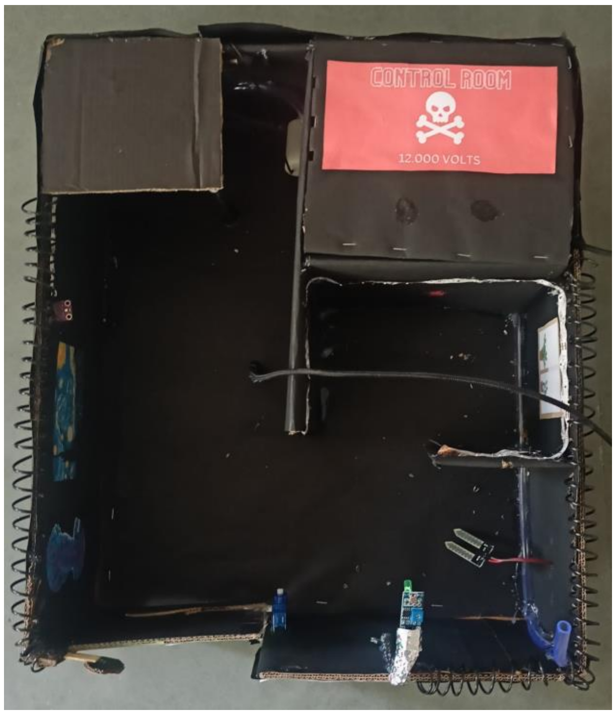


FIG 1.4.2 TOP VIEW

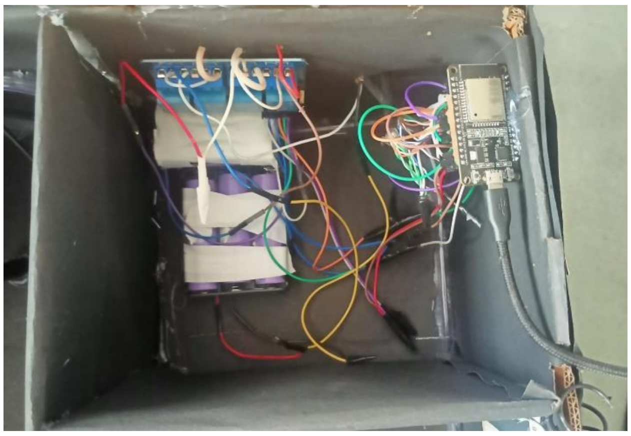


FIG 1.4.3 CONTROL ROOM

Home automation is the process of using technology to automate and control various aspects of a home, such as lighting, heating, air conditioning, and security systems. With the advent of smart home technology, home automation has become more accessible and affordable, making it possible for homeowners to enjoy a more convenient and efficient lifestyle.

The code provided is an example of how home automation can be achieved using an ESP32 board and various sensors and actuators. The code utilizes a Blynk app to provide a graphical user interface for controlling and monitoring the connected devices. In this section, we will explore the different components and functions of the code and how they contribute to home automation.

The first component of the code is the libraries that are used to control the different sensors and actuators. The WiFi library is used to establish a wireless connection to the internet, allowing the ESP32 board to communicate with the Blynk app. The ESP32Servo library is used to control the servo motor that opens and closes the gate. The Adafruit_Sensor and Adafruit_BME280 libraries are used to measure temperature, humidity, pressure, and altitude using the BME280 sensor.

The next component of the code is the setup function, which initializes the different pins and sensors and connects the ESP32 board to the Blynk app. The Blynk.begin function is called to establish a connection to the Blynk server using the authentication token and WiFi credentials. The pinMode function is used to set the direction of the different pins, and the digitalWrite function is used to set the initial state of the relay outputs.

The loop function is the main function of the code, which runs continuously and checks the readings from the different sensors. The Blynk.run function is called to handle incoming commands and update the Blynk app with the latest data. The delay function is used to wait for 2 seconds before executing the next set of commands, which helps to reduce the load on the ESP32 board and prevent it from overheating.

The first set of commands in the loop function checks the state of the IR sensor and controls the gate using the gateServo object. If the IR sensor detects an object, the servo motor is rotated to open the gate, and if there is no object, the servo motor is rotated to close the gate.

The next set of commands reads the analog output from the soil moisture sensor and converts it to a percentage using a simple formula. If the moisture percentage is less than 40, the pump is turned on to water the plants, and if it is greater than 40, the pump is turned off.

The next set of commands reads the temperature and humidity from the BME280 sensor and checks if the values are valid. If the readings are valid, the code checks if the temperature is greater than 35 degrees Celsius and turns on the fan if it is. Otherwise, the fan is turned off.

Finally, the Blynk.virtualWrite function is called to update the Blynk app with the latest data from the sensors. The temperature, humidity, altitude, and pressure readings are sent to virtual pins V0, V1, V2, and V3, respectively, while the moisture percentage is sent to virtual pin V7.

In conclusion, the code provided is an excellent example of how home automation can be achieved using an ESP32 board and various sensors and actuators. The Blynk app provides a user-friendly interface for controlling and monitoring the different devices, while the sensors and actuators automate various aspects of the home, such as watering plants and regulating temperature and humidity levels. By leveraging the power of smart home technology, homeowners can enjoy a more convenient and efficient lifestyle while reducing energy consumption and saving money on utility bills.

1.5 ADVANTAGES

There are several advantages of home automation, including:

1. Convenience: Home automation allows you to control various aspects of your home from a single device, such as your smartphone or tablet. You can turn on/off lights, adjust the thermostat, lock/unlock doors, and even start your coffee maker with the touch of a button.

2. Energy efficiency: By automating your home's lighting, heating, and cooling systems, you can save energy and reduce your utility bills. For example, you can set your thermostat to adjust the temperature based on your schedule or turn off lights in unoccupied rooms to save energy.

3. Improved security: Home automation can help you keep your home safe and secure. You can install smart locks, security cameras, and motion sensors to monitor your home and receive alerts if there is any suspicious activity.

4. Increased comfort: With home automation, you can customize the settings in your home to suit your preferences. For example, you can set the temperature to your preferred level or adjust the lighting to create the perfect ambiance.

5. Remote access: With a home automation system, you can control your home from anywhere in the world as long as you have an internet connection. This means you can check on your home, adjust settings, and receive alerts from your smartphone, tablet, or computer.

Overall, home automation can make your life easier, more comfortable, and more secure while also saving you time and money.