Project Topic:-

Home Automation System

Group Detail-

Priyanshu Gupta (BTECH/10632/20)

Divya Ucheniya (BTECH/10493/20)

1. Introduction

In the evolving landscape of smart technologies, this Home Automation System stands as a testament to the fusion of innovative hardware and software. Leveraging the ESP8266 WiFi module and Arduino Uno, the project aims to create an effortlessly controllable environment for household appliances. The core objective is to empower users to remotely manage lights, fans, and other devices through an intuitive web interface accessible via standard browsers like Google Chrome or Mozilla Firefox.

At the heart of this system is the ESP8266, a cost-effective and widely adopted WiFi module, functioning as both an access point and client. This report unfolds the intricacies of the circuit diagram, illustrating the connections between ESP8266 and Arduino, while delving into the program logic of the Arduino sketch. The HTML code, coupled with JQuery, is examined to reveal the user interface dynamics.

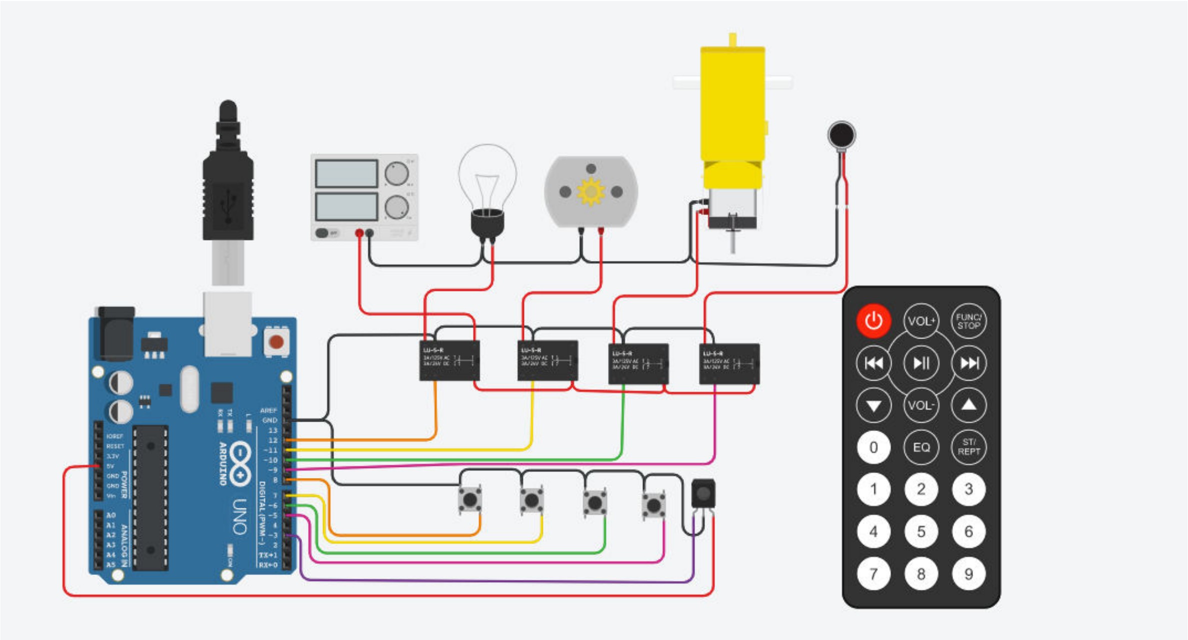
By navigating through these pages, readers will not only comprehend the technical intricacies of the ESP8266-Arduino synergy but also gain insights into potential applications within the burgeoning field of IoT-based home automation. This project epitomizes a practical and forward-looking approach, aligning with principles of efficiency, accessibility, and cost-consciousness in the pursuit of modernized living.

2. Components:

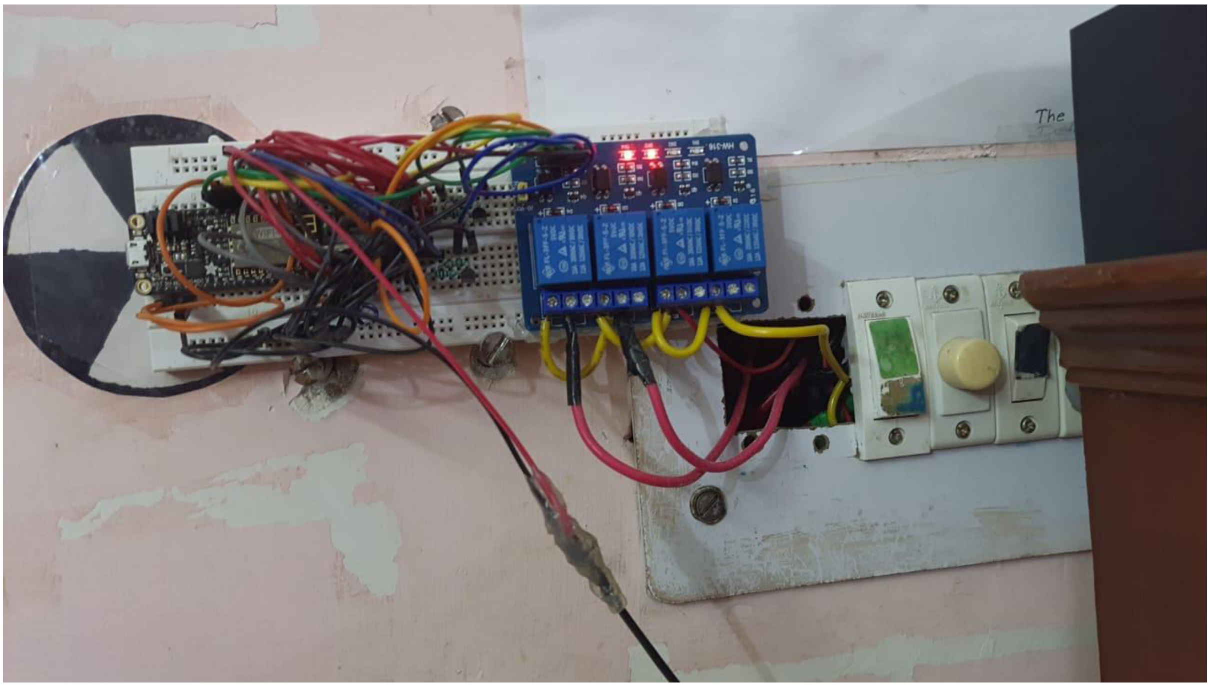
- **ESP8266 Module (ESP-01):**
  Popular WiFi module operating at 3.3V.
- **Voltage Divider:**
  Utilized to adapt the Arduino Tx pin's 5V output to the 3.3V required for the ESP8266 Rx pin.
- **Relays:**
  Electrical switches controlled by Arduino, pivotal for managing AC mains appliances.
- **Arduino Uno:**
  Microcontroller serving as the brain of the system.

3. Circuit diagram

Arduino model:



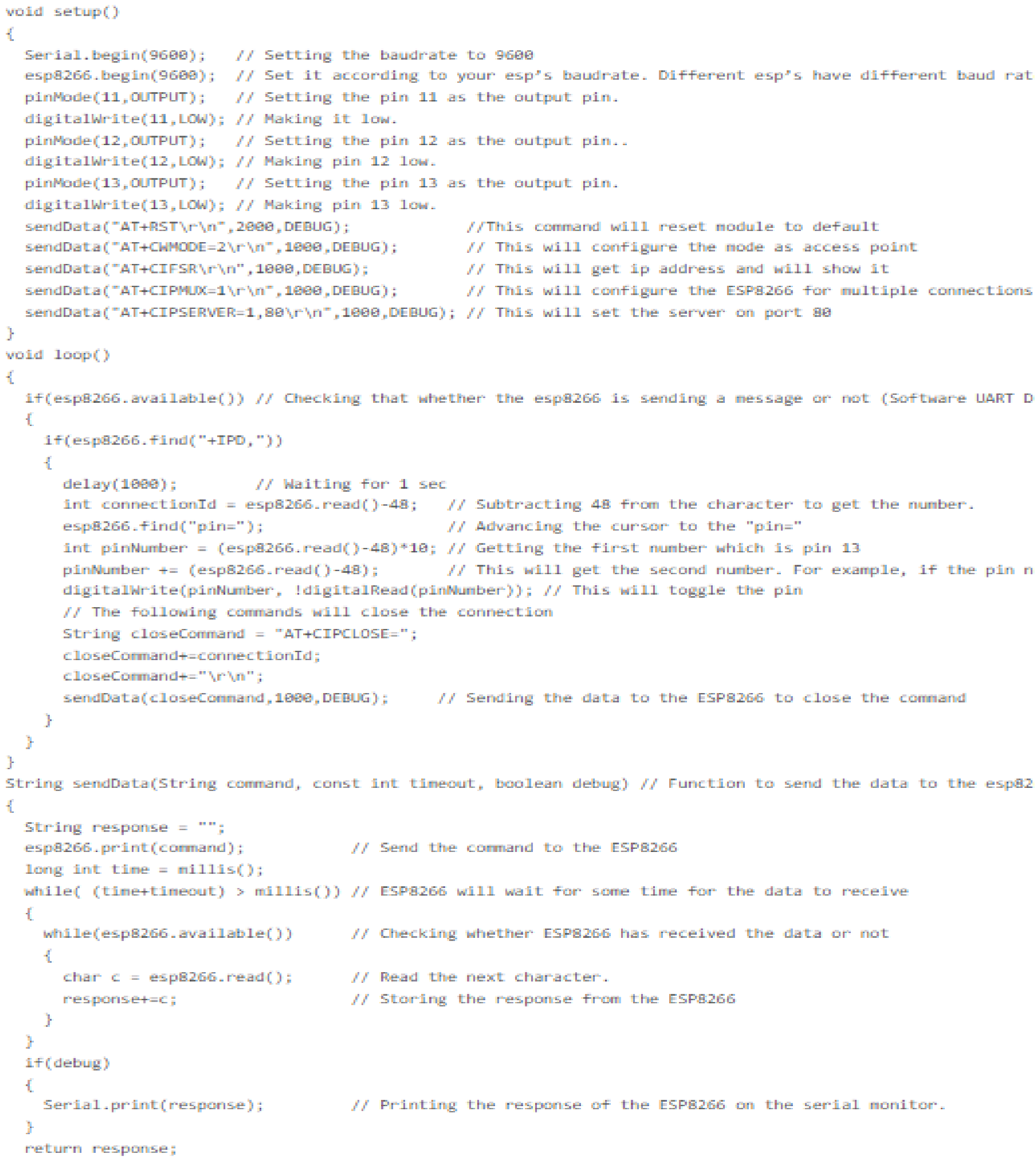
Practical application:



The ESP8266 module, chosen for its functionality, is carefully integrated with the Arduino Uno. The voltage divider ensures compatibility between the different voltage requirements of the ESP8266 and Arduino. Relays, controlled by specific Arduino pins, form the bridge between the microcontroller and AC mains appliances. This meticulous arrangement establishes the foundation for the Home Automation System's effective functionality, emphasizing clarity and reproducibility in the circuit design.

4. Program Logic and Implementation

The core logic of the project is detailed in the provided Arduino sketch. Using SoftwareSerial, the Arduino communicates with the ESP8266 module, and specific commands are sent to set up the system. The sketch configures the ESP8266 as an access point, initializes pins for relay control, and establishes a web server on port 80. The code continually listens for incoming requests, and upon receiving a request, it identifies the target pin and toggles its state. Additionally, a simple HTML code, accompanied by JQuery, facilitates user interaction by presenting toggle buttons on a web page. This section explains the Arduino sketch and HTML code, providing a comprehensive guide for users interested in implementing a similar home automation system.



```
void setup()
{
  Serial.begin(9600);   // Setting the baudrate to 9600
  esp8266.begin(9600);  // Set it according to your esp’s baudrate. Different esp’s have different baud rat
  pinMode(11,OUTPUT);   // Setting the pin 11 as the output pin.
  digitalWrite(11,LOW); // Making it low.
  pinMode(12,OUTPUT);   // Setting the pin 12 as the output pin..
  digitalWrite(12,LOW); // Making pin 12 low.
  pinMode(13,OUTPUT);   // Setting the pin 13 as the output pin.
  digitalWrite(13,LOW); // Making pin 13 low.
  sendData("AT+RST\r\n",2000,DEBUG);            //This command will reset module to default
  sendData("AT+CWMODE=2\r\n",1000,DEBUG);       // This will configure the mode as access point
  sendData("AT+CIFSR\r\n",1000,DEBUG);          // This will get ip address and will show it
  sendData("AT+CIPMUX=1\r\n",1000,DEBUG);       // This will configure the ESP8266 for multiple connections
  sendData("AT+CIPSERVER=1,80\r\n",1000,DEBUG); // This will set the server on port 80
}
void loop()
{
  if(esp8266.available()) // Checking that whether the esp8266 is sending a message or not (Software UART D
  {
    if(esp8266.find("+IPD,"))
    {
      delay(1000);         // Waiting for 1 sec
      int connectionId = esp8266.read()-48;   // Subtracting 48 from the character to get the number.
      esp8266.find("pin=");                   // Advancing the cursor to the "pin="
      int pinNumber = (esp8266.read()-48)*10; // Getting the first number which is pin 13
      pinNumber += (esp8266.read()-48);       // This will get the second number. For example, if the pin n
      digitalWrite(pinNumber, !digitalRead(pinNumber)); // This will toggle the pin
      // The following commands will close the connection
      String closeCommand = "AT+CIPCLOSE=";
      closeCommand+=connectionId;
      closeCommand+="\r\n";
      sendData(closeCommand,1000,DEBUG);     // Sending the data to the ESP8266 to close the command
    }
  }
}
String sendData(String command, const int timeout, boolean debug) // Function to send the data to the esp82
{
  String response = "";
  esp8266.print(command);           // Send the command to the ESP8266
  long int time = millis();
  while( (time+timeout) > millis()) // ESP8266 will wait for some time for the data to receive
  {
    while(esp8266.available())      // Checking whether ESP8266 has received the data or not
    {
      char c = esp8266.read();      // Read the next character.
      response+=c;                  // Storing the response from the ESP8266
    }
  }
  if(debug)
  {
    Serial.print(response);         // Printing the response of the ESP8266 on the serial monitor.
  }
  return response;
```

5. Conclusion

In conclusion, the Home Automation System utilizing the ESP8266 WiFi module and Arduino Uno provides an accessible and practical solution for controlling household appliances. The integration of hardware and software components, as detailed in this report, showcases the versatility and effectiveness of this project. The utilization of ESP8266's capabilities and Arduino's control logic demonstrates a cost-effective and efficient approach to home automation. Users can leverage this report as a guide to replicate or expand upon the system for personalized smart home solutions.