```
// Stand: 22.12.2023
// von Hajo für 1 bis 4 Relais und für feste IP-Adresse erweitert

#include <ESP8266WiFi.h>

const char* comp_date = __DATE__ ", " __TIME__;

int relais = 4;  // Anzahl Relais, 1 bis 4 möglich

 // WAN-Name und Passwort
const char* ssid = ".....";     // WLAN-Name
const char* password = "----------"; // WLAN-Passwort

// und feste IP-Adressen
IPAddress local_IP(192, 168, 1, 37);
IPAddress gateway(192, 168, 1, 1);
IPAddress subnet(255, 255, 255, 0);

WiFiServer server(80);

// Variablen zum Abfragen und zur Beeinflussung der Schaltzustände
String output1State = "off";
String output2State = "off";
String output3State = "off";
String output4State = "off";

// Variablen den einzelnen Pins zuordnen
const int output1 = 0;
const int output2 = 2;
const int output3 = 3;
const int output4 = 4;

void setup() {

  Serial.begin(115200);
  while(!Serial)
    { }  // Warte, bis die serielle Schnittstelle verbunden ist.

  delay(10000); // warten für Monitorausabe

  Serial.print("Sketch-Upload am: "); Serial.println(comp_date);

 // Die Pins als Ausgänge festlegen und auf LOW setzen
 if (relais >= 1) {pinMode(output1, OUTPUT);digitalWrite(output1, LOW);}
 if (relais >= 2) {pinMode(output2, OUTPUT);digitalWrite(output2, LOW);}
 if (relais >= 3) {pinMode(output3, OUTPUT);digitalWrite(output3, LOW);}
 if (relais == 4) {pinMode(output4, OUTPUT);digitalWrite(output4, LOW);}

 if (!WiFi.config(local_IP, gateway, subnet)) {
  Serial.println("STA Failed to configure");
 }

 // WLAN-Verbindung initiieren
 Serial.print("verbinde mit ");
 Serial.println(ssid);
 WiFi.begin(ssid, password);
 while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
 }

 // Den Web-Server starten und die IP-Adresse im Monitor ausgeben
 Serial.println("");
 Serial.print("WiFi verbunden mit ");
 Serial.print("IP address: ");
 Serial.println(WiFi.localIP());
 server.begin();
}

void loop() {

  WiFiClient client = server.available();  // auf Clients reagieren
  if (!client) {
                    // wenn der client nichts tut
                    return;
  }

   // wenn der Client Daten sendet
  Serial.println("new client");
  unsigned long TimeOut = millis() + 250;
  while (!client.available() && (millis() < TimeOut) )
  { delay(1); }
  if (millis() > TimeOut) {Serial.println("client connection time-out!");return;}

  // read the message incoming from one of the clients
  String request = client.readStringUntil('\n');
  Serial.print(request);  // im seriellen Monitor ausgegeben.
  client.flush();
  // Client stoppen, wenn keine Anforderung vorliegen
  if (request == "") {
   Serial.println("empty request! - stopping client");
   client.stop();
   return;
  }

   // Befehle,um die Relais-Pins an-und auszuschalten
   if (relais >= 1) {
     //Relais 1
     if (request.indexOf("GET /1/on") >= 0) {
       Serial.println("D1 ON");
       output1State = "on";
       digitalWrite(output1, HIGH);
       }
     if (request.indexOf("GET /1/off") >= 0) {
       Serial.println("D1 OFF");
       output1State = "off";
       digitalWrite(output1, LOW);
       }
   }
   if (relais >= 2) {
     // Relais 2
     if (request.indexOf("GET /2/on") >= 0) {
       Serial.println("D2 ON");
       output2State = "on";
       digitalWrite(output2, HIGH);
       }
     if (request.indexOf("GET /2/off") >= 0) {
       Serial.println("D2 OFF");
       output2State = "off";
       digitalWrite(output2, LOW);
       }
   }
   if (relais >= 3) {
     // Relais 3
     if (request.indexOf("GET /3/on") >= 0) {
       Serial.println("D3 ON");
       output3State = "on";
       digitalWrite(output3, HIGH);
       }
     if (request.indexOf("GET /3/off") >= 0) {
       Serial.println("D3 OFF");
       output3State = "off";
       digitalWrite(output3, LOW);
       }
   }
   if (relais >= 4) {
     // Relais 4
     if (request.indexOf("GET /4/on") >= 0) {
       Serial.println("D4 ON");
       output4State = "on";
       digitalWrite(output4, HIGH);
       }
     if (request.indexOf("GET /4/off") >= 0) {
       Serial.println("D4 OFF");
       output4State = "off";
       digitalWrite(output4, LOW);
       }
   }

   // Browser
   String Response, Header;
   Response =  "<html><head><title>ESP8266 Steuerung</title>";
   Response += "<meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">";
   Response += "<link rel=\"icon\" href=\"data:,\">";
   Response += "<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}";
   Response += ".button { background-color: #FF3333; border: none; color: white; padding: 10px 20px;";
   Response += "text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}";
   Response += ".button2 {background-color: #00FF00;}</style></head>";

   // Titel der Webseite
   Response += "<h2>WLAN-Schalter</h2>";

   // Schaltvorgänge auf der Webseite optisch sichbar machen (rot-Grün)
   if (relais >= 1) {
     // Button 1
     Response += "<p>R1 GPIO0 " + output1State + "</p>";
     if (output1State == "off") {
       Response += "<p><a href=\"/1/on\"><button class=\"button\">OFF</button></a></p>";
     } else {
       Response += "<p><a href=\"/1/off\"><button class=\"button button2\">ON</button></a></p>";
     }
   }
   if (relais >= 2) {
     //Button 2
     Response += "<p>R2 GPIO2 " + output2State + "</p>";
     if (output2State == "off") {
       Response += "<p><a href=\"/2/on\"><button class=\"button\">OFF</button></a></p>";
     } else {
       Response += "<p><a href=\"/2/off\"><button class=\"button button2\">ON</button></a></p>";
     }
   }
   if (relais >= 3) {
     //Button 3
     Response += "<p>R3 GPIO12 " + output3State + "</p>";
     if (outputt3State == "off") {
       Response += "<p><a href=\"/3/on\"><button class=\"button\">OFF</button></a></p>";
     } else {
       Response += "<p><a href=\"/3/off\"><button class=\"button button2\">ON</button></a></p>";
     }
   }
   if (relais == 4) {
     //Button 4
     Response += "<p>R4 GPIO13 " + output4State + "</p>";
     if (outputt4State == "off") {
       Response += "<p><a href=\"/4/on\"><button class=\"button\">OFF</button></a></p>";
     } else {
       Response += "<p><a href=\"/4/off\"><button class=\"button button2\">ON</button></a></p>";
     }
   }

   Response += "</body></html>";

   Header = "HTTP/1.1 200 OK\r\n";
   Header += "Content-Length: ";
   Header += Response.length();
   Header += "\r\n";
   Header += "Content-Type: text/html\r\n";
   Header += "Connection: close\r\n";
   Header += "\r\n";

   // Reaktionen zum Client senden
   client.print(Header);
   client.print(Response);

   // Client stoppen
   client.stop();
   Serial.println("Client disconnected");

}
```