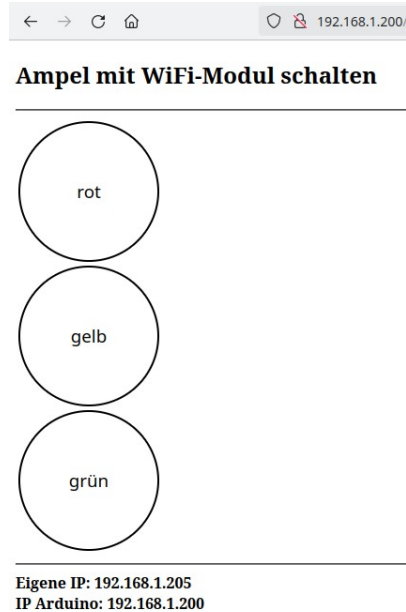


Das Programm "baut" eine Webseite, die mit der Wifi-Funktion des UNO R4 WiFi angezeigt wird. Ein Klick auf einen der Buttons schaltet die dazugehörige LED.

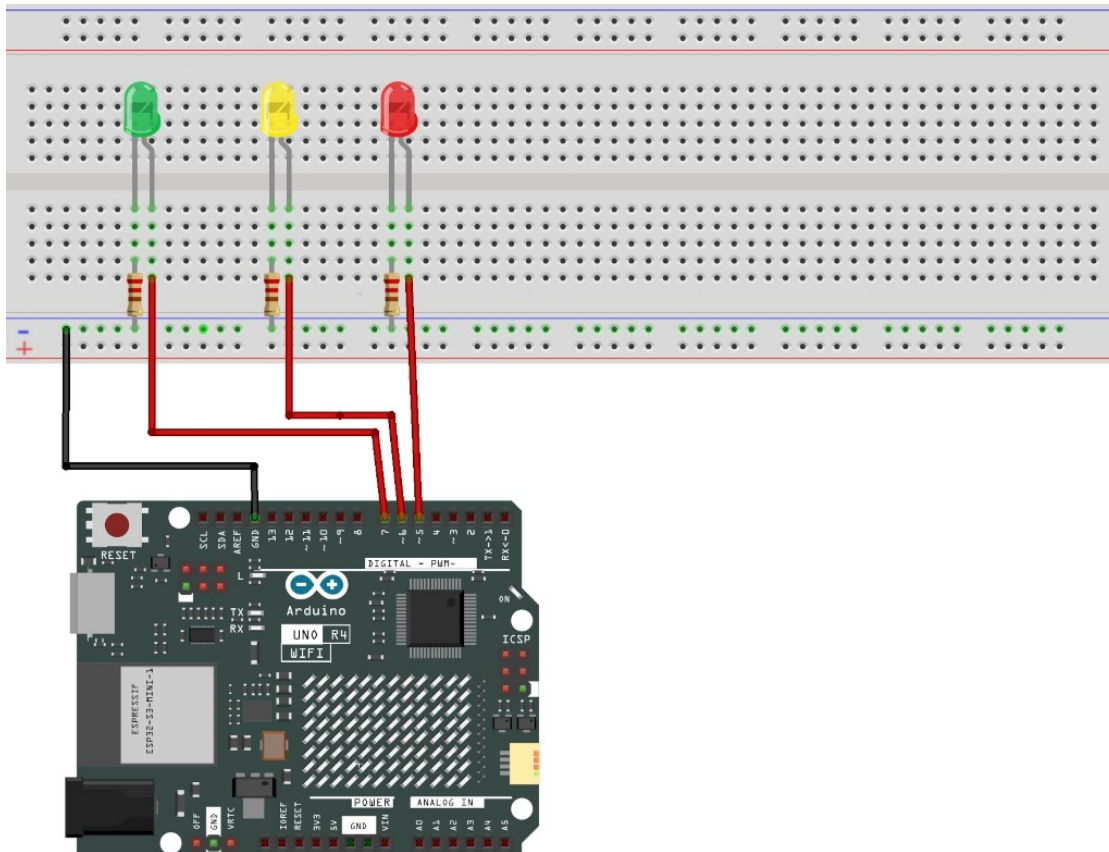
So sieht das Webformular aus:



Benötigte Bauteile:

- ➡ rote, gelbe und grüne LED
- ➡ 3 Widerstände 220 Ω
- ➡ Leitungsdrähte

Baue die Schaltung auf.



Binde die benötigte Bibliothek ein und definiere die Variablen:

```
#include "WiFiS3.h"

// Router-SSID/Passwort anpassen
char Router[] = "Router_SSID";
char Passwort[] = "xxxxxxxx";

// Pins der LEDs
int ROT = 5;
int GELB = 6;
int GRUEN = 7;

// Schalter für den Zustand der LEDs (false->aus, true-> an
bool RotAn = false;
bool GelbAn = false;
bool GruenAn = false;

// statischeIP = false -> IP-Adresse über DHCP vergeben
// statischeIP = true -> statische IP festlegen
bool statischeIP = false;

// statische IP anpassen
IPAddress ip(192, 168, 1, 200);

WiFiServer WiFiServer(80);

WiFiClient WebClient = WiFiServer.available();
```

Der setup-Teil:

```
void setup()
{
  pinMode(ROT, OUTPUT);
  pinMode(GELB, OUTPUT);
  pinMode(GRUEN, OUTPUT);

  Serial.begin(9600);

  // auf serielle Verbindung warten
  while (!Serial);
  delay(1000);

  // statische IP vergeben
  if (statischeIP) WiFi.config(ip);

  // WiFi starten
  WiFi.begin(Router, Passwort);

  Serial.print("Verbindung aufbauen mit ");
  Serial.println(Router);
```

```
// Verbindung herstellen
while (WiFi.status() != WL_CONNECTED)
{
    delay(200);
    Serial.print(".");
}

WiFiServer.begin();

// IP des Servers/des verbunden Computers anzeigen
Serial.print("Server: ");
Serial.println(WiFi.SSID());

// IP des Arduinos anzeigen
if (statischeIP) Serial.print("Statische IP Adresse Arduino: ");
else Serial.print("IP Adresse Arduino DHCP: ");
Serial.println(WiFi.localIP());
}
```

Im Seriellen Monitor wird die verwendete IP-Adresse angezeigt.
Diese musst du in einem Browser deiner Wahl eingeben.



Der loop-Teil:

```
void loop()
{
    // auf WebClients warten ...
    WebClient = WiFiServer.available();
    if (WebClient)
    {
        String SchaltungLesen;

        // solange der WebClient verbunden ist ...
        while (WebClient.connected())
        {
            if (WebClient.available())
            {
                // Anforderung vom WebClienten lesen ...
                char Zeichen = WebClient.read();

                // return (\n) gesendet
                if (Zeichen == '\n')
                {
                    // wenn der String SchaltungLesen leer ist
                    if (SchaltungLesen == "")
                    {
                        /*
                         HTML-Seite aufbauen
                         die folgenden Anweisungen müssen
                         mit print oder println gesendet werden
                         println "verschönert" den Quelltext
                         (erzeugt einen Zeilenumbruch im Quelltext)
                         " müssen mit \ maskiert werden " -> \"
                        */
                        // HTML-Seite aufbauen
                        WebClient.println("HTTP/1.1 200 OK");
                        WebClient.println("Content-type:text/html");

                        // Leerzeile zwingend erforderlich
                        WebClient.println();

                        // Gerüst der HTML-Seite:
                        WebClient.println("<!doctype html>");
                        WebClient.println("<html>");

                        WebClient.println("<body>");

                        // h2 -> Überschrift, hr horizontale Linie
                        WebClient.println("<h2>Ampel mit WiFi-Modul schalten</h2>");
                        WebClient.println("<hr>");

                        // table = Tabelle, tr = Tabelle Zeile, td = Tabelle Zelle
                        WebClient.println("<table><tr>");
```

```
// rote LED ist eingeschaltet -> roter Hintergrund
/*
  style = Stil des Buttons bestimmen:
  background-color = Hintergrundfarbe, width/height = Breite/Höhe
  cursor: pointer = Cursor als Hand darstellen,
  border-radius: 50% = runder Button,
  border = Rand des Buttons als 2 Pixel breite durchgezogene Linie (solid)
  onclick: beim Klick auf den Button wird die URL RotAus übergeben,
  sie wird später ausgewertet
*/
if (RotAn)
{
  WebClient.print("<td><input style='background-color:#FF6565;'");
  WebClient.print(" width:150px;height:150px; cursor:pointer;");
  WebClient.print(" font-size:14pt;");
  WebClient.print(" border-radius:50%;border: 2px solid black;");
  WebClient.print(" type='button'");
  WebClient.println(" value='rot'");
  WebClient.println(" onClick=\"location.href='RotAus'\">");
  WebClient.println("</td><tr>");
}

// rote LED ist ausgeschaltet -> weißer Hintergrund
else
{
  WebClient.print("<td><input style='background-color:white;'");
  WebClient.print(" width:150px;height:150px; cursor:pointer;");
  WebClient.print(" font-size:14pt;");
  WebClient.print(" border-radius:50%;border: 2px solid black;");
  WebClient.print(" type='button'");
  WebClient.println(" value='rot'");
  WebClient.println(" onClick=\"location.href='RotEin'\">");
  WebClient.println("</td><tr>");
}

// gelbe LED ist eingeschaltet -> gelber Hintergrund
if (GelbAn)
{
  WebClient.print("<td><input style='background-color:#FFFB65;'");
  WebClient.print(" width:150px;height:150px; cursor:pointer;");
  WebClient.print(" font-size:14pt;");
  WebClient.print(" border-radius:50%;border: 2px solid black;");
  WebClient.print(" type='button'");
  WebClient.println(" value='gelb'");
  WebClient.println(" onClick=\"location.href='GelbAus'\">");
  WebClient.println("</td><tr>");
}

// gelbe LED ist ausgeschaltet -> weißer Hintergrund
else
{
  WebClient.print("<td><input style='background-color:white;'");
  WebClient.print(" width:150px;height:150px; cursor:pointer;");
  WebClient.print(" font-size:14pt;");
  WebClient.print(" border-radius:50%;border: 2px solid black;");
  WebClient.print(" type='button'");
  WebClient.println(" value='gelb'");
  WebClient.println(" onClick=\"location.href='GelbEin'\">");
  WebClient.println("</td><tr>");
}
```

```

        WebClient.println(" value='gelb'");
        WebClient.println(" onClick=\"location.href='GelbEin'\">");
        WebClient.println("</td><tr>");
    }

    // grüne LED ist eingeschaltet -> grüner Hintergrund
    if (GruenAn)
    {
        WebClient.print("<td><input style='background-color:green;");
        WebClient.print(" width:150px;height:150px; cursor:pointer;");
        WebClient.print(" font-size:14pt;");
        WebClient.print(" border-radius:50%;border: 2px solid black;");
        WebClient.print(" type='button'");
        WebClient.println(" value='gr&uuml;n'");
        WebClient.println(" onClick=\"location.href='GruenAus'\">");
        WebClient.println("</td><tr>");
    }

    // grüne LED ist ausgeschaltet -> weißer Hintergrund
    else
    {
        WebClient.print("<td><input style='background-color:white;");
        WebClient.print(" width:150px;height:150px; cursor:pointer;");
        WebClient.print(" font-size:14pt;");
        WebClient.print(" border-radius:50%;border: 2px solid black;");
        WebClient.print(" type='button'");
        WebClient.println(" value='gr&uuml;n'");
        WebClient.println(" onClick=\"location.href='GruenEin'\">");
        WebClient.println("</td><tr>");
    }

    WebClient.println("</table>");
    WebClient.println("<hr />");

    // IPs anzeigen
    WebClient.print("<b>Eigene IP: ");
    WebClient.print(WebClient.remoteIP());
    WebClient.print("</b>");
    WebClient.print("<br><b>IP Arduino: ");
    WebClient.print(WiFi.localIP());
    WebClient.print("</b>");
    WebClient.println("</body>");
    WebClient.println("</html>");

    // HTTP-Antwort endet mit neuer Zeile
    WebClient.println();

    // Seite vollständig geladen -> loop verlassen
    break;
}

// wenn SchaltungLesen nicht leer ist -> Inhalt löschen
else SchaltungLesen = "";
}

```

```
// bei einem anderen Zeichen als return (\r)
// -> Zeichen dem String SchaltungLesen hinzufügen
else if (Zeichen != '\r') SchaltungLesen += Zeichen;

// indexOf überprüft, ob die Zeichenfolge
// im String SchaltungLesen enthalten ist
// der Zusatz der jeweiligen LED (ein/aus) wird mit ! "umgedreht":
// false->true, true->>false
if (SchaltungLesen.indexOf("RotEin") > 0)
{
    digitalWrite(ROT, HIGH);
    RotAn = !RotAn;
}
if (SchaltungLesen.indexOf("RotAus") > 0)
{
    digitalWrite(ROT, LOW);
    RotAn = !RotAn;
}

if (SchaltungLesen.indexOf("GelbEin") > 0)
{
    digitalWrite(GELB, HIGH);
    GelbAn = !GelbAn;
}

if (SchaltungLesen.indexOf("GelbAus") > 0)
{
    digitalWrite(GELB, LOW);
    GelbAn = !GelbAn;
}

if (SchaltungLesen.indexOf("GruenEin") > 0)
{
    digitalWrite(GRUEN, HIGH);
    GruenAn = !GruenAn;
}

if (SchaltungLesen.indexOf("GruenAus") > 0)
{
    digitalWrite(GRUEN, LOW);
    GruenAn = !GruenAn;
}
}
}

WebClient.stop();
}
```