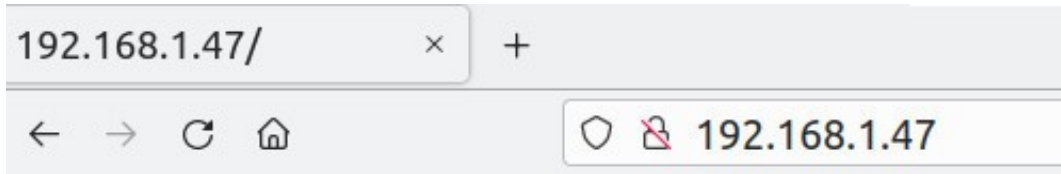




Ziel des Projekts

Mit einem Arduino WiFi „baut“ das Programm "baut" eine Webseite mit verschiedenfarbigen Buttons. Ein Klick auf einen der Buttons schaltet händisch eine Ampel.



Ampel mit WiFi-Modul schalten

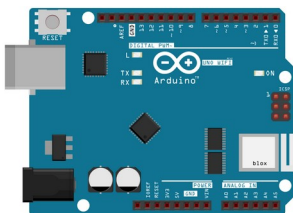


Eigene IP: 192.168.1.79

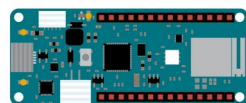
IP Arduino: 192.168.1.47

Die Hardware

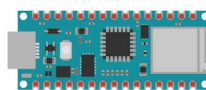
Für diese Anleitung benötigst du einen Arduino mit WiFi:



Arduino WiFi Rev 2



Arduino MKR WiFi 1010

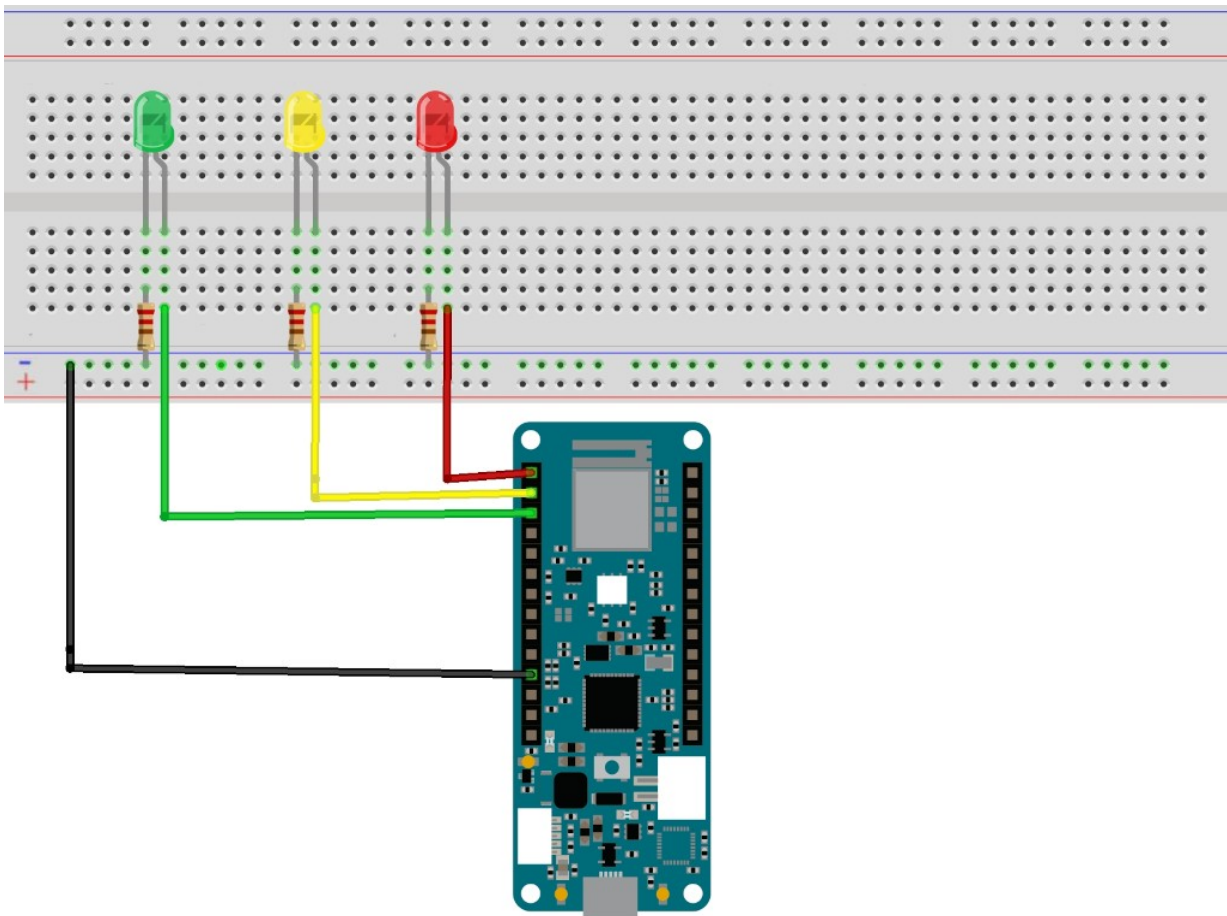


Arduino Nano 33 IoT

Benötigte Bauteile

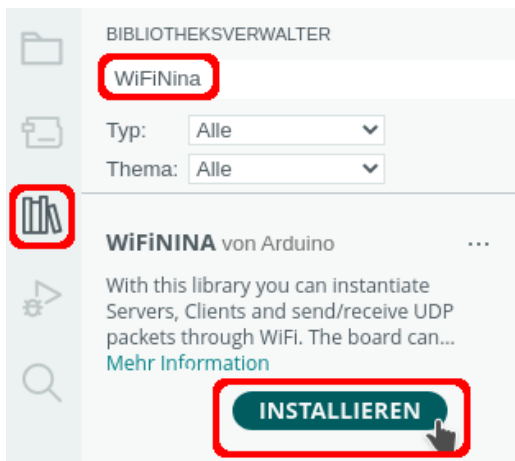
- ➔ 1 rote LED
- ➔ 1 grüne LED
- ➔ 1 gelbe LED
- ➔ 3 Widerstände 220 Ω
- ➔ Leitungsdrähte

Der Schaltplan (mit Arduino MKR WiFi 1010)



fritzing

Benötigte Bibliothek:





Das Programm

Binde die benötigte Bibliothek ein und definiere die Variablen.

```
#include "WiFiNINA.h"

int ROT = 5;
int GELB = 6;
int GRUEN = 7;

// Router: Name des Routers
// Passwort: WLAN-Passwort
char Router[] = "FRITZ!Box";
char Passwort[] = "xxxxxxx";

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

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

// Netzwerkstatus
int status = WL_IDLE_STATUS;

WiFiServer server(80);

WiFiClient client = server.available();
```

Der setup-Teil:

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

  Serial.begin(9600);

  // statische IP
  if (festeIP) WiFi.config(ip);

  // Verbindung aufbauen
  if (WiFi.status() == WL_NO_MODULE)
  {
    Serial.println(F("Verbindungsaufbau gescheitert!"));
  }

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



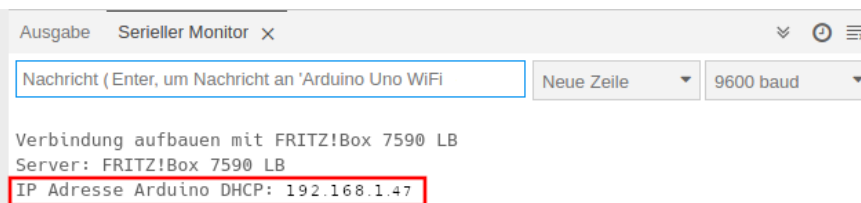
```
while (status != WL_CONNECTED)
{
    status = WiFi.begin(Router, Passwort);

    // Zeit für den Verbindungsaufbau
    // wenn die Verbindung nicht zustandekommt -> Zeit vergrößern
    delay(2000);
}

server.begin();

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

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



Anzeige der IP im Seriellen Monitor

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
    // auf Clienten warten ...
    client = server.available();
    if (client)
    {
        String SchaltungLesen;

        // solange der Client verbunden ist ...
        while (client.connected())
        {
            if (client.available())
            {
                // Anforderung vom Clienten lesen ...
                char Zeichen = client.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
        client.println("HTTP/1.1 200 OK");
        client.println("Content-type:text/html");

        // Leerzeile zwingend erforderlich
        client.println();

        client.println(F("<!doctype html>"));
        client.println(F("<html>"));

        client.println(F("<body>"));
        client.println(F("<h2>Ampel mit WiFi-Modul schalten</h2>"));
        client.println(F("<hr />"));
        client.println(F("<table><tr>"));

        // Button rot schalten
        client.print(F("<td><input style='font-size:14pt;'"));
        client.print(F("font-weight:bold;"));
        client.print(F(" background-color:#FF6565;"));
        client.print(F(" width:200px; cursor:pointer;"));
        client.print(F(" border-radius:5px;border: 2px solid black;"));
        client.print(F(" type='button'"));
        client.println(F(" onClick=\"location.href='R'\""));
        client.println(F(" value='rot schalten'>"));
        client.println(F("</td></tr>"));

        // Button gelb schalten
        client.print(F("<td><input style='font-size:14pt;'"));
        client.print(F("font-weight:bold;"));
        client.print(F(" background-color:#FFFB65;"));
        client.print(F(" width:200px; cursor:pointer;"));
        client.print(F(" border-radius:5px;border: 2px solid black;"));
        client.print(F(" type='button'"));
        client.println(F(" onClick=\"location.href='GE'\""));
        client.println(F(" value='gelb schalten'>"));
        client.println(F("</td></tr>"));
```



```
// Button grün schalten
client.print(F("<td><input style='font-size:14pt;"));
client.print(F("font-weight:bold;"));
client.print(F(" background-color:#76FA5F;"));
client.print(F(" width:200px; cursor:pointer;"));
client.print(F(" border-radius:5px;border: 2px solid black;"));
client.print(F(" type='button'"));
client.println(F(" onClick=\"location.href='GR'\""));
client.println(F(" value='gr&uuml;n schalten'>"));
client.println(F("</td>"));
client.println(F("</table>"));

client.println(F("<hr />"));

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

// HTTP-Antwort endet mit neuer Zeile
client.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;
}
```



```
// endsWith überprüft, ob der String SchaltungLesen mit dem
// entsprechenden Schaltbefehl endet
// R = rot, GE = gelb, GR = grün
if (SchaltungLesen.endsWith("GET /R"))
{
    // wenn ROT an -> ROT ausschalten
    if (!digitalRead(ROT)) digitalWrite(ROT, HIGH);
    else digitalWrite(ROT, LOW);
}

if (SchaltungLesen.endsWith("GET /GE"))
{
    if (!digitalRead(GELB)) digitalWrite(GELB, HIGH);
    else digitalWrite(GELB, LOW);
}

if (SchaltungLesen.endsWith("GET /GR"))
{
    if (!digitalRead(GRUEN)) digitalWrite(GRUEN, HIGH);
    else digitalWrite(GRUEN, LOW);
}
}

client.stop();
}
}
```