

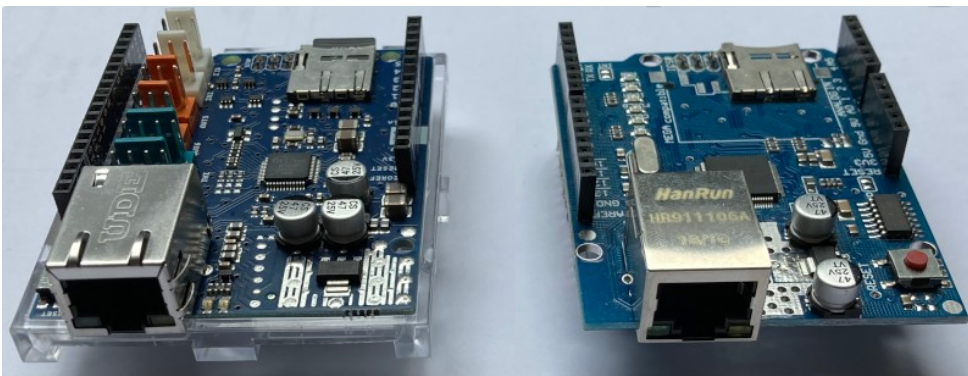


Ziel des Projekts

Das Programm "baut" mit einem Ethernet-Shield eine Webseite mit der zunächst eine LED und anschließend eine Ampel geschaltet werden kann.



Die Hardware



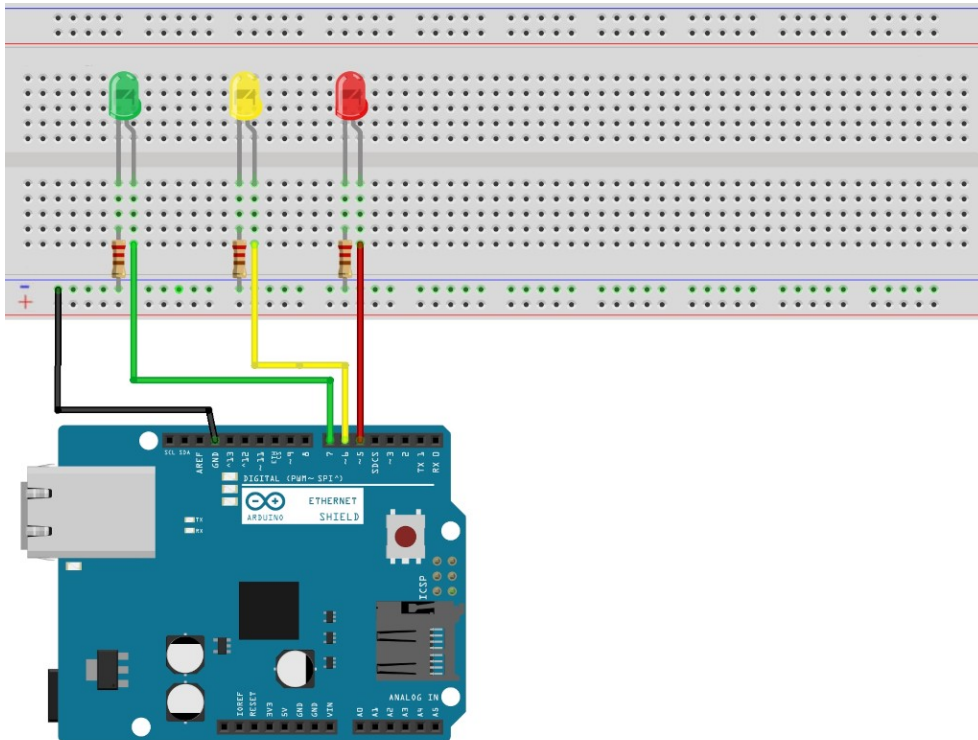
Für diese Anleitung benötigst du ein sogenanntes „Shield“, eine Platine, die einfach auf den Arduino aufgesteckt wird. Auf ihr befindet sich ein LAN-Anschluss (RJ45). Alle digitalen und analogen Anschlüsse stehen auch weiterhin zur Verfügung.

Benötigte Bauteile

- 3 LEDs (rot, gelb, grün)
- 3 Widerstände 220 Ω
- Ethernet-Shield
- Leitungsdrähte



Der Schaltplan



fritzing

Das Programm

Eine LED schalten

Das Programm schaltet die rote LED ein und aus.
So sieht das Programm im Browser aus:



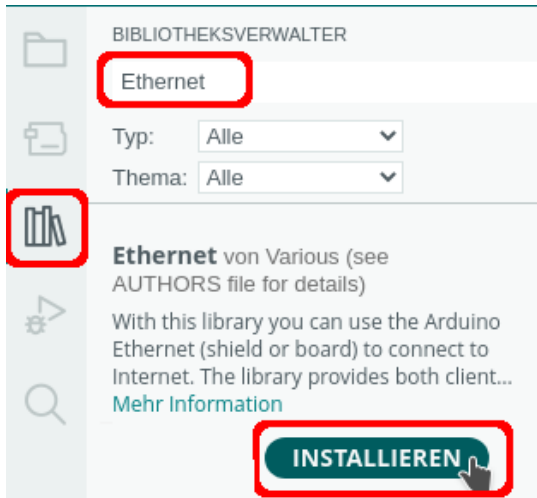
LED mit Ethernet-Shield schalten

rot einschalten**rot ausschalten****Eigene IP: 192.168.1.12****IP des Ethernet-Shields: 192.168.1.36**



Benötigte Bibliothek

Bevor du starten kannst musst du eine Bibliothek installieren:



Binde die benötigte Bibliothek ein und definiere die Variablen.

```
#include "Ethernet.h"

int ROT = 5;

// MAC-Adresse und IP definieren
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};

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

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

// Name des Servers vergeben
EthernetServer server(80);

// SchaltungLesen beinhaltet den Befehl zum Schalten der LED
String SchaltungLesen;
```

Der setup-Teil. Beachte die Kommentare.

```
void setup()
{
  Serial.begin(9600);
  pinMode(ROT, OUTPUT);

  // Ethernet starten feste IP
  if (festeIP) Ethernet.begin(mac, ip);

  // Ethernet starten DHCP
  else Ethernet.begin(mac);

  // Server starten
  server.begin();
}
```



```
// zur Kontrolle IP-Adresse anzeigen
// localIP -> Adresse, die im Browser eingegeben wird
Serial.print(F("IP des Ethernet-Shields: "));
Serial.println(Ethernet.localIP());
}
```

Im Seriellen Monitor wird die IP des Ethernet-Shields angezeigt. Diese Adresse wird in einem Browser eingegeben.

Der loop-Teil.

Beachte die Kommentare.

```
void loop()
{
    // auf Clienten warten ...
    EthernetClient Client = server.available();

    // neue Anfrage
    if (Client)
    {
        // solange der Client (der lokale PC) verbunden ist ...
        while (Client.connected())
        {
            if (Client.available())
            {
                // Daten vom Clienten lesen ...
                char Zeichen = Client.read();

                // ... und die Antwort zusammenbauen
                SchaltungLesen += Zeichen;

                // return gesendet
                if (Zeichen == '\n')
                {
                    /*
                     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 " -> \"
                    */
                    Client.println(F("HTTP/1.1 200 OK"));
                    Client.println(F("Content-Type: text/html"));
                    Client.println(F("Connection: close"));
                    Client.println();

                    Client.println(F("<!doctype html>"));
                    Client.println(F("<html>"));
                    Client.println(F("<body>"));
                    Client.println(F("<h2>LED mit Ethernet-Shield schalten</h2>"));
                    Client.println(F("<hr />"));
                    Client.println(F("<table>"));
                }
            }
        }
    }
}
```



```
// Button rot ein
Client.print(F("<tr><td><input style='font-size:14pt; 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;' type='button'"));
Client.println(F(" onClick=\"location.href='RotEin'\""));
Client.println(F(" value='rot einschalten'>"));
Client.println(F("</td>"));

// Button rot aus
Client.print(F("<td><input style='font-size:14pt; font-weight:bold;');
Client.print(F(" background-color:white;"));
Client.print(F(" width:200px; cursor:pointer;"));
Client.print(F(" border-radius:5px;border: 2px solid black;' type='button'"));
Client.println(F(" onClick=\"location.href='RotAus'\""));
Client.println(F(" value='rot ausschalten'>"));
Client.println(F("</td></tr>"));
Client.println(F("</table>"));
Client.println(F("<hr />"));

/*
  IP-Adressen anzeigen: Client.remoteIP() -> eigene IP,
  Ethernet.localIP() -> IP des Ethernet-Shields
*/
Client.print(F("<b>Eigene IP: "));
Client.print(Client.remoteIP());
Client.print(F("</b>"));
Client.print(F("<br><b>IP des Ethernet-Shields: "));
Client.print(Ethernet.localIP());
Client.print(F("</b>"));
Client.println(F("</body>"));
Client.println(F("</html>"));
delay(1);
Client.stop();

/*
  Buttons auswerten
  indexOf -> untersuchen ob der String RotEin
  in der Antwort enthalten ist
*/
if (SchaltungLesen.indexOf("RotEin") > 0) digitalWrite(ROT, HIGH);
if (SchaltungLesen.indexOf("RotAus") > 0) digitalWrite(ROT, LOW);

// String SchaltungLesen löschen
SchaltungLesen = "";
}
}
}
}
```



Ampelschaltung

Das Programm wird mit den übrigen Buttons ergänzt.

Die Definition der Variablen und der setup-Teil:

```
#include "Ethernet.h"

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

// MAC-Adresse und IP definieren
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};

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

// feste IP
IPAddress ip(192, 168, 1, 200);
// Name des Servers vergeben
EthernetServer server(80);

// SchaltungLesen beinhaltet den Befehl zum Schalten der LED
String SchaltungLesen;

void setup()
{
  Serial.begin(9600);

  pinMode(ROT, OUTPUT);
  pinMode(GELB, OUTPUT);
  pinMode(GRUEN, OUTPUT);

  // Ethernet starten feste IP
  if (festeIP) Ethernet.begin(mac, ip);

  // Ethernet starten DHCP
  else Ethernet.begin(mac);

  // Server starten
  server.begin();

  // zur Kontrolle IP-Adresse anzeigen
  // localIP -> Adresse, die im Browser eingegeben wird
  Serial.print(F("IP des Ethernet-Shields: "));
  Serial.println(Ethernet.localIP());
}
```



Der loop-Teil. Beachte die Kommentare

```
void loop()
{
    // auf Clienten warten ...
    EthernetClient Client = server.available();

    // neue Anfrage
    if (Client)
    {
        // solange der Client verbunden ist ...
        while (Client.connected())
        {
            if (Client.available())
            {
                // Anforderung vom Clienten lesen ...
                char Zeichen = Client.read();

                // ... und dem String hinzufügen
                // SchaltungLesen beinhaltet den Befehl zum Schalten der LED
                SchaltungLesen += Zeichen;

                // return gesendet
                if (Zeichen == '\n')
                {
                    /*
                     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 " -> \"
                    */
                    Client.println(F("HTTP/1.1 200 OK"));
                    Client.println(F("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 Ethernet-Shield schalten</h2>"));
                    Client.println(F("<hr />"));

                    Client.println(F("<table><tr>"));

                    // Button rot ein
                    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='RotEin'\""));
                    Client.println(F(" value='rot einschalten'>"));
                }
            }
        }
    }
}
```



```
Client.println(F("</td>"));

// Button rot aus
Client.print(F("<td><input style='font-size:14pt;'"));
Client.print(F("font-weight:bold;"));
Client.print(F(" background-color:white;"));
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='RotAus'\""));
Client.println(F(" value='rot ausschalten'>"));
Client.println(F("</td></tr>"));

// Button gelb ein
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='GelbEin'\""));
Client.println(F(" value='gelb einschalten'>"));
Client.println(F("</td>"));

// Button gelb aus
Client.print(F("<td><input style='font-size:14pt;'"));
Client.print(F("font-weight:bold;"));
Client.print(F(" background-color:white;"));
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='GelbAus'\""));
Client.println(F(" value='gelb ausschalten'>"));
Client.println(F("</td></tr>"));

// Button grün ein
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='GruenEin'\""));
Client.println(F(" value='gr&uuml;n einschalten'>"));
Client.println(F("</td>"));

// Button grün aus
Client.print(F("<td><input style='font-size:14pt;'"));
Client.print(F("font-weight:bold;"));
Client.print(F(" background-color:white;"));
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='GruenAus'\""));
Client.println(F(" value='gr&uuml;n ausschalten'>"));
Client.println(F("</td></tr>"));
```



```
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 des Ethernet-Shields: "));
Client.print(Ethernet.localIP());
Client.print(F("</b>"));
Client.println(F("</body>"));
Client.println(F("</html>"));

delay(1);

Client.stop();

/*
  ein Klick auf einen der Buttons
  erzeugt eine GET-Anweisung in der Form
  GET /RotEin HTTP/1.1 -> Button rot ein auswerten
  GET /RotAus HTTP/1.1 -> Button rot aus auswerten
  indexOf durchsucht den String nach diesen Zeichen
  (> 0 = Zeichen gefunden)
*/
if (SchaltungLesen.indexOf("RotEin") > 0) digitalWrite(ROT, HIGH);
if (SchaltungLesen.indexOf("RotAus") > 0) digitalWrite(ROT, LOW);
if (SchaltungLesen.indexOf("GelbEin") > 0) digitalWrite(GELB, HIGH);
if (SchaltungLesen.indexOf("GelbAus") > 0) digitalWrite(GELB, LOW);
if (SchaltungLesen.indexOf("GruenEin") > 0) digitalWrite(GRUEN, HIGH);
if (SchaltungLesen.indexOf("GruenAus") > 0) digitalWrite(GRUEN, LOW);
// String SchaltungLesen leeren
SchaltungLesen = "";
}
}
}
}
```