ARDUINO im Netzwerk

Ethernet und Web-Server mit ARDUINO

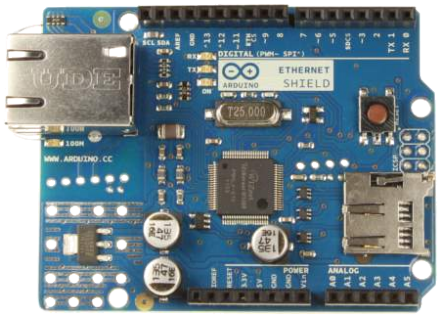
Creative Open Lab - Rainer Keil

FABLAB
REGION NÜRNBERG

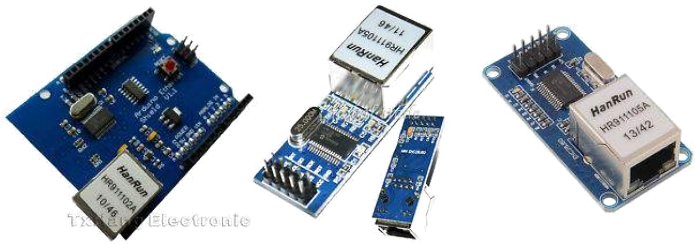
BY NC SA

Voraussetzungen
Was brauche ich an Hardware ?

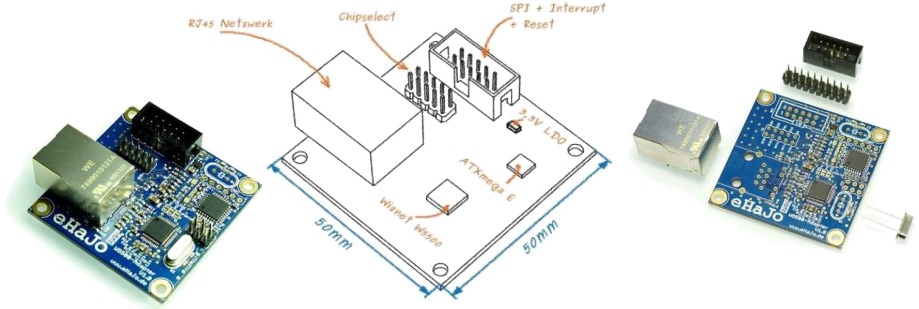
- **Das Original: ARDUINO Ethernet Shield**
  - **WizNet W5100**
  - **10/100 MBit/s**
  - **Shield mit zusätzlichen Slot für SD-Card**
  - **Bibliothek in IDE enthalten**



- **Die verbreitete Alternative:**
  - **ENC28J60**
  - **10 MBit/s**
  - **Als Shield und Modul in diversen Ausführungen**
  - **Verschiedene Bibliotheken verfügbar**



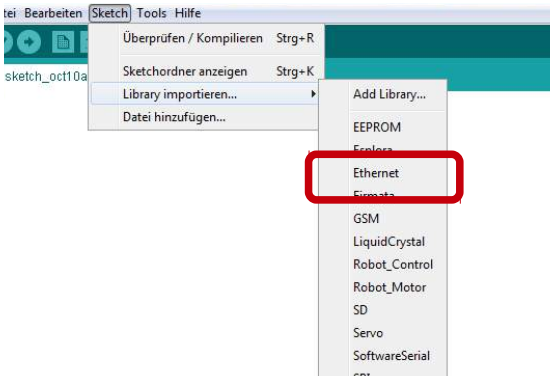
- **Die innovative Lösung von eHajo**
  - **SPI-Netzwerkadapter Wiznet W5500**
  - **10/100 MBit/s**
  - **Modulbausatz mit Controller**
  - **Spannungsregler 3,3V on Board**
  - **Bibliothek bei Wiznet erhältlich**

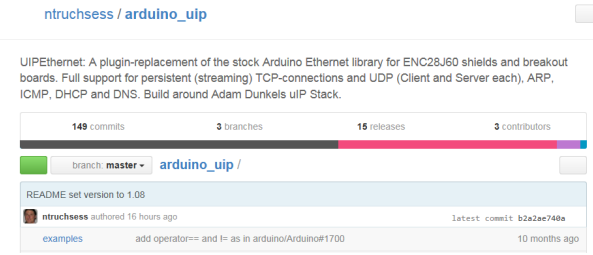


Voraussetzungen
Was brauche ich an Software ?

Welche Bibliothek benötigt wird, ist von der eingesetzten Hardware abhängig:

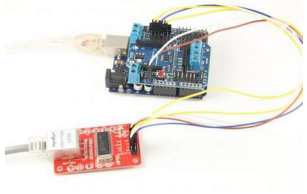
- **ARDUINO Ethernet Shield**
  - **Die erforderliche Lib ist im „Lieferumfang“ der IDE enthalten.**
- **ENC28J60**
  hierfür gibt es verschiedene Libraries die im Web verfügbar sind.
  - **Teilweise sind diese aufrufkompatibel zur Standard-Lib.**
    z.B. https://github.com/ntruchsess/arduino_uip
  - **Im Unterschied zur Standardlib wird der TCP/UDP Stack hier im Atmega ausgeführt. Dies führt zu einem höheren Ressourcenbedarf (Prozessorbelastung, Code, RAM …)**
  - **In Netzwerken mit viel „Broadcastverkehr“ (z.B. hier im FabLab) problematisch (bleibt hängen) → weitere Tests erforderlich**
- **SPI-Netzwerkadapter Wiznet W5500 (eHajo)**
  - **Ersatz der Standard-Lib durch eine W5500 Lib von Wiznet**
    Siehe https://github.com/Wiznet/WIZ_Ethernet_Library
  - **Durch Anpassung der (Datei)-Namen ist auch eine parallele Installation möglich (Demnächst im Wiki des FabLab).**





Verbindungen
Wie verbinde ich die Hardware ?

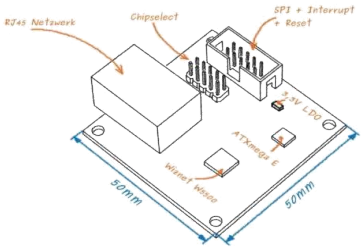
- **ARDUINO Ethernet Shield**
  - **Einfach das Shield auf den Arduino stecken**
- **ENC28J60**
  - **Ist es ein Shield → siehe oben**
  - **Modul: die SPI-Schnittstelle mit dem Arduino verbinden**



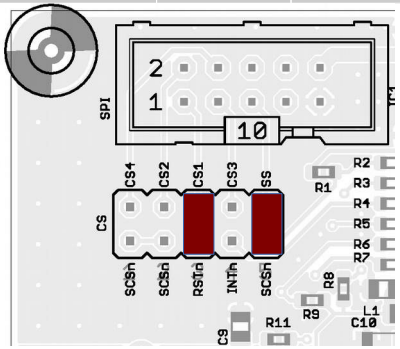
Beispiel

ENC28J60 module	Arduino Uno/Due	Arduino Mega
CS	D10	D53
SI	D11	D51
SO	D12	D50
SCK	D13	D52
RESET	RESET	RESET
INT	D2	D2
VCC	3V3	3V3
GND	GND	GND

- **SPI-Netzwerkadapter Wiznet W5500 (eHajo)**
  - **die SPI-Schnittstelle mit dem Arduino verbinden**
  - **Jumper CS auf dem Adapter setzen:**
    RSTn ↔ CS1 und SCSn ↔ SS
  - **Die Lötjumper CP1-4 und M1-3 bleiben offen**



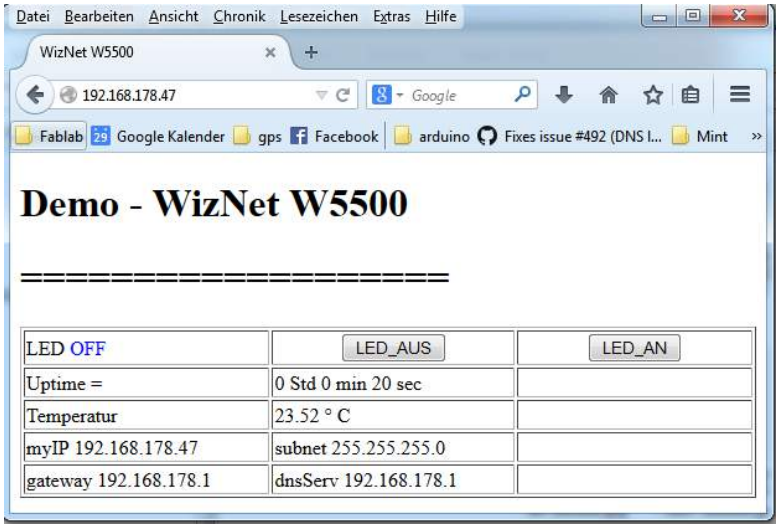
	Arduino	W5500 Board (10 pos Stecker)
SPI SCK	13	8
SPI MISO	12	6
SPI MOSI	11	7
SPI SS	10	2
RST (Reset)	RST	1
VCC (5V)	VCC	10
GND	GND	9

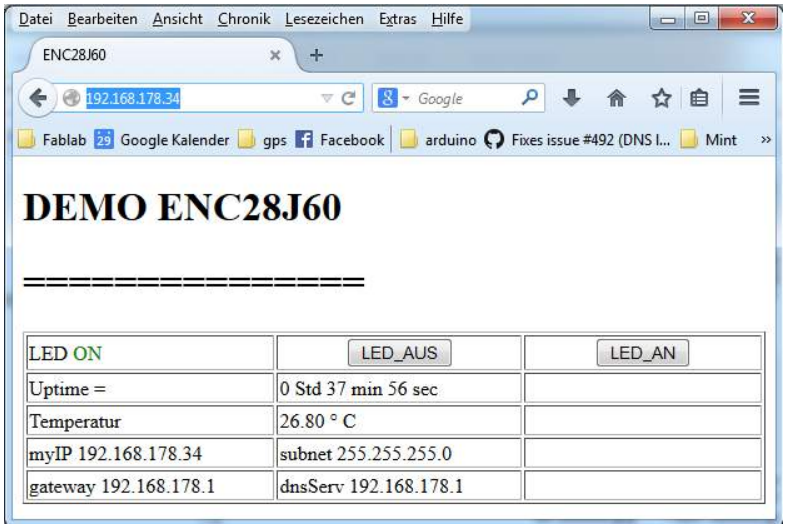


Stromversorgung / Spannungslevel
Die Ethernet Chips werden mit 3,3 V versorgt.
Bitte beim Anschluss beachten sofern kein Spannungsregler auf dem Board verbaut ist !!
Die (SPI-) Datenleitungen sind bei
den oben aufgeführten Chips/Boards „5 V tolerant“

Vergleich: W5500 - ENC28J60

	Wiznet W5500	ENC28J60
Programmierung	Aufrufkompatibel zum Standard	Aufrufkompatibel zum Standard
Netzwerk-Geschwindigkeit	10/100 MBit/s	10 MBit/s
TCP / UDP Stack	Im Chip	Per Software
Anbindung	SPI	SPI
Speicherbedarf (einfacher HTTP Server)	ca. 20 kByte	ca. 30 KByte
CPU Belastung	gering	hoch





Programm für einfachen Webserver

```
#include <Arduino.
#include <SPI.h>
#
```

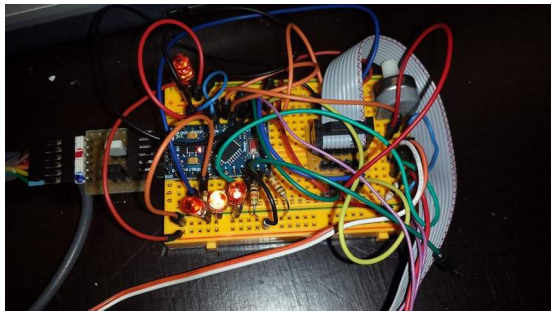
```
void setup() {
  // Open serial
  Serial.begin(11
  while (!Serial)
```

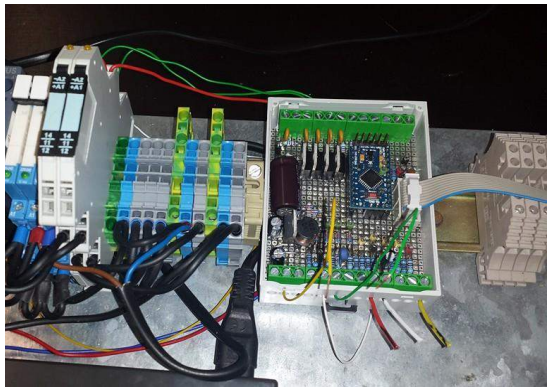
```
EthernetClient client = server.available();
if (client) {
```

```
        // send a standard http response header
        client.println(F("HTTP/1.1 200 OK"));
        client.println(F("Content-Type: text/html"));
        client.println(F("Connection: close"));  // the connection will be closed after completion of the response
        client.println(F("Refresh: 2 ; URL=/"));  // refresh the page automatically every 2 sec - go back to /
        client.println();
        client.println(F("<!DOCTYPE HTML>"));
        client.println(F("<html><head><title>ENC28J60</title>"));
        client.println(F("<html>"));



        client.print(F("<h1> DEMO ENC28J60</h1>"));
        client.print(F("<h1> ===============</h1>"));

        client.print(F("<Table border = '1'> <colgroup><col width='250'> <col width='250'> <col width='250'> </colgroup>"));

// ---------------------------------------------------------------------

        client.print(F("<tr>"));
        client.print(F("<td> LED "));

        if(Led_State & 1) {
                client.print(F("<font color='green'>ON<font color='black'> </td>"));
            }else{
                client.print(F("<font color='blue'>OFF<font color='black'> </td>"));
            }
            client.print(F("<td align='center'> <form method=get><input type=submit name=LED_AUS value='LED_AUS'></form> </td>"));
```

Beispiel: BlueBlub Druckluftsteuerung mit Webinterface

- **Zur Steuerung des Kompressors für die Druckluftversorgung des Fablabs textbasiertes Web-Interface**
- **eHajo Board mit Wiznet W5500**
- **Arduino pro mini**
  - **Regelung des Drucks in der Anlage**
    - **Kompressoransteuerung**
    - **Ventil zwischen Kessel und Leitung**
  - **Temperaturüberwachung**
  - **Spannungsüberwachung**
  - **WebPage**
  - **SYSLOG**





Demnächst im Produktivbetrieb

ARDUINO im Netzwerk

FABLAB
REGION NÜRNBERG

CC BY NC SA