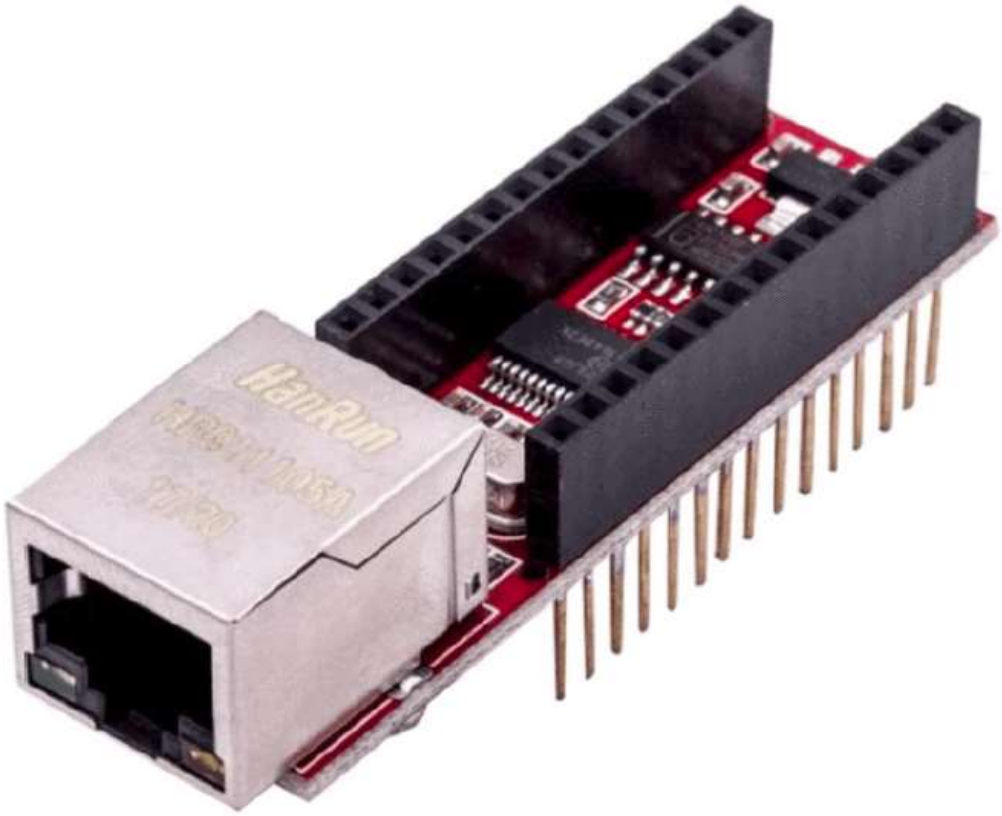
Herzlich willkommen!

Vielen Dank, dass Sie sich für unser AZ-Delivery Ethernet Shield für den Arduino Nano entschieden haben. Auf den folgenden Seiten geben wir Ihnen eine Einführung in den Aufbau und die Benutzung des Shields.

Wir wünschen Ihnen viel Spaß!



Dieses Shield ist das perfekte Gerät, um Arduino Boards mit dem Internet zu verbinden. Ein Anwendungsbeispiel ist die Ausgabe von Sensordaten oder der Position einer Motorwelle auf einer Webseite oder in diversen Apps für Android oder iOS.

Wir möchten Sie nochmal darauf hinweisen, dass das Shield lediglich mit dem Arduino Nano kompatibel ist.

Um das Shield mit anderen Arduino-Boards nutzen zu können, benötigen Sie ein Steckbrett.

Arduino und ENC28J60 Ethernet Controller

„ENC28J60“ entspricht der Modellnummer des von Microchip hergestellten Chip auf diesem Shield. Er besteht aus 28 Pins und beinhaltet einen komplett autarken Ethernet Controller für eine 10BASE-T-Netzwerkverbindung mit einem SPI für den Datenaustausch mit dem Arduino.

Falls Sie an den technischen Details des 10BASE-T interessiert sind, können wir Ihnen folgende Webseite empfehlen:

https://en.wikipedia.org/wiki/Ethernet_over_twisted_pair.

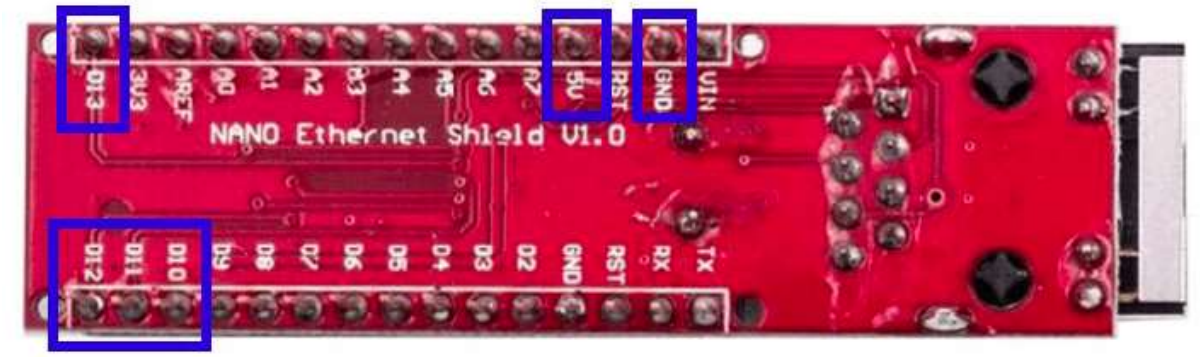
Sollte Ihr Computer eine LAN-Verbindung unterstützen wird für die Buchse der gleiche Stecker verwendet wie für das Shield. Die 10 im Namen steht für die Übertragungsgeschwindigkeit von 10Mbit/s. Obwohl das für heutige Verhältnisse sehr gering ist, sollte es im Rahmen von Arduino-Projekten vollkommen ausreichend sein, da die Datenlast für gewöhnlich gering ist.

Der Ethernet Controller (ENC28J60) beinhaltet ein SPI und verwendet die Arduino Pins 10, 11, 12 und 13. Für weitere Informationen bezüglich des **S**erial **P**eripheral **I**nterface besuchen Sie bitte die Wikipedia-Seite:

https://de.wikipedia.org/wiki/Serial_Peripheral_Interface.

In Folgendem Absatz erklären wir Ihnen die einzelnen Pin-Namen des SPI. Das Ethernet Shield nimmt hierbei die Rolle des Slaves und der Arduino Nano die des Masters ein.

Pin-Name	Erklärung
SS	Slave Select wird genutzt, um das Slave-Gerät zu aktivieren/deaktivieren
MOSI	Master Output Slave Input überträgt den Datenverkehr von Master zu Slave
MISO	Master Input Slave Output überträgt den Datenverkehr von Slave zu Master
SCK/SCLK	Zeitsynchronisierung von Master zu Slave



Das von Arduino angebotene Ethernet Shield verwendet ebenfalls einen All-In-One-Ethernet-Controller. Für diesen werden vorinstallierte Bibliotheken in der Arduino IDE angeboten. Dieses Shield hat allerdings den Wiznet W5100 installiert.

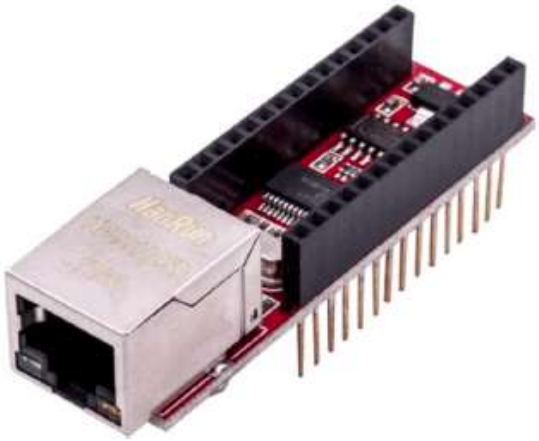
https://store.arduino.cc/arduino-ethernet-shield-2

https://www.wiznet.io/

In diesem eBook wird das MicroChip ENC28J60 erklärt!

Das Modul ist NICHT dasselbe wie der W5100 und ist NICHT mit diesem kompatibel. Aus diesem Grund werden andere Bibliotheken benötigt.

In unserem Webstore sind zwei Modelle des ENC28J69 verfügbar, die so ziemlich gleich funktionieren. Lediglich die Boards und die Pins sind speziell angepasst. Untenstehend sehen Sie beide Varianten.



Ethernet shield for Nano

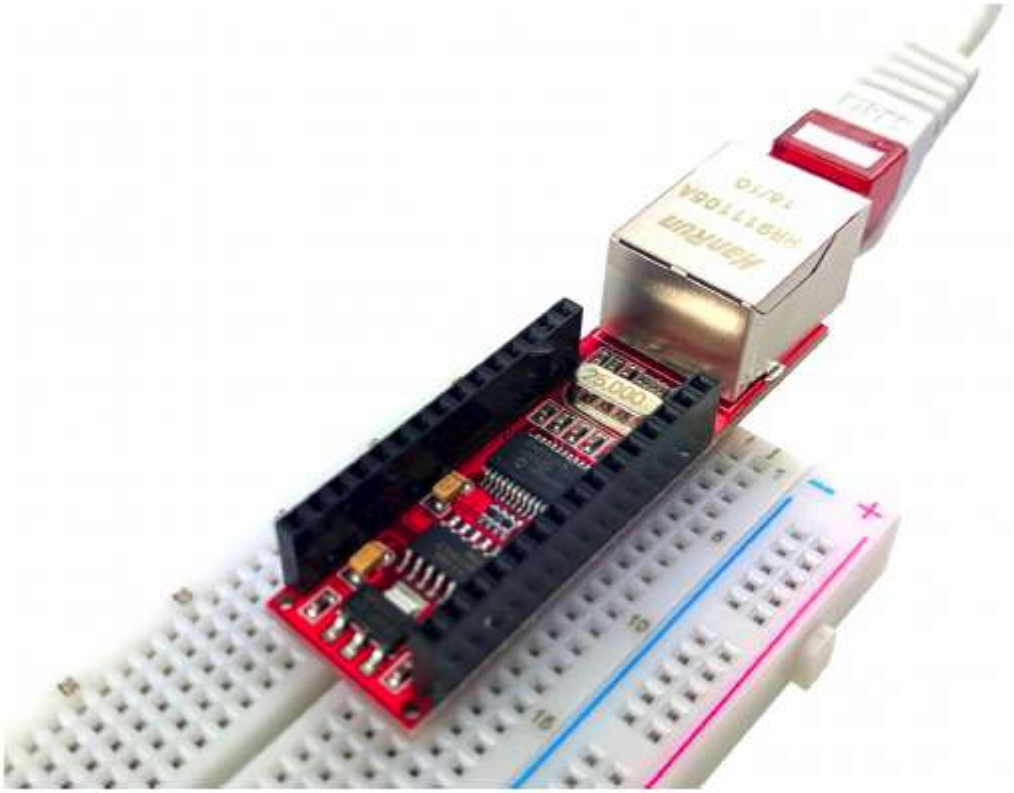
ENC28J60 network module

ENC28J60-Produktseite in unserem Webshop:

https://www.az-delivery.de/products/enc28j60-netzwerkmodul?_pos=2&_sid=2a3831c3f&_ss=r&ls=de

Der Nano und das Shield zusammengesteckt sind äußerst kompakt.

Zusammengesteckt hat der Aufbau ungefähr eine Länge von 67mm, 17mm Breite und 17mm Höhe. Die Abmessungen hängen davon ab, wie das Shield mit dem Nano verbunden wird. Sie können sowohl die oberen Verbindungen verwenden, als auch die unteren zum Experimentieren (Steckbrettverbindung siehe unten). Für den Endaufbau können Sie die unteren Pins abschneiden.



Für das Shield wird grundsätzlich eine Stromverbindung benötigt, sprich die GND- und +3.3V- oder +5V-Pins. Für die Datenverbindung verwenden Sie die Standard SPI-Pins D10, D11, D12 und D13 auf dem Arduino-Board.

In diesem eBook stellen wir Ihnen für zwei Bibliotheken ein „Hallo Welt!“-Beispiel vor. Die „Hallo Welt“-Nachricht kann dabei in einem Webbrowser angezeigt werden.

UIPEthernet Bibliothek

Die Bibliothek kann die vorinstallierte Standardbibliothek in der Arduino IDE ersetzen. Dadurch kann man bereits bestehende Beispiele sehr leicht anpassen um entweder das Arduino Ethernet Shield oder das ENC28J60 Ethernet Shield zu verwenden. Hierfür müssen lediglich die zwei include-Statements im Original („`#include <Ethernet.h>`“ und „`#include <SPI.h>`“) mit „`#include <UIPEthernet.h>`“ ersetzen.

Die Bibliothek hat zusätzlich auch eine eigene „print“-Funktion implementiert, die genauso wie die „print“-Funktion der SPI-Bibliothek. Dadurch bleibt der Code simpel und leicht zu benutzen.

Fortgeschrittene Funktionalitäten sind falls benötigt vorhanden.

UIPEthernet kann von der Plattform GitHub unter folgendem Link heruntergeladen werden:

https://github.com/ntruchsess/arduino_uip.

Diese wird auch auf der Arduino-Seite erwähnt:

https://playground.arduino.cc/Hardware/ArduinoEthernet/.

Zum Hinzufügen zur Arduino IDE verwenden Sie den Menüpunktepfad: *Sketch > Include Library > Add .ZIP Library…*

Sollten Sie die Version 1.8.9 der Arduino IDE verwenden, wird bei Installation der Bibliothek eine Warnung ausgegeben, die Sie allerdings ignorieren können.

Im Folgenden sehen Sie ein Sketch-Beispiel für diese Bibliothek:

```
#include <UIPEthernet.h>
byte mac[] = { 0x54, 0x34, 0x41, 0x30, 0x30, 0x31 };
IPAddress ip(192, 168, 0, 254);
EthernetServer server(80);
void setup() {
  Serial.begin(9600);
  Ethernet.begin(mac, ip);  // start the Ethernet connection and the server:
  server.begin();
  Serial.print("IP Address: ");  Serial.println(Ethernet.localIP());
}
void loop() {
  EthernetClient client = server.available();  // listen for incoming clients
  if (client) {
    Serial.println("-> New Connection");
    boolean currentLineIsBlank = true;  // http request ends with a blank line
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
```

```
(4 tabs)// if you've gotten to the end of the line (received a newline
        // character) and the line is blank, the http request has ended,
        // so you can send a reply
        if (c == '\n' && currentLineIsBlank) {
              client.println("<html><title>Hello World!</title><body><h3>Hello
World!</h3></body>");
          break;
        }
        if (c == '\n') {
          currentLineIsBlank = true;  // you're starting a new line
        }
        else if (c != '\r') {
          currentLineIsBlank = false; // a character on current line
        }
      }
    }
    delay(10);  // give the web browser time to receive the data
    client.stop();  // close the connection:
    Serial.println(" Disconnected\n");
  }
}
```

Folgend erklären wir Ihnen den Code. Zuerst wird die Bibliothek eingebunden, dann die MAC-Adresse des Ethernet Shields eingestellt. Dann wird die statische IP-Adresse des Servers gesetzt. Im Beispiel verwenden wir eine Adresse des Adresspools *„192.168.0.0“* (Netzwerkmaske 255.255.255.0). Dies beinhaltet Adressen von 192.168.0.1 bis 192.168.0.255.

Zuerst sollten Sie überprüfen welche Adresspools in Ihrem lokalen Netzwerk verfügbar sind.

In Linux können Sie die IP-Adresse und die Netzwerkmaske Ihres Computers mit dem Befehl „ifconfig“ im Terminal überprüfen. Für uns ergab sich:

IP-Adresse:	192.168.0.101
Subnetzmaske:	255.255.255.0
Broadcast:	192.168.0.255

In Windows öffnen Sie die Eingabeaufforderung und geben Sie den Befehl „ipconfig“ ein.

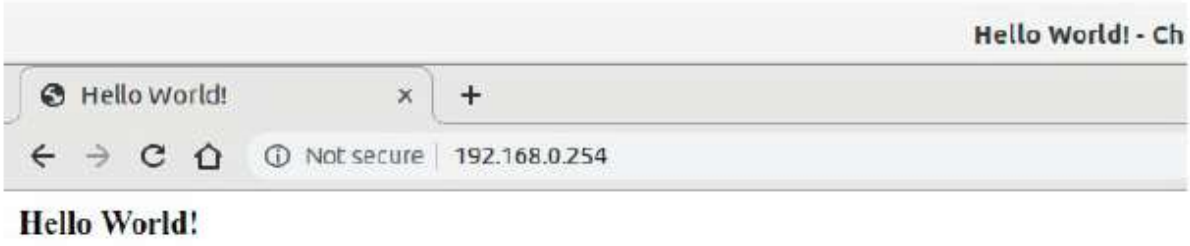
In unserem Fall können wir die Adresse des PCs 192.168.0.101 und die Broadcast-Adresse 192.168.0.255 nicht verwenden. Alle anderen Adressen in dem zuvor genannten Bereich können benutzt werden. Beispielsweise ist die Adresse 192.168.0.254 verwendbar.

Danach wird ein Serverobjekt auf den Port 80 gelegt. (Alle Server verwenden diese Portnummer).

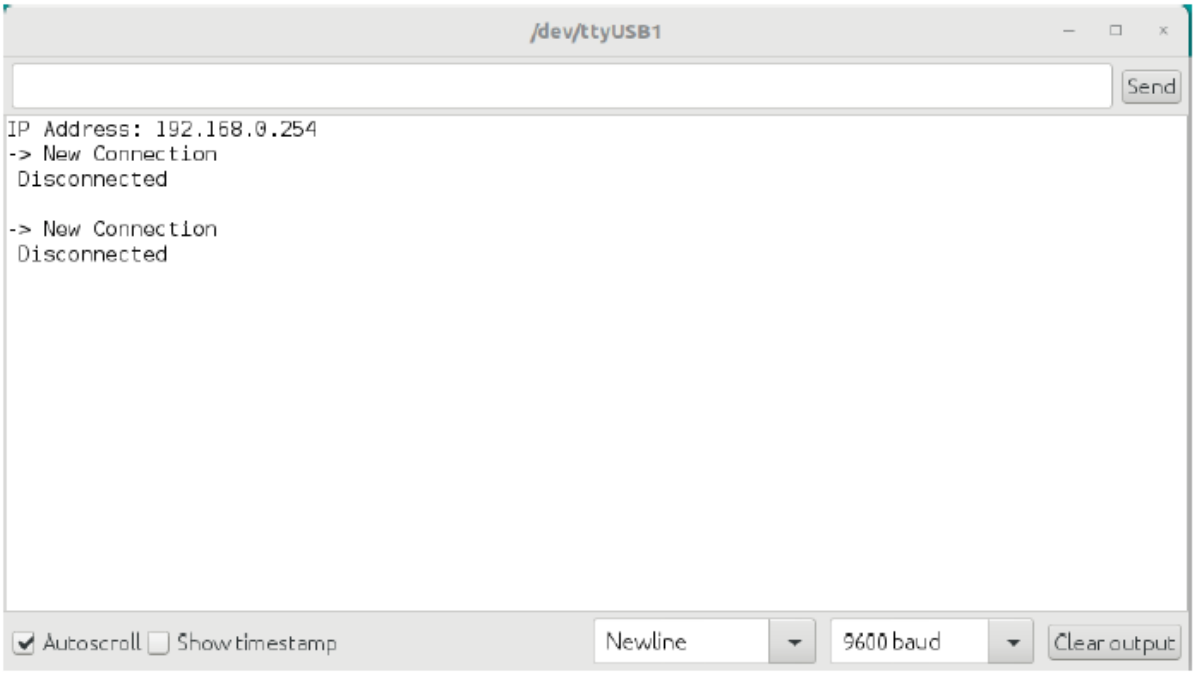
In der `setup`-Funktion wird der Server gestartet und die IP-Adresse an das SPI ausgegeben.

In der `loop`-Funktion wird darauf gewartet, dass sich ein Client verbindet. Sobald das geschieht wird das Ereignis an das SPI weitergegeben und die Webseite ausgegeben. Danach wird der Client getrennt, was ebenfalls an das SPI gegeben wird.

Wenn Sie den Nano anschalten und in Ihrem Browser die von Ihnen eingestellte IP-Adresse eingeben, sollte „Hello World!“ ausgegeben werden.



Im *Serial Monitor* sollte die Ausgabe wie folgt aussehen.



Ethercard-Bibliothek

Ethercard eignet sich hervorragend für fortgeschrittene Nutzer.

Vorsicht: Ethercard scheint Pin 8 anstatt Pin 10 für SS zu verwenden!

Die Bibliothek ist in der Arduino-Gemeinschaft äußerst beliebt, da sie eine der vollständigsten Implementierungen aufweist.

Ein weiteres Plus ist die einfache Umsetzung für grundsätzlich komplizierte Aufgaben wie DHCP und weitere fortgeschrittene Funktionen.

Diese Bibliothek erhalten Sie ebenfalls auf der Plattform GitHub unter dem Link: https://github.com/njh/EtherCard.

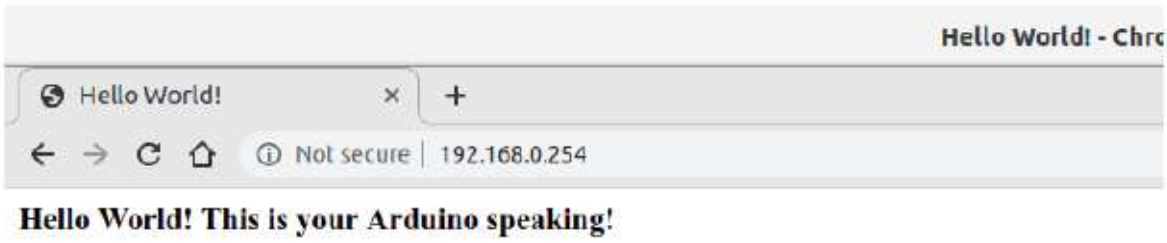
Die Bibliothek wird dann genauso wie die vorhergehende über die Menüpunkte *Sketch > Include Library > Add .ZIP Library…* hinzugefügt.

Das Sketchbeispiel finden Sie auf der nächsten Seite.

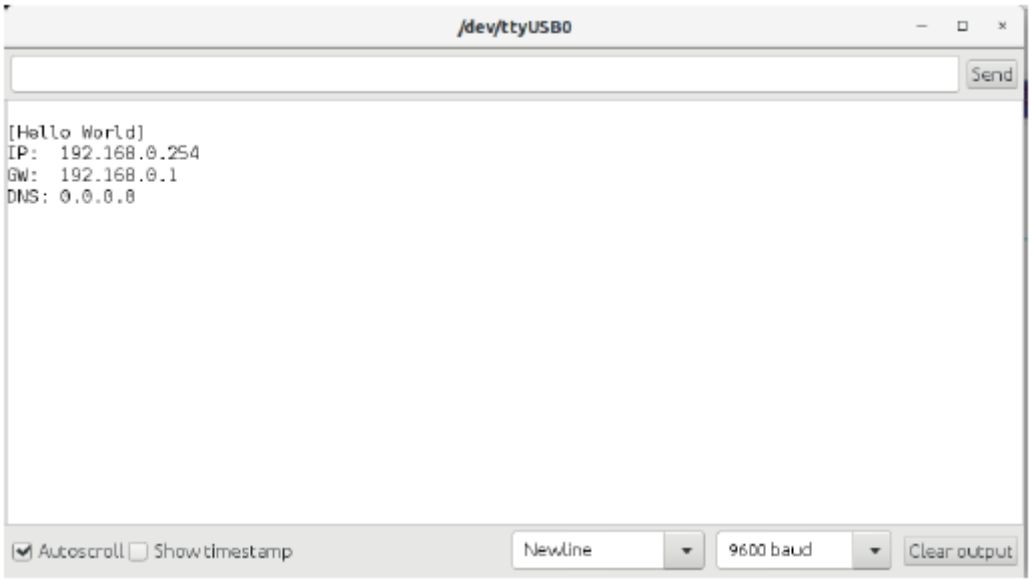
```
#include <EtherCard.h>
// Ethernet IP, default gateway and MAC addresses
static byte myip[] = { 192,168,0,254 };
static byte gwip[] = { 192,168,0,1 };
static byte mymac[] = { 0x74,0x69,0x69,0x2D,0x30,0x31 };
byte Ethernet::buffer[500]; // tcp/ip send and receive buffer

const char page[] PROGMEM =
"HTTP/1.0 503 Service Unavailable\r\n"
"Content-Type: text/html\r\n"
"Retry-After: 600\r\n"
"\r\n"
"<html>"
  "<head><title>"
    "Hello World!"
  "</title></head>"
  "<body>"
    "<h3>Hello World! This is your Arduino speaking!</h3>"
  "</body>"
"</html>";

void setup(){
  Serial.begin(9600);
  Serial.println("\n[Hello World]");
  if (ether.begin(sizeof Ethernet::buffer, mymac) == 0) {
    Serial.println("Failed to access Ethernet controller"); }
  ether.staticSetup(myip, gwip);
  ether.printIp("IP:  ", ether.myip);
  ether.printIp("GW:  ", ether.gwip);
  ether.printIp("DNS: ", ether.dnsip);
}
void loop(){
  // wait for an incoming TCP packet, but ignore its contents
  if (ether.packetLoop(ether.packetReceive())) {
    memcpy_P(ether.tcpOffset(), page, sizeof page);
    ether.httpServerReply(sizeof page - 1);
  }}
```

Im Allgemeinen entspricht dieser Sketch dem vorhergehenden, weshalb wir hier auf eine erneute detaillierte Besprechung verzichten.

Wenn Sie den Nano einschalten und in Ihrem Browser die IP-Adresse, die im Sketch eingestellt ist, eingeben, sollte folgende Webseite angezeigt werden:



Der *Serial Monitor* sollte folgendermaßen aussehen:



Sie haben es geschafft. Das Modul ist eingerichtet und Sie können es für Ihre Projekte verwenden.

Nun ist es an der Zeit neue Projekte selbstständig in Angriff zu nehmen. Dabei unterstützen Sie viele Beispiel-Sketches und Tutorials, die Sie im Internet finden.

Wenn Sie auf der Suche nach hochwertigen Produkten für Arduino und Raspberry Pi sind, sind wir von AZ-Delivery Vertriebs GmbH der richtige Ansprechpartner. Wir unterstützen Sie mit vielen Anwendungsbeispielen, Einrichtungshilfen, eBooks, Bibliotheken und natürlich unseren Technik-Experten!

https://az-delivery.de

Viel Spaß!

Impressum

https://az-delivery.de/pages/about-us