

AZ-Delivery

Herzlich willkommen!

Vielen Dank, dass Sie sich für unser AZ-Delivery Ethernet Shield W5100 für Atmega328P Board und Mega entschieden haben. Auf den nachfolgenden Seiten geben wir Ihnen eine Einführung in das Programmieren und die Nutzung dieses praktischen Gerätes.

Viel Spaß!



Anwendungsbereiche

Bildung und Lehre: Einsatz in Schulen, Hochschulen und Ausbildungseinrichtungen zur Vermittlung von Grundlagen der Elektronik, Programmierung und eingebetteten Systemen. Forschung und Entwicklung: Verwendung in Forschungs- und Entwicklungsprojekten zur Erstellung von Prototypen und Experimenten in den Bereichen Elektronik und Informatik. Prototypenentwicklung: Einsatz in der Entwicklung und Erprobung neuer elektronischer Schaltungen und Geräte. Hobby und Maker-Projekte: Verwendung durch Elektronikenthusiasten und Hobbyisten zur Entwicklung und Umsetzung von DIY-Projekten.

Erforderliche Kenntnisse und Fähigkeiten

Grundlegendes Verständnis der Elektronik und Elektrotechnik. Kenntnisse in der Programmierung, insbesondere in der Programmiersprache C/C++. Fähigkeit, Schaltpläne zu lesen und einfache Schaltungen zu entwerfen. Erfahrung im Umgang mit elektronischen Komponenten und Löten.

Betriebsbedingungen

Das Produkt darf nur mit den im Datenblatt spezifizierten Spannungen betrieben werden, um Beschädigungen zu vermeiden. Eine stabilisierte Gleichstromquelle ist zum Betrieb erforderlich. Bei der Verbindung mit anderen elektronischen Komponenten und Schaltungen sind die maximalen Strom- und Spannungsgrenzen zu beachten, um Überlastungen und Schäden zu vermeiden.

Umweltbedingungen

Das Produkt sollte in einer sauberen, trockenen Umgebung verwendet werden, um Schäden durch Feuchtigkeit oder Staub zu vermeiden. Schützen Sie das Produkt vor direkter Sonneneinstrahlung (UV)

Bestimmungsgemäße Verwendung

Das Produkt ist das für den Einsatz in Bildungs-, Forschungs- und Entwicklungsumgebungen konzipiert wurde. Es dient zur Entwicklung, Programmierung und Prototypenentwicklung von elektronischen Projekten und Anwendungen. Das Sensor Produkt ist nicht als fertiges Verbraucherprodukt gedacht, sondern als Werkzeug für technisch versierte Nutzer, darunter Ingenieure, Entwickler, Forscher und Studenten.

Nicht bestimmungsgemäße vorhersehbare Verwendung

Das Produkt eignet sich nicht für den Industriellen Einsatz oder sicherheitsrelevante Anwendungen. Eine Verwendung des Produkts in Medizingeräten oder für Zwecke der Luft- und Raumfahrt sowie in Fahrzeugen ist nicht zulässig

Entsorgung

Nicht mit dem Hausmüll entsorgen! Ihr Produkt ist entsprechend der europäischen Richtlinie über Elektro- und Elektronik-Altgeräte umweltgerecht zu entsorgen. Die darin enthaltenen, wertvollen Rohstoffe können so der Wiederverwendung zugeführt werden. Die Anwendung dieser Richtlinie trägt zum Umwelt- und Gesundheitsschutz bei. Nutzen Sie die von Ihrer Kommune eingerichtete Sammelstelle zur Rückgabe und Verwertung elektrischer und elektronischer Altgeräte. WEEE-Reg.-Nr.: DE 62624346

Elektrostatische Entladung

Achtung: Elektrostatische Entladungen können das Produkt beschädigen. Hinweis: Erden Sie sich, bevor Sie das Produkt berühren, indem Sie beispielsweise ein antistatisches Armband tragen oder eine geerdete Metalloberfläche berühren.

Sicherheitshinweise

Obwohl unser Produkt den Anforderungen der RoHS-Richtlinie (2011/65/EU) entspricht und keine gefährlichen Stoffe in über den Grenzwerten zulässigen Mengen enthält, können dennoch Rückstände vorhanden sein. Beachten Sie die folgenden Sicherheitshinweise, um chemische Gefährdungen zu vermeiden: Achtung: Beim Löten können Dämpfe entstehen, die gesundheitsschädlich sein können. Hinweis: Verwenden Sie einen Lötdampfabsauger oder arbeiten Sie in einem gut belüfteten Bereich. Tragen Sie gegebenenfalls eine Atemschutzmaske. Achtung: Einige Personen könnten empfindlich auf bestimmte Materialien oder Chemikalien reagieren, die im Produkt enthalten sind. Hinweis: Sollten Hautreizungen oder allergische Reaktionen auftreten, unterbrechen Sie die Nutzung und suchen Sie gegebenenfalls einen Arzt auf. Achtung: Halten Sie das Produkt außerhalb der Reichweite von Kindern und Haustieren, um versehentlichen Kontakt und Verschlucken von Kleinteilen zu vermeiden. Hinweis: Bewahren Sie das Produkt in einem sicheren, geschlossenen Behälter auf, wenn es nicht verwendet wird. Achtung: Vermeiden Sie den Kontakt des Produkts mit Nahrungsmitteln und Getränken. Hinweis: Lagern und verwenden Sie das Produkt nicht in der Nähe von Lebensmitteln, um eine Kontamination zu verhindern. Obwohl unser Produkt den

Anforderungen der RoHS-Richtlinie (2011/65/EU) entspricht und keine gefährlichen Stoffe in über den Grenzwerten zulässigen Mengen enthält, können dennoch Rückstände vorhanden sein. Beachten Sie die folgenden Sicherheitshinweise, um chemische Gefährdungen zu vermeiden: Achtung: Beim Löten können Dämpfe entstehen, die gesundheitsschädlich sein können. Hinweis: Verwenden Sie einen Lötdampfabsauger oder arbeiten Sie in einem gut belüfteten Bereich. Tragen Sie gegebenenfalls eine Atemschutzmaske. Achtung: Einige Personen könnten empfindlich auf bestimmte Materialien oder Chemikalien reagieren, die im Produkt enthalten sind. Hinweis: Sollten Hautreizungen oder allergische Reaktionen auftreten, unterbrechen Sie die Nutzung und suchen Sie gegebenenfalls einen Arzt auf. Achtung: Halten Sie das Produkt außerhalb der Reichweite von Kindern und Haustieren, um versehentlichen Kontakt und Verschlucken von Kleinteilen zu vermeiden. Hinweis: Bewahren Sie das Produkt in einem sicheren, geschlossenen Behälter auf, wenn es nicht verwendet wird. Achtung: Vermeiden Sie den Kontakt des Produkts mit Nahrungsmitteln und Getränken. Hinweis: Lagern und verwenden Sie das Produkt nicht in der Nähe von Lebensmitteln, um eine Kontamination zu verhindern. Das Produkt enthält empfindliche elektronische Komponenten und scharfe Kanten. Unsachgemäßes Umgang oder Montage kann zu Verletzungen oder Beschädigungen führen. Beachten Sie die folgenden Sicherheitshinweise, um mechanische Gefährdungen zu vermeiden: Achtung: Die Platine und die Anschlüsse des Produkts können scharfe Kanten aufweisen. Gehen Sie vorsichtig vor, um Schnittverletzungen zu vermeiden. Hinweis: Tragen Sie bei der Handhabung und Montage des Produkts geeignete Schutzhandschuhe. Achtung: Vermeiden Sie übermäßigen Druck oder mechanische Belastung der Platine und der Komponenten. Hinweis: Montieren Sie das Produkt nur auf stabilen und ebenen Oberflächen. Verwenden Sie geeignete Abstandshalter und Gehäuse, um mechanische Belastungen zu minimieren. Achtung: Stellen Sie sicher, dass das Produkt sicher befestigt ist, um unbeabsichtigtes Verrutschen oder Herunterfallen zu verhindern. Hinweis: Verwenden Sie passende Unterlage oder eine sicheren Befestigung in Gehäusen oder auf Montageplatten. Achtung: Achten Sie darauf, dass alle Kabelverbindungen sicher und korrekt angeschlossen sind, um Zugbelastungen und versehentliches Herausziehen zu vermeiden. Hinweis: Führen Sie Kabel so, dass sie nicht unter Spannung stehen und keine Stolpergefahr darstellen. Das Produkt arbeitet mit elektrischen Spannungen und Strömen, die bei unsachgemäßem Gebrauch zu elektrischen Schlägen, Kurzschlägen oder anderen Gefahren führen können. Beachten Sie die folgenden Sicherheitshinweise, um elektrische Gefährdungen zu vermeiden: Achtung: Verwenden Sie das Produkt nur mit den spezifizierten Spannungen. Hinweis: Die Leistungsgrenzen des Produkts finden Sie im dazugehörigen Datenblatt Achtung: Vermeiden Sie Kurzschlüsse zwischen den Anschlüssen und Komponenten des Produkts Hinweis: Achten Sie darauf, dass keine leitenden Objekte die Platine berühren oder überbrücken. Verwenden Sie isolierte Werkzeuge und beachten Sie die Anordnung der Verbindungen. Achtung: Führen Sie keine Arbeiten am Produkt durch, wenn es mit einer Stromquelle verbunden ist. Hinweis: Trennen Sie das Produkt von der Stromversorgung, bevor Sie Änderungen an der Schaltung vornehmen oder Komponenten anschließen bzw. entfernen. Achtung: Überschreiten Sie nicht die spezifizierten Stromstärken für die Ein- und Ausgänge des Produkts. Hinweis: Die Leistungsgrenzen des Produkts finden sich in den technischen Spezifikationen oder im Datenblatt Achtung: Stellen Sie sicher, dass die verwendeten Stromquellen stabil und korrekt dimensioniert sind. Hinweis: Verwenden Sie nur geprüfte und geeignete Netzteile, um Spannungsschwankungen und Überlastungen zu vermeiden. Achtung: Halten Sie ausreichenden Abstand zu spannungsführenden Teilen ein, um unabsichtlichen Kontakt zu vermeiden. Hinweis: Sorgen Sie entsprechend der verwendeten Spannung für eine sichere und übersichtliche Anordnung der Verkabelung. Achtung: Verwenden Sie isolierende Gehäuse oder Schutzbabdeckungen, um das Produkt vor direktem Kontakt zu schützen. Hinweis: Setzen Sie das Produkt in ein nicht leitendes Gehäuse ein, um versehentliche Berührungen und Kurzschlüsse zu vermeiden. Das Produkt und die darauf befindlichen Komponenten können sich während des Betriebs erwärmen. Unsachgemäßes Umgang oder eine Überlastung des Produkts kann zu Verbrennungen, Beschädigungen oder Bränden führen. Beachten Sie die folgenden Sicherheitshinweise, um thermische Gefährdungen zu vermeiden: Achtung: Stellen Sie sicher, dass das Produkt innerhalb der empfohlenen Betriebstemperaturen verwendet wird. Hinweis: Der empfohlene Betriebstemperaturbereich liegt typischerweise zwischen -40°C und +85°C. Überprüfen Sie die spezifischen Angaben im Datenblatt des Produkts. Achtung: Platzieren Sie das Produkt nicht in der Nähe von externen Wärmequellen wie Heizkörpern oder direkter Sonnenstrahlung. Hinweis: Sorgen Sie dafür, dass das Produkt in einem kühlen und gut belüfteten Bereich betrieben wird. Achtung: Stellen Sie sicher, dass das Produkt gut belüftet ist, um eine Überhitzung zu vermeiden. Hinweis: Verwenden Sie Lüfter oder Kühlkörper, wenn das Produkt in einem geschlossenen Gehäuse betrieben wird oder in einer Umgebung mit eingeschränkter Luftzirkulation. Achtung: Montieren Sie das Produkt auf hitzebeständigen Oberflächen und in hitzebeständigen Gehäusen. Hinweis: Verwenden Sie Materialien für Gehäuse, die hohe Temperaturen aushalten können, um eine Beschädigung oder Feuergefahr zu vermeiden. Achtung: Implementieren Sie eine Überwachung der Temperatur bei Verwendung eines Gehäuses und gegebenenfalls Schutzmechanismen, die das Produkt abschalten, wenn es überhitzt. Hinweis: Verwenden Sie Temperatutfühler und entsprechende Software, um die Temperatur des Produkts zu überwachen und das System bei Bedarf abzuschalten. Achtung: Vermeiden Sie Überlastungen, die zu übermäßiger Erwärmung der Komponenten führen können. Hinweis: Überschreiten Sie nicht die spezifizierten Grenzwerte für Strom und Spannung, um eine Überhitzung zu verhindern. Achtung: Kurzschlüsse können erhebliche Hitze entwickeln und Brände verursachen. Hinweis: Stellen Sie sicher, dass alle Verbindungen korrekt und sicher sind und dass keine leitenden Objekte unbeabsichtigt Kurzschlüsse verursachen können.

Az-Delivery

Das Ethernet Shield verbindet schnell und einfach Ihr Mikrocontroller Board mit dem Internet. Sie müssen nur das Modul auf Ihr Mikrocontroller Board stecken, mit einem LAN-Kabel mit Ihrem Netzwerk verbinden (nicht enthalten) und ein paar einfachen Anweisungen folgen, um Ihr Mikrocontroller Board basiertes Projekt über das Inter-net umzusetzen.

Benötigte Komponenten:

- ein Mikrocontroller Board (nicht enthalten)
- Betriebsspannung 5V (wird vom Mikrocontroller Board bereitgestellt)
- Übertragungsgeschwindigkeit: 10/100Mb (via LAN-Kabel, nicht enthalten)
- Verbindung mit Mikrocontroller Board durch ein SPI Port

Das Ethernet Shield ermöglicht einem Mikrocontroller Board sich mit dem Internet zu verbinden. Es basiert auf einem Wiznet W5100 Ethernet Chip. Der Wiznet W5100 unterstützt sowohl TCP als auch UDP Netzwerk-Verbindungen. Es unterstützt bis zu vier gleichzeitige Netzwerkverbindungen. Nutzen Sie die in der Arduino IDE vorinstallierte Ethernet-Bibliothek, um Sketches zu schreiben, die sich über das Shield mit dem Internet verbinden. Das Ethernet Shield wird über ein Kabel mit dem Mikrocontroller Board verbunden, damit ein weiteres Shield aufgesteckt werden kann. Das Ethernet Shield hat eine Standard RJ-45 Verbin-dung, mit einem integrierten Leitungstransformator und einem aktivierten PoE. Auf dem Board befindet sich ein Micro SD-Kartenslot, der benutzt werden kann, um Dateien über das Netzwerk zu speichern. Er ist mit allen Mikrocontrollers Boards kompatibel.

Az-Delivery

Der auf dem Board verbaute SD-Kartenleser ist durch die SD-Bibliothek erreichbar, die ebenfalls auf dem Arduino IDE vorinstalliert ist. In der Bibliothek liegt SS auf Pin 4.

Das Shield beinhaltet auch einen Reset-Controller, um sicherzustellen, dass sich das W5100 Ethernet Modul ordnungsgemäß nach dem Einschalten zurücksetzt. Ältere Modelle des Shields waren nicht mit dem Mega kompatibel und mussten manuell nach dem Einschalten zurückgesetzt werden.

Eigenschaften:

- Mit dem Ethernet Shield kann sich Ihr Mikrocontroller Board mit dem Internet verbinden
- Kann sowohl als Client als auch als Server benutzt werden
- Direkte Steckverbindung, kein Löten notwendig
- Aufgrund des Wiznet W5100 Ethernet Chips können Sie Ihren Mikrocontroller Board leicht online nutzen
- Direkte Unterstützung durch eine offizielle Ethernet Bibliothek
- Beinhaltet einen Micro-SD Kartenslot, um Dateien über das Netzwerk speichern zu können
- Mit Mikrocontroller Board Duemilanove, Uno und Mega verwendbar,
- Der Wiznet W5100 unterstützt (IP) TCP und UDP Netzwerkverbindungen.
- unterstützt bis zu vier simultane Netzwerkverbindungen
- Das Shield verbindet sich mit dem Internet über Sketches, die die Ethernet-Bibliothek einbinden

Az-Delivery

Der Mikrocontroller Board kommuniziert sowohl mit dem W5100 als auch mit der SD-Karte über den SPI Bus (durch die ICSP Buchse). Dies entspricht den digitalen Pins 10, 11, 12, und 13 auf dem Uno und den Pins 50, 51, und 52 auf dem Mega. Auf beiden Boards wird Pin 10 benutzt, um den W5100 auszuwählen und Pin 4 für die SD-Karte. Diese Pins können generell nicht für I/O genutzt werden. Auf dem Mega wird der Hardware SS Pin 53 nicht benutzt, um den W5100 oder die SD-Karte auszuwählen, muss allerdings als Ausgabe geschaltet sein, ansonsten funktioniert das SPI Interface nicht.

Bitte beachten Sie, dass sich der W5100 und die SD-Karte den SPI Bus teilen. Aus diesem Grund können beide nicht gleichzeitig aktiv sein. Sollten Sie beide Peripheriegeräte in Ihrem Programm nutzen wollen, sollte dieses Problem von den entsprechenden Bibliotheken gelöst werden. Sollten Sie jedoch eines der beiden Peripheriegeräte nicht benutzen, müssen Sie es explizit deaktivieren. Dafür müssen Sie bei der SD-Karte Pin 4 als Ausgabe deklarieren und auf dem Pin ein High schreiben. Für den W5100 setzen Sie den digitalen Pin 10 als Ausgabe und auf High.

Das Shield stellt eine Standard RJ-45 Ethernet Buchse zur Verfügung.

Der Reset-Button auf dem Shield setzt sowohl den W5100 als auch das Mikrocontroller Board zurück.

Az-Delivery

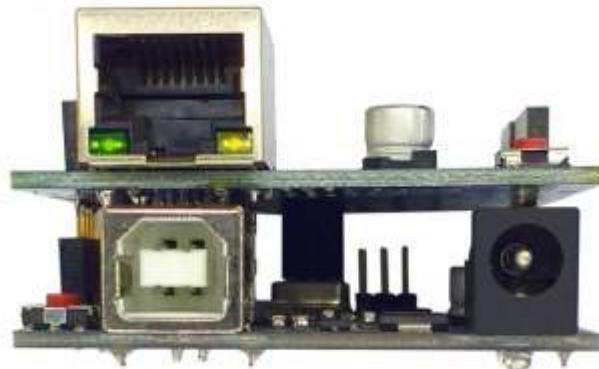
Das Shield beinhaltet einige Status- LEDs:

- **PWR:** Zeigt an, dass Board und Shield an sind
- **LINK:** Zeigt eine stehende Netzwerkverbindung an und blinkt bei Senden oder Empfangen von Daten auf dem Shield
- **FULLD:** Zeigt eine Vollduplex-Netzwerkverbindung an
- **100M:** Zeigt die Verfügbarkeit einer 100 Mb/s Netzwerkverbindung (im Gegensatz zu 10 Mb/s)
- **RX:** Blinkt, wenn das Shield Daten empfängt
- **TX:** Blinkt, wenn das Shield Daten sendet
- **COLL:** Blinkt, wenn Netzwerkkollisionen registriert werden

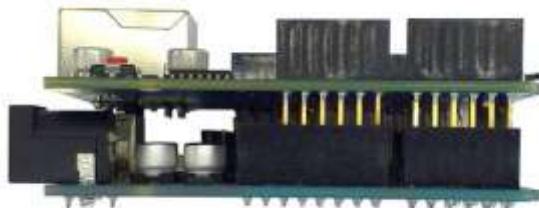
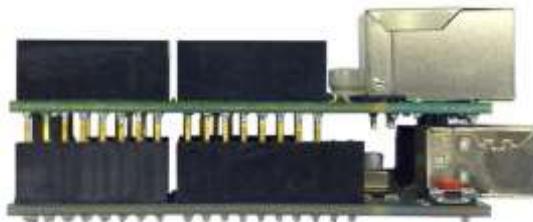
Die mit “INT” markierte Lötstelle kann mit dem Mikrocontroller Board verbunden werden, um Mitteilungen über Zwischenfälle von Programmunterbrechungen auf dem W5100 zu erhalten. Dies wird allerdings nicht von der Ethernet-Bibliothek unterstützt. Die Leitung verbindet den INT-Pin des W5100 mit dem digitalen Pin 2 auf dem Mikrocontroller Board.

Verbinden des Shields mit dem Atmega328P Board

Sie müssen das Shield nur auf das Mikrocontroller Board stecken. Auf den nachfolgenden Bildern wird dies mit einem Atmega328P Board gezeigt:



Hier befindet sich eine Micro SD-Karte (nicht enthalten)!





Die Bibliothek

Die Bibliothek für dieses Shield ist bereits mit Arduino IDE vorinstalliert. Dadurch haben Sie viele Sketch-Beispiele, die Sie ausprobieren können. Sie müssen nur Ihr Shield mit dem Mikrocontroller Board verbinden, ein LAN-Kabel mit dem Shield und ein USB-Kabel mit dem Mikrocontroller Board.

Falls Sie die Arduino IDE nicht installiert haben, können Sie die Software für Ihr Betriebssystem unter folgendem Link herunterladen:

<https://www.arduino.cc/en/main/software>

Folgenden Sie den Anweisungen in der Installationssoftware.

Nach der Installation nutzen Sie die Menüpunkte File > Examples > Ethernet > Webserver und laden den Webserver Sketch herunter. Der Sketch erstellt einen Webserver und eine Webseite, liest alle analogen Eingaben und schreibt die Werte auf die Webseite.

Zuerst müssen Sie die IP-Adresse aus dem IP-Adressen-Pool für Ihr lokales Netzwerk suchen. Hier können wir eine Adresse des Adresspools „192.168.0.0“ (Netzwerkmaske 255.255.255.0) nutzen. 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

Az-Delivery

Für Windows OS 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.

In diesem Sketch ändern wir zuerst diese Codezeile:

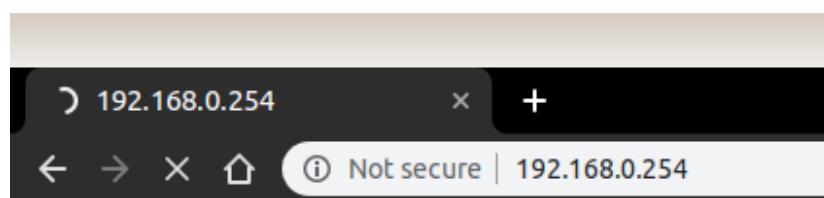
```
IPAddress ip (192, 168, 1, 177);
```

zu

```
IPAddress ip (192, 168, 0, 254);
```

Oder zu einer IP-Adresse, die zu Ihrem IP-Adressen-Pool gehört.

Wenn Sie den Sketch auf Ihren Mikrocontroller Board geladen haben, öffnen Sie Ihren Inter-netbrowser und tippen Sie 192.168.0.254 ein. Folgende Webseite sollte geladen werden:



```
analog input 0 is 327
analog input 1 is 283
analog input 2 is 247
analog input 3 is 228
analog input 4 is 251
analog input 5 is 277
```

Az-Delivery

Wenn Sie Ihren Serial Monitor öffnen, (Tools > Serial Monitor)
sollte die Ausgabe folgendermaßen aussehen:

The screenshot shows a terminal window titled '/dev/ttyUSB0'. The window contains the following text:

```
Ethernet WebServer Example
server is at 192.168.0.254
new client
FDT . HTTP.0.0
Hnrt: 082.068.0.244
Bnnndbthnn: jddp,`lhvd
B`bhd,Bnntrnl: l`x,`fd<0
Tpfr`dd,Hnrdbtrd,Rdptdrtr: 0
DNT: 0
Trdr,@fdnt: Lnhll`4.0 (X00: Lhntx x86^64( @ppldVdbJht.426.26
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,i
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

At the bottom of the window, there are several configuration options:

- Autoscroll Show timestamp
- Both NL & CR
- 9600 baud
- Clear output

Die Webseite aktualisiert sich alle 5 Sekunden. Dies kann durch Ändern der Zahl 5 in dieser Code-Zeile behoben werden:

```
client.println("Refresh: 5"); //refresh the page automatically
every 5
```

Az-Delivery

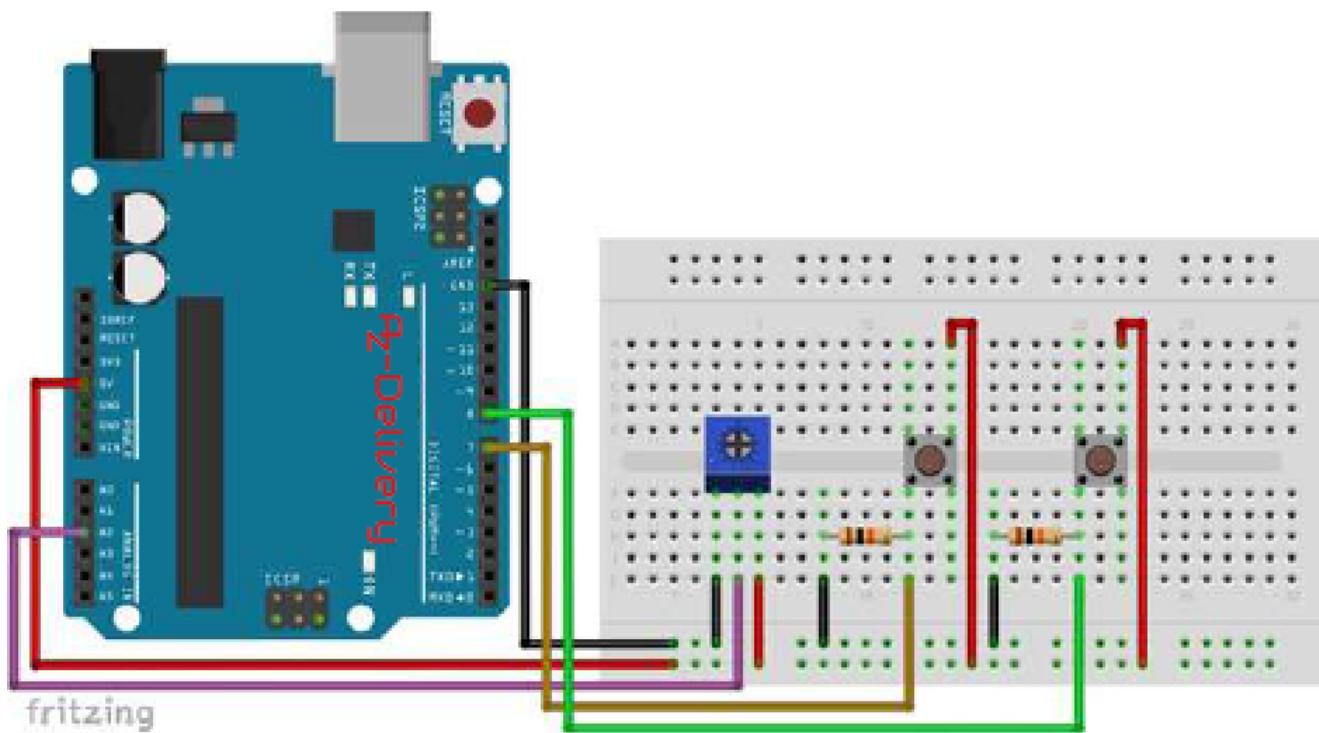
Senden der Webseite von der SD-Karte mit AJAX

Mit diesem Programmbeispiel versuchen wir einen Webserver einzurichten, der eine Webseite ausgibt, die in der Lage ist den Status von zwei Druckknöpfen und die Spannung eines Potentiometers zu lesen.

Die Knöpfe sind mit den digitalen I/O-Pin 7 und 8 und das Potentiometer mit dem analogen Pin 2 auf dem Atmega328P Board verbunden.

In dem Sketch wird AJAX verwendet, um die Werte auf der Webseite zu aktualisieren ohne die ganze Webseite neu zu laden.

Verbinden Sie alles so, wie auf dem Verbindungsdiagramm unten:



Az-Delivery

Wie Sie sehen können benutzen wir für jeden Knopf einen Pull-Down-Widerstand von $10k\Omega$, was bedeutet, dass wenn der Knopf nicht gedrückt wird ein Low-Level-Signal (oder 0 V) an den digitalen Input-Pin ausgegeben wird. Sobald der Knopf gedrückt wird, liest der digitale Input-Pin ein High-Level-Signal.

Wir verbinden das Ethernet Shield mit dem Mikrocontroller Board genau wie in den vorangegangenen Kapiteln gezeigt und benutzen die Micro SD-Karte, die als einzige Datei "index.htm" gespeichert haben sollte. Sie können einen beliebigen Texteditor benutzen, um die Datei zu erstellen und mit folgendem html und javascript Code zu füllen:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Arduino SD Card Web Page using Ajax with XML</title>
        <script>
            function GetArduinoInputs() {
                nocache = "&nocache=" + Math.random() * 1000000; var request = new XMLHttpRequest(); request.onreadystatechange = function() {
                    if (this.readyState == 4) {
                        if (this.status == 200) {
                            if (this.responseXML != null) {
                                // extract XML data from XML file (containing switch states and analog value)
                                document.getElementById("input1").innerHTML = this.responseXML.getElementsByTagName('button1')[0].childNodes[0].nodeValue;
                                document.getElementById("input2").innerHTML = this.responseXML.getElementsByTagName('button2')[0].childNodes[0].nodeValue;
                                document.getElementById("input3").innerHTML = this.responseXML.getElementsByTagName('analog1')[0].childNodes[0].nodeValue;
                            }
                        }
                    }
                }
            }
        </script>
    </head>
    <body>
        <input type="text" id="input1" value=""/>
        <input type="text" id="input2" value=""/>
        <input type="text" id="input3" value=""/>
    </body>
</html>
```

Az-Delivery

```
(3 tabs)    request.open("GET", "ajax_inputs" + nocache, true);  
    request.send(null);  
    setTimeout('GetArduinoInputs()', 1000);  
}  
</script>  
</head>  
  
<body onload="GetArduinoInputs()">  
    <h1>Arduino Inputs from SD Card Web Page using Ajax  
    with XML</h1>  
    <p>Button 1 (pin 7): <span id="input1">...</span></p>  
    <p>Button 2 (pin 8): <span id="input2">...</span></p>  
    <p>Analog (A2): <span id="input3">...</span></p>  
</body>  
</html>
```

Nachdem Sie die Datei erstellt haben, müssen Sie diese auf die Micro SD-Karte und stecken Sie die Karte nachfolgend in das Shield.

Danach müssen Sie noch den Sketch in Ihre Arduino IDE kopieren. Der Code ist auf der nächsten Seite. Sie müssen jedoch sicherstellen, dass Sie die IP-Adresse wie vorhergehend besprochen selbst einstellen. In unserem Beispiel haben wir die Adresse 192.168.0.254 gewählt.

Az-Delivery

Wir erklären Ihnen nun das Programm.

Zu Beginn fügen wir drei Bibliotheken hinzu. Die SPI-Bibliothek ist für die Kommunikation zwischen Shield und SD-Karte mit dem Mikrocontroller Board verantwortlich. Die Zweite, die Ethernet-Bibliothek, ist für das Ethernet-Shield und die Dritte für die SD-Karte auf dem Shield.

Nun stellen wir die MAC-Adresse des Shields ein, da diese für eine erfolgreiche Verbindung zum Internet benötigt wird. Danach wird die IP-Adresse wie vorher besprochen festgelegt. Nachfolgend wird der Server als Objekt eingerichtet und der Serverport auf den Port 80 gelegt.

Im Setup-Bereich wird der Ethernetchip zuerst deaktiviert, da über das SPI-Interface immer nur ein Gerät betrieben werden kann. Das Mikrocontroller-Board ist hierbei der Master, der Ethernetchip slave Nummer 1 und die SD-Karte slave Nummer 2. Um nun die SD-Karte einzurichten, muss zuerst der Ethernetchip deaktiviert werden. Danach wird das serielle Interface mit einer Baudrate von 9600 eingerichtet. Dieses benutzen wir für das Debugging.

Es wird überprüft, ob im SD-Kartenslot eine Karte steckt und falls ja, ob die Datei „index.htm“ existiert. Danach erfolgt eine Ausgabe auf das serielle Interface.

Danach werden die digitalen I/O-Pins 7 und 8 als Eingabepins deklariert und der Webserver gestartet.

Az-Delivery

```
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>
// size of buffer used to capture HTTP requests
#define REQ_BUF_SZ 50
// MAC address from Ethernet shield sticker under board
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 0, 254); // IP address, may need to change
// depending on network
EthernetServer server(80); // create a server at port 80
File webFile; // the web page file on the SD card
char HTTP_req[REQ_BUF_SZ] = {0}; // buffered HTTP request stored
// as null terminated string
char req_index = 0; // index into HTTP_req buffer
void setup() {
    // disable Ethernet chip
    pinMode(10, OUTPUT);
    digitalWrite(10, HIGH);
    Serial.begin(9600); // for debugging
    Serial.println("Initializing SD card..."); // initialize SD
    card
    if(!SD.begin(4)) {
        Serial.println("ERROR - SD card initialization failed!");
        return; // init failed
    }
    Serial.println("SUCCESS - SD card initialized.");
    // check for index.htm file
    if(!SD.exists("index.htm")) {
        Serial.println("ERROR - Can't find index.htm file!");
        return; // can't find index file
    }
    Serial.println("SUCCESS - Found index.htm file.");
    pinMode(7, INPUT); // switch is attached to pin 7
    pinMode(8, INPUT); // switch is attached to pin 8 //
    Ethernet.begin(mac, ip); // initialize Ethernet device
    server.begin(); // start to listen for clients
}
```

Az-Delivery

```
#include <Ethernet.h>
#include <SD.h>
// size of buffer used to capture HTTP requests
#define REQ_BUF_SZ 50
// MAC address from Ethernet shield sticker under board
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 0, 254); // IP address, may need to change
// depending on network
EthernetServer server(80); // create a server at port 80
File webFile; // the web page file on the SD card
char HTTP_req[REQ_BUF_SZ] = {0}; // buffered HTTP request stored
// as null terminated string
char req_index = 0; // index into HTTP_req buffer
void setup() {
    // disable Ethernet chip
    pinMode(10, OUTPUT);
    digitalWrite(10, HIGH);
    Serial.begin(9600); // for debugging
    Serial.println("Initializing SD card..."); // initialize SD card
    if(!SD.begin(4)) {
        Serial.println("ERROR - SD card initialization failed!");
        return; // init failed
    }
    Serial.println("SUCCESS - SD card initialized.");
    // check for index.htm file
    if(!SD.exists("index.htm")) {
        Serial.println("ERROR - Can't find index.htm file!");
        return; // can't find index file
    }
    Serial.println("SUCCESS - Found index.htm file.");
    pinMode(7, INPUT); // switch is attached to Arduino pin 7
    pinMode(8, INPUT); // switch is attached to Arduino pin 8
    Ethernet.begin(mac, ip); // initialize Ethernet device
    server.begin(); // start to listen for clients
}
```

Az-Delivery

```
void loop() {
    EthernetClient client = server.available(); // try to get client
    if(client) { // got client?
        boolean currentLineIsBlank = true;
        while(client.connected()) {
            if(client.available()) { // client data available to read
                char c = client.read(); // read 1 byte (character) from client
                // buffer first part of HTTP request in HTTP_req array (string)
                // leave last element in array as 0 to null terminate string (REQ_BUF_SZ - 1)
                if(req_index < (REQ_BUF_SZ - 1)) {
                    HTTP_req[req_index] = c; // save HTTP request character
                    req_index++;
                }
                // last line of client request is blank and ends with \n
                // respond to client only after last line received
                if(c == '\n' && currentLineIsBlank) {
                    // send a standard http response header
                    client.println("HTTP/1.1 200 OK");
                    // remainder of header follows below, depending on if
                    // web page or XML page is requested
                    // Ajax request - send XML file
                    if(StrContains(HTTP_req, "ajax_inputs")) {
                        // send rest of HTTP header
                        client.println("Content-Type: text/xml");
                        client.println("Connection: keep-alive");
                        client.println();
                        // send XML file containing input states
                        XML_response(client);
                    }
                    else { // web page request
                        // send rest of HTTP header
                        client.println("Content-Type: text/html");
                        client.println("Connection: keep-alive");
                        client.println();
                        // send web page
                        webFile = SD.open("index.htm"); // open web page file
                        if(webFile) {
                            while(webFile.available()) {
                                // send web page to client
                                client.write(webFile.read());
                            }
                            webFile.close(); }
                    }
                }
            }
        }
    }
}
```

Az-Delivery

```
(≡ tabs)        }
    // display received HTTP request on serial port
    Serial.print(HTTP_req);
    // reset buffer index and all buffer elements to 0
    req_index = 0;
    StrClear(HTTP_req, REQ_BUF_SZ);
    break;
}
// every line of text received from the client ends with \r\n
if(c == '\n') {
    // last character on line of received text
    // starting new line with next character read
    currentLineIsBlank = true;
}
else if(c != '\r') {
    // a text character was received from client
    currentLineIsBlank = false;
}
} // end if (client.available())
} // end while (client.connected())
delay(1);      // give the web browser time to receive the data
client.stop(); // close the connection
} // end if (client)
}

// send the XML file with switch statuses and analog value
void XML_response(EthernetClient cl) {
    int analog_val;
    cl.print("<?xml version = \"1.0\" ?>");
    cl.print("<inputs>");
    cl.print("<button1>");
    if(digitalRead(7)) {
        cl.print("ON");
    }
    else {
        cl.print("OFF");
    }
    cl.print("</button1>");
    cl.print("<button2>");
    if (digitalRead(8)) {
        cl.print("ON");
    }
}
```

Az-Delivery

```
(1 tab)
else {
    cl.print("OFF");
}
cl.print("</button2>");
// read analog pin A2
analog_val = analogRead(2);
cl.print("<analog1>");
cl.print(analog_val);
cl.print("</analog1>");
cl.print("</inputs>");

}
// sets every element of str to 0 (clears array)
void StrClear(char *str, char length) {
    for(int i = 0; i < length; i++) {
        str[i] = 0;
    }
}
// searches for the string sfind in the string str
// returns 1 if string found
// returns 0 if string not found
char StrContains(char *str, char *sfind) {
    char found = 0;
    char index = 0;
    char len;
    len = strlen(str);
    if(strlen(sfind) > len) {
        return 0;
    }
    while(index < len) {
        if(str[index] == sfind[found]) {
            found++;
            if(strlen(sfind) == found) {
                return 1;
            }
        }
        else {
            found = 0;
        }
        index++;
    }
    return 0;
}
```

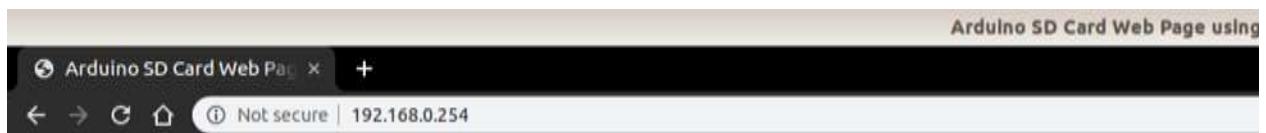
Az-Delivery

In der Schleifenfunktion wird darauf gewartet, dass sich ein Client mit dem Webserver verbindet. Sobald dies geschieht, stellt der Server die Webseite bereit.

Wir gehen nachfolgend nicht ins Detail wie die Webseite übertragen wird, weil wir dafür HTML, Javascript und die Arduino IDE-Sprache verwenden. Zusätzlich erstellen wir eine XML-Datei, um Daten zwischen dem Webserver und der Webseite auszutauschen (Dadurch erfolgt dieser schneller.) Wir erstellen eine XML-Datei in diesem Sketch nur als Beispiel. Wir benutzen die XML_response-Funktion, um die Datei zu erstellen und um die Status der digitalen Input-Pins 7 und 8 und den Wert des analogen Input-Pins 2 zu lesen.

Das Programm ist ausführlich kommentiert und Sie sollten leicht jeden entsprechenden Programmteil finden und verstehen, was für was benötigt wird.

Sobald Sie den Sketch auf das Mikrocontroller-Board geladen haben, öffnen Sie einen Webbrower und geben die von Ihnen im Sketch verwendete IP-Adresse in die Adresszeile ein. In unserem Beispiel ist das die 192.168.0.254. Drücken Sie En-ter und die Webseite sollte in Ihrem Browser wie folgt angezeigt werden:



Arduino Inputs from SD Card Web Page using Ajax with XML

Button 1 (pin 7): OFF

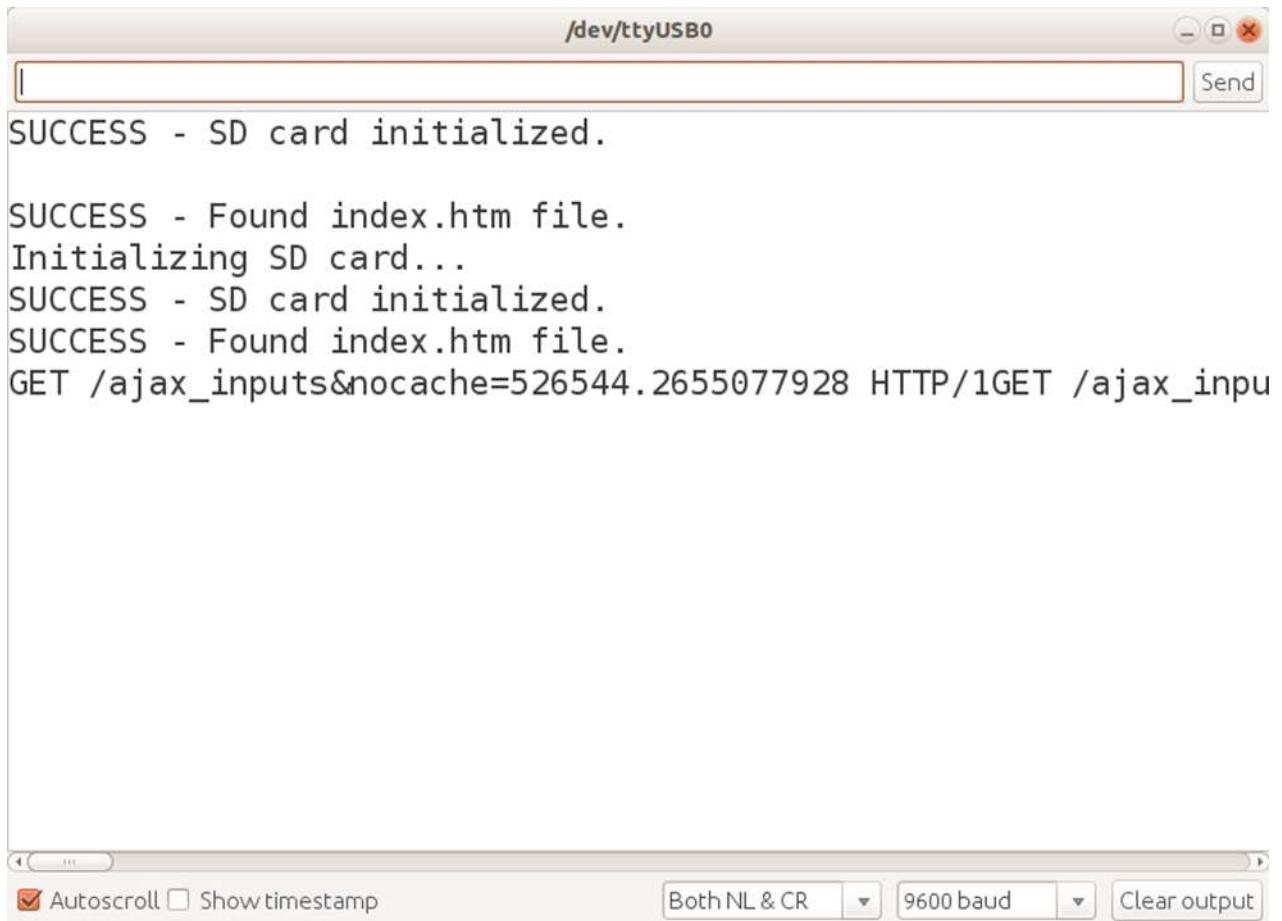
Button 2 (pin 8): OFF

Analog (A2): 203

Az-Delivery

Sobald Sie einen der Knöpfe betätigen, ändert sich der OFF-Status zu einem ON-Status, ohne dass die Webseite neu geladen wird. Sobald Sie den Drehknopf des Potentiometers betätigen sollte sich der Wert der analogen Variable im Bereich von 0 bis 1023 verändern. Der Wert wird jede Sekunde einmal aktualisiert.

Wenn Sie den Serial Monitor (*Tools > Serial Monitor*) öffnen, sollte die Ausgabe wie folgt aussehen.



The screenshot shows the Arduino Serial Monitor window titled '/dev/ttyUSB0'. The main text area displays the following log output:

```
SUCCESS - SD card initialized.  
SUCCESS - Found index.htm file.  
Initializing SD card...  
SUCCESS - SD card initialized.  
SUCCESS - Found index.htm file.  
GET /ajax_inputs&nocache=526544.2655077928 HTTP/1GET /ajax_inpu
```

At the bottom of the window, there are several configuration options: 'Autoscroll' (checked), 'Show timestamp' (unchecked), 'Both NL & CR' (selected), '9600 baud' (selected), and 'Clear output'.

Sie haben es geschafft. Nun können Sie das Modul für Ihre Projekte verwenden.

AZ-Delivery

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 Hochwertige Mikroelektronik und Zubehör sind, sind wir von AZ-Delivery Vertriebs GmbH der richtige Ansprech-partner. Wir unterstützen Sie mit vielen Anwendungsbeispielen, Einrichtungs-hilfen, eBooks, Bibliotheken und natürlich unseren Technik-Experten!

<https://az-delivery.de>

Viel Spaß!

Impressum

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