

Ökoserver

Mathias Gebhardt
(Gymnasium Michelstadt)

Inhaltsverzeichnis:

1. Einleitung.....	2
2. Die Hardware des Ökoservers	3
2.1. Vor- und Nachteile eines Computers.....	3
2.2. Vor- und Nachteile des Mikrocontrollers.....	3
2.3. Die komplette Hardware	4
3. Die Software.....	6
4. Ausblick.....	7
5. Danksagungen	8
6. Literatur	8

1. Einleitung

In einem Zeitalter, in dem es immer wichtiger wird, sich im Internet zu präsentieren, benötigt fast jede Firma aber auch die eine oder andere Privatperson eine eigene Webseite. Webseiten müssen aber auch bereitgestellt und gepflegt werden.

Für eine Webpräsenz kann man einerseits einen Server mit URL bei einer Firma „mieten“, andererseits aber auch einen eigenen Webserver benutzen. Dadurch spart man die Kosten, die durch das Mieten der Hardware entstehen.

Als privater Webserver kann auch ein alter PC benutzt werden, der Nachteil ist nur, dass dieser sehr viel Strom verbraucht. Und so bin ich auf die Idee gekommen, einen Mikrocontroller als Webserver zu benutzen.

Das Ziel dieses Projektes ist es also, einen Webserver auf der Basis eines Mikrocontroller zu bauen und einzurichten.

Dieser Webserver sollte geringe Anschaffungskosten haben, einen geringen Stromverbrauch, aber dennoch hohe Leistung erzielen. Er sollte auch andere Geräte an- und ausschalten können.

Ein Teil dieser Aufgaben wird von Schaltungen übernommen, der andere Teil von der Software (vgl. Abb.1). Die Software besteht auch aus zwei Teilen. Der eine Teil ist die Webseite auf der SD-Karte, der andere das Betriebssystem. Dieses stellt die Webseite ins Internet und ermöglicht das Schalten von Geräten.

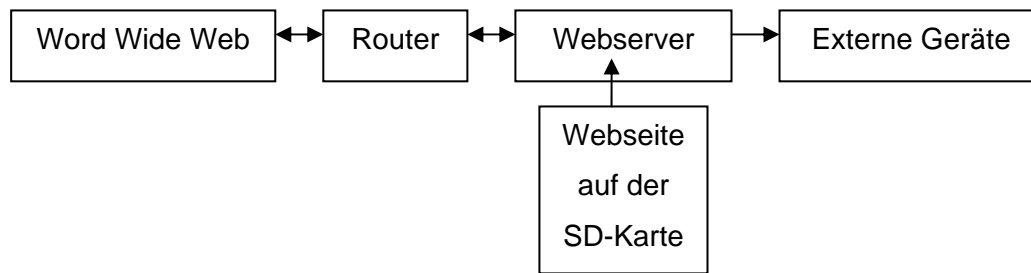


Abb.1: Blockschema des Webservers

2. Die Hardware des Ökoservers

2.1. Vor- und Nachteile eines Computers

Als Hardware für einen Webserver käme ein handelsüblicher Computer oder ein Server in Betracht. Ein großer Vorteil dieser beiden Geräte ist aber, dass zusätzlich zu den Webserverdiensten, wie zum Beispiel IIS oder Apache, auch ein SQL Server bereitgestellt werden kann. Weiterhin ist es möglich diesen Rechner in einer Domäne zu betreiben, um auf deren Benutzer zuzugreifen. Da fast niemand eine private Domäne hat und auch die Bandbreite durch den Internetanbieter begrenzt wird, sind die beiden Rechnertypen fast schon zu leistungsstark für einen privaten Webserver.

Ein weiterer aber noch gravierenderer Nachteil eines Computers oder Server ist es, dass die sehr viel Strom verbrauchen. So hat der Server meiner Schule nach eigenen Messungen einen Stromverbrauch von ca. 200 W. Bei mittelmäßiger bis starker Belastung steigt dieser Wert auf ca. 250 W – 300 W. Ein solcher Stromverbrauch macht sich dann auch auf der Stromrechnung bemerkbar, was einen privaten Webserver auf Computer oder Server aus Kostengründen nicht sehr attraktiv macht, da die Stromkosten wahrscheinlich höher sind, als die „Miete“ für einen Webserver bei einer Firma.

2.2. Vor- und Nachteile des Mikrocontrollers

Ein Mikrocontroller hingegen hat einen viel geringeren Stromverbrauch und geringere Anschaffungskosten als ein Computer oder Server.

Als Mikrocontroller kam für mich zu erst nur ein ATmega32 oder ein anderer der Firma Atmel in Betracht, da ich in vorhergehenden Projekten mit Mikrocontrollern dieser Firma sehr viel Erfahrung gesammelt habe. So bin ich auf einen Webserver von Ulrich Radig [1] gestoßen.

Ein Nachteil eines Webserver auf der Basis eines Mikrocontroller von Atmel ist nur, dass die Leistungsfähigkeit sehr gering ist.

Deshalb habe ich statt einem universell einsetzbaren ATmega ein Mod5270 von NetBurner [2] verwendet. Dieser Mikrocontroller hat der Vorteil, dass er nur für die Verwendung als

Webserver geeignet ist und eine Bandbreite von bis zu 100 Mbits ermöglicht. Diese Bandbreite reicht völlig für den privaten Gebrauch, da die durch den Provider begrenzte Bandbreite auch nicht viel größer ist. Ein weiterer Vorteil dieses Moduls ist, dass es sich, mit einem Stromverbrauch von ca. 0,8 W, fast nicht auf der Stromrechnung bemerkbar macht (vgl. [3]).

2.3. Die komplette Hardware

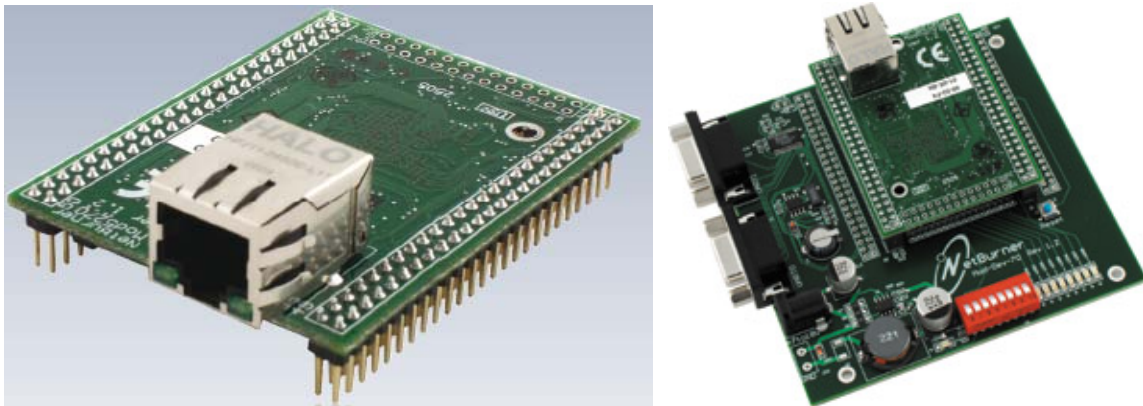


Abb.21: links: MOD5270 (entnommen aus [3]), rechts: MOD5270 mit Entwicklungsboard (entnommen aus [4]).

Zusammen mit dem Entwicklungsboard und einer Programmierumgebung kann der Webserver für \$ 99 bei der Firma NetBurner gekauft werden und ist direkt nach dem Anschließen des Netzkabels und des Stromkabels voll einsatzfähig.

Das Entwicklungsboard verfügt neben 2 seriellen Schnittstellen, 8 LEDs und 8 Schaltern auch über einen SD-Karten-Schlot. So kann man mit der passenden Software die Webseite, die auf der SD-Karte liegt, für das Internet bereitstellen. Jedoch gerät auch dieser in C programmierter Mikrocontroller ab und zu an seine Grenzen oder benötigt Dateien, die nicht auf seiner SD-Karte vorhanden sind. Deshalb soll der „kleine“ Webserver in der Lage sein einen „großen“ Computer anzuschalten und Daten abzufragen. Dies geschieht eine Relais-Schaltung (vgl. Abb.3).

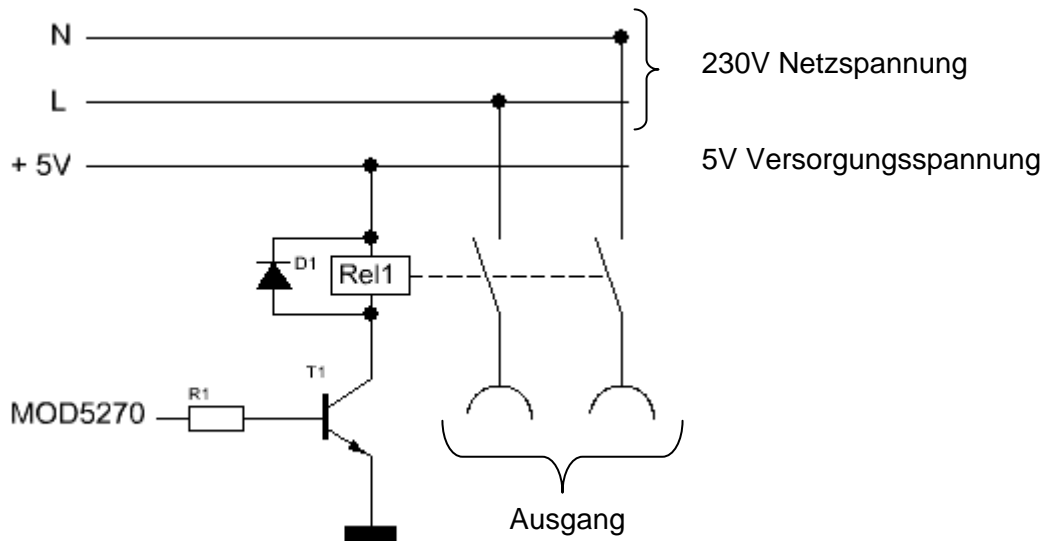


Abb.3: Schaltplan des Netzschalters

Mit dieser Schaltung kann eine Steckdose an oder ausgeschaltet werden. Dadurch ist gewährleistet, dass der Computer oder auch ein anderes Gerät, das gesteuert werden soll, keinen Stand-by-Strom hat und nur dann Strom verbraucht, wenn dies auch geschehen soll. Da die meisten Geräte einen Ein- und Aussteller bzw. –Schalter besitzen, wird noch eine weitere Transistor-Schaltung benötigt, um das Gerät anzuschalten oder auszuschalten (vgl. Abb.4).

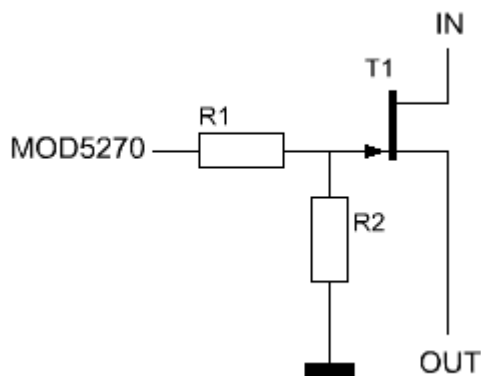


Abb.4: Schaltplan des An- / Ausschalters

Dies ist aber nur eine Möglichkeit eines An- oder Ausschalters. Wenn die Netzspannung vom originalen Schalter geschaltet wird, wird eine Schaltung mit einem Relais benötigt, die ähnlich Abb.2 ist. Man könnte aber auch in diesem Fall den An- und Ausschalter komplett weglassen und das Gerät nur über die Steckdose schalten.

Die zeitliche Abfolge des Schaltvorgangs ist in Abb.4 dargestellt. Auf ihr ist unter anderem zu erkennen, dass die Steckdose angeschaltet werden muss bevor der Anschaltimpuls oder der Anschalter geschaltet wird. Auch darf erst die Steckdose abgeschaltet werden, wenn das Gerät aus ist.

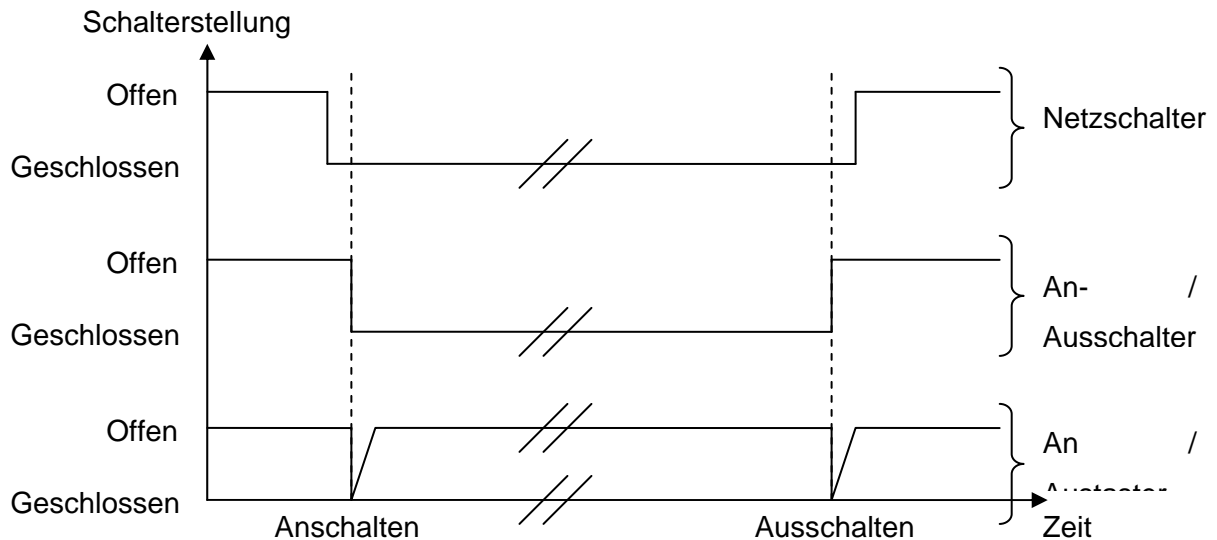


Abb.4: Darstellung der einzelnen Schalterpositionen während des Ein- und Ausschaltens

3. Die Software

Die Hardware wurde komplett und funktionsfähig von NetBurner geliefert. Zusätzlich zu ihr wurden die oben vorgestellten Schaltungen gebaut und mit dem Webserver verbunden.

Jetzt kommen wir zum wichtigsten Teil des Webserver, die Software.

Ohne ein funktionierendes Betriebssystem würde der Webserver nicht einmal auf einen Ping, ein Programm, das Datenpakete an eine IP-Adresse schickt und wieder empfängt, um die Verbindung zu prüfen, reagieren.

Generell sollte hier auch erwähnt werden, dass alle Programmausschnitte in der Programmiersprache C geschrieben sind. Sie wurden entweder aus den Demoprogrammen von NetBurner [6] oder aus dem Betriebssystem von Dr.Dobb's [5] entnommen und von mir selbst an die speziellen Bedürfnisse dieses Projektes angepasst oder komplett selbst geschrieben.

So hat die minimale Version des Betriebssystems folgende Aufgaben zu erledigen (vgl. Listing 1):

- Initialisierung des TCP/IP Stacks
- Beziehung der IP-Adresse über einen DHCP-Server (normalerweise der Router)
- Oder setzen der statischen IP-Adresse
- Starten des http Webservers.

```
InitializeStack();    // Initialize TCP/IP stack

if ( EthernetIP == 0 )
{
    GetDHCPAddress();    // Get the IP-Adress
}

StartHTTP();        // Start the web server
```

Listing 1: Funktionen, die am Start aufgerufen werden müssen (entnommen aus [6]).

Alle drei Funktionen werden von Bibliotheken von NetBurner bereitgestellt.

Da der Webserver auch auf die SD-Karte zugreifen muss, wird am Anfang noch die SD-Karte eingebunden. Diese Funktionen werden von dem Betriebssystem von Dr-Dobb's [5] übernommen.

Nach dem Aufspielen des Programms auf den Webserver ist dieser in der Lage, eine Webseite, die auf der SD-Karte liegt, im LAN anzuzeigen.

Um diese im Internet anzuzeigen, wird noch ein weiteres „Programm“ benötigt. Im eigentlichem Sinne ist es gar kein Programm sondern ein Dienst, der die dynamische IP-Adresse, die sich mit jeder Einwahl ändert, mit einer Internetadresse, die sich ein Mensch merken kann, verknüpft. Ein Anbieter dieses Dienstes ist Dyndns.com. Er bietet Adressen im Format *.dnsalias.net kostenlos an (vgl. [7]).

Mein Webserver ist unter

<http://gyimiweb.dnsalias.net/oekoserver>

zu erreichen.

Jetzt ist der Webserver vollwertig als solcher einsetzbar und kann fast kostenlos eine Webseite anzeigen.

4. Ausblick

Der Webserver kann zurzeit nur statische HTML-Dateien anzeigen. Dies hat den Nachteil, dass die Funktionen nur dann ausgerührt werden können, wenn sie von NBEclipse kompiliert wurden und direkt in das Betriebssystem des Webserver eingebunden werden. Da der Webserver in der Lage sein sollte, dass auch Webseiten auf der SD-Karte Funktionen ausführen können, sollte er auch ASP- oder PHP-Dateien unterstützen. Dies ist notwendig, da die Geräte von einer Webseite auf der Speicherkarte an- und ausgeschaltet werden sollen. Dafür werde ich das Betriebssystem von Dr.Dobb's erweitern.

Eine weitere Idee wäre, den Webserver in eine Domäne fahren zu können, um auf die Benutzer zugreifen zu können, um zum Beispiel Dateien vom Schulaccount abzurufen.

Eine andere Möglichkeit wäre, ASP und .Net Framework auf ihm zu laufen zu bringen und den Webserver meiner Schule durch ihn zu ersetzen. So könnte auch dort ein bisschen Strom gespart werden.

Man könnte aber den Webserver auch zuhause einsetzen, um eine private Webseite ins Netzzustellen und gleichzeitig über eine zweite, passwortgeschützt Seite Verbraucher im Haus an- und ausschalten, die Rollläden steuern oder das Licht schalten. Vielleicht ist sogar eine Kopplung mit einem Bussystem möglich, um Licht, Rollläden, Steckdosen, usw. zu steuern.

5. Danksagungen

Ich danke dem Schulverein des Gymnasiums Michelstadt für die Übernahme der Anschaffungskosten und der Schule selbst für die Bereitstellung von Material.

Auch danke ich meinem Betreuer Herrn Dr. Untergasser, der mich bei Softwarefragen unterstützt hat.

6. Literatur

1. Ulrich Radig: „ETH_M32_EX“ und „AVR Webmodul“, <http://www.ulrichradig.de>, 16.11.09
2. NetBurner: „MOD5270“, http://www.netburner.com/products/core_modules/mod5270.html, 16.11.2009
3. NetBurner: „MOD5270 Datasheet“, http://www.netburner.com/downloads/mod5270/mod5270_datasheet_pinout_diagram.pdf#page=1&view=Fit, 18.11.2009
4. NetBurner: „Embedded Network Development Kit“, http://www.netburner.com/products/development_kits/network_development.html, 18.11.2009
5. Dr.Dobb's: „Building your own web server“, <http://www.ddj.com/embedded/211300170>, 16.11.2009
6. NetBurner: NBEclipse Demoprogramm
7. Dyndns: <http://www.dyndns.com/>, 22.11.09