



AVR Web Server

Version 1.1

erstellt von

Ulrich Radig

Vorwort:

Der von mir aufgebaute AVR Webserver dient zum Steuern Messen und Regeln von diversen Gerätschaften genannt seien Garagentor, Lampen, Heizung und andere Dinge die sich in einer Wohnung finden lassen. Obwohl einige Projekte schon existieren entschloss ich mich im Januar 2004 einen eigenen Webserver aufzubauen und zu programmieren. Im Gegensatz zu den anderen Webservern auf 8 Bit μ Controller Basis sollte dieser ohne externen Speicher (SRAM) auskommen, dieses sichert den einfachen Nachbau und einen geringen Preis. Benötigt werden für dieses Projekt ein RISC μ C von Atmel Type ATmega 103/128 und eine Netzwerkkarte RTL8019 von Realtek. Die serielle Schnittstelle ist optional und wird nicht benötigt allerdings sei darauf hingewiesen dass diese das Finden und Aufspüren von Fehlern sehr vereinfacht. Der ATmega 103/128 hat 4KByte internes SRAM und 128KByte Flash welches auch benötigt wird. Einigen Personen ist es auch noch gelungen mit einigen kleinen Änderungen den Source Code auf einen ATmega32 laufen zu bringen, der nur 2KByte internes SRAM besitzt allerdings bietet dieser dann nicht mehr genügend Speicher für spätere Erweiterungen, wie Anbindung einer MMC/SD Karte mit Fat16. Der Webserver beinhaltet zur Zeit die Protokolle ARP, ICMP, IP, TCP und einen kleinen HTTP Server sowie rudimentäre Funktionen für Telnet. Für die Zukunft ist noch DHCP, SMTP und FTP geplant. Als Programmiersprache entschloss ich mich für C, um auch später andere Typen von μ C zu benutzen. Zum Einsatz kam der freie AVR GCC Compiler den es für Linux sowohl als auch für Windows gibt. Die Webseiten werden mit dem Programmcode im internen Flash gespeichert. Es ist auch ohne weiteres möglich mit etwas Programmiererfahrung das interne oder auch ein externes EEPROM zu benutzen. Welches sich schon auf meinen Experimentierbord für ATmega 103/128 μ C befindet.

Erste Schritte:

Nachdem eine Netzwerkkarte mit dem μ C verbunden wurde, erfolgt das Programmieren des Code in den Flash Speicher. Dazu muss aber vorher der Code an den μ C angepasst und danach neu kompiliert werden.

Der μ C Type wird im Makefile eingetragen.

```
# MCU name
```

```
MCU = atmega32    ← oder atmega103/128
```

Die Frequenz des Quarzes sowie die Baudrate in main.c

```
//Der Quarz auf dem Experimentierboard hat eine Frequenz von 14.318MHz
```

```
#define sysclk 14318000    //Quarz Frequenz in Hz
```

```
//Die Baud_Rate der Seriellen Schnittstelle ist 9600 Baud
```

```
#define baud_rate 9600    //Baud Rate für die serielle Schnittstelle
```

des Weiteren findet man in main.c die MAC und IP Adresse

```
//MAC Adresse des Web Servers
```

```
const unsigned char mymac[] = {30,10,43,43,45,77};
```

```
//IP Adresse des Web Servers
```

```
const unsigned char MYIP[] = {192,168,0,101};
```

Nach dem der Webserver mit Spannung versorgt wurde sollte nun am Ausgang der Seriellen Schnittstelle via Hyperterminal folgendes erscheinen:

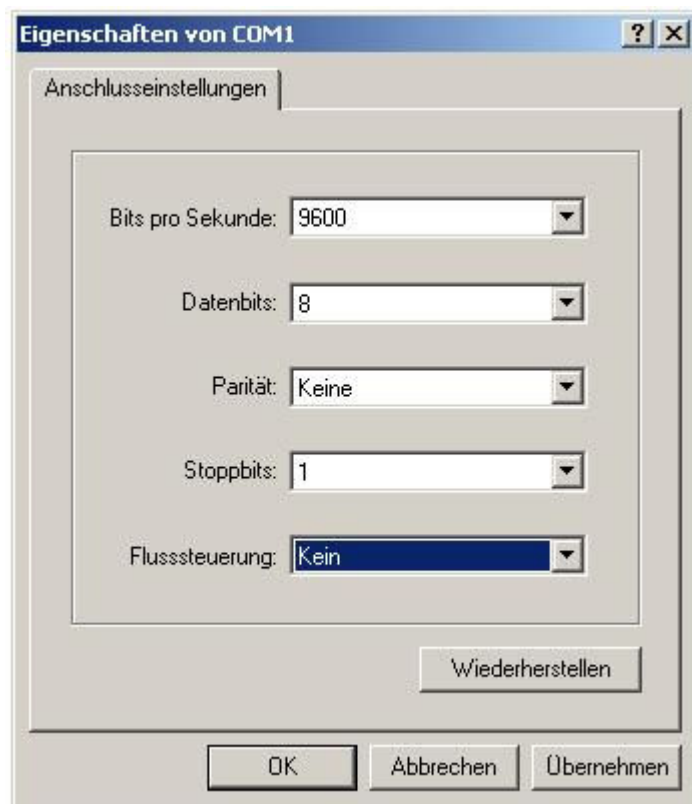
Init Network Card: Init Ready!

Card Init: 34

My Mac: 1e.a.2b.2b.2d.4d

My IP : 192.168.0.101

Es sollten folgende Einstellungen für Hyperterminal vorgenommen werden:



Wenn nicht 34 steht wurde die Netzwerkkarte nicht richtig erkannt, das kann daran liegen das die Netzwerkkarte nicht richtig konfiguriert wurde, bzw. ein Fehler in der Verdrahtung vorliegt. Bei einen Mega103/128 wird der Interrupt4 verwendet. Bei einen Maga32 wird der Interrupt0 benutzt, sowie sollte das Fuse Bits CKOPT richtig programmiert sein (External Quarz 2 OFF). Der erste Test des AVR Webservers erfolgt nach richtiger Einschaltmeldung noch am DOS – Prompt (unter DOS oder im DOS-Fenster). Hier wird versucht den Webserver „anzupingen“. Dazu gibt man folgendes ein:

```
ping 192.168.0.101
```

Sollte dieser Test erfolgreich verlaufen, kann man nun den Internet Explorer öffnen und folgende Seite aufrufen.

<http://192.168.0.101>

Danach sollte die Webseite vom Webserver erscheinen.

Benötige ich wirklich eine Realtek8019 Netzwerkkarte?

Der Source Code spricht nur NE2000 kompatible Register an, somit sollte dieser mit allen gängigen ISA Netzwerkkarten laufen. Folgende Netzwerkkarten habe ich bisher getestet.

- Netzwerkkarten mit dem Realtek Chipsatz 9019 verschiedener Hersteller
- Accton Netzwerkkarten mit dem MPX2 Chipsatz z.B. Accton 166x
- Netzwerkkarten mit einem UM9008 Chipsatz

Wie muß ich die Netzwerkkarte einstellen?

Die Netzwerkkarte muß auf IRQ 9 und auf die I/O Adresse 300 sowie RJ45 eingestellt werden. Dieses kann auf zwei Arten geschehen, erstens mit Hilfe eines PCs und für die Netzwerkkarte zugehörige Programm oder man entfernt das EEPROM von der Netzwerkkarte und diese erhält automatisch die benötigten default Werte (IRQ9 I/O 300).

Gibt es auch ein Netzwerkkarten Modul?

Es gibt auch ein Netzwerkkarten Modul mit RTL8019 Chipsatz welches nicht konfiguriert werden braucht.

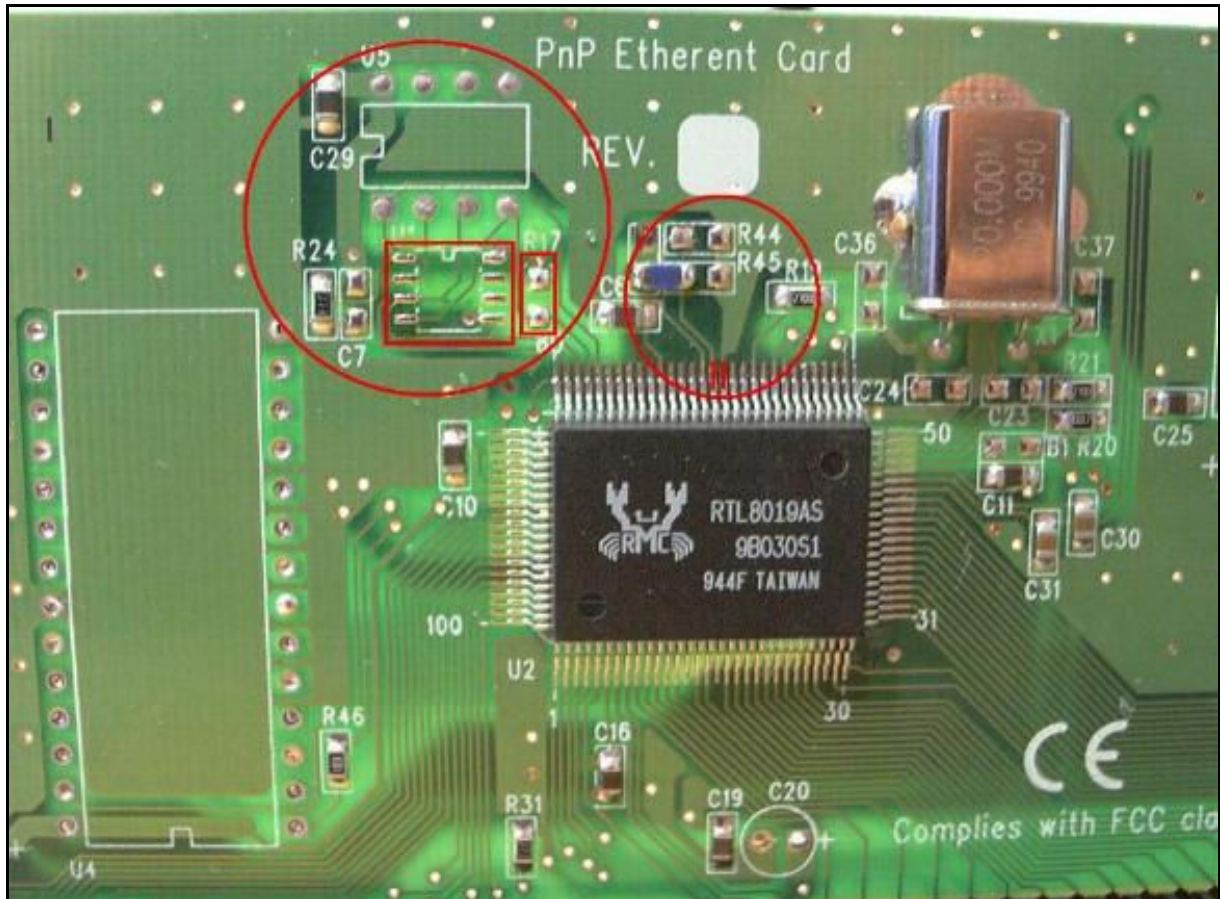
Anschluß der Netzwerkkarte an das Atmel Experimentierboard

Ansicht auf die Platine von Oben

Slotblech						
	Signal	Pin Nr.	ISA-Bus	Pin Nr.	Signal	S
GND	GND	B1	■.....■	A1	/IO CH CK	NC
PORT_C.7	RES DRV	B2	■.....■	A2	D7	PORT_A.7
+5V(VCC)	+5V	B3	■.....■	A3	D6	PORT_A.6
INT4(PORTE.4)	IRQ9	B4	■.....■	A4	D5	PORT_A.5
NC	-5V	B5	■.....■	A5	D4	PORT_A.4
NC	DRQ2	B6	■.....■	A6	D3	PORT_A.3
NC	-12V	B7	■.....■	A7	D2	PORT_A.2
NC	/OWS	B8	■.....■	A8	D1	PORT_A.1
NC	+12V	B9	■.....■	A9	D0	PORT_A.0
GND	GND	B10	■.....■	A10	IO CH RDY	NC
+5V(VCC)	/SMEMW	B11	■.....■	A11	AEN	GND
+5V(VCC)	/SMEMR	B12	■.....■	A12	A19	GND
PORT_C.6	/IOW	B13	■.....■	A13	A18	GND
PORT_C.5	/IOR	B14	■.....■	A14	A17	GND
NC	/DACK3	B15	■.....■	A15	A16	GND
NC	DRQ3	B16	■.....■	A16	A15	GND
NC	/DACK1	B17	■.....■	A17	A14	GND
NC	DRQ1	B18	■.....■	A18	A13	GND
NC	/REFRESH	B19	■.....■	A19	A12	GND
NC	BCLK	B20	■.....■	A20	A11	GND
NC	IRQ7	B21	■.....■	A21	A10	GND
NC	IRQ6	B22	■.....■	A22	A9	+5V(VCC)
NC	IRQ5	B23	■.....■	A23	A8	+5V(VCC)
NC	IRQ4	B24	■.....■	A24	A7	GND
NC	IRQ3	B25	■.....■	A25	A6	GND
NC	/DACK2	B26	■.....■	A26	A5	GND
NC	T/C	B27	■.....■	A27	A4	PORT_C.4
NC	ALE	B28	■.....■	A28	A3	PORT_C.3
+5V(VCC)	+5V	B29	■.....■	A29	A2	PORT_C.2
NC	OSC	B30	■.....■	A30	A1	PORT_C.1
GND	GND	B31	■.....■	A31	A0	PORT_C.0
NC	/MEM CS 16	D1	■.....■	C1	/SBHE	NC
NC	/I - O CS 16	D2	■.....■	C2	LA23	NC
NC	IRQ10	D3	■.....■	C3	LA22	NC
NC	IRQ11	D4	■.....■	C4	LA21	NC
NC	IRQ12	D5	■.....■	C5	LA20	NC
NC	IRQ15	D6	■.....■	C6	LA19	NC
NC	IRQ14	D7	■.....■	C7	LA18	NC
NC	/DACK0	D8	■.....■	C8	LA17	NC
NC	DRQ0	D9	■.....■	C9	/MEMR	NC
NC	/DACK5	D10	■.....■	C10	/MEMW	NC
NC	DRQ5	D11	■.....■	C11	SD8	NC
NC	/DACK6	D12	■.....■	C12	SD9	NC
NC	DRQ6	D13	■.....■	C13	SD10	NC
NC	/DACK7	D14	■.....■	C14	SD11	NC
NC	DRQ7	D15	■.....■	C15	SD12	NC
+5V(VCC)	+5V	D16	■.....■	C16	SD13	NC
NC	/MASTER	D17	■.....■	C17	SD14	NC
GND	GND	D18	■.....■	C18	SD15	NC
Lötseite von der Einsteckkarte				Bauteilseite von der Einsteckkarte		

Netzwerkkarte ohne EEPROM

Hier die Änderung die man an der Netzwerkkarte vornehmen kann um sich die Konfiguration mit dem PC zu sparen (Nur Realtek 8019 Netzwerkkarten)



Zum betreiben der Netzwerkkarte an dem AVR Experimentierboard muss diese noch vorerst modifiziert werden.

1. Auslöten des EEPROMs sowie alle damit verbundenen PULL UP/DOWN Widerstände.
2. Anschluss PIN64 verbinden nach GND (Wenn nicht vom Hersteller schon geschehen).
3. Anschluss PIN65 verbinden nach VCC.