

Webserver zur Temperaturmessung

1 Inhalt

1	Inhalt	1
2	Überblick	2
2.1	Schnellstart.....	2
2.2	Hardware.....	2
2.3	Hinweise zum Übersetzen der Webserversoftware.....	2
2.4	Bekannte Probleme	3
2.4.1	TCP-Stack.....	3
2.4.2	Ausgabe von Daten aus Variablen.....	3
2.4.3	IP-Adresse wird an zwei Stellen geführt.....	3
2.4.4	Sendepufferüberlauf.....	3
2.4.5	Uhr geht nach.....	3
2.4.6	NIC Adressierung beschreibt Bit 6 und 7 von Port A	3
2.5	Optimierung des Speicherbedarfs	3
3	Leistungsmerkmale des Systems.....	4
3.1	Ein- und Ausgabe von Daten über HTML-Seiten	4
3.1.1	Ausgabe von Daten aus einer Variablen auf einer HTML-Seite	4
3.1.2	Einlesen von Client-Daten aus einer HTML-Seite in eine Variable.....	4
3.1.3	Die Steuerung von Hardware mittels einer Variablen	4
3.1.4	Zusammenfassung zu Bezeichnungen und Namen	4
3.1.5	Übersicht der verwendeten Platzhalter.....	5
3.1.5.1	Variablen	5
3.1.5.2	Steuermarken	6
3.1.6	Anleitung zur Erweiterung.....	6
3.2	Nutzung des PWM-Betriebs	6
3.3	Auslesen des EEPROM der Netzwerkkarte.....	7
3.4	Ausgabe des AVR-EEPROM-Inhaltes auf der seriellen Schnittstelle.....	7
3.5	Einstellen der IP-Adresse über das Webinterface.....	7
3.6	Ausgabe eines Favourite Icon	8
3.7	Einrichtung einer one-wire Schnittstelle.....	8
3.7.1	Schaltung	8
3.7.2	einsetzbare Temperatursensoren	9
3.7.2.1	Eingabeport mit dem Temperatursensor DS1820	9
3.7.3	Datenverwaltung	9
3.7.3.1	Verwaltung der statischen Daten im EEPROM	10
3.7.3.2	Verwaltung der dynamischen Daten im SRAM	11
3.7.4	Ein- und Ausschalten der one-wire Busleitungen	11
3.7.5	Ausgabe der one-wire Daten mit der Webserver.....	11
3.8	Komprimierte Speicherung von HTML-Seiten.....	12
3.9	Temperaturaufzeichnungen	12
3.10	Betrieb einer Webkamera	12
3.10.1	Anschaltung der Kamera	13
3.10.2	Betrieb der Kamera	13
3.11	Automatische Einstellung der Uhr.....	13
3.11.1	Der Langwellenempfänger.....	14
3.11.2	Die Verbindung zwischen Empfänger und Dekoder mit einer Stromschleife	14
3.11.2.1	Die Auslegung der Stromschleife	14
3.11.2.2	Weitere Anzeigeräte.....	14
3.11.2.3	Die Versorgung der Stromschleife und die Anschaltung des Microcontrollers	14
3.11.3	Die Dekodierungssoftware.....	15
3.11.3.1	Die Störsicherheit des Empfangs	15
3.11.3.2	Die Prüfung der Daten	16

2 Überblick

Der beschriebene Webserver läuft auf einem ATmega32 Prozessor von Atmel. Als Netzwerkcontroller wird eine Netzwerkkarte mit einem Controller RTL8019 (NE2000- oder DP8390- kompatibel) verwendet. Das Interface des Netzwerkcontrollers zur CPU ist ein ISA-BUS, der aus einem umschaltbaren 8bit/16bit-CPU-Interface besteht und damit von einem Mikrocontroller einfach angesteuert werden kann. Ein PCI-Interface wäre erheblich schwieriger anzusteuern.

Bei der Zusammenstellung des Systems wurden folgende Quellen genutzt:

- Die verwendete Leiterplatte ist bei Holger Buss (www.mikrocontroller.com) erhältlich.
- Die Software basiert auf der Webserversoftware von Ulrich Radig (www.ulrichradig.de).
- Die Software für das one-wire Interface stammt aus dem Applikationsbericht AVR318 der Fa. Atmel (www.atmel.com).
- Die Ansteuerung der Webkamera wurde von Simon Schulz (avr.auctionant.de/avr-ip-webcam) entwickelt.
- Die Funktion des DCF77-Dekoders ist in Funkschau 22/1984, Seite 67 ff. beschrieben.

2.1 Schnellstart

In dem Paket ist die Datei **main.hex** enthalten. Diese Datei ist auf der Webserverhardware von Holger Buss lauffähig. Die Taktfrequenz muß dazu 14,7456 MHz betragen. Das erstmalige Ansprechen des Servers erfolgt über die Adresse 192.168.99.199. Die Adresse kann dann interaktiv geändert und mit einem Reset aktiviert werden.

Die Fuses CKOPT und EESAVE sollten programmiert (d.h. auf „0“ gesetzt) sein.

Der Rest sollte sich spätestens nach dem Lesen dieser Anleitung intuitiv bedienen lassen.

2.2 Hardware

Die Webserverplatine wird gemäß Schaltung ISA_CTRL_V1_0 Version 1.0 aufgebaut. Abweichend vom Bestückungsplan wird R12 mit 10 kΩ bestückt.

Die Erweiterungen können alle über den 14-poligen Pfosten-Steckverbinder S1 angeschlossen werden.

Hier die Belegung des Steckverbinders mit Signalbezeichnungen und Nutzung.

Pin	Signal	Nutzung
1	B4 (SS)	Port B4 wird als schaltbare Ausgangsleitung genutzt.
2	D3 (INT1)	D3 ist ein one-wire-Bus. Die Verwendung als externer Interrupt ist möglich.
3	D4 (OC1B)	D4 ist ein one-wire-Bus. (OC1B nicht nutzbar, Timer 1 wird für die Uhr benötigt)
4	D5 (OC1A)	D5 ist ein one-wire-Bus. (OC1A nicht nutzbar, Timer 1 wird für die Uhr benötigt)
5	D6 (ICP)	D6 wird für das Reset-Signal der Kamera verwendet.
6	D7 (OC2)	OC2 ist als PWM-Ausgang geschaltet und treibt die rote LED auf der Leiterplatte.
7	A5 (Analogeingang)	A5 wird als schaltbare Ausgangsleitung genutzt
8	A6 (Analogeingang)	A6 wird als schaltbare Ausgangsleitung genutzt
9	A7 (Analogeingang)	A7 wird als Analogeingang geschaltet und als DCF77-Signaleingang genutzt
10	+5V	stabilisierte Betriebsspannung des Reglers auf der Webserver-Leiterkarte
11	D1 (TXD)	TXD-Signal der seriellen Schnittstelle (für Kamera- und Systemmonitorzwecke)
12	+12V	unstabilisierte Spannung des Servers hinter der Verpolungsschutzdiode
13	D0 (RXD)	RXD-Signal der seriellen Schnittstelle (für Kamera- und Systemmonitorzwecke)
14	GND	Ground - Masse

2.3 Hinweise zum Übersetzen der Webserversoftware

Vor dem Übersetzen muß eventuell die Konfiguration angepaßt werden. Die Fundstellen einiger Definitionen und wichtiger Einstellungen sind nachstehend aufgeführt.

Bezeichnung	Datei	Voreinstellung	Beschreibung
Prozessortyp	makefile	ATmega 32	MCU auf Prozessortyp setzen
IP-Adresse	main.c	192.168.99.199	IP-Adresse nach Reset bei leerem EEPROM festlegen
Systemtakt	main.h	14.745.600 Hz	SYSCLK auf die Quarzfrequenz setzen, die bei Webcambetrieb 14,7456 MHz betragen sollte. (auch 1/4, 1/2 oder 3/4 davon möglich).
Hardwareplattform	rtl8019.h	gesetzt	ISP_CTRL setzen, wenn die Hardware von Holger Buss verwendet wird. Vorsicht: In einigen Quellen wird eine Marke mit gleicher Funktion abweichend ISA_CTRL genannt.
Baudrate der seriellen Schnittstelle	uart.h	9600	Baudrate nach Wunsch ändern. Achtung: Bei Webcambetrieb muß 9600 bd gewählt werden. Die Webcam-Routinen schalten dann auf 460800 bd um.

SD-Card Nutzung	mmc.h	nicht gesetzt (keine SD-Card)	USEMMC auf 0 setzen, wenn keine SD-Card genutzt wird
TELNET Nutzung	telnet.h	0	USE_TELNET auf 0 setzen, wenn TELNET nicht genutzt wird
FTP Nutzung	ftpd.h	0	USE_FTPD auf 0 setzen, wenn FTP nicht genutzt wird
MAX_DEVICES	main.h	17	Anzahl der verwaltbaren Dallas-Bausteine + 1
erlaubte Leitungen	main.h	Pin 3, 4 und 5	mögliche one-wire Busleitungen an Port D

Im **makefile** wurde zusätzlich der Menüpunkt „ISP“ eingerichtet. Durch Anwählen dieses Schalters wird das Programm Ponyprog gestartet, die aktuelle Datei **main.hex** geladen und die EEPROM-Daten des AVR ausgelesen. Anschließend wird der AVR programmiert. Damit die im EEPROM vorhandenen Einstellungen bei Neuprogrammierung nicht verloren gehen, wird die Fuse EESAVE programmiert.

2.4 Bekannte Probleme

2.4.1 TCP-Stack

Der TCP-Stack ist nicht vollständig implementiert. Bei fehlerhaften Verbindungen findet keine automatische Paketwiederholung statt. Das sollte bei kabelgebundenen Netzwerkverbindungen unproblematisch sein.

2.4.2 Ausgabe von Daten aus Variablen

HTML-Seiten, die auf einer SD-Card abgelegt wurden, dürfen keine dynamischen Elemente (gemäß 3.1) enthalten, da die Daten von der SD-Card unter Umgehung des Parsers direkt in den Netzwerkpuffer kopiert werden.

Bei der Verlagerung von HTML-Seiten in die SD-Card ist dies zu berücksichtigen.

2.4.3 IP-Adresse wird an zwei Stellen geführt

In der Seite Index.htm der Originalversion 1.38 (zu finden in webpage.c) wird die IP-Adresse der Reload-Seite im HTML-Text genannt. Die Angabe ist nicht erforderlich, muß aber bei Wechsel der IP-Adresse geändert werden. (Dieser Punkt ist in der hier beschriebenen Version nicht mehr relevant)

2.4.4 Sendepufferüberlauf

In der Originalversion 1.38 kann der Sendepuffer überlaufen, weil die Puffergrenze beim Ersetzen von Platzhaltern nur einmal geprüft wird. Wenn ein Platzhalter durch mehrere Zeichen ersetzt wurde, konnte über das Pufferende hinaus geschrieben werden. Zur Lösung wird der Puffer um 80 Zeichen verlängert. Damit lassen sich Strings mit einer Länge von bis zu 80 Zeichen in den Buffer schreiben. Das System läuft damit stabil.

2.4.5 Uhr geht nach

Beim Kamerabetrieb kann der Uhreninterrupt nicht zeitnah bearbeitet werden, weil die Operationen zur Bildausgabe mehrere Sekunden dauern und im NIC-Interruptprogramm abgewickelt werden. Bei einer Bildübertragung gehen so regelmäßig 2 s verloren. Um dieses Problem zu beheben, wurde die Uhr auf Pollingbetrieb umgestellt. Der 16-bit-Zähler „Counter 1“ wird dazu freilaufend betrieben und gelegentlich mit dem Sollwert des nächsten Sekundenbeginns verglichen. Das Problem ist damit beseitigt.

2.4.6 NIC Adressierung beschreibt Bit 6 und 7 von Port A

Der Network Interface Controller benötigt 5 Adressbits, die verschaltet sind. Die Webserversoftware addiert zur Adresse die Konstante „96“ und schreibt dann 7 Adressbits auf den Ausgangsport. Das Problem ist beseitigt.

2.5 Optimierung des Speicherbedarfs

Der Speicher im ATmega 32 teilt sich auf in 32 kByte Flash, 2 kByte RAM und 1 kByte EEPROM. Der gcc-Compiler gibt nach dem Übersetzen für die Speichernutzung folgende Werte aus:

Bereich	Beschreibung
.text	Programmcode im Flash
.data	Variable im RAM, die initialisiert werden müssen. Die Initialisierungsdaten liegen im Flash.
.bss	Variable im RAM, die nicht initialisiert werden müssen

Die Belegung des Flash errechnet sich zu `.text + .data`, weil die Initialisierungsdaten für die Variablen mit dem Umfang „data“ hinter dem Programmcode „txt“ abgelegt sind.
Der belegte RAM-Bereich berechnet sich zu `.data + .bss`.

In der Funktion `memcmp (str1,str2,n)` werden die ersten `n` Stellen des Strings `str2` mit `str1` verglichen. Zu Beginn des Programms wird `str2` im Zuge der Initialisierung im RAM abgelegt und dort für den Vergleich bereitgehalten. Die Funktion `strcmp (str1,PSTR(str2),n)` leistet den selben Vergleich mit dem Unterschied, dass `str1` mit dem im Flash liegenden `str2` verglichen wird. Damit wird weniger RAM benötigt. Im Flash verhält sich die Änderung neutral, weil die Daten im Bereich `.text` abgelegt werden und im Bereich `.data` entfallen.

3 Leistungsmerkmale des Systems

3.1 Ein- und Ausgabe von Daten über HTML-Seiten

Die Ausgabe einer Variablen als fünfstellige Dezimalzahl ist bereits im Quelltext der Version 1.38 vorgesehen. Die Ausgabe von Variablen wurde um weitere Varianten ergänzt und es wurde eine Eingabe von Daten über die Webschnittstelle hinzugefügt.

3.1.1 Ausgabe von Daten aus einer Variablen auf einer HTML-Seite

Die Variable muß, wie in C üblich, deklariert und definiert werden. Zusätzlich muß ein Platzhalter für die HTML-Seite eingerichtet werden. Die Platzhalter wurden nach dem Muster `%Vx` angelegt.
In **webpage.c** wird der Platzhalter anstelle des auszugebenden Datums in die HTML-Seite aufgenommen.
In **tcp.c** befindet sich der Parser für die HTML-Seite. Hier muß ein Programm bzw. ein Programmaufruf zur Interpretation des Platzhalters der Variablen angelegt werden. In **tcp.c** wird der HTML-Text Byte für Byte in den Sendepuffer übertragen und dabei nach Platzhaltern gescannt. Anstelle der Platzhalter werden die Daten aus den zugehörigen Variablen ausgegeben. Es sind verschiedene Beispiele realisiert.

Bei binären Variablen (Schalter mit zwei Stellungen), die über einen „radio-button“ geschaltet werden, muß im HTML-Text für die Voreinstellung des Schalters das Wort „checked“ ausgegeben werden können.
Bei Bedarf können für einen Platzhalter auch Datenfelder ausgegeben werden, z.B. die IP-Adresse in der Form `a.b.c.d`, die MAC-Adresse in der Form `u:v:w:x:y:z` oder eine Temperatur `xx.yy°C`.

3.1.2 Einlesen von Client-Daten aus einer HTML-Seite in eine Variable

Zum Einlesen von Daten, die der Client (Browser) an den Server schickt, muß die HTML-Seite in **webpage.c** einen „<form method=get>“ Block enthalten, der ein Formularelement für die Variable und zur Ausgabe der Voreinstellung den Platzhalter der Variablen enthält. Sinnvollerweise wählt man das Formularelement der Variablen in Anlehnung an den Namen des Platzhalters. Als Formularelemente sind Textfelder, Radio-buttons und select-Elemente sinnvoll einsetzbar.

Bei Checkboxen ist als Besonderheit zu beachten, daß der Inhalt nur übertragen wird, wenn die Checkbox markiert ist. Das Löschen der Markierung einer Checkbox wird nicht zum Server übertragen! Bei der Auswertung im PC ist deshalb eine Umgehungslösung anzuwenden. Als Lösung wurde ein „hidden“-Element eingeführt, das im HTML-Text vor den Checkboxen angeordnet ist und alle Checkboxen vor dem erneuten Setzen löscht.

In **httpd.c** wird der GET-request ausgewertet. Hier wird die URL-Adresszeile aus der Seitenanforderung des Client auf die möglichen Variablen untersucht.

3.1.3 Die Steuerung von Hardware mittels einer Variablen

Die Steuerung von Hardware durch den Inhalt einer Variablen erfolgt nach dem Verändern der Variablen in **httpd.c**. Es ist bei AVR-Prozessoren möglich, den Wert einer Ausgabeleitung in der Hardware des Prozessors zu speichern, in dem der Pegel der Ausgabeleitung bei Bedarf eingelesen wird. Hier wurde die universellere Softwaremethode gewählt, bei der alle Informationen in Variablen gespeichert werden.

3.1.4 Zusammenfassung zu Bezeichnungen und Namen

Für ein dynamisches Element sind folgende drei Bestandteile erforderlich:

a. Variable im C-Quelltext.

Die Variable ist Bestandteil des Mikrocontrollerprogramms und speichert die Information.
z.B. bei der Variablen Minutenzähler: `unsigned char mm`

b. Platzhalter im HTML-Text.

Der Platzhalter steht in **webpage.c**. Bei der Ausgabe einer zu sendenden Datei in **tcp.c** wird anstelle des Platzhalters der Inhalt der Variablen ausgegeben.
z.B. wird der Platzhalter `%VAm` durch den Inhalt der Variablen Minutenzähler ersetzt.

c. Formularelement im HTML-Text.

Dem Formularelement wird über den Browser ein „value“ zugewiesen und beides wird mit dem Get-request an den Server übertragen. Der Server durchsucht die Antwort des Client auf das Vorkommen des Formularelementes, ermittelt den Wert und setzt die Variable entsprechend.
z.B. wird bei der Variablen Minutenzähler das Formularelement name=„VAm“ gesetzt.

Alle drei Bezeichnungen sind prinzipiell unabhängig voneinander. Eine gleiche oder ähnliche Bezeichnung vereinfacht jedoch die Handhabung und senkt die Fehlerwahrscheinlichkeit.

3.1.5 Übersicht der verwendeten Platzhalter

3.1.5.1 Variablen

Platzhalter	Ausgabeformat	Eingabe	Funktion
%VA0	-	0	temporäre Variable
%VA1	5-stellige Dezimalzahl	-	Seitenzähler
%VA2 (+/-)	Zahl oder „checked“	2-stellige Dezimalzahl	
%VA3 (n)	„selected“, wenn VA3=n	2-stellige Dezimalzahl	
%VA4 (+/-)	Bit-Test, „checked“	0 oder 1	
%VA5 (+/-)	Bit-Test, „checked“	0 oder 1	
%VA6 (+/-)	Bit-Test, „checked“	0 oder 1	
%VA9	-	0	temporäre Variable
%VAd(*°)	25,23 °C	-	Temperatur des DS1820
%VAe(*)	#	-	External-PWR-Bit des DS1820
%VAh	2-stellige Dezimalzahl	2-stellige Dezimalzahl	Stunden
%VAi	Dezimalzahl	Dezimalzahl 0..255	IP-Adresse 1. Byte
%VAj	Dezimalzahl	Dezimalzahl 0..255	IP-Adresse 2. Byte
%VAk	Dezimalzahl	Dezimalzahl 0..255	IP-Adresse 3. Byte
%VAL	Dezimalzahl	Dezimalzahl 0..255	IP-Adresse 4. Byte
%VAm	2-stellige Dezimalzahl	2-stellige Dezimalzahl	Minuten
%VAs	2-stellige Dezimalzahl	2-stellige Dezimalzahl	Sekunden
%VAA	Dezimalzahl 0.255	-	Signal des ADC7
%VAB (n)	„selected“, wenn VAB=n	Dezimalzahl 1..3	Schalter für Dump
%VAD(*)	16-stellige Hexzahl	-	Seriennummer des Dallas-Bausteins
%VAE	-	Satznummer	Satz löschen
%VAF	Dezimalzahl	Dezimalzahl 0..8	Funlayer
%VAI	192.168.99.199	-	Komplette IP-Adresse
%VAL	2-3-stellige Dezimalzahl	2-3-stellige Dezimalzahl	Index
%VAM	00:01:02:03:04:CA	-	Komplette MAC-Adresse
%VAN(* /)	„String“	„String“ mit Token	Ein/Ausgabe eines Bausteinnamens
%VAO	0..9 A..Z	-	Index zur Basis 36
%VAP (n)	„selected“, wenn VAP=n	Dezimalzahl 1..3	Schalter für Bildformat
%VAR (* n)	Dezimalzahl	Dezimalzahl 0..255	Index, dezimal, „selected“, wenn VAR=n
%VAS	Dezimalzahl	Dezimalzahl 0..255	Schaltschwelle für den DCF77-Eingang
%VAT(*)	„String“	-	Ausgabe eines Tokens
%VAX	Bit-Test, „checked“	0 oder 1	Port B4
%VAY	Bit-Test, „checked“	0 oder 1	Port A5
%VAZ	Bit-Test, „checked“	0 oder 1	Port A6

Die Platzhalter sind teilweise mit zusätzlichen Optionen ausgestattet.

- (*) Der Platzhalter betrifft ein Datenfeld. Zur Auswahl des Elementes aus diesem Feld muß z. B. auf die Bezeichnung %VAD genau ein Zeichen folgen. Wenn dieses Zeichen ein * ist, wird der Schleifenindex zur Auswahl benutzt. Mit den Zeichen [0..9,A..Z] kann alternativ ein Index im Bereich 0..36 angegeben werden.
 - (*°) Wie bei (*). Wenn das Symbol „°“ folgt wird die Einheit „°C“ angehängt.
 - (*|/)
- Anstelle der Option „*“ ist die Option „/“ möglich. Bei der Option / wird das Feldelement ebenfalls durch den Schleifenindex ausgewählt. Die Ausgabe erfolgt jedoch in einer Tokenschreibweise, die kompatibel zur Dateneingabe im Eingabefeld der Variablen ist. Dieses Format dient zur Vorbesetzung eines Bearbeitungsfeldes.

- (+|-) Auf den Platzhalter muß ein Zeichen + oder – folgen. Abhängig vom Inhalt der Variablen und dem Zeichen + bzw. – wird gemäß nebenstehender Tabelle der Text „selected“ ausgegeben. Bei der Verwendung einer Checkbox wird die Voreinstellung durch das Wort „checked“ vorgegeben. Im Programm ist das mit gleicher Syntax umgesetzt. Wenn das Zeichen + bzw. – fehlt, wird eine Dezimalzahl ausgegeben.

	VAn+	VAn-
Variable wahr	selected	
Variable unwahr		selected

- (n) Wenn der Inhalt der Variablen der Ziffer n entspricht, wird der Text „selected“ ausgegeben. n hat den Wertebereich 0..36, dargestellt als [0..9;A..Z].

3.1.5.2 Steuermarken

%ANFT	Anfang eines Wiederholblockes über alle Token von Index=0 bis Index=MAX_TOKEN
%ENDT	Ende eines Wiederholblockes über alle Token
%ANFN	Anfang eines Wiederholblockes über alle Dallasbausteine von Index=0 bis Index=MAX_... Bestimmte Bausteingruppen können ausgeblendet werden
%ENDN	Ende eines Wiederholblockes über alle Dallasbausteine
%ANFY	Anfang eines Wiederholblockes für einen Speicherdump
%ENDY	Ende eines Wiederholblockes für einen Speicherdump. Mit der Ende-Kennung wird jeweils eine Zeile ausgedruckt.
%END.	Ende einer HTML-Datei

3.1.6 Anleitung zur Erweiterung

Wenn für bestimmte Zwecke eine weitere Variable benötigt wird, muß die Software entsprechend erweitert werden. Da Eingriffe an verschiedenen Stellen erforderlich sind, wird die Vorgehensweise hier an einem Beispiel detailliert dargestellt.

Ziel ist es, den Wert einer Variablen in einer HTML-Seite auszugeben und den Wert über eine HTML-Seite einstellen zu können.

Zuerst wird die Webseite in **webpage.c** erweitert. Zur Ausgabe der Variablen wird der Platzhalter „%VAh“ eingeführt, der im Quelltext dort stehen muß, wo später der Inhalt der Variablen ausgegeben werden soll. Zusätzlich wird das Formularelement „VAh“ eingeführt, das der Browser mit dem vom Benutzer eingegebenen Wert füllt und mit dem „Get-request“ in der Adresszeile an den Server weitergibt.

Im Programm **tcp.c** wird die Webseite Zeichen für Zeichen in den Ausgabepuffer geschrieben. Dabei wird der Platzhalter „%VAh“ durch den Inhalt der Variablen ersetzt. Die Zeichen „%VA“ werden am Anfang der Abfragekette geprüft. Falls diese vorhanden sind, wird das Folgezeichen geprüft. In unserem Beispiel muß auf das „m“ geprüft werden. Wenn das „m“ erkannt wurde, wird der Inhalt der zugehörigen Variablen in den Puffer geschrieben. In der Abfragekette finden sich Beispiele aller realisierten Ausgabevarianten, die alternativ verwendet werden können. Es ist darauf zu achten, daß anstelle des Platzhalters maximal 80 Zeichen eingesetzt werden.

Die eigentliche Variable muß vor der Benutzung deklariert werden. Das kann z.B. in **main.h** analog zu den dort aufgeführten Beispiele erfolgen.

Die Initialisierung der Variablen kann je nach Erfordernis z.B. in **main.c** durch das Zuweisen einer Konstanten, oder aber in dem Applikationsunterprogramm, wie z.B. der Uhr, erfolgen.

In **httpd.c** werden die Daten aus dem GET-request ausgewertet. Es wird geprüft, ob das Formularelement „VAh“ vorhanden ist und welchen Wert das Formularelement hat. Die zugeordnete Variable wird an dieser Stelle mit dem Datum gesetzt, das im Browserfenster eingegeben wurde. Wenn durch das Datum Aktionen erforderlich werden, dann können diese nach dem Abfrageblock ausgelöst werden. Es ist jedoch zu beachten, das die Aktionen für alle Variablen bei jedem Get-request ausgeführt werden. Andernfalls können erforderliche Aktionen vom Applikationsunterprogramm ausgeführt werden. Unter Aktionen wird hier z.B. das Einstellen der LED-Helligkeit, oder das Setzen und Löschen von Variablen verstanden.

Damit sind alle erforderlichen Schritte erledigt.

3.2 Nutzung des PWM-Betriebs

Die rote Leuchtdiode auf der Leiterplatte ist an Leitung 7 von Port D angeschlossen. Diese Leitung kann auch als PWM-Ausgang programmiert werden. Zur Erprobung wurde der 8-bit-Counter 2 in den PWM-

Betrieb geschaltet. Die Helligkeit der LED läßt sich nun durch Schreiben in das Timerregister von 0 bis 255 einstellen. Die gewählte PWM-Frequenz ergibt sich bei einer Taktfrequenz von 16 MHz zu 980 Hz. Beim Kamerabetrieb wird das Statussignal mit der LED im Form einer Helligkeitsänderung wiedergegeben. Wenn ein DCF-Signal empfangen wird, wird die LED zu Beginn einer Sekunde für 150 ms ausgetastet.

Bei der Option des Servobetriebes wird Port D Leitung 7 als Impulsleitung für einen Modellbauservo zur Drehung der Kamera benutzt. Modellbauservos haben drei Anschlüsse. Neben der Versorgungsspannung gibt es einen Pulseingang zur Steuerung. Die Pulsdauer legt fest in welcher Stellung der Servo verharrt. Bei einer Pulsdauer von 1,0 ms befindet sich der Servo am Linksanschlag, bei 2,0 ms am Rechtsanschlag. Der Puls wird im Modellbaubetrieb alle 20 ms ausgelöst. Bei Funkfernsteuerungen werden in einem Rahmen von 20ms so neben der Rahmensynchronisation 8 Servosignale übertragen. An komfortabelsten ist die PWM-Steuerung mit einem freilaufenden Oszillator zu realisieren, der nur bei Bedarf eingestellt werden muß.

Der im ATmega 32 vorhandene 8 bit Counter kann mit Vorteilereinstellungen von 1, 8, 256 und 1024 betrieben werden. Bei 14,7456 MHz Takt und einer Vorteilereinstellung von 1024 ergibt sich für einen PWM-Zyklus eine Zeit von 18ms. In einer Millisekunde werden dabei 14,4 Schritte durchlaufen, die der Servo zwischen Links- und Rechtsanschlag wiedergibt. Jeder Einstellschritt hat bei Nutzung eines Standardservos mit 60° Ausschlag einen Stellwinkel von 4,2°.

Frequenz	Vorteiler	Zyklusdauer in ms	Schritte pro ms	Grad pro Schritt	Vorteiler	Zyklusdauer in ms	Schritte pro ms	Grad pro Schritt
3.686.400	1024	71,1	3,6	16,7	256	17,8	14,4	4,2
7.372.800	1024	35,6	7,2	8,3	256	8,9	28,8	2,1
11.059.200	1024	23,7	10,8	5,6	256	5,9	43,2	1,4
14.745.600	1024	17,8	14,4	4,2	256	4,4	57,6	1,0

Wegen der groben Stufung der Vorteilereinstellungen ist es nicht möglich, bei einer Zählerlänge von 8 Bit einen 2 ms Puls mit höherer Auflösung, d.h. einer höheren Zahl von Schritten pro ms darzustellen. Eine genauere Einstellung wäre nur durch Nutzung des 16 Bit PWM-Zählers möglich, der jedoch vom Uhrenprogramm belegt wird.

Frequenz	Vorteiler	Zyklusdauer in ms	Schritte pro ms	Grad pro Schritt
3.686.400	8	142,2	460,8	0,130
7.372.800	8	71,1	921,6	0,065
11.059.200	8	47,4	1382,4	0,043
14.745.600	8	35,6	1843,2	0,033

3.3 Auslesen des EEPROM der Netzwerkkarte

Das serielle EEPROM auf der Netzwerkkarte enthält die vom Kartenhersteller festgelegte eindeutige MAC-Adresse, die Daten zur Grundinitialisierung der Netzwerkkarte und Daten zur PnP-Funktion. Beim Initialisieren der Netzwerkkarte wird der Inhalt des EEPROM ausgelesen und in Form eines Speicherdumps auf der seriellen Schnittstelle ausgegeben. Die MAC-Adresse aus dem EEPROM wird in das Feld MYMAC[] geladen und vom Webserver benutzt.

Damit ist sichergestellt, daß mehrere Webserver im gleichen Netz betrieben werden können, ohne daß es wegen gleichlautender MAC-Adressen zu Kommunikationsproblemen kommt.

3.4 Ausgabe des AVR-EEPROM Inhaltes auf der seriellen Schnittstelle

Der Inhalt des EEPROMs wird als Dump auf der seriellen Schnittstelle ausgegeben. Die ersten 7 Byte werden von den SD-Card-Routinen benutzt. Weitere 5 Byte sind mit der IP-Adresse belegt. Ab Adresse 0010h wird der Speicher in 16 Byte-Blöcken mit den Thermometerdaten belegt. Das Dump-Programm wird für den Dump des AVR-EEPROM und des seriellen EEPROM der Netzwerkkarte benutzt. Die Unterprogramme zum Lesen der Daten unterscheiden sich und werden dem DUMP-Programm als Parameter übergeben.

3.5 Ausgabe des AVR-EEPROM oder AVR-RAM über den Browser

Die Unterprogramme für den Speicherdump wurden universell ausgelegt. Im Webbrowser kann eingestellt werden, ob der RAM-Inhalt oder der EEPROM-Inhalt ausgegeben werden soll.

3.6 Einstellen der IP-Adresse über das Webinterface

Durch diese Möglichkeit können mehrere Server mit dem gleichen Softwareimage geladen werden und ohne Konflikte im selben Netz zusammenarbeiten.

Im Quelltext wird eine IP-Adresse vorgegeben (z.B. 192.168.99.199). Unter dieser Adresse wird der Server nach der ersten Inbetriebnahme angesprochen. Über die Webschnittstelle läßt sich dann eine beliebige andere Adresse einstellen. Diese Adresse wird zusammen mit den Einstellungen für das Meßintervall, die

aktiven Busleitungen und die Anzeigeoptionen mit einer Prüfsumme gesichert und im EEPROM abgelegt. Nach einem Reset werden die Daten aus dem EEPROM geprüft und bei Gültigkeit benutzt.

Eine aufwendigere Alternative wäre die Einrichtung einer zweiten einstellbaren IP-Adresse für den Webserver. Der Server wäre dann über zwei verschiedene Adressen erreichbar.

Der Lösungsansatz, die IP-Adresse mit einem DHCP-Client oder dem RARP-Protokoll zu beziehen, hat den Nachteil, dass der Server schwieriger zu finden ist, weil die zugeteilte IP-Adresse nicht bekannt ist. Hier wäre weiterer Aufwand erforderlich (z.B. mit einem dynamischen DNS, dem manuellen Ermitteln der IP-Adresse durch ein an die Broadcastadresse gerichtetes Ping-Signal oder durch die Anmeldung des Servers in einem Logfile).

3.7 Ausgabe eines Favourite Icon

Wenn der Browser das Bild favorite.ico anfordert, erhält er vom Webserver ein Icon mit der Typangabe „text/plain“. Dieses Icon wird dann anstelle des Microsoft „e“-Explorerlogos für die Web-Adresse benutzt. Wenn die Webadresse als Favorit gespeichert wird, wird das Icon von Browser angefordert und gespeichert. Die Quittungsausgabe für den Get-request wurde dazu ergänzt. Nun stehen die Typen „text/html“, „image/jpg“ und „text/plain“ zur Verfügung.

3.8 Einrichtung einer one-wire Schnittstelle

Die one-wire Schnittstelle (bzw. der one-wire Bus) besteht aus einer Signalleitung, die im one-wire System auch mit DQ bezeichnet wird und einer Masseleitung (GND).

Es gibt im Netz verschiedenste Realisierungen von one-wire Schnittstellen für AVR-Prozessoren. Für diese Anwendung wurde der Applikationsbericht AVR318 von Atmel herangezogen. Einige Funktionen wurden an den GNU-GCC-Compiler angepaßt. Die Funktion des aktiven Pullup wurde ergänzt. Die Namen der Dateien zur one-wire Schnittstelle beginnen mit „OWI“ und sind im Order „QWI“ des Quelltextes abgelegt.

Die Buslänge kann, abhängig von der Ansteuerelektronik und der Leitungskapazität, bis zu 500m betragen. An einer Busleitung können über 100 Bausteine angeschlossen werden. Da der Temperatursensor während der Temperaturmessung einen Strombedarf von etwa 1mA hat, wird bei der gleichzeitigen Temperaturwandlung mehrerer Bausteinen auf dem selben Bus die Bausteinanzahl durch die Stromversorgung und das Schaltverhalten der Ausgangsstufe begrenzt.

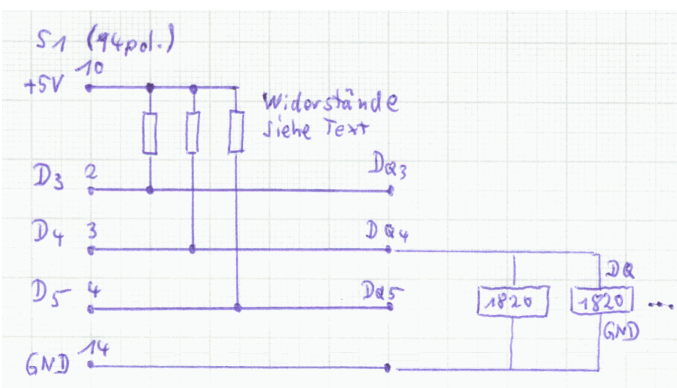
Um diese Begrenzung zu umgehen wurden drei Busleitungen D_{Q3}, D_{Q4} und D_{Q5} vorgesehen, die sich über das Webinterface einzeln ein- und ausschalten lassen. Auf diese Weise kann die Leitungsanzahl erhöht werden, sofern alle Leitungen auf dem selben I/O-Port liegen (hier Port D). Für Details zur Busansteuerung und -auslegung, besonders in Grenzbereichen, wird auf die Applikationsberichte der Fa. Dallas verwiesen.

In der Praxis sind für den Bausteinumfang einer Heizungssteuerung folgende Werte zu erwarten:
8 Raumsensoren für Fußbodenheizkreise, 11 Sensoren für technische Zwecke ergeben in Summe 19 Sensoren.

Die in dieser Lösung maximal verwaltbare Bausteinanzahl wird durch die Größe des zur Verfügung stehenden Speichers begrenzt. Für den ATmega32 wurden 63 Datensätze vorgesehen. (In einem EEPROM mit einer Größe von 1024 Byte finden 64 Sätze zu 16 Byte Platz. Die 16 Byte des Satzes „0“ werden für andere Zwecke benötigt. Deshalb können nur 63 Bausteine verwaltet werden.

3.8.1 Schaltung

Die durchgeführten Schaltungserweiterungen werden mit einer Flachbandleitung am 14-poligen Erweiterungsstecker S1 der Leiterplatte angeschlossen.



Die Leitungen D3, D4 und D5 sind mit der Bezeichnung D_{Q3}, D_{Q4} und D_{Q5} als Busleitungen für den 1-wire Bus vorgesehen. Der Pull-upwiderstand sollte abhängig von der Leitungslänge mit 1 kΩ bis 4,7 kΩ dimensioniert werden. Der interne Pullup-Widerstand des Prozessors ist mit 20-50 kΩ zu hoch ohmig. An den Busleitungen D_{Qn} werden die Sensoren parallel betrieben, indem die Leitung DQ der Bausteine an die Signalleitung und GND an die Masseleitung angeschlossen wird.

3.8.2 einsetzbare Temperatursensoren

Zur Zeit werden die folgenden Thermometerbausteine angeboten.

Typ	von der Software unterstützt	Toleranz	Betriebsspannung	Verfahren	family code	Read Power Supply	Anmerkungen
DS1820	X	+/- 0,5°C	4,3 - 5,5 V	count remain	10h	X	ältester Baustein der Serie
DS18S20	X	+/- 0,5°C	3,0 - 5,5 V	count remain	10h	X	aktueller Ersatz für DS1820
DS1821		1°C	2,7 - 5,5 V	count remain			hochauflösende Messung möglich, keine Seriennummer vorhanden, Betrieb als Stand-alone-Thermostat möglich
DS1822		+/- 2°C	3,0 - 5,5 V	9-12 bit	22h	X	
DS18B20		+/- 0,5°C	3,0 - 5,5 V	9-12 bit	28h	X	Kompatibel zum DS1822

Der DS1820 und der DS18S20 werden von der Software unterstützt. Die Verwendung des DS1821 macht keinen Sinn, weil der Baustein nicht multi-drop-fähig ist und deshalb nur als einziger Baustein an einer Busleitung betrieben werden kann.

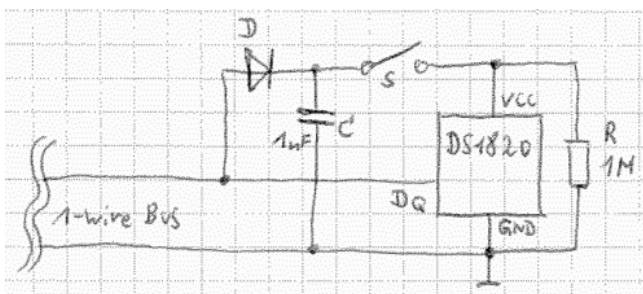
Die Bausteine DS1822 und DS18B20 sind auch einsetzbar. Dazu muß die Software erweitert werden, weil die Temperaturdaten anders zu interpretieren sind.

Zur Zeit ist nur das „count remain“-Verfahren realisiert. Danach werden die Temperatur mit einer Auflösung von 0,5 °C ausgelesen und die weiteren Bruchteile über den verbleibenden Zählerstand und die Zählschritte pro Grad errechnet. In der Variante 9-12 bit wird dagegen die Auflösung im Baustein eingestellt und vollständig im 2-er Komplement geliefert. Bei Bedarf muß diese Variante programmiert werden.

Mit dem „Read Power Supply“-Kommando kann bei busversorgten Bausteinen abgefragt werden, ob der Baustein über den Bus versorgt wird, oder eine separate Versorgung hat. Mit einer Zusatzschaltung kann auch ein „Bit“ eingelesen werden.

3.8.2.1 Eingabeport mit dem Temperatursensor DS1820

Mit dem Kommando Read Power Supply (B4) läßt sich abfragen, ob das IC über den Bus (parasitär) oder über den Betriebsspannungsanschluß versorgt wird. Mit dieser Abfrage läßt sich indirekt die Stellung eines Schalters ermitteln.



Die Betriebsspannung wird dazu mit der Siliziumdiode **D** und dem Kondensator **C** (1nF) aus dem one-wire Bussignal **DQ** abgeleitet. Der **Vcc**-Anschluß des DS1820 wird über einen Widerstand **R** (1MΩ) an Masse gelegt, damit ein definiertes Signal anliegt und nach dem Abschalten die Eingangskapazität sicher entladen wird. Die Betriebsspannung am Kondensator **C** wird dann mit dem Schalter **S** auf den DS1820 geschaltet.

Anwendungsbeispiele:

- Wenn der Schalter als Öffnungskontakt eines Fensters ausgebildet ist kann eine Steuerung dafür sorgen, daß beim Öffnen des Fensters die Heizung ausgeschaltet wird.
- Ein Schalter an einem Raumtemperaturfühler kann zum manuellen Ein- und Ausschalten der Heizung benutzt werden.
- Mit einem Relaiskontakt zur Potentialtrennung kann festgestellt werden, ob eine Umwälzpumpe ein- oder ausgeschaltet ist.

In der Software wird das Power-Supply-Bit ausgelesen und für weitere Auswertungen zur Verfügung gestellt. Bei gesetztem Power-Supply-Bit wird auf der HTML-Seite hinter der Temperatur das Zeichen „#“ ausgegeben.

3.8.3 Datenverwaltung

Für jeden one-wire Baustein wird ein Datensatz angelegt. Der Datensatz hat einen eindeutigen Index von 1 bis zur maximalen Satzanzahl. Die Daten werden auf das EEPROM und das SRAM verteilt.

Dabei werden die praktisch konstanten Daten, wie die Seriennummer des Bausteins und die Bitmaske wegen der begrenzten Schreibzyklen im EEPROM abgelegt. Im SRAM werden die Meßergebnisse und einige Statusinformationen gespeichert.

3.8.3.1 Verwaltung der statischen Daten im EEPROM

Die one-wire Suchroutine sucht die angeschlossenen Bausteine, bestimmt die Seriennummer und den Anschluß am Prozessor. Die Seriennummer ohne CRC-Summe und die Bitmaske zum Feststellen des Anschlusses werden im EEPROM des Prozessors abgelegt.

Im EEPROM ist für jeden Baustein ein 16 Byte großes Feld vorgesehen, das sich im Speicherdump gut kontrollieren läßt. In den ersten 7 Byte steht die Seriennummer ohne Prüfsumme. Im 8. Byte steht die Bitmaske der Anschlußleitung. In der Bitmaske ist das Bit der korrespondierenden Portleitung gesetzt, an der der Baustein angeschlossen ist. Wenn das Feld unbenutzt ist, ist die Bitmaske im EEPROM unprogrammiert und hat den Wert 0xff.

Die verbleibenden 8 Byte des Feldes sind für den Namen vorgesehen. Damit auch lange Namen benutzt werden können, wurden für bestimmte Worte und Silben Token angelegt.

Bezeichnung	Token	Token	Token	Token	Eingabe
Heizkörper Vorlauf	Heizkörper	Vorlauf			*0*2
Heizkörper Rücklauf	Heizkörper	Rücklauf			*0*3
Fußbodenheizung Vorlauf	Fußbodenheizung	Vorlauf			*1*2
Fußbodenheizung Rücklauf	Fußbodenheizung	Rücklauf			*1*3
Kesseltemperatur		temperatur			Kessel*9
Luftwärmetauscher Lufteinlaß	Luft	wärmetauscher	Luft	einlaß	*4*7*4*5
Luftwärmetauscher Luftauslaß	Luft	wärmetauscher	Luft	auslaß	*4*7*4*6
Luftwärmetauscher Vorlauf	Luft	wärmetauscher	Vorlauf		*4*7*2
Luftwärmetauscher Rücklauf	Luft	wärmetauscher	Rücklauf		*4*7*3
Außentemperatur Vorn	Außen	temperatur	Vorn		*8*9*A
Außentemperatur Hinten	Außen	temperatur	Hinten		*8*9*B

Damit lassen sich alle üblichen Kurzbezeichnungen in 8 Byte unterbringen. Nebenstehend eine Zusammenstellung der Token:

Die Eingabe der Bezeichnung „Kesseltemperatur“ erfolgt als: Kessel*9

Die Eingabe der Bezeichnung „Fußbodenheizungswärmetauscher Vorlauf“ erfolgt als: *1s*7*2

Zur Ausgabe einer Bedienungshilfe in Form einer Liste der zur Verfügung stehenden Abkürzungen, wurde eine Schleifenstruktur mit folgende Marken eingeführt:

%ANFN Start eines wiederholbaren Blockes. Beim Erreichen der Marke wird die Startbedingung für die Wiederholung gesetzt. Bei jeder Wiederholung wird der Datenzeiger unmittelbar hinter die Marke gesetzt. Die Schleife beginnt beim Index 0 und endet mit dem Index des letzten Tokens.

%VAT* Ausgabe des durch den Schleifenzähler indizierten Tokens. Wenn das Zeichen „*“ durch den Index des Tokens ausgetauscht wird, wird dieses Token ausgegeben. z.B. %VAT3 = „Rücklauf“; %VADB = „Hinten“. Der Index wird in diesem Fall durch ein Zeichen [0..9,A..Z] gebildet (zur Basis 36).

%VAL Ausgabe des Index, also der Laufvariablen der Schleife. Der Schleifenzähler läuft von 0 bis zur Tokenanzahl.

%ENDN Ende des zu wiederholenden Blockes. An dieser Stelle wird die Wiederholbedingung getestet, die Schleifenvariable erhöht und bei Bedarf zurückgesprungen.

Zur Eingabe des Namens wurde ein Webseite angelegt, in der sich die Bausteindaten auch löschen lassen. Die Webseite kann in der Temperaturmeßansicht über die Datensatznummer angewählt werden.

Token	Kürzel
Heizkörper_	*0
Fußbodenheizung	*1
Vorlauf	*2
Rücklauf	*3
Luft	*4
einlaß	*5
auslaß	*6
wärmetauscher_	*7
Außen	*8
temperatur_	*9
Vorn	*A
Hinten	*B
zimmer	*C
Warm	*D
Kalt	*E
Wasser	*F

3.8.3.2 Verwaltung der dynamischen Daten im SRAM

Der Webserver führt im SRAM eine Liste der im EEPROM eingetragenen Bausteine mit den zuletzt ermittelten Meßwerten. Diese Liste enthält für jeden Baustein 2 Byte.

1. Byte: Temperatur des Thermometers

0..125	Betrag der Temperatur in °C für positive Temperaturen,
126	unbenutzt,
127	„dieser Eintrag ist leer“,
128..183	Betrag der Temperatur in °C für negative Temperaturen,
184..255	unbenutzt.

2. Byte: Nachkommaanteil der Temperatur mit Information zur externen Stromversorgung

00..99	Nachkommaanteil in 0,01 °C,
100..126	unbenutzt,
127	„dieser Baustein ist nicht aktiv“ (0x7f), d.h. dieser Baustein ist z. Zt. nicht angeschlossen. Wir merken uns jedoch die Seriennummer um eine evtl. vorhandene Namenszuordnung nicht zu verlieren,
128..227	Nachkommaanteil in 0,01 °C mit gesetztem Power Supply Bit des Sensors,
228..254	unbenutzt,
255	Dieser Baustein ist kein Thermometer.

Um Fließkommaoperationen zu vermeiden, werden der Vorkommaanteil und der Nachkommaanteil der gemessenen Temperatur als Integerwerte behandelt und gespeichert.

Die Temperatur wird aus dem Sensor im 2-er-Komplement auf 0,5 °C genau eingelesen. Die Darstellung wird dann in Betrag und Vorzeichen umgewandelt und auf ganze Grade genau als Vorkommaanteil im ersten Byte gespeichert. Der Nachkommaanteil wird aus dem im Sensor verbliebenen Zählerstand (count remain) und den Zählschritten pro Grad berechnet und vorzeichenkorrigiert gespeichert.

Vor der Ausgabe wird geprüft, ob die Temperatur negativ ist und gfs. ein '-' Zeichen ausgegeben. Der Nachkommaanteil ist immer eine zweistellige Zahl, die auf das Komma folgt.

Der Listenindex entspricht dem Index der Daten im EEPROM. Der Index 0 bleibt im SRAM unbenutzt, weil im EEPROM an dieser Stelle andere Daten gespeichert sind. Der Speicherplatz im SRAM kann anderweitig genutzt werden.

3.8.4 Ein- und Ausschalten der one-wire Busleitungen

Im Programm besteht eine Bitmaske, in der alle als one-wire Bus benutzbaren Leitungen markiert sind. Mit dieser Bitmaske wird verhindert, daß bei der Suche nach one-wire Bausteinen anders verwendete Leitungen des gleichen Ports beeinträchtigt werden.

Daneben gibt es eine Bitmaske, mit der die one-wire Busleitungen ein- oder ausgeschaltet werden. Diese Auswahlmaske läßt sich über das Webinterface verändern, um einzelne Busleitungen im Betrieb ein- und auszuschalten zu können.

3.8.5 Ausgabe der one-wire Daten mit der Webserver

Zur Auswahl der auszugebenden Daten wird ein Optionsfeld mit Schaltern eingesetzt. Die Schalter lassen sich über das Webinterface stellen.

Folgende Optionen stehen zur Verfügung:

- -0- Ausgabe aller Datensätze (vom ersten bis zum letzten Platz)
- -1- Ausgabe aller Datensätze in denen eine Seriennummer notiert, also ein Eintrag vorhanden ist
- -2- Ausgabe aller Datensätze mit Thermometerbausteinen (erstes Byte der Seriennummer = 10h)
- -3- Ausgabe aller Datensätze aktiver Thermometerbausteine (aktiv bedeutet: am Bus angeschlossen)

Die Bedingungen werden anhand der folgenden Daten geprüft:

SRAM [Index][0] = 7Fh; Eintrag unbenutzt

SRAM [Index][1] = 7Fh; Baustein nicht aktiv (nicht angeschlossen)

SRAM [Index][1] = FFh; Der Baustein ist kein Thermometer

Die Nutzung von Platzhaltern als Steuerelement in HTML-Seiten war bereits mit dem Steuerelement „%END“ realisiert, daß die Webseite abschließt.

Um eine komfortable Auswahlmöglichkeit bei der Datenausgabe zu erhalten, wurde eine Schleifenstruktur mit folgende Marken eingeführt:

- %ANFT** Start eines wiederholbaren Blockes. Beim Erreichen der Marke wird die Startbedingung für die Wiederholung gesetzt. Bei jeder Wiederholung wird der Datenzeiger unmittelbar hinter die Marke gesetzt. Die Schleife beginnt beim Index 1 und endet mit dem Index der letzten Tabellenposition.
- %VAD*** Ausgabe der durch den Schleifenzähler indizierten Seriennummer. Wenn das Zeichen „*“ durch den Index des Sensors ausgetauscht wird, wird nur diese Seriennummer ausgegeben. z.B. %VAD3 = Sensor 3; %VADB = Sensor 11; %VADV = Sensor 32. Der Index wird in diesem Fall durch ein Zeichen [0..9, A..Z] gebildet. Der Index wird damit einstellig zur Basis 36 dargestellt.
- %VAd*** Ausgabe der durch den Schleifenzähler indizierten Temperatur.
- %VAL** Ausgabe des Index, also der Laufvariablen der Schleife. Der Schleifenzähler läuft von 1 bis zur Satzanzahl. Je nach eingestellter Option werden Indizes ausgelassen.
- %ENDT** Ende des zu wiederholenden Blockes. An dieser Stelle wird die Wiederholbedingung getestet, die Schleifenvariable auf den nächsten gültigen Wert gesetzt und bei Bedarf zurückgesprungen. Je nach eingestellter Option werden bestimmte Datensätze übersprungen.

Die Anzeigeaktualisierungszeit (refresh-Zeit) der HTML-Seite kann ebenfalls über das Webinterface eingestellt werden. Die Eingabe wird mit einem select-Element durchgeführt, in dem die Zeit aus einer vorgegebenen Liste wählbar ist. Damit läßt sich der Wertebereich bei der Eingabe einschränken. Eine refresh-Zeit von 0 s macht das System unbedienbar und wird nicht vorgeschlagen.

3.9 Komprimierte Speicherung von HTML-Seiten

Da die HTML-Seiten bei kleineren Anwendungen im Programmspeicher abgelegt werden, ist es erforderlich, mit dem endlichen Speicherplatz sparsam umzugehen. Für die Speicherung der Bausteinnamen im EEPROM steht ein Programm zur Verfügung mit dem Token durch Bezeichnungen ausgetauscht werden können. Dieses Programm kann auch genutzt werden, um bei Wiederholungen im HTML-Text den Platzbedarf zu verringern und den entsprechenden Textbereich nur einmal im Speicher abzulegen.

Der Tokenbereich wurde dazu unterteilt. Die Token von „0“ bis „MAX-TOKEN“ werden für die Bausteinbezeichnungen der Dallasbausteine verwendet. Alle weiteren bis zum Index 36 stehen für wiederholt genutzte HTML-Texte zur Verfügung.

3.10 Temperaturaufzeichnungen

Es bietet sich an, die gemessenen Temperaturen aufzuzeichnen, grafisch darzustellen und auszuwerten. Die Aufbereitung für eine grafische Darstellung erfordert einen hohen Speicheraufwand und ist deshalb nur schwierig mit den Mitteln des benutzten AVR-Prozessor zu erledigen. Deshalb wird dazu ein externer Rechner verwendet, der regelmäßig die Messdaten erhält. Wenn der externe Rechner permanent zur Verfügung steht, müssen die Daten nicht im AVR zwischengespeichert werden. Die Datenübertragung findet dann häufiger statt, ist dafür aber einfacher zu programmieren.

Die Datenübertragung muß entweder vom Meßsystem oder vom Host initiiert werden. Das Meßsystem könnte z.B. UDP Pakete senden, die im Host in ein Logfile geschrieben werden. Dazu steht unter Linux das Werkzeug syslog zur Verfügung. Das regelmäßige Senden von UDP-Paketen aus einer Anwendung heraus, ist jedoch auf dem Mikrocontroller noch nicht realisiert.

Alternativ könnte der Host die Daten des Meßsystems zyklisch abfragen. Die Datenübertragung kann über das Telnet-, das FTP- oder das HTML-Protokoll abgewickelt werden.

Für die realisierte Lösung werden die Daten von einem Linux-Rechner zyklisch per HTML-Protokoll abgefragt. Die Abfrage geschieht mit einem zeitgesteuerten (Cron-)Skript, das mittels „wget“ die Seite „log.htm“ vom Prozessor liest, die Daten extrahiert und dem Programm RRDTOOL übermittelt.

3.11 Betrieb einer Webkamera

Als Kamera wird eine Aufsteckkamera für Mobiltelefone von Sony Ericsson, Typ MCA-25 benutzt. Die Kamera wird über die serielle Schnittstelle angesprochen.

Die benutzten Datenraten sind 9600 bd und 460800 bd (= 4 * 115200 bd). Der Mindestwert für den USART-Vorteiler des

Baudratenteiler 16	
Faktor	Frequenz
1	7,3728 MHz
2	14,7456 MHz

ATmega 32 beträgt acht und legt so die Mindesttaktfrequenz auf 8 * 460800 = 3,6864 MHz fest.

Wegen des einstellbaren Vorteilers sind auch ganzzahlige Vielfache bis zur maximalen Betriebsfrequenz zulässig. Der ATmega bietet die Möglichkeit, den Baudratenteiler auf 16 zu setzen. Damit ist eine präzisere Abtastung des

Baudratenteiler 8	
Faktor	Frequenz
1	3,6864 MHz (Mindestfrequenz)
2	7,3728 MHz
3	11,0592 MHz
4	14,7456 MHz (höchste Frequenz)
5	18,4320 MHz (>16 MHz!)

Signals möglich. Diese Betriebsart ist jedoch nur bei zwei Quarzfrequenzen möglich.

Die Kamera muß zum Anschluß zerlegt werden. Das Gehäuse besteht aus vier Kunststoffteilen. Nach dem Abhebeln der oberen Blende (mit der Firmenbezeichnung), das zweckmäßigerweise mit einem Messer, ausgehend von der gerundeten Gehäusekante, durchgeführt wird, werden vier Schrauben sichtbar, die die Oberschale mit der unteren Blende verbinden. Die Unterschale der Kamera liegt zwischen Oberschale und unterer Blende. Nach dem Ausbau der Leiterplatte wird der Stecker einige Male hin- und hergebogen, bis die angelöteten Kontakte am Stecker abbrechen.

Ausführliche, bebilderte Umbauanleitungen finden sich unter:

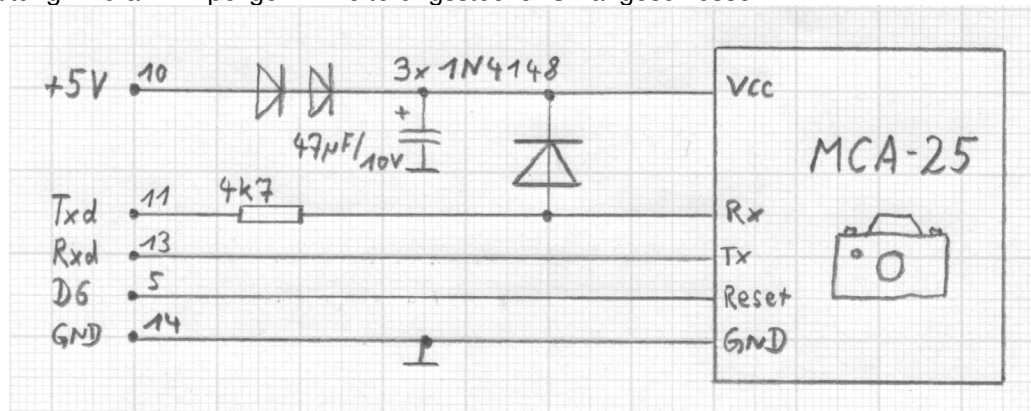
- www.mikrocontroller.com
- www.ulrichradig.de
- avr.auctionant.de/avr-ip-webcam

Mit der dort veröffentlichten Software gehen bei der Ausgabe eines Bildes zwei Sekundenimpulse verloren, weil durch die im Interrupt abgewinkelte Bildausgabe der Timerinterrupt nicht rechtzeitig bearbeitet wird. Die Uhr geht dadurch nach. Zur Lösung dieses Problems wird der 16 bit Timer freilaufend betrieben und die Zeit wird durch regelmäßiges Abfragen des Timers (polling) auf dem aktuellen Stand gehalten.

3.11.1 Anschaltung der Kamera

Die Versorgungsspannung der Kamera kann über zwei Dioden aus der 5V-Betriebsspannung abgeleitet werden. Die Betriebsspannung der Kamera sollte nach Erfahrungsberichten im Bereich von 3,3 bis 4,2 V liegen. Der Siebkondensator kann auch größer dimensioniert werden. Die Signalspannungen dürfen die Betriebsspannung der Kamera nicht überschreiten, weil andernfalls Ströme über die Eingangsschutzdioden fließen. Der Ausgang Port D6 zur Ansteuerung der Reset-Leitung wird als open collector Ausgang mit Pullup-Widerstand programmiert. Der Pegel der Txd-Leitung wird über einen Widerstand 4,7 k Ω mit einer Siliziumdiode auf die Kamerabetriebsspannung geklemmt.

Die Schaltung wird am 14-poligen Erweiterungsstecker S1 angeschlossen.



Schaltskizze zum Anschluß der Kamera und der Temperatursensoren

3.11.2 Betrieb der Kamera

Die Kamera wird mit dem Start des Webserver initialisiert und konfiguriert. In der Konfiguration können die Parameter „Funlayer“ und „Bildgröße“ geändert werden. Für „Funlayer“ sind die Ziffern von 0 bis 8 zulässig. Bei „0“ erfolgt keine Einblendung.

Die Bildgrößen 640*480, 320*240 und 160*120 werden in einem Auswahlfeld vorgegeben.

3.12 Automatische Einstellung der Uhr

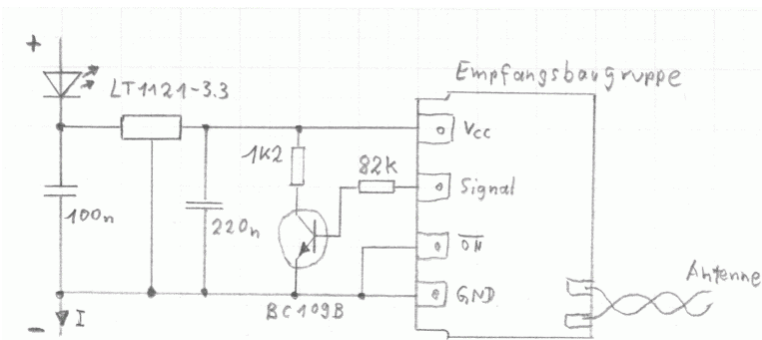
Damit die Zeit für jedermann erreichbar ist, wird sie von der Physikalisch Technischen Bundesanstalt (PTB) dargestellt und über den Zeitzeichensender DCF-77 auf der Frequenz 77,5 kHz verbreitet.

Dieser Zeitzeichensender sendet Sekundenmarken in Form von Trägerabsenkungen (Amplitudenmodulation mit einem Rechtecksignal). Mit jeder Sekundenmarke wird jeweils ein Bit übertragen.

Das System besteht aus einem Langwellenempfänger, der die Modulation als logisches Signal herausgibt und der Dekodierung, die im Webserver erfolgt.

Um Störungen des Datenempfangs zu vermeiden ist es wichtig den Empfänger am optimalen Empfangsort, möglichst weit entfernt von Störquellen, aufzustellen.

3.12.1 Der Langwellenempfänger



Der Empfänger wird aus einer preiswerten Uhr ausgebaut. So wird z. B. von einer Tankstellenkette für eine Rabattkarte und 3,00 € ein geeigneter Funkwecker aus chinesischer Produktion angeboten. Beim Lebensmitteldiscounter sind vergleichbare Uhren für 5,99 € erhältlich. Nach dem Zerlegen verbleibt eine mit 3V zu versorgende Leiterplatte mit einem gebondeten Chip und einer Ferritstabantenne. Prozessor und Empfänger sind mit zwei Leitungen verbunden. Eine dient zum

Einschalten des Empfängers und auf der Anderen liefert der Empfänger das DCF77-Signal.

Der Empfänger wird auf Dauerbetrieb gestellt und mit einem durch die Stromschleife versorgten 3,3V-Regler betrieben. Das Ausgangssignal des Empfängers schaltet einen 1,2 kΩ Lastwiderstand, der den Stromfluß auf etwa 2,8 mA erhöht.

3.12.2 Die Verbindung zwischen Empfänger und Dekoder mit einer Stromschleife

Die Leitungen zur Signalübertragung und zur Energieversorgung des Empfängers dürfen keine Störungen zum Empfänger übertragen. Da die Frequenzanteile im Zeitzeichensignal unter 100 Hz liegen, reicht zur Verbindung zwischen Empfänger und Rechner eine zweiadrige verdrehte Leitung aus (Stromschleife). Damit kann der Betriebsstrom zum Empfänger geführt, das Signal zum Rechner übertragen, eine hohe Störuneempfindlichkeit erreicht und die Leitungslänge bis in den km-Bereich ausgedehnt werden. Gleichtaktstörungen auf der Leitung können mit einer Ferritperle gemindert werden. Der Empfänger benötigt keine separate Stromversorgung.

3.12.2.1 Die Auslegung der Stromschleife

Die Stromschleife führt abhängig vom empfangenen Signal zwei verschiedene Ströme.

Bei 100% der Trägeramplitude fließen nur der Betriebsstrom des Empfängers und der Ruhestrom des Spannungsreglers in Höhe von zusammen 90 µA. Bei der Trägerabsenkung wird ein Lastwiderstand zugeschaltet, der den Strom auf 2,8 mA erhöht. Das Stromverhältnis liegt damit bei etwa 1:30 und läßt eine Rückführung in ein Logiksignal mit hohem Störabstand zu.

Durch das IC zur Betriebsspannungsstabilisierung des Empfängers wird die maximale Betriebsspannung des Empfängers auf 30 V begrenzt. Die minimale Betriebsspannung des Empfängers beträgt 3,8V.

Durch Vorschalten eines Brückengleichrichters kann die Empfängerzuleitung polungsunabhängig ausgeführt werden, der Spannungsfall am Empfänger erhöht sich jedoch um zwei Diodendurchlaßspannungen. Die Stromschleife kann mit einer unregelmäßigen, gesiebten Gleichspannung versorgt werden.

3.12.2.2 Weitere Anzeigeräte

Zur Signalanzeige kann im einfachsten Fall eine Leuchtdiode benutzt werden, die dazu in die Stromschleife geschaltet wird und die Sekundenimpulse des Empfangssignals durch eine Helligkeitsänderung anzeigt. An einer roten Leuchtdiode fallen dann etwa 1,5 V ab.

Es ist möglich in die Stromschleife eine empfängerlose Funkuhr als reines Anzeigerät (Tochteruhr) aufzunehmen. Dabei wird die Betriebsspannung der Tochteruhr durch einen Parallelregler gewonnen und das Datensignal aus der Stromhöhe abgeleitet.

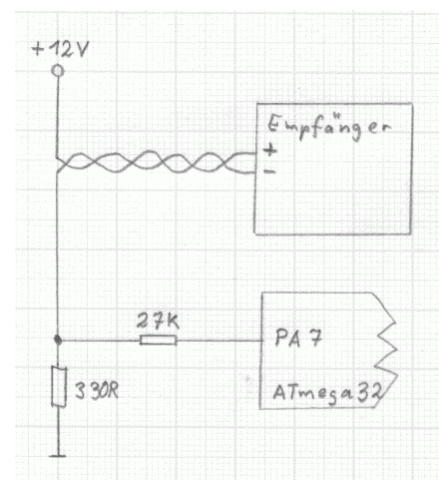
3.12.2.3 Die Versorgung der Stromschleife und die Anschaltung des Microcontrollers

Am ATmega 32 steht zur Auswertung des Signals der Analogeingang PA 7 zur Verfügung, mit dem die Spannung an einem Stromfühlerwiderstand gemessen wird. Der Analog-Digital Wandler wird dazu für den 8 Bit Betrieb konfiguriert und die Betriebsspannung dient als Referenz. Bei einer Dimensionierung des Stromfühlerwiderstands mit 330 Ω ergibt sich für den „1“-Pegel eine Spannung von

$$2,8 \text{ mA} \cdot 330 \Omega = 0,92 \text{ V}$$

und für den „0“-Pegel eine Spannung von

$$90 \mu\text{A} \cdot 330 \Omega = 0,03 \text{ V}.$$



Die zugehörigen Zählerstände sind 47 und 7. Mit der gemessenen Spannung könnten bei Bedarf auch die Signale „Schleife unterbrochen“ und „Empfänger kurzgeschlossen“ ermittelt werden. Mit dieser Beschaltung ist es bei Verzicht auf eine Anzeige-LED möglich, die Stromschleife mit 5 V zu versorgen. Durch Einstellung der Schaltschwelle kann an den Eingang entweder ein Logiksignal (high/low) oder eine Stromschleife mit deutlich geringerem Spannungshub angeschlossen werden.

3.12.3 Die Dekodierungssoftware

In dem hier beschriebenen System werden neben dem DCF-Dekoder ein Temperaturneßsystem und ein Webserver mit Webcam betrieben.

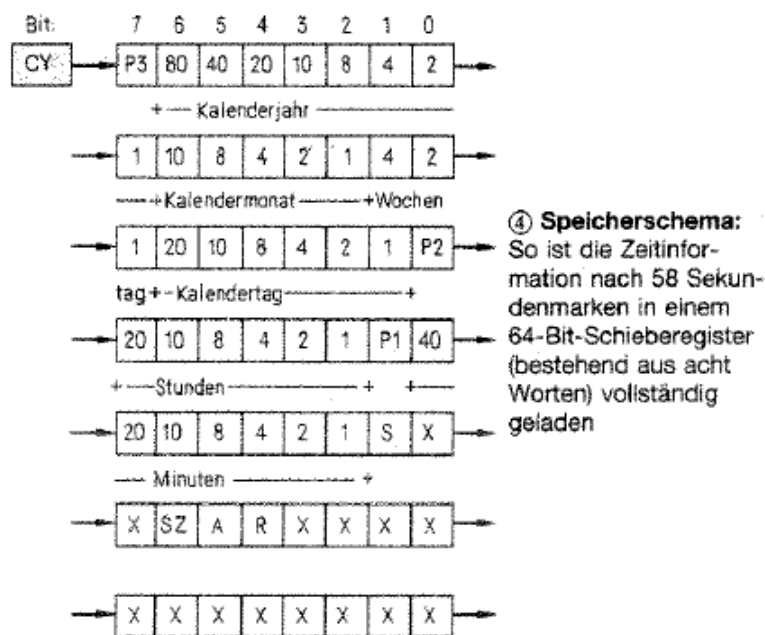
Um Interruptkonflikte zu vermeiden wird das Empfangssignal regelmäßig abgetastet. Eine Abtastfrequenz von 10 Hz würde dabei bereits die Nyquistbedingungen erfüllen. Um die Genauigkeit bei der Erfassung des Sekundenanfangs zu erhöhen wird eine Abtastfrequenz von 100 Hz gewählt. Das dazu erforderliche 10 ms-Zeitraster wird mit dem 16-bit-Zähler aus dem Systemtakt gewonnen.

3.12.3.1 Die Störsicherheit des Empfangs

Die über das DCF-Signal bekannten Daten werden zur Erhöhung der Störsicherheit eingesetzt. Sobald ein Sekundenimpuls empfangen wurde kann der Zeitpunkt zu dem der nächste Sekundenimpuls zu erwarten ist genau bestimmt werden. Um Toleranzen auszugleichen, wird der nächste Sekundenanfang in einem Zeitfenster von 100 ms gesucht. Mit dem Sekundenanfang kann der optimale Abtastzeitpunkt für das Datenbit, 150 ms nach dem Sekundenanfang, bestimmt werden. Das synchronisierte System wertet das Signal 50 ms vor dem erwarteten Sekundenbeginn bis maximal 50 ms danach und zum Abtastzeitpunkt aus. Störungen in der Zwischenzeit bleiben ohne Einfluß.

Dieses Verfahren wurde von der PTB erstmals 1974 in Hardware realisiert und veröffentlicht.

Die Zeit-Daten werden in Form eines 59 Bit langen Daten-Telegramms gesendet. Zur Synchronisation wird in der Sekunde 59 kein Datenbit gesendet.



Die empfangenen Bits werden in einem Schieberegister abgelegt. Zur 59. Sekunde sind dann alle empfangenen Datenbits in der richtigen Position. Damit ist die Zeit auswertbar, wenn mindestens die letzten 38 Datenbits einer Minute empfangen wurden. Die Initialisierungszeit liegt zwischen 40 und 99 Sekunden (mittlere Initialisierungszeit ca. 70 s). Dekodieralgorithmen, die erst ein komplettes Telegramm mit einer Dauer von einer Minute empfangen müssen, benötigen 60 bis 119 Sekunden bis zur ersten Dekodierung. (mittlere Initialisierungszeit ca. 90 s) Weitere Details zum Dekodieralgorithmus sind in Funkschau 22/1984, Seite 67 ff. beschrieben.

In der Regel verlangen die Dekoder nach einem korrekt „gepolten“

Datensignal. Der Trägerabsenkung muß dabei ein fester Logikpegel zugeordnet werden. Die Polungsbezeichnungen normal und invertiert sind jedoch nicht eindeutig. Eine an der Sekundenmarke orientierte Signalbezeichnung würde den „1“-Pegel für die Sekundenmarke als normal festlegen. Bei einer von der Modulationshüllkurve ausgehenden Bezeichnung wäre das gleiche Signal als invertiert zu bezeichnen. Der hier realisierte Dekoder arbeitet unabhängig von der Polarität des Datensignals.

Der Dekoder arbeitet mit einer Zustandsmaschine, die 5 Zustände (0..4) kennt:

0. Signalpegel feststellen und als Pegel des ausgetasteten Signals („Datenbit-Pegel“) annehmen. Anschließend zu Zustand 1.
1. Auf die erste Signaländerung (Flanke) warten und den Zeitpunkt festhalten. Anschließend zum Zustand 2.
2. Auf die nächste Flanke warten und dann die Pulsdauer berechnen. Wenn die Pulsdauer größer als 250 ms ist, dann liegt jetzt ein Sekundenbeginn vor. Nach 150 ms weiter mit Zustand 3. Wenn die Pulsdauer kleiner als 250 ms ist, befinden wir uns am Ende eines Datenbits. Die Zuordnung des

Signalpegels war falsch und wird korrigiert. Außerdem wird der Zeitpunkt wie im Zustand 1 festgehalten. Weiter mit Zustand 2.

3. Dieser Zustand wird 150 ms nach einem Sekundenbeginn erreicht. Jetzt wird das Signal in der Bitmitte abgetastet. Das Ergebnis wird in ein 64 Bit langes Schieberegister geschoben. Jetzt wird auf den nächsten Sekundenbeginn gewartet. Der ideale Zeitpunkt ist in 850 ms. Wir wählen jedoch ein Abtastfenster von +/- 50 ms und fangen deshalb in 800 ms mit der Abtastung an. Der nächste Zustand ist Nr. 4.
4. In diesem Status wird das Signal 10 mal abgetastet. Wenn der Sekundenbeginn gefunden wurde, wird in 150 ms Zustand 3 ausgeführt. Wenn 50 ms nach dem erwarteten Sekundenanfang kein Signal vorliegt, befinden wir uns in der 59. Sekunde. Wenn genügend Datenbits (38) gelesen wurden, wird das Signal dekodiert. Nach 900 ms wird das Signal wieder abgetastet, um den Minutenbeginn zu finden (im Zustand 4 bleiben). Wenn zu wenige Datenbits vorliegen, wird der Empfänger neu synchronisiert (Zustand 1).

3.12.3.2 Die Prüfung der Daten

Zur Übertragungssicherung wird die Minuteninformation mit dem Prüfbit P1 auf gerade Parität ergänzt. Die Stunde wird mit dem Prüfbit P2 und die Datumsinformation mit P3 abgesichert. Neben der Paritätsprüfung werden zur Erkennung von Doppelbitfehlern weitere Tests durchgeführt. Dazu werden die BCD-Daten auf Richtigkeit geprüft, in dem der zulässige Bereich der Zahlen (59 Minuten, 23 Stunden, 31 Tage und 12 Monate) geprüft wird. Wochentag, Tag und Monat müssen zudem ungleich 0 sein. Noch nicht realisiert, aber mit vertretbarem Aufwand machbar ist die Prüfung der Ziffern auf den zulässigen Bereich 0..9.

Weitere Prüfungen auf Redundanz am Einzeltelegramm werden unverhältnismäßig aufwendig. So könnte z. B. errechnet werden, ob der Tag zum Monat gehört („31. Februar“). Weiterhin ließe sich mit einem ewigen Kalender prüfen ob der Wochentag zum Kalenderdatum gehört und die Zeitzonebits zueinander (MEZ <> MESZ) und zum Datum passen.

Eine noch höhere Sicherheit kann auf Kosten der Dekodierzeit erreicht werden, wenn die korrekte Folge zweier aufeinanderfolgender Telegramme geprüft wird.

* * *

17.01.2006

Joachim Börke

email: Joachim.Boerke@gmx.de