

Webserver zur Temperaturmessung

1 Inhalt

1	Inhalt	1
2	Überblick	2
2.1	Hinweise zum Übersetzen der Webserversoftware	2
2.2	Bekannte Probleme	2
2.2.1	TCP-Stack	2
2.2.2	Ausgabe von Daten aus Variablen	2
3	Leistungsmerkmale des Systems	2
3.1	Dynamische Elemente für die Ausgabe auf HTML-Seiten	2
3.1.1	Ausgabe von Daten aus Variablen	2
3.1.2	Einlesen von Client-Daten	3
3.1.3	Die Steuerung von Hardware mittels einer Variablen	3
3.1.4	Zusammenfassung zu Bezeichnungen und Namen	3
3.2	Übersicht der verwendeten Platzhalter	3
3.3	Nutzung des PWM-Betriebs	4
3.4	Auslesen des EEPROM der Netzwerkkarte	4
3.5	Ausgabe des AVR-EEPROM-Inhaltes auf der seriellen Schnittstelle	5
3.6	Einstellen der IP-Adresse über das Webinterface	5
3.7	Ausgabe eines Favourite Icon	5
3.8	Einrichtung einer one-wire Schnittstelle	5
3.8.1	Hardware	5
3.8.2	Datenverwaltung	5
3.8.2.1	Verwaltung der statischen Daten im EEPROM	6
3.8.2.2	Verwaltung der dynamischen Daten im SRAM	7
3.8.3	Ein- und Ausschalten der one-wire Busleitungen	7
3.8.4	Ausgabe der one-wire Daten	7
3.9	Komprimieren von HTML-Seiten	8

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.

Die verwendete Leiterplatte ist bei Holger Buss (www.mikrocontroller.com) erhältlich. Die Software basiert auf der von Ulrich Radig (www.ulrichradig.de). Die Software für das one-wire Interface stammt aus dem Applikationsbericht AVR318 der Fa. Atmel (www.atmel.com).

2.1 Hinweise zum Übersetzen der Webserversoftware

Vor dem Übersetzen muß eventuell die Konfiguration angepaßt werden. In der Originalversion 1.38 sind die wichtigen Einstellungen in den folgenden Dateien zu finden.

Datei	Einstellung
makefile	MCU auf Prozessortyp setzen
main.c	IP-Adresse festlegen
main.h	SYSClk auf die Quarzfrequenz setzen
rtl8019.h	ISP_CTRL setzen, wenn die Hardware von Holger Buss verwendet wird.
mmc.h	USEMMC auf 0 setzen, wenn keine SD-Card genutzt wird
uart.h	Baudrate nach Wunsch ändern
webpage.c	In der Seite Index.htm 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 nach den Änderungen nicht mehr relevant)

Nach den Änderungen und Erweiterungen ist die folgende Konfiguration voreingestellt.

MCU	Atmega 32
IP-Adresse	192.168.99.199
SYSClk	8 MHz
ISP_CTRL	Hardware von Holger Buss
USEMMC	0, d.h. keine SD-Card-Nutzung
Baudrate	19200

2.2 Bekannte Probleme

2.2.1 TCP-Stack

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

2.2.2 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.

3 Leistungsmerkmale des Systems

3.1 Dynamische Elemente für die Ausgabe auf HTML-Seiten

Die Ausgabe einer Variablen als fünfstellige Dezimalzahl war bereits im Quelltext 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 Variablen

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 %VAX angelegt.

In **webpage.c** wird der Platzhalter 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

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 Checkboxes wird der Inhalt nur übertragen, wenn die Checkbox markiert ist. Das Löschen der Markierung einer Checkbox wird deshalb nicht zum Server übertragen. Das ist bei der Auswertung im PC zu beachten. Als Lösung wurde ein „hidden“-Element eingeführt, das alle Checkboxes 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 %VA_m 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=“VA_m“ gesetzt.

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

3.2 Übersicht der verwendeten Platzhalter

Variablen			
Platzhalter	Ausgabeformat	Eingabe	Funktion
%VA0	-	0	temporäre Variable
%VA1	5-stellige Dezimalzahl		Seitenzähler
%VA2 (+/-)	2-3 Ziffern oder „checked“	2-stellige Dezimalzahl	
%VA3 (n)	„checked“, 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	
%VA _h	2-stellige Dezimalzahl	2-stellige Dezimalzahl	Stunden
%VA _m	2-stellige Dezimalzahl	2-stellige Dezimalzahl	Minuten
%VA _s	2-stellige Dezimalzahl	2-stellige Dezimalzahl	Sekunden
%VA _i	2-3-stellige Dezimalzahl	-	IP-Adresse 1. Byte
%VA _j	2-3-stellige Dezimalzahl	-	IP-Adresse 2. Byte

%VAk	2-3-stellige Dezimalzahl	Dezimalzahl 0..255	IP-Adresse 3. Byte
%VAI	2-3-stellige Dezimalzahl	Dezimalzahl 0..255	IP-Adresse 4. Byte
%VAR (* n)	2-3-stellige Dezimalzahl	Dezimalzahl 0..255	Index, dezimal, „selected“, wenn VAR=n
%VAO	0..9 A..Z	-	Index zur Basis 36
%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
%VAD(*)	16-stellige Hexzahl	-	Seriennummer des Dallas-Bausteins
%VAd(*)	25,23 °C #	-	Temperatur des DS1820
%VAT(*)	„String“	-	Ausgabe eines Tokens
%VAN(* /)	„String“	„String“ mit Token	Ein/Ausgabe eines Bausteinnamens
%DELETE	-	Satznummer	Satz löschen
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		

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.
- (*|/) Anstelle der Option * ist die Option / möglich. Bei der Option / wird das Feldelement durch den Schleifenindex ausgewählt. Die Ausgabe erfolgt jedoch in einer Tokenschreibweise, die kompatibel zur Dateneingabe der Variablen ist. Dieses Format dient zur Vorbesetzung des 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 bei gleicher Syntax umzusetzen. Je nach Verwendung wird eine Dezimalzahl ausgegeben, wenn das Zeichen + bzw. – fehlt.
- | | VAn+ | VAn- |
|-----------------|----------|----------|
| Variable wahr | selected | |
| Variable unwahr | | selected |
- (n) Wenn der Inhalt der Variablen der Ziffer n entspricht, wird der Text „checked“ ausgegeben. n hat den Wertebereich 0..36, dargestellt als [0..9;A..Z].

3.3 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 beträgt bei 16 MHz Takt 980 Hz.

3.4 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.

3.5 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.

3.6 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. 192.168.0.66). Unter dieser Adresse wird der Server nach der ersten Inbetriebnahme angesprochen. Über die Webschnittstelle läßt sich dann eine andere Adresse einstellen, wobei die Adresseingabe auf den privaten Adreßbereich 192.168.n.m beschränkt ist. Die Adresse wird zusammen mit einer Prüfsumme im EEPROM abgelegt. Nach einem Reset wird die IP-Adresse aus dem EEPROM geprüft und bei Gültigkeit benutzt.

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

Der Lösungsansatz, die IP-Adresse mit einem DHCP-Client zu beziehen, hat den Nachteil, das der Server schwieriger zu finden ist, weil die 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

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“.

3.8.1 Hardware

Mehrere Sensoren werden parallel an einem Bus betrieben. Die Busleitung wird dazu mit einem 4,7 kOhm Pullup-Widerstand an 5V gelegt, weil der interne Pullup-Widerstand des Prozessors mit 20-50 kOhm zu hochohmig ist.

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 maximal mögliche Bausteinanzahl wird in der Anwendung durch die Größe des zur Verfügung stehenden Speichers begrenzt. Für den Atmega32 wurden 63 Datensätze vorgesehen.

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

Deshalb wurden drei Busleitungen vorgesehen, die sich über das Webinterface ein- und ausschalten lassen. Für Details zur Busansteuerung und -auslegung besonders in Grenzbereichen wird auf die Applikationsberichte der Fa. Dallas verwiesen.

3.8.2 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 Schreibzyklenzahl im EEPROM abgelegt. Im SRAM werden die Meßergebnisse und einige Statusinformationen gespeichert.

3.8.2.1 Verwaltung der statischen Daten im EEPROM

Die one-wire Suchroutine findet angeschlossene Bausteine, bestimmt den Anschluß am Prozessor und die Seriennummer. 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 (siehe nebenstehende Tabelle). Wenn der Eintrag unbenutzt ist, ist die Bitmaske im EEPROM unprogrammiert und hat den Wert 0xff.

0x01	Bit 0 des Ports
0x02	Bit 1
0x04	Bit 2
0x08	Bit 3
0x10	Bit 4
0x20	Bit 5
0x40	Bit 6
0x80	Bit 7

Die verbleibenden 8 Byte des Feldes sind für den Namen vorgesehen.

Damit auch lange Namen benutzt werden können wurden bestimmte Worte und Silben zu Token verkürzt.

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.

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

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

3.8.2.2 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	Eintrag ist leer,
128..183	Betrag der Temperatur in °C mit negativem Vorzeichen im Bit 7,
184..255	unbenutzt.

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

00..99	Nachkommaanteil in 0,01 °C ohne externe Stromversorgung,
100..126	unbenutzt,
127	Baustein 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 externer Stromversorgung (Bit 7 gesetzt),
228..254	unbenutzt,
255	Der 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 und den Zählsschritten pro Grad berechnet und vorzeichenkorrigiert gespeichert.

Vor der Ausgabe wird geprüft, ob die Temperatur negativ ist und bei Bedarf wird ein '-' Zeichen ausgegeben. Der Nachkommaanteil ist immer eine zweistellige Zahl, die nach dem Komma ausgegeben wird.

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

3.8.3 Ein- und Ausschalten der one-wire Busleitungen

Zusätzlich zur Bitmaske, mit der die one-wire Busleitung ausgewählt wird, besteht eine Bitmaske, in der alle als one-wire Bus benutzten Leitungen markiert sind. Mit dieser Bitmaske wird verhindert, dass anders verwendete Leitungen des gleichen Ports bei der Suche nach one-wire Bausteinen beeinträchtigt werden. Die Maske läßt sich über das Webinterface verändern, um einzelne Busleitungen im Betrieb ein- und auszuschalten zu können.

3.8.4 Ausgabe der one-wire Daten

Zur Ausgabe der Daten wird ein Optionsfeld mit Schaltern eingesetzt. Die Schalter lassen sich über das Webinterface bedienen.

Folgende Optionen stehen zur Verfügung:

- -0- Ausgabe aller Datensätze (vom ersten bis zum letzten Platz)
- -1- Ausgabe aller Datensätze, auf denen eine Seriennummer notiert ist, 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 mit den 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 über das Webinterface eingestellt werden. Die Eingabe wird mit einem select-Element durchgeführt, in dem die Zeit aus einer vorgegebenen Auswahl 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 Komprimieren von HTML-Seiten

Da die HTML-Seiten bei kleinen Anwendungen im Programmspeicher abgelegt werden, ist es erforderlich, mit dem Speicherplatz sparsam umzugehen. Mit der Verwendung von Token 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 die Anzahl von Wiederholungen im HTML-Text zu verringern und den entsprechenden Textbereich nur einmal im Speicher abzulegen.

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