



Messung von Temperatur und Luftfeuchtigkeit

Vernetzungsfähige Sensoren für lokale Erfassungsaufgaben

Dr. Claus Kühnel

1. Vorbemerkung

Bei der Sicherung von verschiedenen Prozessen kommt der Überwachung von Umweltdaten, wie Temperatur und relativer Luftfeuchtigkeit, eine hohe Bedeutung zu.

Für die Messung der Temperatur sind zahlreiche Verfahren und Bauelemente bekannt und es können bereits mit einfachen Mitteln recht präzise Ergebnisse erzielt werden. Die Messung der relativen Luftfeuchtigkeit hingegen ist eine messtechnisch schwierigere Aufgabe und Messergebnisse mit Messfehlern im Prozentbereich müssen meist akzeptiert werden.

2. Sensoren

Zur Lösung der im Titel des Beitrags angegebenen Aufgabenstellung gibt es nicht den Sensor an sich. Mit dem Ziel der Vernetzbarkeit mehrerer Sensoren soll aus dem verfügbaren Angebot an Sensoren eine Auswahl getroffen werden.

2.1. Sensoren zur Temperaturmessung

Zur Umsetzung von Temperaturwerten in elektrische Signale steht eine Vielzahl von Sensoren zur Verfügung. Jeder Sensor weist spezifische Merkmale auf, die maßgebend für den praktischen Einsatz sind. Am häufigsten verwendet werden:

- Halbleitersensoren
- Widerstandssensoren
- Thermistoren (NTC)
- Thermoelemente

Von besonderem Interesse an dieser Stelle sind Halbleitersensoren. Diese sind häufig bereits Bestandteil integrierter Schaltungen und sind so einfach in ein Messsystem zu integrieren.



2.1.1. Physikalisches Messprinzip

Die Temperaturabhängigkeit der Strom-Spannungs-Kennlinie des Bipolar-Transistors bildet die Grundlage für die Ableitung eines Temperaturmesswertes. Für die Transistorgleichung gilt in guter Näherung:

$$I_C = I_{ES} \left(\exp \frac{U_{BE}}{U_T} - 1 \right) \quad (1)$$

I_C steht für den Kollektorstrom. I_{ES} bezeichnet den stark temperaturabhängigen Emitter-Sperrersättigungsstrom. U_{BE} steht für die angelegte Basis-Emitterspannung und U_T für die zur absoluten Temperatur proportionale Temperaturspannung.

Für die Temperaturspannung gilt:

$$U_T = \frac{kT}{e} \quad (2)$$

Die Differenz zweier Basis-Emitterspannungen ist proportional zur absoluten Temperatur, da sich die Temperaturabhängigkeiten der Sperrersättigungsströme eliminieren. Führt man in die Widlar-Stromspiegelschaltung eine Gegenkopplung ein, dann erhält man eine linear von der absoluten Temperatur abhängige Referenzspannung. Abbildung 1 zeigt einen solchen Stromspiegel auf der Basis zweier npn-Transistoren.

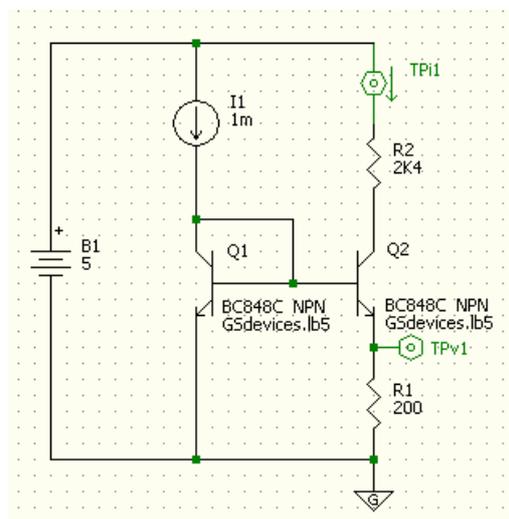


Abbildung 1 Widlar-Stromspiegel mit Gegenkopplung

Am Widerstand R1 kann die zur absoluten Temperatur proportionale Referenzspannung abgegriffen werden. Es gilt die Beziehung:

$$U_{REF} = R_1 I_{C2} = U_T \ln \frac{I_{C1}}{I_{C2}} \quad (3)$$



Der sehr temperaturabhängige Sperrsättigungsstrom tritt durch die Quotientenbildung bei eng benachbarten Transistoren auf einem Chip nicht nachteilig in Erscheinung. Bei diskreten Transistoren ist diese Übereinstimmung der Kennlinien nicht gut, so dass diskrete Stromspiegelschaltungen nur bedingt funktionsfähig sind.

Abbildung 2 zeigt die Ergebnisse einer PSpice-Simulation der mit Abbildung 1 gegebenen Schaltungsparameter.

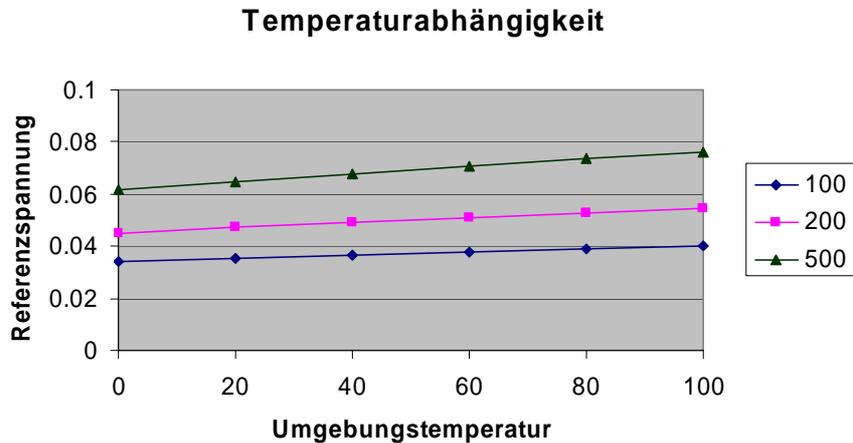


Abbildung 2 Temperaturabhängigkeit der Referenzspannung

Als Bipolar-Transistor wurde auf beiden Seiten ein BC848C eingesetzt, da dessen Modell in der Bibliothek des verwendeten Simulationsprogramms 5SpiceAnalysis (V. 1.05.07) vorhanden war. Wichtig ist an dieser Stelle nur, dass das verwendete Modell die Temperaturabhängigkeit widerspiegelt und beide Transistoren gleich sind. Das Programm 5SpiceAnalysis selbst kann von der URL www.5Spice.com gratis herunter geladen werden.

Über einen Temperaturbereich von 0 °C bis 100 °C wird die Größe der Referenzspannung (abgreifbar über dem Widerstand R₁) berechnet. Der Emitterwiderstand R₁ nimmt Werte von 100 Ω, 200 Ω bzw. 500 Ω ein.

Aus Abbildung 2 erkennt man, dass mit steigendem Wert des Gegenkopplungswiderstands auch die Höhe der Referenzspannung und deren Temperaturkoeffizient zunehmen. Tabelle 1 zeigt die Zusammenhänge quantitativ.

T = 293 K	U _{REF}	TK
R ₁ = 100 Ω	0.035 V	- 0.612 mV/K
R ₁ = 200 Ω	0.047 V	- 0.940 mV/K
R ₁ = 500 Ω	0.065 V	- 1.462 mV/K

Tabelle 1 Einfluss der Gegenkopplung auf die Referenzspannung

Von der zur absoluten Temperatur proportionalen Referenzspannung kann damit durch schaltungstechnische Maßnahmen ein Messsignal für die Temperatur abgeleitet werden.



2.1.2. Sensor zur Temperaturmessung DS2438

Die Überschrift dieses Abschnitts nimmt die Entscheidung für einen Sensor scheinbar bereits vorweg. Da das im vorangegangenen Abschnitt beschriebene Prinzip der Spannungserzeugung sich bei zahlreichen Bausteinen wieder findet, sind es in der Tat andere Eigenschaften, die die Auswahl bestimmt haben.

Der Baustein DS2438 ist ein so genannter *Smart Battery Monitor*, der die Messung von Temperatur, Spannung, Strom, und Zeit ermöglicht sowie 40 Bytes EEPROM und eine unikate Seriennummer zur Verfügung stellt.

Der funktionell recht komplexe Baustein DS2438 wird in Abschnitt 3 gesondert vorgestellt.

2.2. Sensoren zur Messung der relativen Luftfeuchtigkeit

2.2.1. Begriffsbestimmung

Unter der relativen Luftfeuchtigkeit versteht man das Verhältnis der tatsächlich enthaltenen, zur maximal möglichen Masse des Wasserdampfes in der Luft. Die relative Feuchte wird üblicherweise in Prozent angegeben.

Mit steigender Temperatur nimmt die zur Sättigung benötigte Wasserdampfmenge zu. Das hat zur Folge, dass die relative Luftfeuchtigkeit eines gegebenen Luftvolumens bei Erwärmung abnimmt.

Unter http://de.wikipedia.org/wiki/Luftfeuchtigkeit#Relative_Luftfeuchtigkeit können weitere Details nachgelesen werden.

Die Luftfeuchtigkeit ist in einer Vielzahl von Anwendungen von Bedeutung, die unter http://de.wikipedia.org/wiki/Luftfeuchtigkeit#Bedeutung_und_Anwendungsbereiche zusammengestellt sind – hier aber aus Platzgründen nicht weiter betrachtet werden können.

2.2.2. Physikalisches Messprinzip

Die Messung der Luftfeuchtigkeit erfolgt mit Hilfe eines feuchtigkeitsempfindlichen Kondensators, der aus zwei flachen Elektroden besteht, zwischen denen sich eine elektrisch isolierende, hygroskopische Kunststoffschicht (Dielektrikum) befindet. Dieses Dielektrikum kann das in der Luft befindliche Wasser absorbieren. Mit steigender Luftfeuchte steigt so auch die Kapazität des feuchtigkeitsempfindlichen Kondensators.

Zur Messung der elektrischen Kapazität wird eine hochfrequente Wechselspannung angelegt und der fließende Strom gemessen.

In den letzten Jahren werden zunehmend kapazitive Feuchtigkeitssensoren in Messgeräten eingesetzt, die eine hohe Messgenauigkeit ermöglichen und wartungsfrei sind.

Der Kapazitätsunterschied bei der Feuchtigkeitsmessung ist vergleichsweise klein gegenüber der Kapazität schon einiger weniger Meter Kabel. Dies bedeutet, dass die elektronische Auswertung nahe beim Sensor erfolgen muss oder der Sensor selbst Bestandteil eines integrierten Bausteins ist.

2.2.3. Feuchtigkeitssensor HIH-3610

Der von Honeywell angebotene Feuchtigkeitssensor HIH-3610 ist ein speziell für OEM-Anwendungen im Bereich großer Stückzahlen entwickelter Baustein. Der Sensor liefert ein Ausgangssignal, das einen linearen Zusammenhang zwischen Ausgangsspannung und relativer Luftfeuchtigkeit repräsentiert.



Der Feuchtigkeitssensor HIH-3610 misst die relative Luftfeuchtigkeit in einer für die Instrumentierung hinreichenden Genauigkeit. Die Feuchtigkeit wird mit einem lasergetrimmten, thermisch stabilisierten Kapazitätssensor auf Polymerbasis gemessen. Eine On-Chip Signalanpassung sorgt für eine lineare Ausgangsspannung des Sensors.

Die relative Luftfeuchtigkeit berechnet sich gemäß den folgenden Beziehungen:

$$RH_{raw} = 161.29 \frac{V_{AD}}{V_{DD}} - 25.80$$

$$RH_{true} = RH_{raw} / (1.0546 - 0.00216 * T_c)$$

RH_{raw} bezeichnet den Wert der relativen Luftfeuchtigkeit ohne Korrektur der Temperaturabhängigkeit. RH_{true} ist der um die Temperaturabhängigkeit bereinigte Wert der relativen Luftfeuchtigkeit.

Abbildung 3 zeigt die Bauform des Feuchtigkeitssensors HIH-3610.

Die Pinbelegung des abgebildeten Sensors ist

<i>links</i>	<i>GND</i>	<i>Masse</i>
<i>Mitte</i>	<i>V_{AD}</i>	<i>Ausgangsspannung</i>
<i>rechts</i>	<i>V_{DD}</i>	<i>Versorgungsspannung</i>



Die Produktspezifikationen des Feuchtigkeitssensors HIH-3610 sind in Tabelle 2 zusammen gestellt.

Abbildung 3 HIH-3610

Produktspezifikationen

RH Genauigkeit	± 2% RH, 0-100 % RH nicht kondensierend, 25 °C, 5 V DC
RH Austauschbarkeit	± 5% RH, 0-60% RH; ± 8% @ 90% RH typ.
RH Linearität	± 0.5% RH typ.
RH Hysterese	± 1.2% of RH Bereich
RH Wiederholbarkeit	± 0.5% RH
RH Antwortzeit, 1/e	15 s in leicht bewegter Luft @ 25 °C
RH Stabilität	± 1% RH typ. bei 50% RH in 5 Jahren
Versorgungsspannung	4.0 V DC bis 5.8 V DC
Stromaufnahme	200 µA typ.
Messbereich	0 bis 100% RH, nicht kondensierend
bei	-40 °C to 85 °C
Temperaturkompensation	$RH_{true} = RH_{raw} / (1.0546 - 0.00216T)$ T in °C
Bemerkung	Light sensitive, shield from bright light.

Tabelle 2 Produktspezifikationen HIH-3610



2.3. Gesamtschaltung

Die Bausteine DS2438 und HIH-3610 bieten die komplette Funktionalität für die Messung der Raumtemperatur und der relativen Luftfeuchte. Eine Vernetzung mehrerer Sensoren über den 1Wire-Bus ist möglich.

Wenn die Genauigkeit des internen Temperatursensors im DS2438 (+/- 2%) ausreicht, dann kann dieser Sensor allein die Funktionen

- Seriennummer
- Temperaturmessung
- Feuchtigkeitsmessung mit HIH-3610

abdecken.

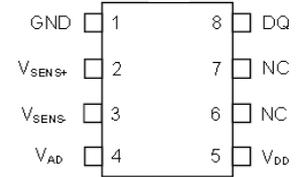


3. Smart Battery Monitor DS2438

Der DS2438 ist als Smart Battery Monitor im 8-Pin-SOIC-Gehäuse (Abbildung 4) zur Messung von Temperatur, Spannung, Strom und Zeit vorgesehen, weist 40 Byte EEPROM und eine unikate Seriennummer auf.

Für die zu betrachtende Applikation wird die interne Temperaturmessung, die Spannungsmessung am Pin V_{AD} sowie die Abfrage der Seriennummer verwendet.

Das Blockschema des DS2438 zeigt Abbildung 5.



DS2438Z, DS2438AZ
8-Pin SOIC (150-mil)

Abbildung 4
Pinout DS2438

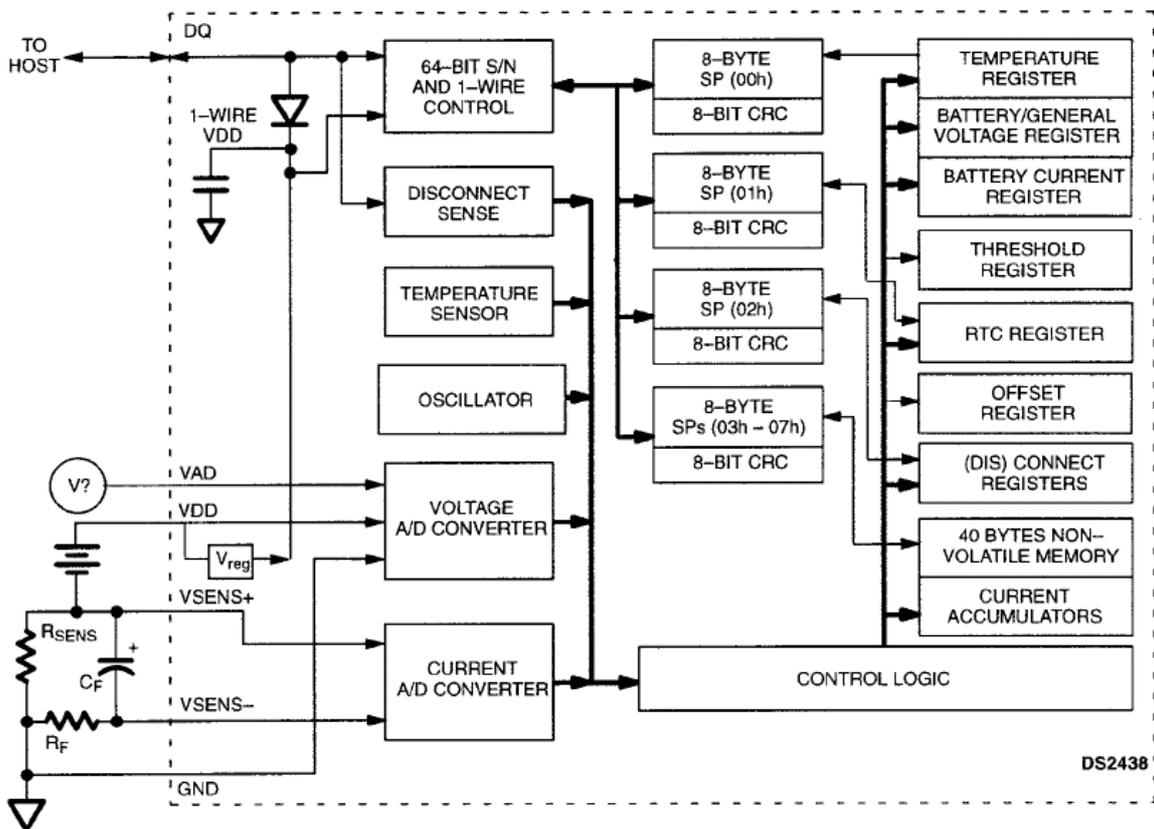


Abbildung 5 DS2438 Blockschema

3.1. Temperaturmessung

Die gemessene Temperatur wird intern mit 13 Bit als 2er-Komplement dargestellt. Die Auflösung beträgt $0.03125\text{ }^{\circ}\text{C}$, der Temperaturbereich $-55\text{ }^{\circ}\text{C}$ bis $125\text{ }^{\circ}\text{C}$.

Tabelle 3 zeigt den Zusammenhang zwischen gemessener Temperatur und dem Ausgabeformat des Messwerts. Das Ergebnis der Temperaturmessung steht in den Bytes 1 und 2 von Page 0 (Tabelle 5)



TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	01111101 00000000	7D00h
+25.0625°C	00011001 00010000	1910h
+0.5°C	00000000 10000000	0080h
0°C	00000000 00000000	0000h
-0.5°C	11111111 10000000	FF80h
-25.0625°C	11100110 11110000	E6F0h
-55°C	11001001 00000000	C900h

Tabelle 3 Zusammenhang Temperatur - Ausgabeformat

3.2. Spannungsmessung

Der DS2438 kann so konfiguriert werden, dass die Spannung an Pin VAD an den internen AD-Umsetzer geschaltet wird. Die Spannungsmessbereich liegt zwischen 1.5 V und 10 V DC bei einer Versorgungsspannung von 5 V.

Tabelle 4 zeigt den Zusammenhang zwischen gemessener Spannung und dem Ausgabeformat des Messwerts. Das Ergebnis der Spannungsmessung steht in den Bytes 3 und 4 von Page 0 (Tabelle 5)

BATTERY VOLTAGE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
0.05V	0000 0000 0000 0101	0005h
2.7V	0000 0001 0000 1110	010Eh
3.6V	0000 0001 0110 1000	0168h
5V	0000 0001 1111 0100	01F4h
7.2V	0000 0010 1101 0000	02D0h
9.99V	0000 0011 1110 0111	03E7h
10V	0000 0011 1110 1000	03E8H

Tabelle 4 Zusammenhang Spannung – Ausgabeformat

3.3. Memory Map

Der Speicher des DS2438 besteht aus Scratchpad RAM sowie SRAM/EEPROM. Geschrieben wird zur Sicherung der Datenkonsistenz immer über das Scratchpad RAM (Schreiben ins Scratchpad RAM und anschließendes Kopieren des Scratchpad RAMs ins RAM/EEPROM).

Der Speicher ist in acht Pages zu acht Bytes organisiert. Jede Page hat ein eigenes Scratchpad RAM. Tabelle 5 zeigt einen Auszug aus dem Memory Map



PAGE	BYTE	CONTENTS	R/W	NV
0	0	STATUS/ CONFIGURATION	R/W	YES
	1	TEMPERATURE LSB	R	NO
	2	TEMPERATURE MSB	R	NO
	3	VOLTAGE LSB	R	NO
	4	VOLTAGE MSB	R	NO
	5	CURRENT LSB	R	NO
	6	CURRENT MSB	R	NO
	7	THRESHOLD	R/W	YES

Tabelle 5 Memory Map (Auszug)

3.4. Konfiguration

Der DS2438 wird über das Status/Configuration Byte konfiguriert (Abbildung 6).

X	ADB	NVB	TB	AD	EE	CA	IAD
X	BF	BF	BF	0	x	x	0
MSB				LSB			

Abbildung 6 Status/Configuration Byte

In Abbildung 6 bedeuten:

ADB	A/D Converter Busy Flag.	(0 = Umsetzung abgeschlossen)
NVB	Nonvolatile Memory Busy Flag	(0 = Schreibvorgang abgeschlossen)
TB	Temperature Busy Flag	(0 = Messung abgeschlossen)
AD	Voltage A/D Input Select Bit	(0 = Pin V_{AD} selektiert)
EE	Current Accumulator Shadow Selector Bit	
CA	Current Accumulator Configuration.	
IAD	Current A/D Control Bit	(0 = keine Strommessung)

Die für den betrachteten Einsatzfall relevante Initialisierung des Status/Configuration Byte ist in Abbildung 6 angegeben.

3.5. Seriennummer

Jeder DS2438 enthält eine unikate Seriennummer (ROM Code) von 64 Bit (Abbildung 7). Der Zugriff auf diese Daten erfolgt über die ROM Commands.

8-BIT CRC CODE	48-BIT SERIAL NUMBER	8-BIT FAMILY CODE (26h)
MSb	LSb	MSb
	LSb	MSb
		LSb

Abbildung 7 ROM Code



3.6. Speicherzugriff

Die Zugriffe beim DS2438 erfolgen nach dem folgenden Protokoll (Transaction Sequence):

- Initialization
- ROM Function Command
- Memory Function Command
- Transaction/Data

Speicherzugriffe (Scratchpad, Register) sind erst möglich, wenn eine der ROM Funktionen (Read ROM, Match ROM, Skip ROM, Search ROM) aufgerufen wurde.

Nach einer erfolgreichen ROM Funktion ist das betreffende 1Wire Device adressiert und die spezifischen Funktionen des DS2438 (Write Scratchpad, Read Scratchpad, Copy Scratchpad, Recall Memory, Convert T, Convert V).

Tabelle 6 zeigt beispielhaft eine Kommandosequenz für Temperatur- und Spannungsmessung und anschließendes Auslesen der Ergebnisdaten.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Reset pulse
RX	Presence	Presence pulse
TX	CCh	Skip ROM
TX	44h	Issue Convert Temperature command, Read Slots
TX	Reset	Reset pulse
RX	Presence	Presence pulse
TX	CCh	Skip ROM
TX	B4h	Issue Convert Voltage command, Read Slots
TX	Reset	Reset pulse
RX	Presence	Presence pulse
TX	CCh	Skip ROM
TX	B8h00h	Issue Recall Memory page 00h command
TX	Reset	Reset pulse
RX	Presence	Presence pulse
TX	CCh	Skip ROM
TX	BEh00h	Issue Read SP 00h command
RX	<9 data bytes>	Read scratchpad data and CRC. This page contains temperature, voltage, and current measurements.
TX	Reset	Reset pulse
RX	Presence	Presence pulse, done

Tabelle 6 Kommandosequenz für Temperatur- und Spannungsmessung

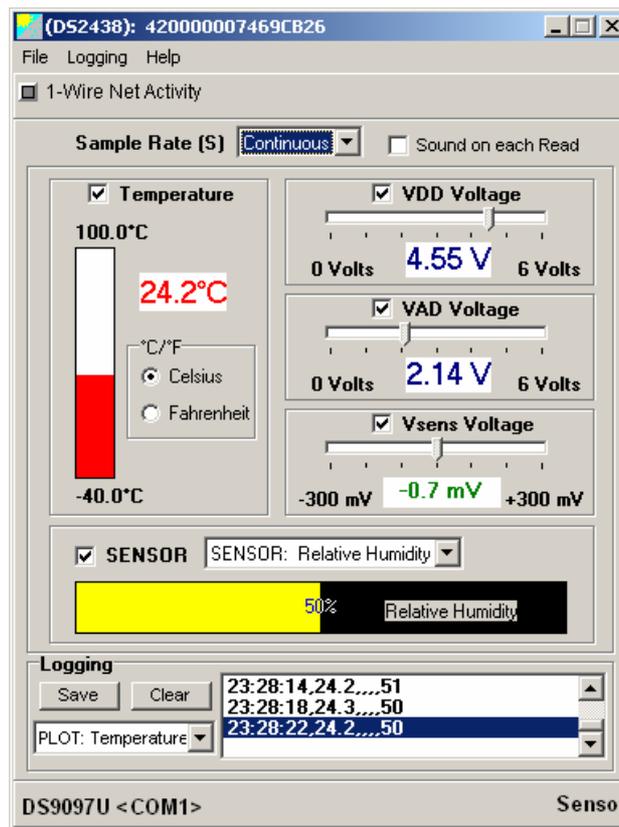


Abbildung 9 DS2438 Messwerte

Ein qualitativer Funktionstest der Temperaturmessung erfolgte durch Besprühen des DS2438 mit Kältespray. Im Temperaturplot (Abbildung 10) ist die abrupte Temperaturänderung und die anschließende Erwärmung deutlich ablesbar. Auf diese Weise kann im konkreten Messaufbau auch die thermische Zeitkonstante ermittelt werden.

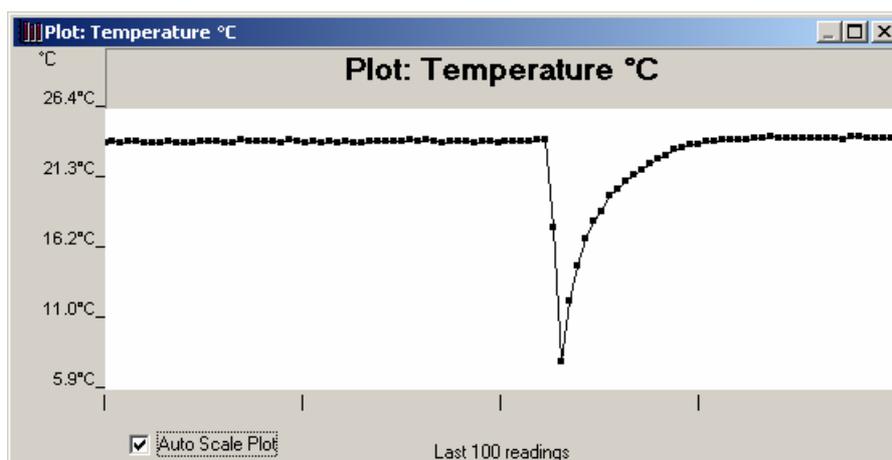


Abbildung 10 Temperaturverlauf nach Besprühen mit Kältespray



4.3. BASCOM-AVR Programmbeispiel

Für eine rasche Umsetzung in Hard- und Software bieten sich Mikrocontroller an, die den 1Wire-Bus direkt unterstützen.

Das folgende Programmbeispiel wurde mit BASCOM-AVR für einen ATmega32 von Atmel erstellt. Abbildung 11 zeigt das verwendete Mikrocontroller-Modul stAVeR-24-M32. Das Modul ist In-System programmierbar (ISP), d.h. es wird kein Programmiergerät benötigt. Arbeitet man mit dem in Abbildung 12 gezeigten und vom Hersteller Lawicel [www.lawicel.com] als stAVeR-24-M32 Activity Board bezeichneten Entwicklungsboard, dann hat man die komplette Infrastruktur einschließlich LCD zur Anzeige und RS-232-Interface zum Entwicklungs-PC beieinander.

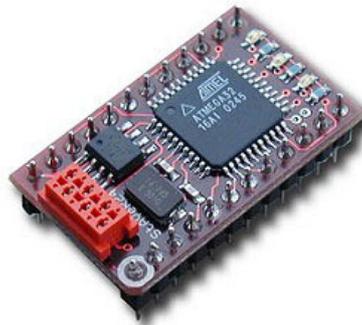


Abbildung 11 stAVeR-24

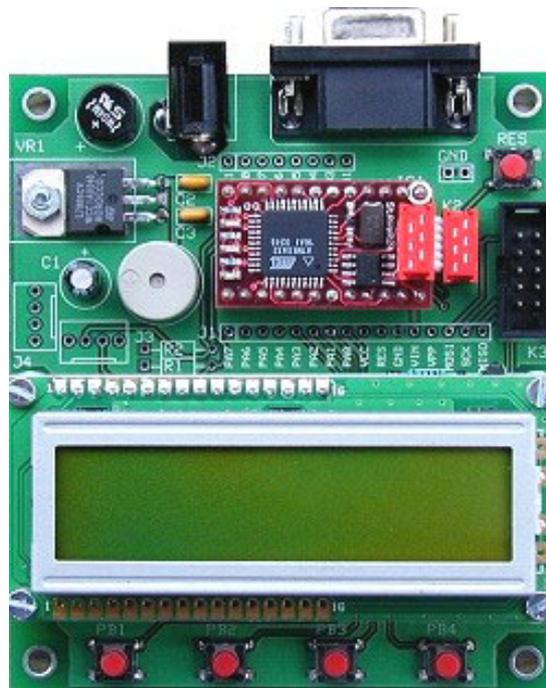


Abbildung 12 stAVeR-24-M32 Activity Board

Listing 1 zeigt den Quelltext des Programmbeispiels RH_TEMP_MEASURE.BAS, der strikt der durch den Baustein DS2438 vorgegebenen Zugriffshierarchie folgt.

Die Datenleitung der Sensorbaugruppe ist hier mit der I/O-Leitung PD4 verbunden. Die Wahl der zu verwendenden I/O-Leitung muss unter Beachtung der vom Entwicklungsboard



benutzten Ressourcen getroffen werden. Hierzu ist am Ende des Beitrags das Schema des stAVeR-24 Activity Boards angegeben (Abbildung 17).

Die Konstante *Dbg* dient als Compilerschalter und sorgt für eine mehr oder weniger geschwätzigte Ausgabe über die serielle Schnittstelle.

Eine Reihe von Konstanten definiert die DS2438-Kommandos, während die Zugriffsoperationen als Subroutines definiert sind. Um die Zugriffsoperationen auf die Sensorbaugruppe so einfach wie möglich zu halten, wird von nur einer angeschlossenen Sensorbaugruppe ausgegangen.

Bei der Initialisierung wird getestet, ob nur ein 1Wire-Baustein angeschlossen ist. Werden mehrere 1Wire-Bausteine oder keiner gefunden, dann erfolgt ein Abbruch des Programms.

Ist die Initialisierung ohne Beanstandungen erfolgt, tritt das Programm in eine Endlosschleife, die alle ca. fünf Sekunden die Sequenz *Measure* anstößt, die vom stAVeR-24 mit einer blinkenden, gelben LED signalisiert wird.

Für die erforderlichen Umrechnungen von Vorteil ist, dass BASCOM-AVR Gleitkomma-Arithmetik beherrscht. So ist es ohne große Umstände möglich, mit exakten Zahlenwerten zu arbeiten. Die Funktion *Fusing()* erlaubt zudem noch eine formatierte Ausgabe der berechneten Zahlenwerte über LCD und RS-232. Die Nachkommastellen wurden hier wegen der zu erwartenden Messfehler von kleiner 2% auf eine Stelle beschränkt.

Die einzelnen Schritte der eigentlichen Messsequenz können an Hand des Quelltextes und der in Abbildung 13 gezeigten Debug-Ausgaben nachvollzogen werden. Läuft das Programm fehlerfrei, dann sind an der RS-232-Schnittstelle nur noch die eigentlichen Messwerte von Interesse. Diese werden nach Änderung der Konstanten *Dbg* auf 0 sowie Compilation und Download dann in der in Abbildung 14 dargestellten Form ausgegeben. Schneidet man im Terminalprogramm diese Werte mit, dann erhält man eine CSV-Datei.

Eine CSV-Datei ist eine Textdatei zur Speicherung oder zum Austausch einfach strukturierter Daten. Das Kürzel CSV steht dabei für *Character Separated Values* oder *Comma Separated Values*, weil die einzelnen Werte durch ein spezielles Trennzeichen, beispielsweise das Komma, getrennt werden. Ein allgemeiner Standard für das Dateiformat existiert nicht. [<http://de.wikipedia.org/w/index.php?title=CSV-Datei&oldid=11021872>]

```
-----
' Measuring Temperature and Humidity by stAVeR24
'
' Sensor for Temperature DS2438 (MAXIM/Dallas)
' Sensor for Humidity HIH-3610 (Honeywell)
'
' 2005-12-12 Claus Kuhnel
-----

$regfile = "M32DEF.DAT"           ' StAVeR-24M32 has the ATmega32 on board
$crystal = 14745600              ' 14.7456 MHz oscillator
$baud = 38400                   ' USART (RS232) Speed
'$sim                          ' uncomment for simulation

Yellowled Alias Portc.3         ' YELLOW LED

Config 1wire = Portd.4

Const Dbg = 0                   ' Switch Compiler Condition (0 = no debug info)

Const Ds2438 = &H26
```



```
Const Skip_rom = &HCC          ' Skip ROM Commands
Const Read_rom = &H33          ' Read ROM
Const Match_rom = &H55        ' Match ROM
Const Search_rom = &HF0       ' Search ROM

' this commands must follow a page byte [0...7]
Const Convertt = &H44         ' Start Temperature Conversion
Const Convertv = &HB4         ' Start Voltage Conversion
Const Read_scratch = &HBE     ' Read Scratchpad RAM
Const Write_scratch = &H4E    ' Write Scratchpad RAM
Const Copy_scratch = &H48     ' Copy Scratchpad RAM
Const Recall_memory = &HB8    ' Copy Memory to Scratchpad RAM

Const Measurevdd = &B00001000 ' Setup for Status Register
Const Measurevad = &B00000000

Dim Scratch(9) As Byte       ' Scratchpad RAM Copy

Dim I As Byte , W As Word

Dim Done As Byte
Dim Itemp As Integer
Dim Temp As Single
Dim Vdd As Single
Dim Vad As Single
Dim Rh_raw As Single
Dim Rh_true As Single
Dim Count As Word

Declare Sub _convertt()
Declare Sub _convertv()
Declare Sub _recallmemory(byval Page As Byte)
Declare Sub _readscratch(byval Page As Byte)
Declare Sub _writescratch(byval Page As Byte)
Declare Sub _copyscratch(byval Page As Byte)
Declare Sub _clearscratch(byval Page As Byte)

Declare Function Initialize() As Byte
Declare Sub Measure()
Declare Sub Crc8_check()

Gosub Init_staver24

Cls : Waitms 500
Reset Yellowled
Upperline
Lcd "DS2438 Test"
Set Yellowled

#if Dbg
    Print : Print "DS2438 Test"
#endif

Wait 1

Done = Initialize()
If Done <> 1 Then
    Cls
    Lcd "Program aborted"
    #if Dbg
```



```
Select Case Done
  Case 0 : Print "No 1wire device found"
  Case Else : Print "Only one 1wire device supported by this program version"
End Select
#endif
End
End If

Wait 3 : Cls

Do
  Reset Yellowled
  Waitms 50
  Measure
  Set Yellowled
  Wait 5
Loop

End

$include "init_staver24.bas"

'-----
' Subroutines & Functions
'-----

Function Initialize() As Byte
  Count = 0
  lwreset                                ' Reset 1Wire Device
  If Err = 1 Then Print "1Wire ERROR"

  lwwrite Read_rom                        ' Read ROM Command
  #if Dbg
    For I = 1 To 8
      Scratch(i) = lwread()
      Print Hex(scratch(i));              ' Output ROM Data
    Next
    Crc8_check                             ' Check CRC-8
    Print

    If Scratch(1) <> Ds2438 Then
      Print "No DS2438 connected"
    End If
  #endif
  W = 1wirecount()
  #if Dbg
    Print "Number of 1wire devices is " ; W
  #endif
  Initialize = W
End Function

Sub Measure()
  #if Dbg
    Print "Initialize Status Register"
  #endif
  _writescratch 0
  lwwrite Measurevdd
  #if Dbg
    Print "Read Scratch"
  #endif
  _readscratch 0
```



```
For I = 1 To 9
  Scratch(i) = lwread()
  Print Hex(scratch(i)) ; Spc(1);
Next
Print

Print "Copy Statusregister to Memory"
_copyscratch 0

_clearscratch 0

Print "Read Scratch"
_readscratch 0
For I = 1 To 9
  Scratch(i) = lwread()
  Print Hex(scratch(i)) ; Spc(1);
Next
Print

Print "Recall Status Register"
_recallmemory 0

Print "Read Scratch"
_readscratch 0
For I = 1 To 9
  Scratch(i) = lwread()
  Print Hex(scratch(i)) ; Spc(1);
Next
Print

Print "Spannungsmessung VDD"
#endif
_writescratch 0
lwwrite Measurevdd
_copyscratch 0
_convertv
_recallmemory 0
_readscratch 0
For I = 1 To 9
  Scratch(i) = lwread()
  #if Dbg
    Print Hex(scratch(i)) ; Spc(1);
  #endif
Next
Itemp = Scratch(5)
Shift Itemp , Left , 8
Itemp = Itemp + Scratch(4)
Vdd = Itemp * 0.01
#if Dbg
  Print : Print Vdd ; " V"

Print "Spannungsmessung VAD"
#endif
_writescratch 0
lwwrite Measurevad
_copyscratch 0
_convertv
_recallmemory 0
_readscratch 0
For I = 1 To 9
  Scratch(i) = lwread()
```



```
#if Dbg
    Print Hex(scratch(i)) ; Spc(1);
#endif
Next
Itemp = Scratch(5)
Shift Itemp , Left , 8
Itemp = Itemp + Scratch(4)
Vad = Itemp * 0.01
#if Dbg
    Print : Print Vad ; " V"
#endif
Rh_raw = Vad * 161.29
Rh_raw = Rh_raw / Vdd
Rh_raw = Rh_raw - 25.8
#if Dbg
    Print "Rel. Luftfeuchte (nicht korrigiert)"
    Print Fusing(rh_raw , "###.#") ; " %"

    Print "Temperaturmessung"
#endif
_convertt
_recallmemory 0
_readscratch 0
For I = 1 To 9
    Scratch(i) = lwread()
    #if Dbg
        Print Hex(scratch(i)) ; Spc(1);
    #endif
Next
Itemp = Scratch(3)
Shift Itemp , Left , 8
Itemp = Itemp + Scratch(2)
Itemp = Itemp / 8
Temp = Itemp * 0.03125
#if Dbg
    Print : Print Fusing(temp , "###.#") ; " grd C"
#endif
Rh_true = Temp * 0.00216
Rh_true = 1.0546 - Rh_true
Rh_true = Rh_raw / Rh_true          ' RH_true = RH_raw/(1.0546-0.00216*T)
#if Dbg
    Print "Rel. Luftfeuchte (korrigiert)"
    Print Fusing(rh_true , "###.#") ; " %"
#else
    Print Count ; ", ";
    Print Fusing(temp , "###.#") ; ", ";
    Print Fusing(rh_true , "###.#")
#endif
Incr Count
Upperline : Lcd Fusing(temp , "###.#") ; " grd C"
Lowerline : Lcd Fusing(rh_true , "###.#") ; " %"
End Sub

Sub _convertt()
    lwreset          ' Reset 1Wire Device
    If Err = 1 Then Print "1Wire ERROR"
    lwwrite Skip_rom
    lwwrite Convertt

    Waitms 20
End Sub
```



```
Sub _convertv
  lwreset ' Reset 1Wire Device
  If Err = 1 Then Print "1Wire ERROR"
  lwwrite Skip_rom
  lwwrite Convertv

  Waitms 20
End Sub

Sub _recallmemory(byval Page As Byte)
  lwreset ' Reset 1Wire Device
  If Err = 1 Then Print "1Wire ERROR"
  lwwrite Skip_rom
  lwwrite Recall_memory ' Recall Page x
  lwwrite Page
End Sub

Sub _readscratch(byval Page As Byte)
  lwreset ' Reset 1Wire Device
  If Err = 1 Then Print "1Wire ERROR"
  lwwrite Skip_rom
  lwwrite Read_scratch ' Read Scratchpad Page 0
  lwwrite Page
End Sub

Sub _writescratch(byval Page As Byte)
  lwreset ' Reset 1Wire Device
  If Err = 1 Then Print "1Wire ERROR"
  lwwrite Skip_rom
  lwwrite Write_scratch
  lwwrite Page
End Sub

Sub _copyscratch(byval Page As Byte)
  lwreset ' Reset 1Wire Device
  If Err = 1 Then Print "1Wire ERROR"
  lwwrite Skip_rom
  lwwrite Copy_scratch ' Copy Scratchpad Page x
  lwwrite Page
End Sub

Sub _clearscratch(byval Page As Byte)
  lwreset ' Reset 1Wire Device
  If Err = 1 Then Print "1Wire ERROR"
  lwwrite Skip_rom
  lwwrite Write_scratch
  lwwrite Page

  lwwrite &H00
  lwwrite &H11
  lwwrite &H22
  lwwrite &H33
  lwwrite &H44
  lwwrite &H55
  lwwrite &H66
  lwwrite &H77
End Sub
```



```
Sub Crc8_check()  
  $external _crc8  
  Dim Result As Byte  
  
  Loadaddr Scratch(1) , Z  
  ldi R24, 8  
  !Call _crc8  
  Loadaddr Result , X  
  st X, R16  
  If Result = 0 Then  
    Print " CRC-8 OK";  
  Else  
    Print " CRC-8 ERROR";  
  End If  
End Sub
```

Listing 1 Programmbeispiel RN_TEMP_MEASURE.BAS



Abbildung 13 Debug-Ausgaben

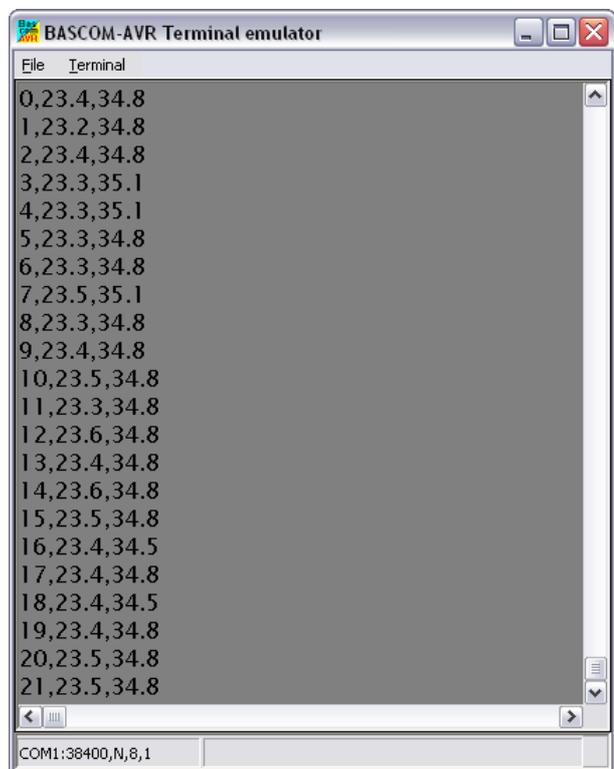


Abbildung 14 Ausgabe von Temperatur und Feuchtigkeit

Mit der vorgestellten Sensorbaugruppe wurde abschließend ein Tagesprofil von Temperatur und relativer Luftfeuchtigkeit aufgenommen. Hierzu wurde eine, wie eben beschrieben, erzeugte CSV-Datei in Microsoft Excel importiert und als grafische Darstellung aufbereitet.

Abbildung 15 zeigt das Resultat der vorgenommenen Messungen in einem Innenraum. Durch Nutzung einer Echtzeit-Uhr könnte noch ein eindeutiger Zeitbezug hergestellt werden. Hier steht nur der Index, der alle ca. fünf Sekunden nach Auslösung der Messung erhöht wird, als Abszisse zur Verfügung.

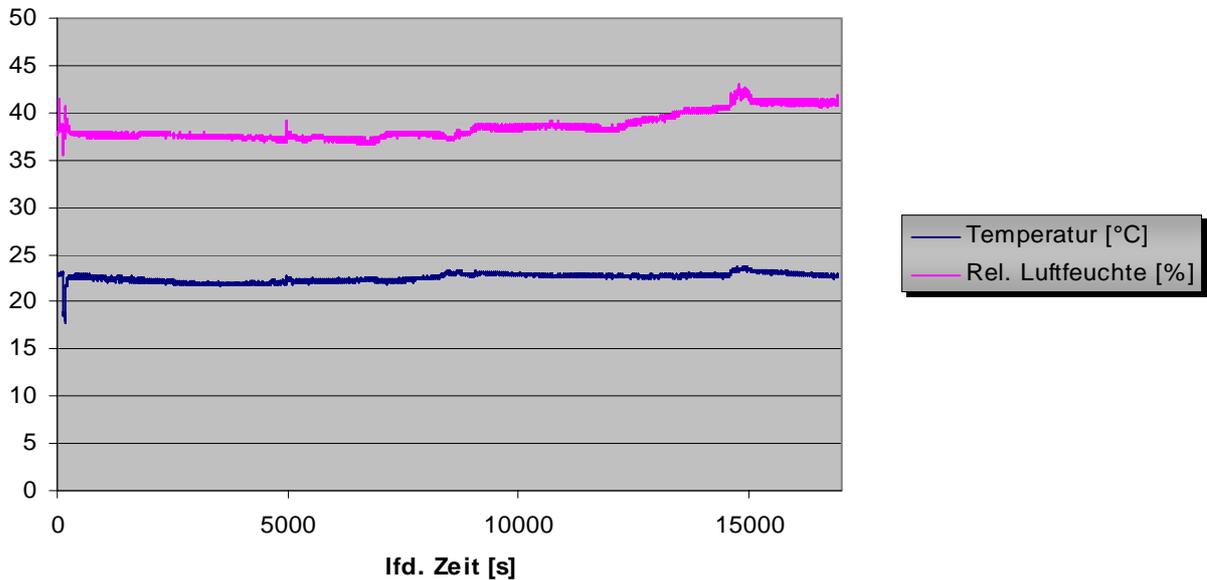


Abbildung 15 Tagesprofil von Temperatur und relativer Luftfeuchtigkeit

Abbildung 14 hatte die Ausgabe der Daten über die RS-232-Schnittstelle gezeigt. Am LCD des stAVeR-24 Entwicklungsboards zeigen sich die Messwerte in der in Abbildung 16 gezeigten Form.

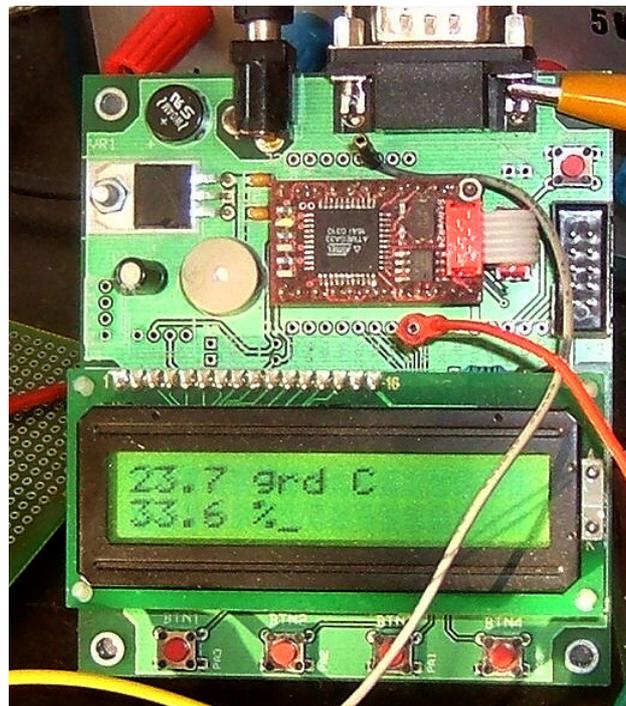


Abbildung 16 Anzeige von Temperatur und relativer Luftfeuchtigkeit

