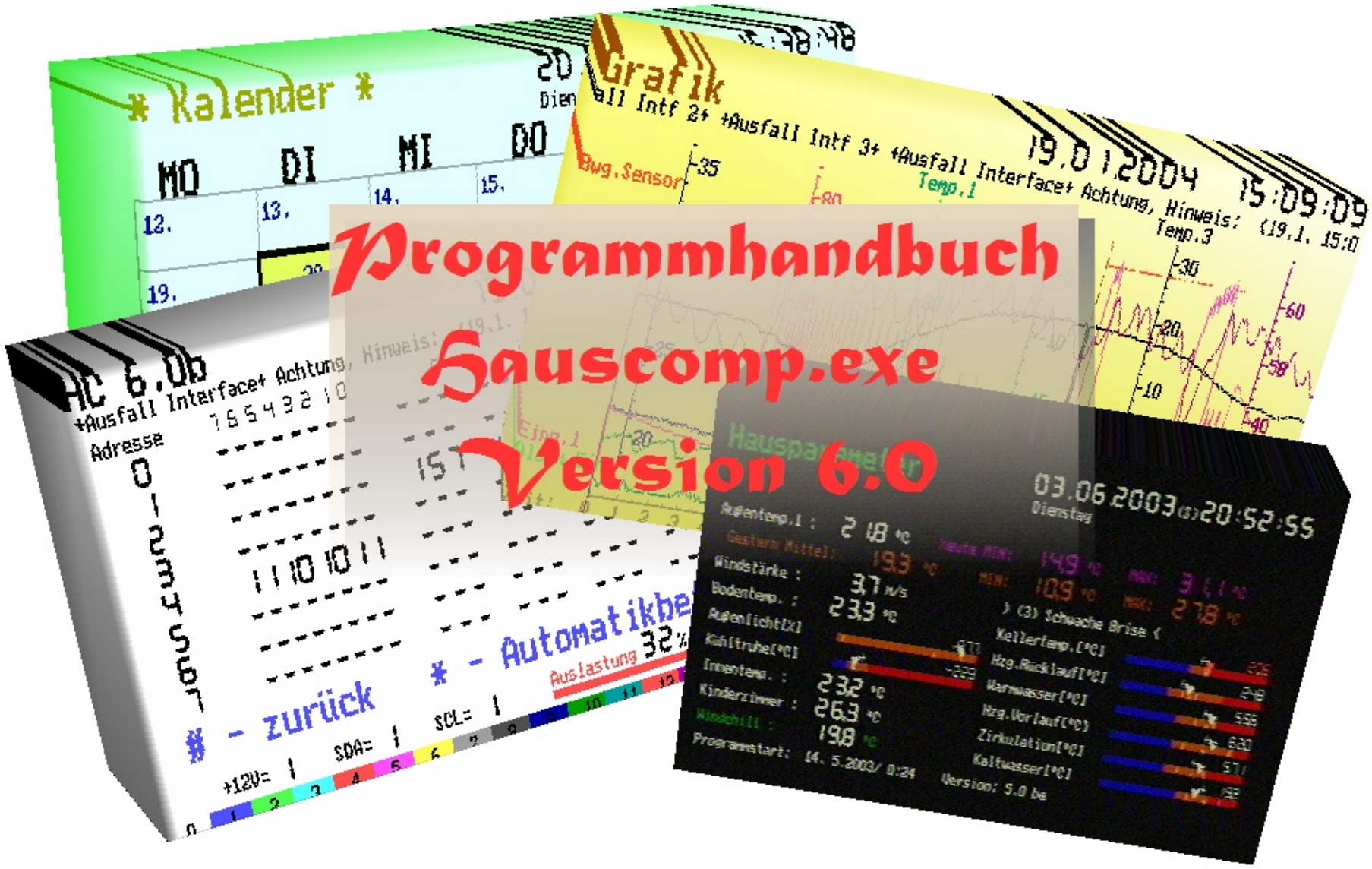


Programmhandbuch Hauscomp.exe Version 6.0

Inhaltsverzeichnis

1. Einführung....6
1.1. Systemvoraussetzungen....7
1.2. Programmstruktur/ Signalverbindungen....8
1.3. Hausbus/ I²C-Bus....9
1.3.1. Anschluss des PCs an den I²C-Bus....9
1.3.2. Die I2C-Bussequenz, Zeitschlitze (ab Version 6.0.38)....10
1.3.3. Reaktionszeiten von Sensoren und Aktoren am I²C-Bus (ab Version 6.0.38)....11
1.3.4. Mögliche Probleme am I²C-Bus/Hausbus, Störerkennung (ab Version 6.0.44)....13
1.4. Typografie und Sprachelemente in diesem Handbuch und in den ini-Dateien....14
1.5. Adressierung der I²C-Schaltkreise und ihre Zuordnung im Programm (Anzeigen, SPS)....16
1.6. Verwendung von I²C Temperatursensoren (LM75, DS1631 AM2320) und Raumsensoren....17
1.7. Die Astrofunktionen....20
1.8. Die Verwendung der Software-Beispiele....22
1.9. Besonderheiten *hauscomp.exe* unter verschiedenen Betriebssystemen....22
1.10. *Hauscomp.exe* in der „DosBox“ bzw. „Dosemu“ Dank an dieser Stelle an Herrn Höfler!....27
1.10.1. „DosBox“ für Windows....27
1.10.2. „DosBox“ für Linux (Ubuntu 9.10)....29
1.10.3. „Dosemu“ für Linux (Ubuntu 9.10)....29
1.10.4. Spezielle Linux Zusammenstellungen für das Projekt Hauscomputer....33
1.11. Netzwerkfähigkeit unter DOS....35
1.11.1. Installation DOS inklusive Netzwerk für GX1LCD (Geode von Kontron) mit Ergänzungen 2014 und 2022....35
1.11.2. Installation DOS inklusive Netzwerk für einen IGEL-3/2 (mit Realtek RTL8100B Netzwerkchip)....39
1.11.2. Installation DOS inklusive Netzwerk für einen IGEL-5/4 (mit VIA VT6102 Netzwerkchip)....39
2. Programmbedienung....40
2.1. Programmstart und Bedienung....40
2.2. Bedienung/ Tastaturbelegung....41
2.3. Die Bildschirmelemente....42
2.3.1. Der Kopfteil....43
2.3.2. Die Laufschrift....44
2.3.3. die restliche Anzeigefläche....44
2.4. Die Menü-Struktur....45
2.4.1. Der Menüauswahlmodus....45

2.4.2. Untermenüs ... 46
2.4.3. Analogsensoren-Untermenü ... 47
2.4.4. SPS-Untermenü ... 48
2.4.5. Die Tagesinfoseite ... 48
2.4.6. Funktionen-Untermenüs ... 49
2.4.7. Betriebsstunden, Energiezähler oder auch „lifing“ ... 54
2.4.8. Die Automatikseiten ... 55
2.4.9. Übersicht der Bildschirmseiten ... 56
2.4.10. Menüsteuerung ... 57
2.4.11. „versteckte“ Seiten ... 57
2.5. Die Korrektur der Uhrzeit des PCs ... 60
2.6. Sprachausgabe mit RN – SPEAK (http://www.robotikhardware.de/) ... 60
2.7. Der Echtzeitmonitor ... 60
2.8. Der Zeitkern ... 61
2.9. Die Ausfall- und Ereignisüberwachung ... 62
2.10. Die Hardwarekopplung des Signals „HINWEIS!“ ... 63
2.11. Die Syntax der Datei „taste.txt“ ... 63
2.12. Die Ordner- bzw. Verzeichnisstruktur ... 68
2.13. Zusammenarbeit hauscomp.exe mit anderen Programmen ... 69
3. Syntax der hardware.ini – Datei ... 70
3.1. Allgemeines: ... 70
3.2. Schlüsselwörter (Bei Nichtangabe werden intern die Standardwerte verwendet): ... 70
4. Syntax der hauscomp.ini - Datei ... 83
4.1. Allgemeines: ... 83
4.2. allgemeine Befehle [GLOBAL]: ... 83
4.3. Befehle zur Verwaltung der Hardware [HAUSBUS]: ... 88
4.4. Befehle zur Programmierung eigener Bildschirmanzeigen [Anzeigen]: ... 99
4.4.1. Darstellen von (beliebigen) Text auf (beliebiger) Bildschirm-Seite unter (beliebigen) SPS-Bedingungen: ... 99
4.4.2. Gestaltung von eigenen Bildschirmmenüs (ausschließlich reserviert für Bildschirmseiten 110 – 119 sowie 200 – 254) ... 99
4.4.3. Programmierung von Bildschirm- „Tasten“ in den selbst gestalteten Menüs (Softkey): (Erweiterung ab 6.0.38) ... 100
4.4.4. Programmierung einer Bildschirm- „Tastengruppe“ in den selbst gestalteten Menüs (Softkeys): (Erweiterung ab 6.0.38) ... 104
4.4.5. Gestaltung von Bildschirmanzeigen (möglich für alle Bildschirmseiten) : ... 108
4.5. Programmierung von Kalendereinträgen [KALENDER]: ... 124
4.6. Befehle zur Programmierung der Bildschirmseiten „Grafik..“ [GRAFIK]: *(letzte Änderungen: V. 6.0.46)* ... 127

4.7. Befehle zur Verwaltung der Modulmatrix (Programmierung Softwaremodule) [SPS]:....129
4.7.0. Speicherung von Betriebsstunden:....129
4.7.1. Allgemeine Befehlsstruktur für die Programmierung der SPS-Module in der Datei *hauscomp.ini*:....129
Parameterzuordnung:....130
4.7.2. Ausgänge der SPS-Module....131
4.7.2. Binäre Verknüpfungs- Module, Zähler, Speicher....132
4.7.3. Verknüpfung der SPS mit Hardware-Eingängen oder der Uhrzeit....146
Der Zustand des Raumsensors wird durch eine UND-Verknüpfung der Zustandsart mit dem Status-Byte des Raumsensors ermittelt.....155
4.7.4. Verknüpfung der SPS mit Hardware-Ausgängen....156
4.7.5. Analoge Module....162
4.7.6. Spezielle Module....173
4.7.7. Festgelegte, interne Konstanten und Variable zur freien Verwendung in der SPS....194
4.7.8. Übersicht über alle Betriebsstunden bzw. „lifing“- relevanten SPS Module und sonstige Zeit-/Speicherfunktionen....197
4.7.9. Übersicht über alle verwendeten Tastaturkodes....199
5. Einschränkungen in der Programmbenutzung / Die Lizenz.RPT / rechtliche Hinweise /Updates....200
5.1. Rechtliche Hinweise....200
5.2. Schreibgeschützte Datenträger....200
5.3. Das Jahr 2038 bzw. 2065 Problem....201
5.4. Updates....201
6. Anwendungsbeispiele....202
6.1. Der individuelle Kalender (keine zusätzliche Hardware notwendig!)....202
6.2. Das Geisterhaus …....204
6.3. Die Steuerung einer Warmwasser-Zirkulationspumpe....207
6.4. Test/ Monitoring I^2C-Hardware....209
6.5. Eine einfache Ampelsteuerung (keine Hardware notwendig)....211
6.6. Das Webinterface....213
6.7. Ein Testprogramm für die Funktionsprüfung aller Interfaces (außer ATtiny26)....215
6.8. Ein Testprogramm für die Funktionsprüfung aller Interfaces mit ATtiny26....216
6.9. Auswertung von Heizungs- oder Solardaten....218
6.10. Statistische Auswertung von Sensordaten....227
6.11. Das erweiterte Webinterface....228
6.12. Ein Testprogramm für Raumsensoren / Außensensor / Multisensoren....229
7. Fehlersuche / Korrekturen am System / bekannte Probleme / Tipps....231
8. Sprechphrasen für Sprachausgabe mit RN – SPEAK Board (nur in Version 6.0.7/8 implementiert!)....236

9. Stichwortverzeichnis..... 237

Weitere Dokumentation:

[1]: Hauscomputer Hardware Band 0 (Allgemeines, Interfaces mit ATtiny 26)
[2]: Hauscomputer Hardware Band 1 (Hausbus, Interfaces, Bussensoren)
[3]: Hauscomputer Hardware Band 2 (Sensoren, Schaltmodule)
[4]: Hauscomputer Hardware Band 3 (praktische Anwendungen)
[5]: Hauscomputer Hardware Band 4 (kombinierte Heizungssteuerung)
[7]: Hauscomputer Hardware Band 5 (I^2C-Wetterstation)
[6]: Hauscomputer und Linux

Die jeweils aktuelle Version ist verfügbar unter http://www.hauscomputer.gmxhome.de/

Zur besseren Lesbarkeit dieses Dokumentes bitte im PDF Reader (z. B. PDF-XChange Viewer) unter Ansicht „Lesezeichen“ aktivieren, man erhält dann ein zusätzliches bedienbares Paneel mit der Gliederungsübersicht.

1. Einführung

Das hier beschriebene Programm „Hauscomp.exe“ entstand aus dem Bedürfnis, gängige PC-Technik in einem Haus zur einfachen Steuerung alltäglicher Vorgänge zu nutzen. Das Programm arbeitet mit den I^2C basierten Schaltkreisen PCF8574 (8-Bit In/Out) sowie PCF8591 (4fach A/D Wandler + D/A Wandler) sowie LM75 und DS1631 (Temperatursensoren), MAX1236 (4 Kanal 12bit A/D Wandler) sowie ATtiny26 (10 bit A/D Wandler) zusammen. Es kann aber auch ohne jegliche zusätzliche Hardware als Kalender eingesetzt werden (Anwendungsbeispiel 6.1). Die eingebaute speicherprogrammierbare Steuerung (SPS) erlaubt fast beliebige Verknüpfungen von analogen und binären Signalen. Über mehrere Jahre wurde das Programm verfeinert und die Stabilität weiter erhöht. Der grundsätzliche Widerspruch zwischen Windows/DOS/Linux - basierten Programmen und der notwendigen Echtzeitfähigkeit konnte grundsätzlich gelöst werden. Durch einigen Aufwand in der Theorie der automatischen Steuerung läuft das Programm mittlerweile mit variabler Zykluszeit, automatischer Eigenkontrolle angeschlossener Hardware, automatischer Filterung von Störsignalen und gleitender Zeitkorrektur des PC-Timers. Es läuft seit 2010 sogar sparsam und schnell unter Linux mit Netzwerkfähigkeit. Die Palette der möglichen Anwendungen reicht von simplen Zeitsteuerungen (Kaffeemaschine!) über Temperaturregelungen bis zur komplexen Steuer- und Regelung von Haustechnik mit gleichzeitiger grafischer Überwachungsfunktion und umfangreichem Datenlogging. Das Programm kann somit mit einem ausgedienten PC/Laptop als Kernstück eines voll automatisierten, wartungsfreien Eigenheims verwendet werden. Es sind keine speziellen Treiber (außer für die Netzwerkfähigkeit unter DOS) notwendig. Bei mir steuert ein 386iger DX40 mit einer Flashdisk seit Jahren das Haus - weitgehend störungsfrei bei einem Eigenverbrauch des PCs von 12W. Das Programm ist Freeware, solange es nicht ununterbrochen über viele Tage genutzt werden soll.

Die vorliegende Beschreibung soll ein Angebot zur Umsetzung eigener Ideen sein. Auch wenn die Vielfalt der Möglichkeiten des Programms einen zunächst erschlägt: man kann das Programm auch ohne Programmierkenntnisse nutzen und erst später allmählich seinen Wünschen anpassen. Durch die Makrosprache ist es möglich geworden, das Programm sehr klein und damit mögliche Fehler gering zu halten – eine gute Voraussetzung für eine zuverlässige Funktion eines Hauscomputers!

Viel Spaß beim Basteln!

Dipl.-Ing. Uwe Behrndt, 2010 *E-Mail: ubehrndt@web.de*

Ergänzung März 2011:

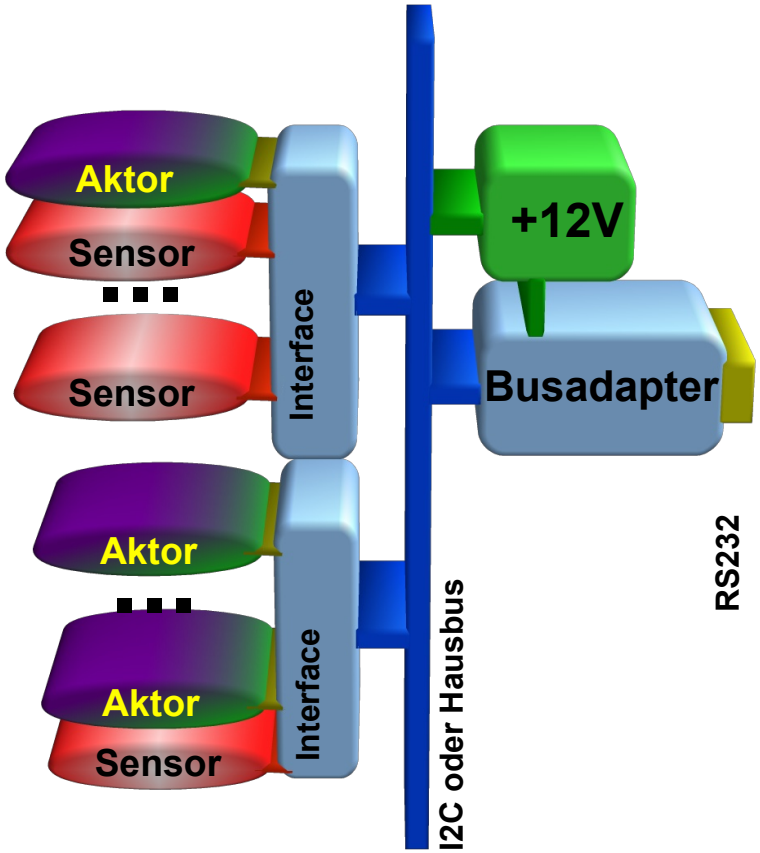
Ab Version 6.0.33 wurde die letzte technische Sperre entfernt, das Programm ist ab sofort für die nicht kommerzielle Nutzung komplett freigegeben (Freeware), zudem ist jetzt ein bootfähiges Linux-Image verfügbar, mit dem volle Netzwerkfähigkeit für unterschiedliche PCs möglich ist!

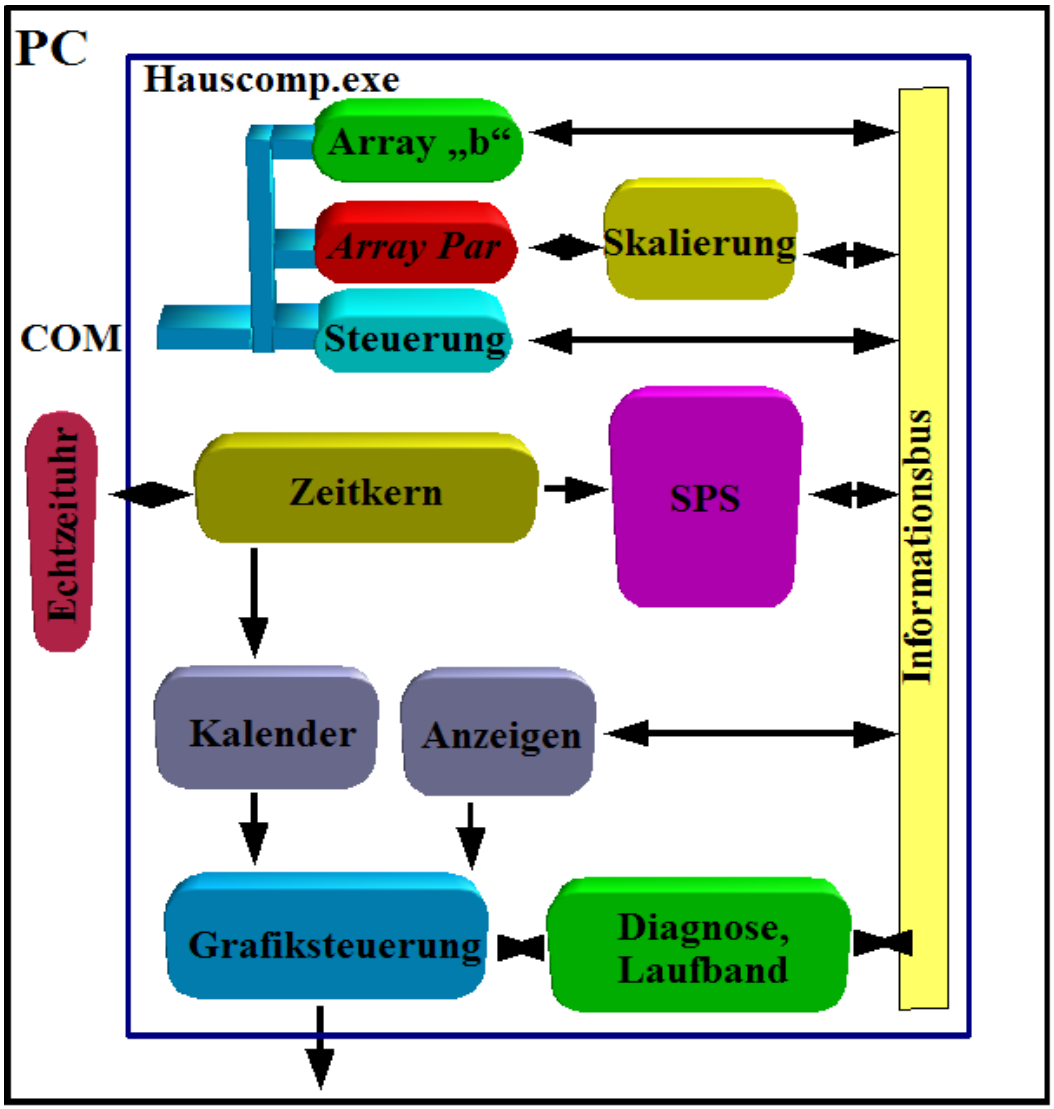
1.1. Systemvoraussetzungen

- IBM- kompatibler PC/Laptop min. 386, 40 MHz; empfohlen: 486 oder höher mit mindestens 640 kB RAM (erfüllt fast jeder Schrott – PC). Für die flüssige Nutzung **aller** Funktionen des Programms (Netzwerk, Datenlogging, Live-Modus) ist eine Geschwindigkeit ab 100 Mhz sinnvoll.
- Diskettenlaufwerk (alternativ auch bootbarer USB-Stick oder Flashdisk/Soliddisk) für die Programmdateien und das Betriebssystem, EGA/VGA kompatible Grafikkarte bzw. „shared“ Grafik;
- **1x serieller RS232-**Anschluss (im Folgenden COM genannt) für den Anschluss der notwendigen Hardware. Prinzipiell kann der serielle Anschluss auch über einen aktiven USB-RS232 Adapter realisiert werden, allerdings wird sich das Programm dabei enorm verlangsamen (hängt von der Qualität des Adapters ab) und es sind Eingriffe am Betriebssystem notwendig. An diesem seriellen Anschluss sollte entsprechende I2C- Hardware angeschlossen sein, wie z. B. PCF8574, PCF8591, LM75, DS1631, MAX1236 usw. Die Hardware Bände 0 – 4 liefern die dafür erforderlichen Infos.
- Keine Festplatte, keine Soundkarte notwendig;
- Das Programm läuft direkt unter folgenden Betriebssystemen: MS-DOS 5 oder höher, PTS-DOS o. ä., Windows 95/98, sowie W2K/XP; unter Windows kann man allerdings entweder Windows-Funktionen oder dieses Steuerprogramm nutzen. Das Programm läuft emuliert unter DOS-Box bzw. Dosemu unter fast allen Betriebssystemen. Das Programm *hauscomp.exe* ist nicht Multitasking fähig. Empfohlen wird MS-DOS (siehe auch liesmich.txt Pkt. 1b) oder Linux mit „Dosemu“ und Netzwerkfähigkeit (siehe auch 1.10.ff.).
- Das Programm läuft nicht direkt auf einem MAC sowie nicht direkt unter „Windows XP Professional x64 Edition“, „Windows Vista“ oder „Windows 8“ ff. Alternative: Linux-Version HC_V2A [6].
- Folgende Dateien sollten sich beim Programmstart im gleichen Verzeichnis wie die *hauscomp.exe** befinden:
  - *hauscomp.ini** {vom Anwender erstellte Textdatei mit Scriptbefehlen für die eigene Anwendung}
  - *hardware.ini** {vom Anwender erstellte Textdatei mit Scriptbefehlen für den Anwendungs-PC}
  - gestern.dai/gestern.dab/gestern.dar {enthält die gestrigen Grafik+Sensordaten, wird vom Programm selbst erzeugt und genutzt}
  - heute.dab/heute.dab/heute.dar {enthält die aktuellen Grafik+Sensordaten, wird vom Programm selbst erzeugt und genutzt}
  - gesternx.daz {enthält die Betriebsstunden der SPS-Module, wird vom Programm selbst erzeugt und genutzt}
  - Lizenz.rpt {Datei mit Registrierungsschlüssel; ist diese Datei schreibgeschützt, so wird der ganze Datenträger als nicht beschreibbar interpretiert; ab Version 6.0.33 wird diese Datei nicht mehr benötigt}
  - Font.abc* {Datei mit Font- Informationen für Windows XP bzw. nicht ISO- konformen Grafikadaptern; ab Version 6.0.33: ist diese Datei schreibgeschützt, so wird der ganze Datenträger als nicht beschreibbar interpretiert}
  - Hauscomp.pif {nur zum Programmstart unter WINDOWS eventuell notwendig}

  * Dateien sind zwingend notwendig

1.2. Programmstruktur/ Signalverbindungen





Zusätzlich: bei Bedarf Netzwerk (LAN/WLAN) für Monitoring bzw. Steuerung

1.3. Hausbus/ I²C-Bus

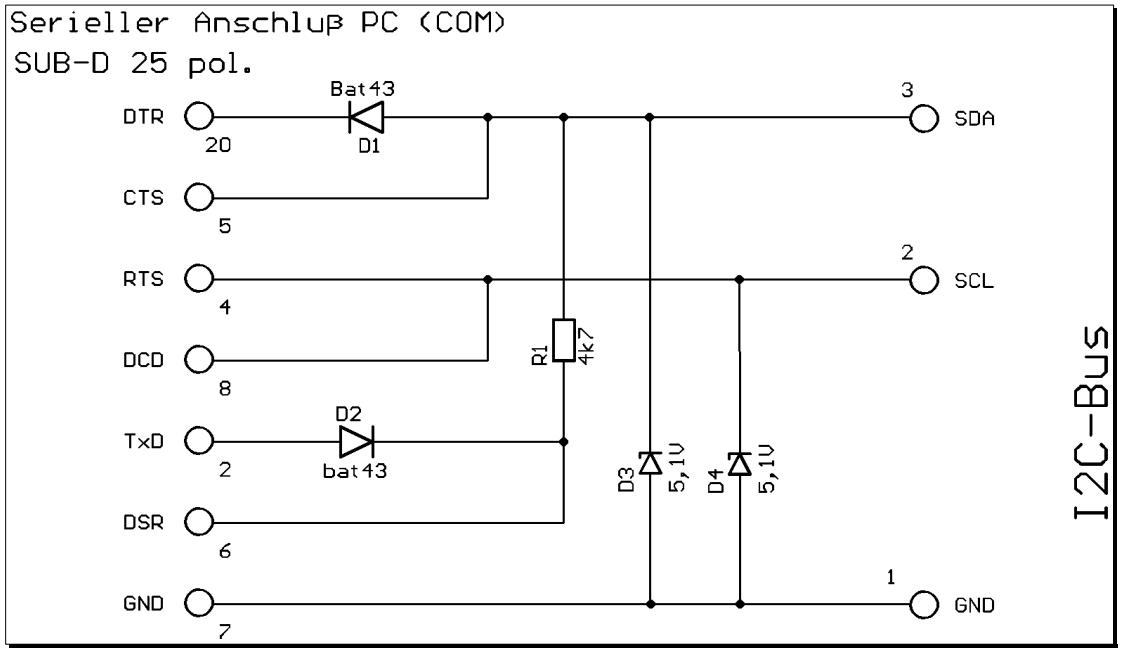
Diesem Projekt liegt der I²C Bus zugrunde. Sowohl in seiner ursprünglichen, als auch in der erweiterten Form (als Hausbus) verbindet er den steuernden PC mit der Peripherie, also den Sensoren und Aktoren. Der I²C-Bus, in den 80er Jahren von Philips erdacht, diente ursprünglich nur der platz- und kostensparenden Verbindung von Schaltkreisen auf einer Platine. Er ist einfach und robust und wurde für den Einsatz auf längere Distanzen von mir als Hausbus modifiziert (siehe auch Hauscomputer Hardware Bände 0 und 1). Die logische Struktur des I²C Busses blieb dabei im Wesentlichen erhalten, lediglich die neuen ATtiny ICs sind mittlerweile mit einem eigenen sicheren Protokoll mit Prüfsummen angebunden. Als Verbindung am PC fungiert die serielle Schnittstelle. Dabei wird hardwaremäßig auch eine Leitungsüberwachung realisiert. Da der Bustakt beim Hausbus sehr stark reduziert ist, sind spezielle Bustreiber nicht erforderlich.

1.3.1. Anschluss des PCs an den I²C-Bus

Notwendige Voraussetzung für die Nutzung des Programms ist natürlich die Kopplung des PCs mit den Peripherie-Bausteinen über den I2C-Bus. Das Programm unterstützt sowohl die einfache, nachfolgend abgebildete Ankopplung, als auch die erweiterten Versionen mit größerer Reichweite (Hardware Bände 0 und 1).

Die Zuordnung für einen seriellen COM Anschluss sind wie folgt:

Bezeichnung	*25- pol.*	*9- pol.*
DTR	20	4
CTS	5	8
RTS	4	7
DCD	8	1
TxD	2	3
DSR	6	6
GND	7	5

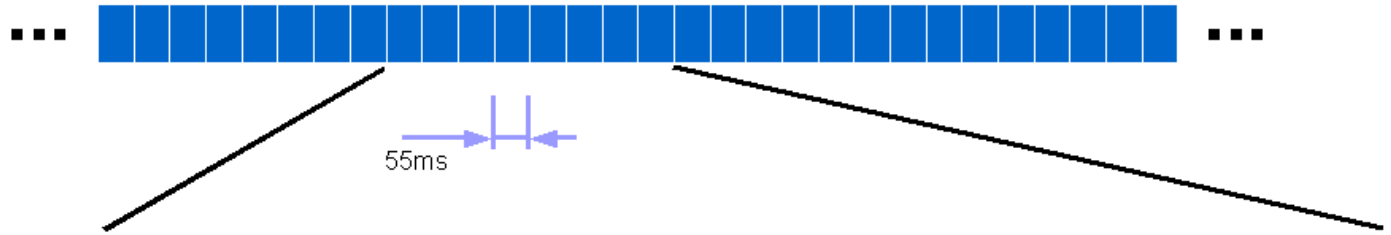


Zur Funktion: Die Z-Dioden D3 und D4 begrenzen die Spannungen am I2C-Bus auf eine ungefährliche Größe. Der Widerstand R1 ist der nur einmal vorhandene Buswiderstand. Die notwendige

Versorgungsspannung liefert TxD (max. 5 mA). Da am Bus nur ein Master (der PC) angeschlossen ist und die verwendeten Bus-IC's ohne Clock-Stretching auskommen, kann auf den zweiten Bus-Widerstand und der entsprechenden Diode an RTS verzichtet werden. An der nachfolgenden Schnittstelle können bereits vorhandene PCF8574 oder PCF8591 angeschlossen werden. Die erzielbare Buslänge liegt bei ca. 20 m. Beim Elektronikversand ELV oder der Firma Horter&Kalb kann man entsprechende Selbstbausätze ordern. Weitere, wesentlich leistungsfähigere Hardware wird in der Dokumentation „Hauscomputer-Hardware“ beschrieben. Die Signale im Einzelnen: DTR und RTS geben Infos auf den Bus; CTS liest aus der SDA- Leitung die Infos von den Slaves; DCD und DSR lesen die SCL- und +12V Leitung zum PC zurück, um Kabelunterbrechungen zu erkennen.

1.3.2. Die I2C-Bussequenz, Zeitschlitze (ab Version 6.0.38)

Die Funktion eines I^2C Busses setze ich einmal als bekannt voraus. Das Programm *hauscomp.exe* greift dabei als Master über die serielle Schnittstelle kontinuierlich auf den Bus zu. Bus-Streching, sowie andere Master sind nicht zugelassen. Jeder in sich abgeschlossene Zugriff auf einen Slave, also z. B. die Adressierung und Abfrage eines analogen Kanals eines PCF8591 erfolgt mit einem relativ geringem Bustakt im unteren kHz-Bereich (Impulsbreite einheitlich ca. 160µs) und in einem Zeitschlitz von 55 ms. Erst dieser sehr niedrige Bustakt erlaubt die imposante maximale Länge des Hausbusses von bis zu mehreren hundert Metern. Die periodisch wiederkehrende Zuordnung der Zeitschlitze erfolgt nach folgendem Muster:



Schlitz Nr.:	1	2	3	4	5	6	7	8
Zeit:	0 ms	55 ms	110 ms	165 ms	220 ms	275 ms	330 ms	385 ms
Zugriff auf:	Binäre Adresse 0	Binäre Adresse 1..x	Abfrage Dimmer, wenn vorhanden – sonst binäre Adresse 1..x	ein Kanal der analogen Adresse 0..y	Zugriff Dimmer, wenn Anfrage – sonst binäre Adresse 0	Binäre Adresse 1..x	ein Kanal der analogen Adresse 0..z (2. Satz), wenn vorhanden – sonst binäre Adresse 1..x	ein Kanal der analogen Adresse 0..y

Zusätzlich zu der Belegung der Zeitschlitze gibt es eine sogenannte „Tunnelung“. Hierbei werden Anforderungen vom SPS Modultyp 34 im aktuellen und im folgenden Zeitschlitz ausgegeben. Die regulären Zeitschlitze verschieben sich entsprechend.

1.3.3. Reaktionszeiten von Sensoren und Aktoren am I²C-Bus (ab Version 6.0.38)

Die im vorangegangenen Abschnitt beschriebene Bussequenz organisiert den sinnvollen zeitlichen Zugriff der einzelnen Treiber auf den Hausbus. Das Auslesen von Analogsensoren kann dabei natürlich in größeren Zeitabständen erfolgen, als z. B. das Einschalten einer Lampe nach Tastendruck. Wenn kein Dimmer und kein zweiter Satz analoger Sensoren vorhanden sind (diese Möglichkeiten gibt es nur in der Industrieversion) ergeben sich aus der Bussequenz folgende Reaktionszeiten der internen Lese- bzw. Schreibroutinen:

- **Auslesen eines Analogsensors:** **max. 8,8 s** (8 Adressen zu je 4x IN und 1x OUT =40 Kanäle in 40 * 220ms)
- **Schreiben oder Lesen eines Binärkanales (Adresse 0):** **max. 0,**2 s (1 Adresse aller 220 ms)
- **Schreiben oder Lesen eines Binärkanales (Adresse 1..7):** **max. 0,9 s** (8 Adressen in 440 ms / 4 Zugriffe)

Zusätzlich zu dieser Reaktionszeit müssen noch die Zeiten der „Anfrage-Formulierung“ durch das Hauptprogramm bzw. der SPS berücksichtigt werden.

Bis zur Version 6.0.38 betrugen die Reaktionszeiten für die Binärkanäle 0 bzw. 1–7: 0,1 s und 1,6 s. Das Controlinterface (Pkt. 8 in [2]) reagiert empfindlich auf die nun geringere Abfragefrequenz der Adresse null, was zu einem Flackern der Status LED führt. Der Kondensator C4 muss deshalb auf 10µF erhöht werden. Alternativ kann man den alten Zustand über den Scriptbefehl „Bussequenz=alt“ wieder herstellen. Allerdings dann ohne die neuen Funktionen.

Für besonders zeitkritische Anwendungen stehen die Befehle „Anzahl_PCF8591=“ bzw. „Anzahl_PCF8574P=“ zur Verfügung. Mit ihrer Hilfe lässt sich die Busabfrage auf wenige Schaltkreise reduzieren, was natürlich entsprechend die Reaktionszeit verkürzt aber auch die Zahl der verfügbaren Adressen und Kanäle schmälert. Ab Version 6.0.49 werden alle nicht in der *hauscomp.ini* initialisierten analogen Kanäle automatisch übersprungen, was den Scriptbefehl „Anzahl_PCF8591=“ mittlerweile überflüssig macht.

Ab Version 6.0.51 wird der Zeitschlitz Nr. 5 für Analogkanäle zur Verfügung gestellt und nicht mehr dem Binärkanal 0 zugeordnet, wenn außer dem Kanal 3 (Signal Hinweis!) kein weiterer Kanal initialisiert wurde, es also kein Controlinterface o. ä. gibt. Das reduziert die maximale Zugriffszeit für das Auslesen eines Analogsensors bzw. die Ansteuerung eines D/A Ausganges von 9 auf 6 s verdoppelt aber natürlich das Schreiben und Lesen eines Binärkanals der Adresse 0 von 0,2 auf 0,4 s.

Ab Version 6.0.51 gibt es die Möglichkeit mit dem Scriptbefehl „Vorrang01=..“, die analogen Adressen 0 und 1, sowie alle Raumsensoren in der Hälfte der Zeit abzufragen. Das geht dann aber zulasten der restlichen analogen Adressen. Bei einem voll ausgebauten System (40 Sensoren, 8 analoge Ausgänge) mit einem Stromsensor an den Adressen 0 und 1 kann die Abfrage je Kanal so von max. 8,8 auf ca. 4s (wenn kein Controlinterface vorhanden ist, sogar auf max. 3s) reduziert werden. Die Abfrage der restlichen analogen Adressen erhöht sich dann auf max. 18s (max. 13s ohne Controlinterface).

Eine weitere Voraussetzung für eine schnelle Steuerung ist eine kurze Zykluszeit der SPS. Die gesamte Zykluszeit setzt sich aus der relativ schnellen Abarbeitungszeit der SPS für einen kompletten Durchlauf aller SPS Module, sowie der Zeit zusammen, die für die grafische Ausgabe benötigt wird. Diese Zykluszeit hängt also vom verwendeten Rechner und der Komplexität der programmierten Grafik ab und sollte möglichst unter 1,5 Sekunden liegen. Lediglich der Bildaufbau unter „Grafik a&b“ bzw. „Grafik analog“ kann mehrere Sekunden dauern. Praktische Beispiele sind tabellarisch im Abschnitt 1.9. gelistet.

Die aktuell gemessene Zykluszeit ist übrigens als Analogwert auch im internen SPS-Modul Nr. 19 zu finden. Zudem liefert das interne SPS-Modul 54 die Anzahl der Zyklen bzw. Programmschleifen je Minute.

Einen Anhaltswert für geringe grafische Belastungen liefert der Menüpunkt „System/Zykluszeit“, der ausschließlich diesen Analogwert ständig aktualisiert anzeigt. Das nebenstehende Bild entstand unter Windows XP mit einem Intel Pentium-M mit 2 GHz Taktfrequenz.



1.3.4. Mögliche Probleme am I²C-Bus/Hausbus, Störerkennung (ab Version 6.0.44)

Der I2C-Bus wird in diesem Projekt extrem langsam getaktet um erst ein langes Leitungsnetz möglich zu machen. Aber es gibt auch Grenzen. Der Bus kann praktisch unter folgenden Bedingungen Probleme machen:

- Ein Slave kann den SDA-Pegel nicht ausreichend gegen Masse ziehen, die Daten dieses Slaves gehen verloren.
  - Ursache: 1. Fehlerhafte Erkennung des Pegelwechsels bei zu niedrigen Spannungen (hängt von der seriellen Schnittstelle ab); 2. Der Spannungsabfall über der SDA-Leitung ist zu groß.
  - Abhilfe: Verwendung des Busadapters „Deluxe“ (Hardware Band 0). Damit wird der Pegelwechsel beim einem höheren Schwellwert um +3V erkannt. Eine weitere Möglichkeit ist das Parallelschalten mehrerer Adern für die SDA-Leitung.
- Ein Slave zieht den SDA- oder SCL-Pegel irgendwann dauerhaft gegen Masse und blockiert so den kompletten Bus, alle Daten gehen verloren.
  - Ursache: Fehlerhaftes Design des Slaves (Beispiel: LM75).
  - Erkennung: Im Hardware-Status sichtbar; das Signal „Busproblem“ erscheint dauerhaft.
  - Abhilfe: Die einzige Möglichkeit ist, auf derartige Hardware zu verzichten!
- Wegen zu großer Leitungskapazität gehen Daten verloren.
  - Ursache: Bei Leitungslängen über 250 m (CAT-5) bzw. 150 m (normale verdrillte Leitungen) beträgt die Leitungskapazität einer Ader ca. 15nF gegen Masse, die Daten-Impulse werden bereits sichtbar verschliffen. Der Bus bleibt aber funktionsfähig bis etwa 33nF (bei 160µs Impulsbreite) unter Verwendung des Busadapters „Deluxe“ (Hardware Band 0).
  - Erkennung: Slaves/Interfaces mit Attiny melden sich öfter nicht (Hardwareübersicht, entsprechende rote Balken sichtbar).
- Einkopplung von Störimpulsen auf die SDA und/oder SCL Leitungen.
  - Ursache: Hochfrequente Störungen von z. B. „chinesischen“ Schaltnetzteilen oder Lampen.
  - Erkennung: Das Signal „Busproblem“ wird häufig ausgelöst (als „Problem“ in der Laufschrift, kleines rotes Kreuz bzw. ausgelöstem internen Modul 59).
  - Abhilfe: Schaltnetzteil ersetzen, Lampen entstören, Verkabelung anders legen. abgeschirmtes Kabel für den Bus verwenden.
- Ausfall der +12V Stromversorgung.
  - Ursache: Ausfall Netzteil, fehlerhafter Slave/Kurzschluss, Kabelproblem.
  - Erkennung: im Hardware-Status sichtbar.

Das Signal „Busproblem“ kann man auch in der Grafik (auch in Echtzeitgrafik) sichtbar machen, um einen Verlauf über die Zeit zu sehen:

graph_b= [1..16] 8 59 4 /Busproblem

1.4. Typografie und Sprachelemente in diesem Handbuch und in den ini-Dateien

Beschreibung

Das vorliegende Handbuch verwendet einheitliche Schreibweisen (Typografie) und Darstellungsarten. Zur Verdeutlichung von Zusammenhängen im Handbuch werden verschiedene Schriftarten und Elemente verwendet:

`Schriftart steht für Definitionen oder_Benutzereingaben;`

Schriftart steht für reservierte Scriptbefehle (Schlüsselwörter);

Schriftart steht für Script-Beispiele.

[] - Parameter

{[]} - Parameter kann entfallen

`{rot, grün}` - Beispiele für die Syntax

_ - in der Befehlssyntax (mit orange-brauner Hintergrundfarbe geschrieben) als Tiefstrich dargestellt bedeutet, an der entsprechenden Stelle in der Befehlszeile muss mindestens ein Leerzeichen oder ein Tabulatorzeichen als Trennzeichen (!) vorhanden sein.

0..7 - zulässiger Wertebereich sind ganze Zahlen im Bereich 0,1,2..7

Real - Zahlenwert stellt eine Gleitkommazahl dar, die Genauigkeit beträgt ca. 7 Dezimalstellen (1538.3452; 5E2; -2; +0;1.4 usw. sind gültige Werte), Komma wird als Punkt geschrieben!
Wertebereich: ± $2{,}9 \cdot 10^{-39} .. 1{,}7 \cdot 10^{38}$

Integer - Zahlenwert ist eine ganze Zahl im Bereich -32768 bis 32767

Byte - Zahlenwert ist eine ganze Zahl im Bereich 0 bis 255

String [20] - Zeichenkette mit maximal 20 Zeichen lang, ohne Apostroph. Es dürfen allerdings keine Leerzeichen enthalten sein. z. B. ist „Lüfter_I“ zulässig, „Lüfter II“ aber nicht)

Führende nullen können grundsätzlich weggelassen werden. Ebenso das Pluszeichen „+“. Manchmal trägt es aber zur Übersichtlichkeit bei, wenn sie trotzdem angegeben werden. Werden im weiteren Standardwerte angegeben, so beziehen sie sich auf die interne Voreinstellung, falls der betroffene Befehl nicht programmiert wird.

*.ini Dateien

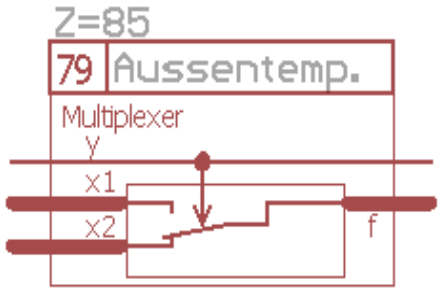
Die Programmierung des Hauscomputers (*hauscomp.exe* + Hardware) erfolgt über Scriptbefehle in *.ini Dateien, deren Syntax im Folgenden erläutert wird. Diese Dateien müssen mit gängigen Textbearbeitungsprogrammen (z. B. Notepad++, Totalcommander) bearbeitet und als unformatierter Text gespeichert werden.

Alle Schlüsselwörter müssen linksbündig mit unmittelbar folgendem Gleichheitszeichen geschrieben werden. **Dabei ist unbedingt die Groß-**

Kleinschreibung zu beachten! *Die Abarbeitung der Befehle im Programm erfolgt zeilenweise (**max. 80 Zeichen je Zeile!**) von oben nach unten. Ungültige Zeilen wie Kommentare, Zwischenüberschriften usw. sollten mit Leerzeichen, „#“, „/“, „*“, „;“ oder ähnlichem beginnen.* Als **Trennzeichen** zwischen Parametern wird das **Leerzeichen** verwendet. Ab Version 6.0.29 ist auch der **Tabulator** zulässig. Einzelne Sektionen wie z. B. [GLOBAL] dienen damit nur der Übersichtlichkeit, sie werden vom Programm ignoriert.

SPS

Bei der Auflistung der SPS-Befehle bzw. Module sind Zeichnungen eingefügt, die die Funktion verdeutlichen sollen. Diese grafische Form der SPS-Darstellung ist an die Funktionsbausteinsprache (EN 61131–1) angelehnt und kann mithilfe der Bibliothek „..\Eagle_Projekte\lbr\hauscomp.lbr“ unter EAGLE für die Dokumentation eigener Projekte genutzt werden. Leider ist eine direkte Umwandlung vom Funktionsbausteinplan in die entsprechenden Scriptzeilen für die *hauscomp.ini* nicht möglich.



Oben steht die SPS-Modulnummer, im Rechteck links der Modultyp und rechts daneben die Modulbezeichnung. Dünne Signallinien kennzeichnen Zustände (EIN/AUS), dickere – analoge Signale. Direkt vorgegebene Parameter sind innerhalb des Moduls dargestellt. Indirekt programmierbare Parameter haben entsprechende Anschlüsse nach außerhalb.

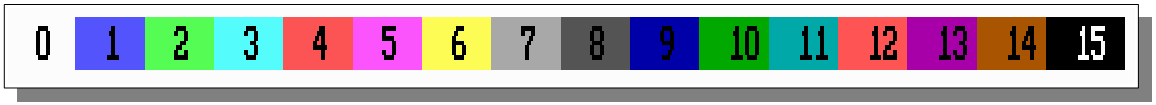
Grafik

Einige grafische Beispiele in dieser Beschreibung wurden als Hardcopy vom Bildschirm übernommen. Normalerweise ist der Hintergrund schwarz, zur besseren Darstellung wurde allerdings vor der Hardcopy die Taste <i> (Invertierung) betätigt. Die üblichen Farben ohne Invertierung (dunkler Hintergrund) auf dem Bildschirm sind folgende:



Hinweis: Die Farbe 12 blinkt rot.

Mit Invertierung (weißer Hintergrund):



Bei Wahl der Hintergrundfarbe „hellgrau“ gibt es einen Kompromiss zwischen beiden Varianten: Die Farben werden normal, also nicht invertiert angezeigt, allerdings sind Schwarz und Weiß vertauscht → also Darstellung auf hellem Hintergrund ohne Farbverfälschungen.

1.5. Adressierung der I²C-Schaltkreise und ihre Zuordnung im Programm (Anzeigen, SPS)

Am I²C-Bus melden sich alle Schaltkreise mit einer bestimmten Adresse, die sich aus einem Typ-abhängigen Anteil und einem, mittels Jumper festlegbaren Adressanteil zusammensetzen. Beim E/A Schaltkreis PCF8574 (feste Adresse = 32) z. B. sieht das z. B. folgendermaßen aus:

S	0	1	0	0	A2	A1	A0	0	A

S: Startbedingung, A: Acknowledge, A0-A2: Adressjumper auf der Leiterplatte.

Zusätzlich zu den üblichen I²C-Schaltkreisen werden auch selbst programmierte ATtiny-ICs eingesetzt. Sie verwenden zusätzlich im Datentelegramm eine Prüfsumme. Das verbessert die Zuverlässigkeit des Systems.

Im Programm *hauscomp.exe* wird diese komplizierte Technik stark vereinfacht:

Als **Adressen** der binären PCF8574 werden für die maximal 8, mit Jumper adressierbaren IC's **0..7** verwendet. Gleiches gilt für die analogen PCF8591 und alle Attiny_xx. Demzufolge gibt es z. B. die Adresse 4 zweimal: einmal als analoge und einmal als binäre Adresse. Da im vorliegenden Programm je nach Adresstyp verschiedene Script Befehle zur Anwendung kommen, sollte es keine Verwechslungen geben.

Systemfremde Schaltkreistypen können vom Programm *hauscomp.exe* über den SPS Modultyp 34 angesprochen werden.

Jeder Schaltkreis hat mehrere Ein- bzw. Ausgänge. Diese Leitungen werden **Kanäle** genannt. Beim PCF8574 gibt es dementsprechend die Kanäle 0..7. Beim PCF8591/ATtiny bzw. MAX1236 die Kanäle 0..3 für die 4 analogen Messeingänge und Kanal 4 für den analogen Ausgang (nur PCF8591/ATtiny).

Sowohl die Anzeigen, als auch die SPS verwenden diese einheitlichen Adressen:

- binär: Adr 0, Kan 0 ... Adr 7, Kan 7
- analog: Adr 0, Kan 0 ... Adr 7, Kan 4
- Raumsensoren: Subadr. 0 .. Subadr. 15

Die PCF8574 werden entsprechend der Bussequenz regelmäßig oft abgefragt. Die Adresse 0 ist dabei eine besondere: Hier werden auch Systemereignisse aufgeschaltet. Näheres dazu siehe auch Abschnitt 2.10.

Neben der binären PCF8574P wurden auch ATtiny26 mit entsprechender Firmware implementiert. Die Adressierung erfolgt dort sinngemäß zu den PCFs. Die zugehörigen Treiber werden automatisch je nach Konfiguration in der *hauscomp.ini* ausgewählt.

1.6. Verwendung von I²C Temperatursensoren (LM75, DS1631 AM2320) und Raumsensoren

Gegenüber analogen Temperatursensoren, die über einen Operationsverstärker an einen analogen Messeingang PCF8591 angeschlossen werden müssen, erlauben einige Temperatursensoren wie zum Beispiel der LM75 einen direkten Anschluss am I2C-Bus. Das Programm *hauscomp.exe* unterstützt dabei den LM75 (9 Bit) und den DS1631 (wird mit 12 Bit initialisiert). Diese Schaltkreise (im weiteren Bussensoren genannt) besitzen einen Thermostat Ausgang, er wird aber in diesem Projekt nicht verwendet.

Vorteile von Bussensoren:

- Geringerer Hardwareaufwand
- Keine Kalibrierung notwendigen – Genauigkeit ist besser ±2°C (je nach Temperatur sowie Spannungsversorgung)
- Höhere Auflösung: LM75 – 0,5 °C Ds1631 – 0,0625 °C
- Großer Temperaturmessbereich von -55 °C bis +125 °C

Nachteile von Bussensoren:

- Da die Sensoren die gleiche feste Adresse wie die PCF8591 haben, blockiert jeder Temperatursensor eine entsprechende Adresse. Anstelle von 4 analogen Eingängen und 1 analogen Ausgang beim PCF8591 kann also nur 1 Bussensor (z. B. DS1631) abgefragt werden.
- Die Bussensoren sind bei Anbindung an den I²C Bus thermisch selbst aktiv, erzeugen also geringfügig Wärme und verfälschen daher die gemessene Temperatur um bis zu 2 °C. Zudem ist der thermische Widerstand des Gehäuses nicht optimal. Daher sollte der zu messende Körper in direkten Kontakt mit dem Gehäuse gebracht werden.
- Der Hausbus kann wegen der Pegeldifferenz nicht unmittelbar an die Bussensoren angeschlossen werden, es sind weitere Bauelemente notwendig.
- Es wurde beim LM75 ein gelegentliches „Hängenbleiben“ des „Bus Traffics“ beobachtet: mit einem entsprechenden Totalausfall des Busses. Er wird daher in der Praxis des Projektes Hauscomputer nicht mehr eingesetzt.

Einbindung der Bussensoren in ein Projekt:

Praktisch ist die Einbindung der I2C Bussensoren sehr einfach. Mit dem Befehl „Bussensor=“ in der *hauscomp.ini* wird der entsprechende Schaltkreis im Programm angemeldet. Die weitere Verwendung der gemessenen Temperatur erfolgt genauso wie bei einem analogen Sensor: mit der entsprechenden analogen Adresse. Der analoge Kanal ist dabei immer 0 (es gibt ja auch nur einen). Als Bonbon kann über den genannten Befehl der Temperatursensor präzise geeicht werden. Durch Angabe eines Korrektur-Wertes wird der gemessene Temperaturwert entsprechend verändert. Für die Fehlersuche und Inbetriebnahme wird aber auch weiterhin in der Menüseite 018 (Bus-Anzeige) immer der original gemessene Wert vom Bussensor angezeigt!

Als Beispiel sollen je zwei LM75 und DS1631 im Programm angemeldet werden.

Zunächst die *hardware.ini*:

Es erfolgen die üblichen Eintragungen. Die Anzahl der analogen Bus-IC's incl. Bussensoren wird hier im Beispiel auf 5 begrenzt.

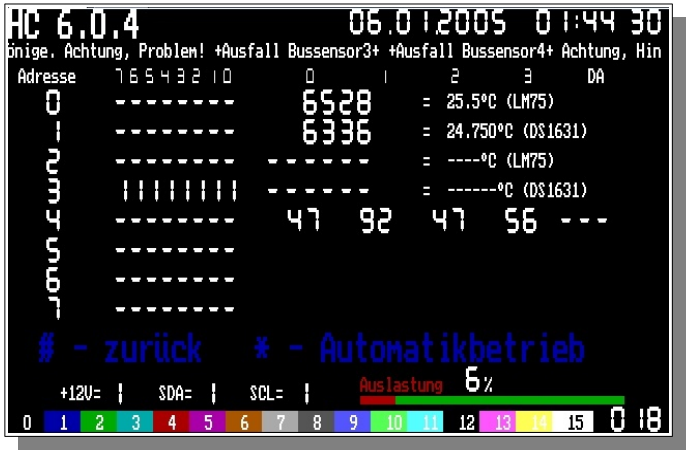
```
COM=2
Sommerzeit=off
Laufschrift=on
Anzahl_PCF8591=5
```

Einträge in der *hauscomp.ini*:

Die Zeile „Test=95“ startet das Programm mit der Seite 95 ohne automatische Umschaltung. Die Zeile „Sensor=“ ist eingerückt und damit inaktiv. Die 4 Zeilen „Anzeige=“ organisieren die Anzeige der gemessenen Temperaturen auf der Hausparameterseite 95. Die Einbindung der Sensoren erfolgt mit den Zeilen „Bussensor=“ auf die Adressen 0, 1, 2, 3. Der erste LM75 ist mit einem Korrekturwert von -1,5°C versehen. Die Schaltkreisbezeichnungen aller analogen IC's und damit auch der Bussensoren werden in der letzten Zeile definiert.

```
Norm=ANSI
Test=95
 Sensor=test          Einh 1 0  30    0     0       255  255
Anzeige=95 2           0 0   1 1   0 0 -20 +350 0  ;test
Anzeige=95 0           1 0   1 2   0 0 -20 +350 0  ;test
Anzeige=95 2           2 0   1 3   0 0 -20 +350 0  ;test
Anzeige=95 0           3 0   1 4   0 0 -20 +350 0  ;test
graph_a=0              0 0   9   0 +350
Bussensor=LM75/1   °C 0 2 44 -1.5
Bussensor=DS1631/1 °C 1 3 44  0
Bussensor=LM75/2   °C 2 2 44  0
Bussensor=DS1631/2 °C 3 3 44  0
PCF8591=Bussensor1 Bussensor2 Bussensor3 Bussensor4 PCF8591/4 PCF8591/5
```





Die Screenshots der Hausparameterseite und der Testseite zeigen das Ergebnis obiger ini- Dateien an einer realen Testumgebung mit je einem LM75 und DS1631 und einem PCF8591 auf Adresse 4. Die Temperaturauflösung des DS1631 ist frappierend hoch.

Temperatur- und Feuchtesensor AM2320/2321:

Der Sensor AM2320 misst die Lufttemperatur im Bereich -40..+80°C mit einer Genauigkeit von +/- 0,5°C und einer Auflösung von 0,1°C, sowie die Luftfeuchtigkeit im Bereich 0..99,9%RH mit einer Genauigkeit von +/- 3%RH und einer Auflösung von 0,1%RH. Der Treiber für diesen Sensor ist automatisch aktiviert, wenn an der I2C-Adresse 0B8h kein anderes Bauteil initialisiert wird (z. B. Binär26=xxxx 4 ..). Die übertragenen Daten sind per CRC-16 (Modbus mit Polynom=0A001h) gesichert.

Die gültigen Temperatur- und Feuchtigkeitswerte sind unter den internen SPS-Modulen 57 und 58 abrufbar.

Raumsensor, Außensensor, Multisensor (ab Version 6.0.46):

Es können bis zu 16 Multisensor-Baugruppen (und damit bis zu 100 Einzelsensoren/Aktoren) am Bus angeschlossen werden. Sie messen ohne zusätzliche Kalibrierung verschiedene Gasparameter in einem Raum. Die Erkennung und die Abfrage erfolgen vollautomatisch, sie müssen daher nicht initialisiert werden. Es existiert aber der Scriptbefehl „Raumsensor=“ um einen exklusiven Namen und eine Kalibrierung dem jeweiligen Sensor zuzuweisen. Ist ein Interface mit MAX1236 initialisiert, wird der, sich auf der gleichen Adresse befindende Multisensor, automatisch deaktiviert. Als Temperatur/Feuchte Sensoren sind dabei sowohl AM232x, als auch SHT3x (ab Firmware M2.0) verwendbar.

Die Abfrage der Raumsensoren erfolgt im Treiber zeitlich zusammen mit der Abfrage von analogen Sensoren. Sind mehr analoge Sensoren initialisiert, so ist das Abfrageintervall der Raumsensoren entsprechend größer. Bei 35 analogen Sensoren und analogen Ausgängen beträgt das Abfrageintervall der Raumsensoren z. B. 65 s bis 90 s, je nach Anzahl der Raumsensoren. Ist dagegen kein analoger Sensor implementiert, so liegt das Intervall bei 7-12 s. Da die Gasparameter in einem Raum sich nur langsam ändern, sind diese Abfragezeiten sicher akzeptabel, zudem beträgt das interne Abfrageintervall des T/F Sensors in der Firmware der Raumsensoren (ab M1.1) ca. 30 s bzw. (ab M2.0) ca. 24 s. Ab Version 6.0.53 konnte das Abfrageintervall der Raumsensoren in der hauscomp.exe nochmal halbiert werden, sodass die Werte spürbar schneller aktualisiert werden.

Für eine unmittelbare Anzeige der verschiedenen Gasparameter stehen die Anzeigetypen 25 und 28 mit der Adresse 9 zur Verfügung. Per SPS-Modultyp 105 und 106 können die gemessenen Werte in der SPS gespiegelt und weiter verarbeitet werden.

Für die Aktivierung verschiedener Zustände in den Raumsensoren (Relais, Hinweis, Alarm, Zustand „LAMP“) sind die SPS-Modultypen 30 und 31 (Binärkanal schalten) vorbereitet. Die Ansteuerung bzw. Übertragung der Schaltbefehle an die Raumsensoren erfolgt bei Änderungen in der SPS zunächst direkt per „Tunnelung“, dann aber (bei fehlerhafter Übertragung) unmittelbar nach der nächsten planmäßigen Abfrage des Raumsensors. Da bei der Methode der „Tunnelung“ immer nur ein Busteilnehmer angesprochen wird, kann die Ausgabe z. B. des Signals „Alarm“ an mehrere Raumsensoren (bzw. die Löschung des Signals) durchaus >2 min dauern.

1.7. Die Astrofunktionen

Das Programm errechnet im laufenden Betrieb eine Reihe von astronomischen Daten. Dazu notwendig sind lediglich die korrekte Uhrzeit und die Vorgabe der Standortkoordinaten mit den Scriptbefehlen „Geo=“ sowie „NN=“ in der *hardware.ini*. Die Bios-Uhr des PCs muss daher zwingend das ganze Jahr über die mitteleuropäische Zeit bereitstellen, d. h. ohne Umstellung auf die Sommerzeit, da das Programm keinerlei Möglichkeit hat den Unterschied zu erkennen. Demzufolge schreibt man in die *hardware.ini* „Sommerzeit=auto“ um die automatische Umschaltung auf die Sommerzeit (nur der Anzeige, nicht aber der BIOS- Uhr!) zu aktivieren.

Die Astrofunktionen bestehen aus folgenden Elementen: Der Seite „Tagesinfo“, dem „Sternenhimmel“ im Untermenü „Grafik“, dem Vollmondzeichen im Kalender und einigen laufend errechneten frei verwendbaren SPS-Parametern. Die Genauigkeit ist relativ hoch: ein Vergleich der Daten mit http://www.calsky.com/ ergab eine Abweichung kleiner 2 min für Sonnenauf- bzw. Sonnenuntergang. Auf der Seite „sonstiges“ im Untermenü „Funktionen“ sind zusätzlich einige laufend errechnete Werte gelistet. Die Darstellung des Sternenhimmels mit den hellsten Fixsternen und Planeten sollte der Orientierung für Laien dienen, während die Angabe der lokalen Ortssternzeit „Local Apparent Sidereal Time“ (LAST) für die Ausrichtung von Fernrohren nach Sternkarten verwendet werden kann. Ein Dank dafür an http://www.ottmarlabonde.de/L1/Astr06.html und alle Arcussinusfreaks im Internet.



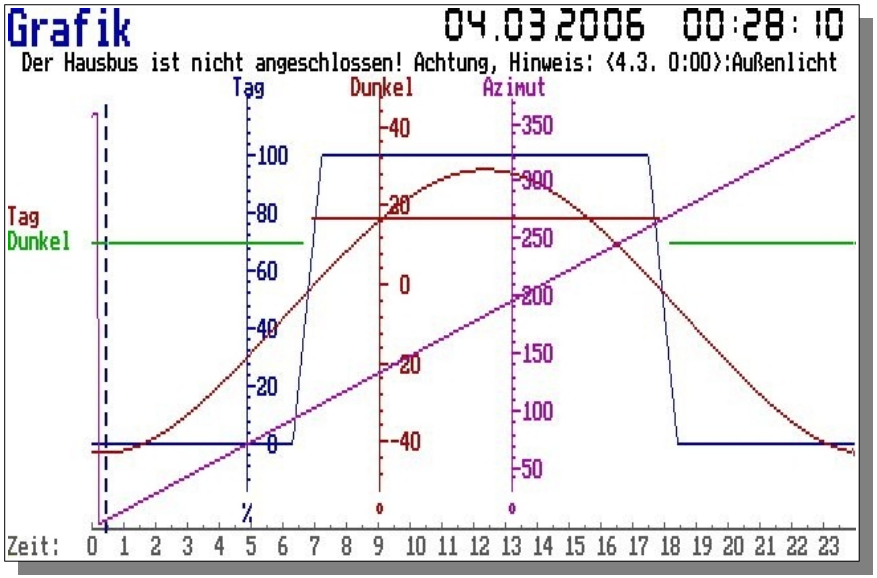
Die Energieintensität und der UV-Index werden in Abhängigkeit von der aktuellen Sonnenhöhe sowie der Höhe des Standortes über dem Meeresspiegel nach folgender Formel errechnet:

$$Intensität = e^{k1 - k\frac{(1 - H_{[km]}/k2)}{\sin(h)}}$$

h: Sonnenhöhe über Horizont; H: Höhe über NN; k, k1, k2: Konstanten

Nicht berücksichtigt werden die aktuelle Bewölkung sowie die schwankende Dicke von Atmosphäre und Ozonschicht. Die errechneten Werte stellen also Höchstwerte dar, die für Mitteleuropa optimiert sind. Die Energieintensität ist dabei die, aus der Solarkonstanten (1,367 kW/m²) abgeleitete Leistung, die die Erdoberfläche je nach Sonnenhöhe erreicht. Sie kann mit tatsächlich gemessenen Leistungen (nicht Energiemengen!)

von Solaranlagen verglichen werden.





Die wichtigste Anwendung der ganzen Himmelsmathematik ist der virtuelle Helligkeitssensor- programmiert als analoger Ausgang des internen SPS-Moduls 40 (in obiger Grafik die blaue trapezförmige Kurve). Er liefert in den Phasen Sonnenauf- und Untergang einen Wert in %, den man für z. B. für die Außenbeleuchtung nutzen kann. Der Schaltausgang des internen SPS-Moduls 41 (das grüne Signal) schaltet dafür bereits angepasst bei Werten unter 30 % auf „Dunkel“. Der Schaltausgang des internen SPS-Moduls 40 (das braune Signal) schaltet dagegen bei einer virtuellen Helligkeit >64,58 % „Ein“. Grundlage der Berechnung ist die Sonnenhöhe [°] als analoges Signal im internen Modul 41 (braunes, sinusförmiges Signal in der Grafik).

Die Analogausgänge der Module 41 und 42 berechnen laufend Sonnenhöhe und Azimut für den vorgegebenen Standort. Damit ließe sich durchaus gradgenau ein mechanisches Folgesystem für Solaranwendungen steuern.

In der Kalenderanzeige wird der Zeitpunkt des Vollmondes automatisch als kleines Smiley eingeblendet. Die Berechnung erfolgt vereinfacht mit einer Genauigkeit von +/- 1 Tag.



1.8. Die Verwendung der Software-Beispiele

Im Verzeichnis ../Software/ befindet sich ein kompletter Satz aller notwendigen Programm - Dateien. Nach Doppelklick auf die *hauscomp.exe* in diesem Verzeichnis wird mit den dort enthaltenen *hauscomp.ini* und *hardware.ini* das Programm gestartet. Um die Beispiele im Abschnitt 6 auszuprobieren, müssen nur die entsprechenden ini-Dateien in das Verzeichnis ../Software/ kopiert und dort gestartet werden.

1.9. Besonderheiten *hauscomp.exe* unter verschiedenen Betriebssystemen

Zunächst etwas zum Verständnis der Funktionsweise des Programms. Das Programm *hauscomp.exe* besteht intern aus einem Hauptprogramm und einer Interruptroutine, die über den Timer-Interrupt 1C aufgerufen wird. Die Interruptroutine organisiert den Datenverkehr auf dem Hausbus und stellt die bereits umgerechneten Sensordaten in Tabellen dem Hauptprogramm in Echtzeit zur Verfügung. Das Hauptprogramm wiederum ist zuständig für die SPS, die Anzeigen auf dem Bildschirm und das gesamte Dateimanagement. Normalerweise wird der Timer-Interrupt 1C unter DOS aller 55 ms, d. h. 18,2 mal je Sekunde aufgerufen. Damit ist die Funktion des Programms super gewährleistet. Der Bus wird regelmäßig abgefragt und in der Zeit dazwischen läuft das Hauptprogramm mit seinen Rechnungen und Anzeigen. Wird ein längeres Ereignis, wie z. B. eine Dateiausgabe gestartet, so unterbricht das Hauptprogramm kurzzeitig die Interruptroutine. Unter DOS und unter Windows 98SE war das alles auch kein Problem, erst neuere Betriebssysteme halten die genannten 55 ms nicht mehr genau ein. Unter Windows XP schaltet das Betriebssystem z. B. unregelmäßig zwischen verschiedenen Zeitspannen um. Unter der DosBox/DosEmu unter LINUX hängt es u.a. davon ab, ob das Programm im Vordergrund oder im Hintergrund läuft bzw. welche Aufgaben die anderen Programme zu bewältigen haben. Eine optimale Performance ergibt sich mittlerweile mit einem FUTRO 450-2 unter Linux (HC_V2A) incl. Netzwerk (siehe auch [6]).

Diese Besonderheiten führten zu zahlreichen Modifikationen im Projekt „Hauscomputer“. Hier die wichtigsten:

- Unter Windows XP wird die Umschaltung der Timer Betriebsart softwaremäßig gemessen und in den internen Berechnungen so gut es geht berücksichtigt. Die Genauigkeit der SPS-Zeitmesser (Betriebsstundenzähler usw.) kann daher um maximal 10 % vom wirklichen Wert abweichen. Da das Zeitfenster für die Busroutine z. T. nur halb so groß ist wie unter DOS, wird der Takt auf dem Hausbus prophylaktisch verdoppelt – was wiederum die maximale Länge der Leitungen halbiert.
- In der DosBox unter LINUX auf langsamen Rechnern, wenn die *hauscomp.exe* im Hintergrundbetrieb läuft, funktioniert zwar die Interruptroutine und damit der Hausbus, es ist aber praktisch keine Rechenzeit mehr für das Hauptprogramm vorhanden, sodass die SPS und alle Anzeigen „einfrieren“ würden, und zwar ohne Ansprechen der hardwaremäßigen Watchdogs! Aus diesem Grund wurde eine automatische Erkennung dieses Zustandes eingeführt. Ab Version 6.0.28 schaltet das Programm 3 s nach Stillstand des Hauptprogramms in den sogenannten „Notbetrieb“ um. Dabei wird die Interruptroutine unterbrochen, um ausreichend Rechenzeit für 1 – 2 Durchläufe des Hauptprogramms zu gewährleisten, d. h. alle regelmäßigen Berechnungen durchzuführen. Anschließend übernimmt wieder die Interruptroutine. Ein zyklischer Durchlauf aller Funktionen kann damit mehr als 7 s betragen. Eine weitere Funktion ist der „statische Anzeigemodus“, dabei werden grafische Elemente seltener gezeichnet, ebenfalls um auf langsamen Rechnern Rechenzeit zu sparen.
- Eine Analyse der Übertragungsfehler auf dem Hausbus (I^2C ist immerhin ein Jitter- tolerantes Übertragungsprotokoll) zeigte doch sporadische Fehler unter allen neueren Betriebssystemen, zudem noch abhängig von der PC Hardware. Ursache ist die so nicht vorgesehene Nutzung der

RS232 Schnittstelle. HP Laptops neigen z. B. zu einer Fehlerhäufigkeit von ca. 0,2 % aller übertragenen Datenpakete. Die Einführung einer mit Prüfsumme gesicherten Übertragung unter Verwendung des Attinys26 war die Folge. Nun sind sowohl die Software, als auch die Hardware ausreichend fehlertolerant für den Langzeitbetrieb unter Windows XP bzw. LINUX.

- Unter Linux Puppy-Slacko 5.7 wurde ein Fehler in der RAM-Verwaltung (tmpfs) festgestellt, dass dazu führt, das beim Löschen von Files im RAM etwa 1 kByte je File zu wenig wieder freigegeben wird. Dieser Speicher geht intern verloren und kann nur durch ein reboot wiederhergestellt werden. Letztendlich führt dieser Fehler bei intensiver Nutzung der EDB-Datenbanken und des Befehls „Netzwerk=“ zu einem Speicherverlust von mehr als 10 MB pro Tag und damit zu einem zwangsläufigen Neustart nach einigen Tagen, da der Rechner ohne freien Arbeitsspeicher „einfriert“! Da Puppy-Slacko 5.7 aber als letzte Slacko-Version hervorragend alte Hardware unterstützt, wurde ab Version 6.0.51 folgende Lösung gefunden: Zusätzlich zum bisherigen Speicherkonzept in der *hauscomp.exe* werden nun alternativ alle zeitkritischen Dateien unter ihrem Namen mit vorangestellten Tiefstrich gespeichert und zum Schluss mit dem DOS-Attribut „Hidden“ gekennzeichnet. Ein Linux-Script kopiert die aktualisierten Dateien in einen Datensatz ohne Tiefstrich und verhindert so eine Löschung von Files. Dieses Speicherkonzept wird aktiv, wenn bei Programmstart im Datenpfad die Datei „version.txt“ gefunden wird (unabhängig von ihrem Inhalt). Zusätzlich werden dann auch die Dateien „sps.sps“ und „klima.txt“ erzeugt.

Letztlich ist es möglich, das DOS Programm *hauscomp.exe* mit funktionierendem Hausbus auch unter neueren Betriebssystemen zu betreiben. Dabei müssen allerdings Kompromisse eingegangen werden. Die nachfolgende Tabelle stellt die wesentlichen Unterschiede dar.

Betriebssystem	***Funktion Hausbus an COM (RS232)***	***maximale Buslänge [1] [%]***	***Netzwerk***	***+12V aus COM-Port [2]***	***SDA- Korrektur-adapter [3]***	***Genauigkeit Betriebsstunden/ Energiezähler/ Frequenzzähler***	***Besonderheiten***
MS-DOS, PTS-DOS o.ä.	stabil	100	spezielle DOS-Treiber notwendig	ja	möglich, wird automatisch erkannt	< 0,2 %	Programm ist schnell und zuverlässig aber andere Software nicht gleichzeitig möglich; Netzwerk ist problematisch[8]. Super kurzer Programmstart
Windows 98 SE	instabil [4]	100	ja	Ja[4]	Ja[4]	< 0,2 %	Entweder es läuft oder aber nicht
Windows XP	instabil [4]	50	ja	ja	möglich, wird automatisch erkannt	< 10 %	„Wochenkorrektur=“ nicht möglich. Programm läuft nur im Vollbildmodus, andere Software nicht gleichzeitig möglich, sporadische Dateizugriffsfehler möglich
DosBox (V.0.72) unter Windows 98 SE	stabil	100	z.T. Win 98 Treiber notwendig	nein	nur mit „SDAKorrektur=on“ in *hardware.ini*	< 0,2 %	Befehl „ver set 6.0“ unter DOS notwendig; Programm ist langsam wegen Emulation[5].
DosBox (V.0.72)	stabil	50	ja	nein	nur mit	< 10 %	„Wochenkorrektur=“ nicht möglich. Andere

Betriebssystem	***Funktion Hausbus an COM (RS232)***	***maximale Buslänge [1] [%]***	***Netzwerk***	***+12V aus COM-Port [2]***	***SDA- Korrektur-adapter [3]***	***Genauigkeit Betriebsstunden/ Energiezähler/ Frequenzzähler***	***Besonderheiten***
unter Windows XP					„SDAKorrektur=on“ in *hardware.ini*		Software gleichzeitig möglich. Laufschrift „ruckelt“ möglicherweise; Programm ist langsam wegen Emulation[5]. In *hardware.ini* Scriptbefehl „BS=XP“ notwendig
DosBox (V.0.73) unter Linux	stabil	100	ja	ja	möglich, wird automatisch erkannt	< 20 % (je nachdem, wie viele Programme parallel laufen)	Unter Ubuntu 9.10 getestet; Netzwerk ohne Probleme; „Wochenkorrektur=“ nicht möglich. Laufschrift „ruckelt“ möglicherweise;Programm ist langsam wegen Emulation[5]; Bildschirmschoner muss deaktiviert sein. Programm sollte im Vordergrund laufen; Laufschrift muss deaktiviert werden[7]; Editieren der ini – Dateien problematisch[6]
DosEmu unter Linux **Absolute Empfehlung!** Siehe auch Linux HC_V2A	stabil	100	ja	ja	k.A.	< 10% (< 0,2%[9])	Unter Ubuntu 9.10 u.a. getestet; Netzwerk ohne Probleme; „Wochenkorrektur=“ ist möglich[10]. Programm läuft fast so schnell wie unter DOS! Bildschirmschoner sollte deaktiviert sein. Laufschrift muss deaktiviert werden[7];INT21 muss deaktiviert werden[8]; Editieren der ini – Dateien problematisch[6]
Windows 7 (32 Bit)	-	-	-	-	-	-	Programm lässt sich nicht im Vollbildmodus starten
Windows 8 (32 Bit)	stabil	k.A.	ja	k.A.	k.A.	k.A.	Programm läuft zwar, aber keine Bildschirmausgabe, da Vollbildmodus nicht möglich
DosBox (V.0.74) unter Win 8 (32 Bit)	stabil	50	ja	nein	k.A.	k.A.	Dosbox-0.74.conf befindet sich unter c:\users\ich\AppData\local\DosBox\

1) Maximale Länge geschätzt bei 100%: ca. 300 m. Kein MAX 1236; ohne Scriptbefehl „Bustakt=“ in *hardware.ini*.
2) Für geringe Belastungen, z. B. für 4-Kanal Digitalmultimeter mit MAX1236
3) Z. B. bei Nutzung des SDA-Korrekturadapters oder des PC Anschlusses am 230V Schaltinterface
4) Hängt von Initialisierung des COM - Anschlusses durch Windows ab: Hardware Abstraktionslayer (HAL) verhindert möglicherweise für DOS – Programme den Zugriff auf die serielle Schnittstelle

5) Langsamer Grafikaufbau, Zykluszeit der SPS vergrößert sich, sehr langsames lesen und Schreiben von Dateien; der Datenverkehr auf dem Hausbus ist nicht betroffen.
6) Die ini.* Dateien erwarten ein Zeilenende wie unter DOS/Windows üblich (OD + OA am Ende). Unter Linux editierte Dateien werden u.U. vom Programm *hauscomp.exe* nicht erkannt.
7) Ist die Laufschrift aktiviert, kommt es bei Dateizugriffen unter Linux bzw. Windows XP zu zufälligen Datei Zugriffsfehlern und damit zum Programmabsturz. Ab Version 6.0.38 behoben.
8) Im DOS Betrieb sorgt der regelmäßige (interne) Aufruf von Interrupt 21Funktion 19h für die saubere Ausführung von TSR Programmen, die ein eventuell vorhandenes Netzwerk verwalten. Deshalb muss mit dem Scriptbefehl „INT21=on“ diese Funktion aktiviert werden. Unter LINUX ist das nicht notwendig, im Gegenteil: unter Dosemu führt diese Funktion zum Programmabsturz (ab Version 6.0.38 nicht mehr!). Deshalb darf der Scriptbefehl „INT21=“ nicht verwendet oder sollte auf „off“ gesetzt werden.
9) Nach Programmstart unter Linux führen regelmäßige Dateizugriffe (im Live-Modus), sowie die Kernelverwaltung selbst zu einer Abweichung des Zeitkerns bis max. 5%. Bei „langsamen“ PCs (bis ca. 700 MHz) kann die Abweichung bis 10% betragen. Nach dem Durchlauf eines ganzen Tages (von 0:00 bis 24:00) ohne manuelle Zeitkorrektur und ohne Umstellung von oder zur Sommerzeit wird diese Abweichung automatisch im Programm erkannt und ab dem nächsten Tag korrigiert. In der Folgezeit verbessert sich die Abweichung damit bis auf den theoretischen Wert von < 0,2%. Möglich ist eine Vorgabe bereits ab Programmstart, siehe auch Scriptbefehl „zkonst=“ für *hardware.ini*. Ab Version 6.0.39e erfolgt die Korrektur nicht mehr täglich, sondern ab Programmstart bereits stündlich, sodass nach spätestens 4h die Abweichung <1% ist.
10) Die Uhrzeit wird unter Ubuntu nur im laufendem Betriebssystem korrigiert, nicht aber im BIOS. Das hat zur Folge, dass nach einem Systemneustart eine nicht korrigierte Uhrzeit geladen wird. Unter Puppy Linux (Versionen Wary und Slacko) wurde die Zeit selbst im Betriebssystem nicht korrigiert. Besser ist es deshalb, die Uhrzeit direkt von LINUX über das Netzwerk automatisch korrigieren zu lassen und den Scriptbefehl „Wochenkorrektur=“ in der *hardware.ini* auf „0“ zu setzen.

Programmversion	„MemAvail“ [Byte] Seite 148 unter Linux HC_V2A (PS/nach 8 min)	„MemAvail“ [Byte] Seite 148 unter Windows XP	Bemerkungen
6.0.53 vom 11.03.2021	130920/126824	79160	
6.0.54 vom 12.04.2021	97080/92984	45320	Erweiterung Anzahl SPS-Module
6.0.55 vom 17.11.2021	93928/89832	42168	Programmkode ausgemistet
6.0.56 vom 31.01.2022	94184/90088	42424	
6.0.57 vom 04.12.2022	94104/90008	42344	
6.0.58B1 vom 17.02.2023	93464/89368	41704	Erweiterung HCdata
6.0.59B2 vom 29.08.2024	92968/88872	41208	
6.0.60 vom 29.08.2026	88344/84248	36584	Programmkorrekturen

Software über Jahrzehnte funktionsfähig zu halten, ist eine lösbare Aufgabe!

Praktische Erfahrungswerte mit verschiedenen PC Plattformen (hauscomp.exe Version 6.0.37):

Hinweis: Ab Version 6.0.38 konnten die Zykluszeiten für umfangreichere Grafik gegenüber den Tabellenwerten um ca. 30-50% verkürzt werden, allerdings verringert sich der freie Hauptspeicher (MemAvail) auch um 12 KByte.

PC-Typ	*Prozessor*	*Stromaufnahme*	*Betriebssystem*	*Netzwerk*	*zk-Wert*	*Zykluszeit ohne Grafik (BS255)*	*Zykluszeit geringe Grafik (BS022)*	*Zykluszeit maximale Grafik (Zeitkern, 189/148)*	*Freier Haupt-Speicher (MemAvail)*	*Besonderheiten*
386SX33	386SX, 33 MHz	12 VA	DOS 6.2	nein	14	0,5 s	0,7 s	1,4 s	146 KByte	
386SX40	386SX, 40 MHz	15 VA	DOS 6.2	nein	22	0,2 s	0,4 s	0,9 s	146 KByte	
GX1-LCD Kontron	Geode, 300 MHz	6 W [1)]	DOS, Installation siehe 1.11.1	ja	399	0,3 s	0,4 s	2,5 s	24 KByte	CPU: 133Mhz im BIOS (lowest)
Thinkpad 755C	486DX, 75MHz	k.A.	DOS 6.2	ja	306	0,055 s	0,165 s	0,3 s	32 KByte	
Thinkpad 755CD	486DX, 75MHz	k.A.	DOS 6.2	ja	408	0,055 s	0,165 s	0,3 s	97 KByte	
IGEL 3/2	VIA C3, 400 MHz	16 VA	DOS (Pkt. 1.11.2)	ja	3364	0,055 s	0,055 s	0,2 s	93 KByte	
IGEL 3/2	VIA C3, 400 MHz	16 VA	Linux HC_Basic	ja	1056	< 0,01 s	0,4 s	2,0 s	200 KByte	
IGEL	800 MHz	15 VA								
IGEL 5/4	VIA C7, 1 GHz	17 VA	Linux HC_Basic	nein	6432	< 0,01 s	0,1 s	0,6 s	196 KByte	Netzwerkchip wird nicht erkannt
IGEL 5/4	VIA C7, 1 GHz	17 VA	DOS (Pkt. 1.11.2)	ja	7952	< 0,01 s	0,055 s	0,2 s	119 KByte	
IGEL 5/4	VIA C7, 1 GHz	17 VA	Linux HC_advanced (Puppy-Wary)	ja	4437	0,1 s	0,2 s	1,2 s	206 KByte	Laufschrift=on, zkonst=17.34
FUTRO S450-2	AMD Sempron 200U 1 GHz	19 VA	Linux HC-V2 (Puppy-Slacko 5.7)	ja	3328	0,1 s	0,113 s	0,17 s	147 KByte **(V6.0.50)**	Laufschrift=off, zkonst=17.36
FUTRO S450-2	AMD Sempron 200U 1 GHz	19 VA	Linux HC-V2A (Puppy-Slacko 5.7)	ja	7011	0,1 s	0,118 s	0,17 s	140 KByte **(V6.0.51)**	Laufschrift=off, zkonst=17.36
IBM T30	Pentium 4, 2 GHz	51 VA	Windows XP	ja	13017	< 0,01 s	< 0,01 s	0,2 s	164 KByte	

[1)] ohne Netzteil

1.10. *Hauscomp.exe* in der „DosBox“ bzw. „Dosemu“ Dank an dieser Stelle an Herrn Höfler!

Das Programm „DosBox“ wurde entwickelt, um ältere Spiele unter neueren Betriebssystemen wie Windows oder Linux lauffähig zu bekommen. Der große Vorteil moderner Betriebssysteme gegenüber DOS Anwendungen ist die weitgehende Netzwerkfähigkeit, die das Update und den Zugriff auf gesammelte Daten im Projekt Hauscomputer erlauben; aber auch andere Programme können quasi gleichzeitig parallel betrieben werden. Das Programm *hauscomp.exe* läuft so allerdings in einer Emulation und ist damit wesentlich langsamer als normal. Ein Pentium 4 mit 3,05 GHz braucht im Live Modus unter LINUX für die Speicherung der „live.bmp“ nach Wechsel der Bildschirmseite ca. 5 s. Das ist natürlich erheblich mehr als ohne Emulation (unter XP sind dafür bei gleichem Rechner nur 0,3 s notwendig). In dieser Zeit wird auch nicht der Hausbus abgefragt. Für die normale und vor allem sichere Funktion ist deshalb ein Kompromiss zwischen schnellem Rechner und Netzwerkfähigkeit gefragt. Der Eintrag „cycles=max“ unter [cpu] in der dosbox.conf (alternativ auch <Ctrl>+F12/F11) kann das Programm zwar wesentlich beschleunigen, verschlechtert aber auch die Genauigkeit verschiedener SPS – interner Funktionen.

Eine einfache Kontrolle erlaubt die Seite 186 (I2C-Bus Analyse), auf der die Zeit für einen kompletten Bus Zyklus sekundengenau angezeigt wird. Normalerweise beträgt sie 255 Interrupts mal 55 ms = 14 s. Werden mehr als 15 s angezeigt, so gibt es also ein Problem mit dem Interrupt. Eine Lösungsmöglichkeit ist die Verbesserung der Rechner Performance, das Schließen anderer offener Fenster/Programme. Eine Erhöhung des Bustaktes (Scriptbefehl „Bustakt=“) ist in der Regel nicht möglich, da das Programm bereits am zeitlichen Limit läuft.



Ab Version 6.0.30 läuft das Programm *hauscomp.exe* auch unter „Dosemu“ unter Linux. Die Geschwindigkeitsbeschränkungen entfallen damit. Einzige Einschränkungen: Die Laufschrift muss mit dem Scriptbefehl „Laufschrift=off“ deaktiviert werden und der Interrupt 21 muss mit dem Befehl: „INT21=off“ abgeschaltet werden. Diese Einschränkungen entfallen ab Version 6.0.38 – ab jetzt wird die Dosemu über die Enviroment-Variable „HELLPATH“ automatisch erkannt.

Grundsätzlich sind nun Updates der *hauscomp.ini* und Einblicke in Echtzeit sowie die Steuerung über das Netzwerk, unspektakulär möglich.

1.10.1. „DosBox“ für Windows

Da in der DosBox die serielle Schnittstelle unter Windows trotz Hardware-Abstaktions-Layer (HAL) wieder funktioniert, kann man das Programm *hauscomp.exe* fast ohne Einschränkungen nutzen. Daneben bietet Windows 98 SE (gegenüber XP) noch die DOS- kompatible Timer Schnittstelle und somit die maximale Hausbuslänge. Problematisch ist einzig die DOS – Version, die die DosBox dem laufenden Programm vorgaukelt: V. 5.0. Das Programm *hauscomp.exe* hält die Box so für Windows XP und reduziert den Bustakt prophylaktisch auf die Hälfte. Eine Korrektur auf V. 6.0 behebt dieses Problem unter Windows 98 SE. Ein weiteres Problem ist die Behandlung des TxD Ausganges. Praktisch kann er nicht mehr stabil auf +12V geschaltet werden. Die Versorgung externer Komponenten aber auch die automatische Erkennung des SDA Korrekturadapters funktionieren nicht mehr. Deshalb muss in der *hardware.ini* bei Bedarf (SDA Korrekturadapter physikalisch vorhanden) der Scriptbefehl „SDAKorrektur=on“ eingefügt werden.

Bevor die DosBox installiert wird, sollten die Hauscomputer Dateien in einem beliebigen Verzeichnis zur Verfügung gestellt werden. Das Verzeichnis sollte zum Lesen und Schreiben im Netzwerk freigegeben sein.

Installation: Zunächst lädt man die DosBox von der Homepage des Herstellers oder von



hauscomp.exe in der DosBox unter Windows XP

http://www.chip.de/downloads/DOSBox-0.72_13015039.htm herunter. Man folgt dann der Standard - Installation (c:\programme\DOSBox-0.72\). Danach werden mit einem Editor in der Datei „dosbox.conf“ in den Abschnitten [serial] und [autoexec] die entsprechenden Zeilen ergänzt.

Angenommen, das Programm *hauscomp.exe* und die anderen notwendigen Dateien befinden sich im Verzeichnis „c:\wz\hc\test\dosbox\“ und der reale COM Port soll 1 sein:

Windows 98 SE	***Windows XP***
[serial] serial1=directserial realport:COM1	[serial] serial1=directserial realport:COM1
[autoexec] mount c: c:\wz	[autoexec] mount c: c:\wz

Windows 98 SE	*Windows XP*
c: ver set 6.0 cd c:\HC\test\Dosbox hauscomp	c: cd c:\HC\test\Dosbox hauscomp

Zum Schluss wird die DosBox in die Autostart-Gruppe aufgenommen, um den automatischen Start nach dem Booten des Rechners zu gewährleisten.

1.10.2. „DosBox“ für Linux (Ubuntu 9.10)

Dieser Abschnitt wurde ab Version 6.0.33 entfernt, da die Funktionalität von Dosemu für das Programm *hauscomp.exe* wesentlich effektiver ist.

1.10.3. „Dosemu“ für Linux (Ubuntu 9.10)

Es gibt eine Alternative zur Dosbox: „Dosemu“, welches einige wesentliche Vorteile gegenüber Dosbox bietet: Es emuliert nicht nur, sondern überlässt dem DOS Programm die volle Souveränität über die CPU. Damit läuft alles fast so schnell wie unter einem reinen DOS, zusätzlich erhält man aber auch die Netzwerkfähigkeit. Dieses Programm ist aber nur unter Linux vorhanden.

Infos zu Dosemu gibt es unter http://de.wikipedia.org/wiki/Dosemu. Ein zweiter Vorteil ist, dass ein freies Dos zur Anwendung kommt. Dosemu liefert nur die Umgebung, in der man ein Dos, eigener Wahl, verwenden kann (MS-Dos, Freedos, DR-Dos). In der aktuellen Version von Dosemu 1.4.0 (findet man in den Paketquellen) ist FreeDos bereits enthalten und wird automatisch mit installiert.

Die Installation erfolgt am einfachsten über die Paketquellen. Nach der Installation hat man bereits eine volle Dos-Umgebung. Es erscheint unter Anwendungen # Systemwerkzeuge # Dosemu und kann mit einem Klick gestartet werden.

Im sich öffnenden Terminal erscheint ein übliches Dos-Fenster. In den Vollbildmodus kann mit Strg-Alt-F wechseln. Sollte das Programm nicht starten, ist es hilfreich ein Terminal zu öffnen und dort das Programm mit Dosemu zu starten.

c:\ ist ein Verweis auf /home/username/.dosemu/drive_c

d:\ ist das allgemeine Homeverzeichnis /home/username/

Man kann den bereits für Dosbox angelegten Hauscomputer Ordner im Homeverzeichnis übernehmen.

z. B.: ~/Dosbox od. /home/username/Dosbox. Somit kann schön Dosbox oder Dosemu getestet werden. Der gleichzeitige Betrieb ist natürlich nicht möglich

Einstellungen der dosemu.conf und autoexec.bat:

Mit der Datei dosemu.conf, welche unter /etc/dosemu/ liegt, kann man Dosemu konfigurieren.

Um Änderungen an dieser Datei vornehmen zu können muss man diese Datei mit Rootrechten öffnen.

Dabei starten wir das Terminal über Anwendungen: Zubehör: Terminal und geben dort dann sudo gedit /etc/dosemu/dosemu.conf ein. Die Passwortabfrage mit dem Passwort bestätigen. Danach öffnet sich der Editor mit der Datei dosemu.conf. Dabei interessant sind die Punkte - CPU settings und Serial port settings:

```
## CPU settings: define the CPU features to DOSEMU
$_cpu = "80586"
$_hogthreshold = (5)
## Serial port settings
$_com1 = "/dev/ttyS0"
```

$_cpu = "80586" - dieser Wert verändert von außen nichts Ersichtliches.

$_hogthreshold = (5) - damit wird Rechenpower an andere parallelllaufende Prozesse abgegeben, denn im Gegensatz zur Dosbox will Dosemu keine Rechenleistung teilen. Mit diesem Wert kann man dies verbessern. 5 ist ein guter Wert und funktioniert bei mir ganz gut.

Serial port settings - Einstellungen für die serielle Schnittstelle:

$_com1 = "/dev/ttyS0" – damit wird Linux „durchlässig“ für die direkte Ansteuerung der seriellen Schnittstelle.

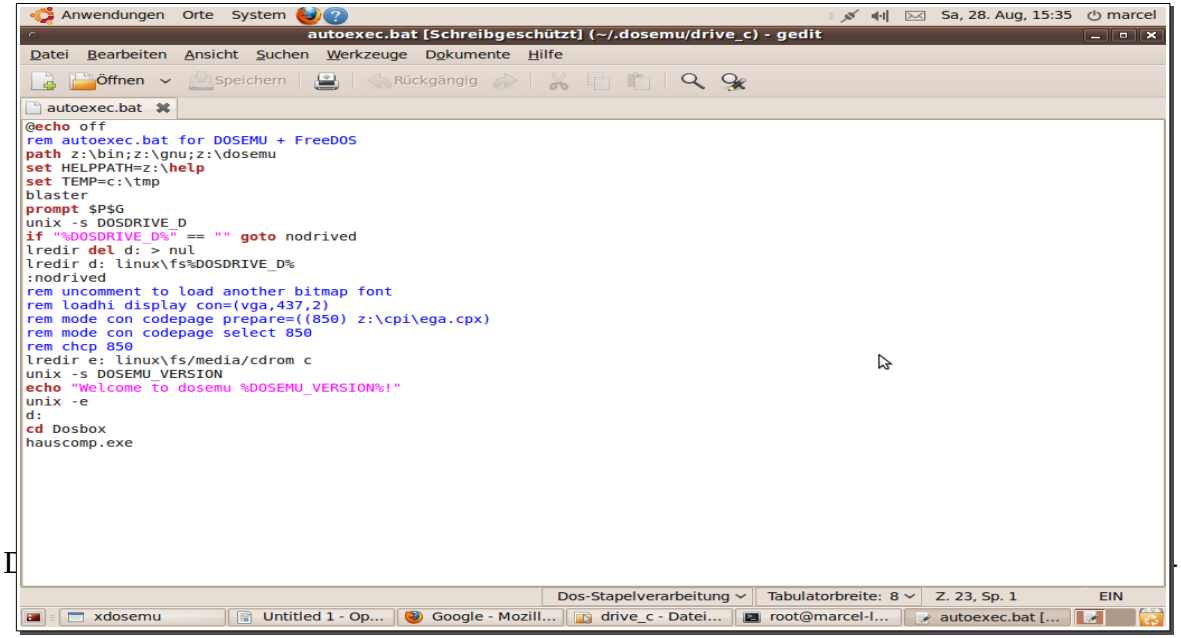
Die autoexec.bat wird ebenfalls um einige Zeilen ergänzt. Die Datei liegt im versteckten Dosemu Ordner ~/.dosemu/drive_d/autoexec.bat und wird mit gedit geöffnet. Da dieser Ordner im eigenen Homeverzeichnis liegt, braucht man keine Rootrechte, also nur

gedit ~/.dosemu/drive_d/autoexec.bat

Da wir den vorhandenen Dosbox Ordner verwenden wollen, fügen wir folgendes in der autoexec.bat hinzu:

d:

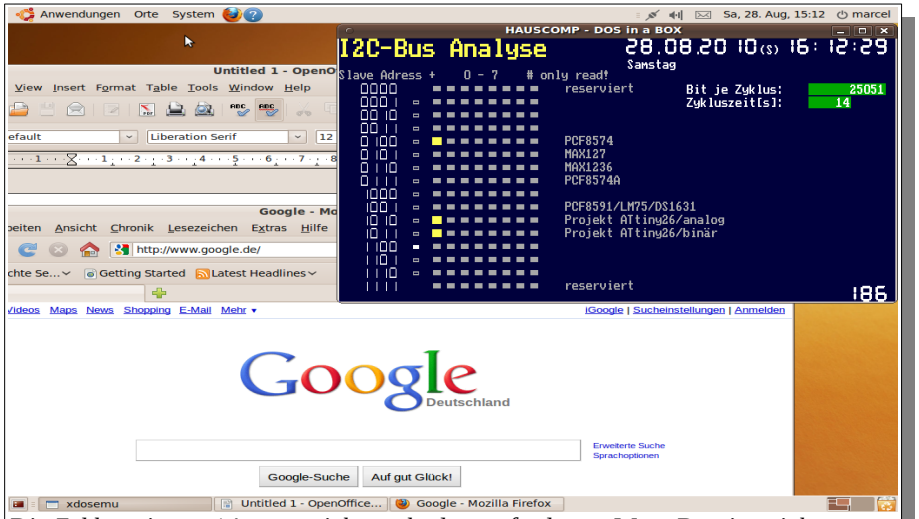
cd Dosbox



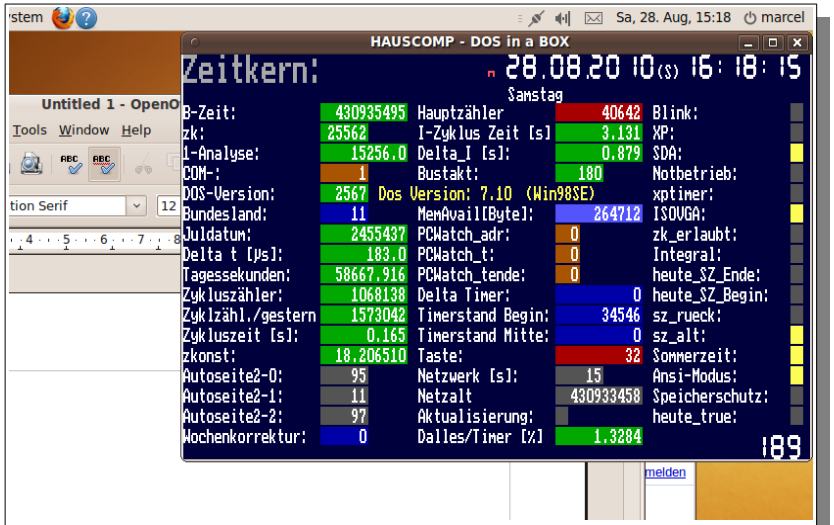
hauscomp.exe

Danach startet man den Hauscomputer über Anwendungen: Systemwerkzeuge: Dosemu od. über das Terminal.

Bilder Hauscomp-Software unter Dosemu:



Die Zykluszeit von 14s entspricht exakt dem geforderten Wert. Damit weichen Betriebsstundenzähler um weniger als 5% vom Soll ab.



Interessant ist der "zk" Wert - hier mit 25562 wesentlich besser als nur 34 unter der DosBox

Autostart:

Damit die DosBox nach Einschalten des Rechners automatisch gestartet wird, muss zunächst der Anmeldebildschirm abgeschaltet werden:

- Systemverwaltung: Anmeldebildschirm, dort „entsperren, als <username> automatisch anmelden und schließen.

Im Anschluss wird die Box im Autostart-Menü aufgenommen System: Einstellungen: Startprogramme:

- „Hinzufügen“, „Durchsuchen“
- Rechte Maustaste – verborgene Dateien anzeigen,
- Dateisystem/usr/bin, dort Doppelklick auf Programm „dosemu“
- Name: Hauscomputer
- Kommentar: Autostart

- „Hinzufügen“ drücken und Fenster schließen.

Der Samba Server sollte automatisch starten. Wenn nicht, gibt es hier eine Anleitung: http://forum.ubuntuusers.de/topic/samba-automatisch-starten/

Nun ist nur noch der Bildschirmschoner über System: Einstellungen: Bildschirmschoner zu deaktivieren. Damit ist alles konfiguriert, der Rechner ist im lokalen Netzwerk sichtbar und man kommt über das Netzwerk von anderen Rechnern an alle Dateien heran. Sollte es Probleme geben, kann man das Thema SAMBA unter http://http.//www.goldmann.de/linux-mac-und-windows-in-einem-netzwerk_tipp_70.html nachlesen. Sollte der Start von SAMBA aus irgendeinem Grund von Hand notwendig sein, ist im Terminalfenster „sudo etc/init.d/samba start“ einzugeben. Es erscheint dann am Ende der Eingabezeile ein „[OK]“.

Damit später auch aus dem Netzwerk auf alle Hauscomputer-Dateien schreibend und lesend zugegriffen werden kann, muss noch folgende Zeile im Terminalfenster (Anwendungen – Zubehör – Terminal) eingegeben werden:

sudo_chomod_777_/home/username/Dosbox/_-R <Enter>

„_“ steht dabei für ein Leerzeichen und als username ist natürlich der jeweilige anzugeben. Ich hoffe, ich habe nichts Wesentliches vergessen. Für mich als Linux Neuling war diese Prozedur gar nicht so einfach, da viele kleine Details Fallstricke beherbergten. Dosemu sollte übrigens immer im Vordergrund laufen, da so mehr Rechenpower zur Verfügung gestellt wird. Wird die Rechenleistung zu gering, so läuft das Programm im sogenannten Notbetrieb. Es funktioniert dabei zwar noch alles, aber es wird sehr, sehr langsam…

Es gibt noch eine Einschränkung: Sollte unter Linux eine *.ini editiert werden, so ist zu beachten, dass das Programm *hauscomp.exe* am Zeilenende die Kodierung OD + OA erwartet. Mit dem Programm Gedit (Linux) wird aber standardmäßig nur „OA“ angehängt, mit der Folge, dass diese Zeile später nicht gelesen werden kann. Unter Linux also nur Parameter ändern, keine neuen Zeilen mit Enter einfügen! Alternativ dazu kann man mit dem Programm unix2dos die editierte *ini Datei in eine Dos- kompatible umwandeln:

Im Terminal in den Ordner wechseln in dem die Datei liegt und unix2dos ausführen oder gleich den gesamten Pfad angeben.

unix2dos Dateiname od.

unix2dos Pfadname eingeben.

unix2dos_/home/username/Dosbox/*.ini

Bei Fernzugriff über das Netzwerk: man kopiert die Datei auf den (Linux-) Rechner, führt den Befehl aus und kopiert sie wieder zurück in den Dosbox Ordner. Die Datei beim Zurückkopieren ersetzen lassen.

Dieser ganze Abschnitt zum Thema Dosemu ist also reichlich komplex. Für die praktische Nutzung wurde daher aus den bisherigen Erfahrungen ein fertiges Linux-Betriebssystem entwickelt, dass auf Anhieb funktioniert (nächster Abschnitt).

1.10.4. Spezielle Linux Zusammenstellungen für das Projekt Hauscomputer

Die Erfahrungen mit Dosemu unter Ubuntu führten folgerichtig zu einem angepasstem Linux System, dass lizenzfrei, mit einer aktuellen Hardwareerkennung, zudem boot- und netzwerkfähig ist. Auf der Grundlage von Debian und Dosemu wurde ein Paket geschnürt, dass auch sparsam ausgestattete PCs unterstützt. Als Festplatte kommt jeweils eine CF Karte/oder USB-Stick zum Einsatz. Das System ist über Jahre erprobt und zuverlässig. Die Software läuft weitgehend im RAM und belastet nicht die Lebensdauer des Speichers. Nach dem Aufspielen des Images auf einen bootfähigen Datenträger erhält man ein vollautomatisch arbeitendes Betriebssystem mit ebenfalls automatisch startender DOS Emulation inklusive *hauscomp.exe*. Die Ansteuerung des Hausbusses über die serielle Schnittstelle ist genauso gewährleistet wie die Nutzung der Netzwerkmöglichkeiten, sowie eventuell vorhandener USB-Anschlüsse. Geringe Einschränkungen müssen bei der Echtzeitfähigkeit, sowie der Geschwindigkeit (nur etwa halb so schnell wie reines DOS) hingenommen werden, dafür gibt es aber ein leistungsfähiges Web-Interface!

- Die Beschreibung der aktuellen Linux- Zusammenstellungen befindet sich im Band „Hauscomputer und Linux“ -

Tabellarischer Vergleich der beiden ersten Linux- Zusammenstellungen (historisch, nicht mehr aktuell!):

	Image HC_Basic (2011)	***Image HC_advanced (2013)***
Hauptspeicher (RAM) Mindestgröße	64 MB, >= 128 MB empfohlen	512MB
CF-Card/Festplatten Mindestgröße	1 GB	1 GB
Mindestanforderung an Prozessortakt	100 Mhz	400 Mhz
Beispiel für PC	IGEL 3/2 VIA C3 400 Mhz	IGEL 5/4 VIA 1GHz
Autostart *hauscomp.exe* nach Einschalten	Ja	Ja, (2min 12s)
Automatische Zeitkorrektur	Autonom (wöchentlich durch *hauscomp.exe*)	Übers Internet (NTP)
CF-card geeignet (wenig Zugriffe auf Flash während des Betriebes)	Ja, wenn Logs nur 1x täglich aufgezeichnet werden und siehe 1)	Ja, da Betriebssystem im RAM läuft
Automatische Hardwareerkennung	System, Netzwerk, USB, (eingeschränkt)	System, Netzwerk, USB, WLAN (Debian kompatibel)
IP Adresse	Statisch 192.168.1.233, änderbar, dhcp möglich	Statisch 192.168.1.233, änderbar, dhcp möglich
Netzwerkprotokolle/Server	VNC, SSH, SCP Port 22, automatisch startend, Passwort wird unterstützt	Http, Https, FTP, Passwort wird unterstützt, CGI / Samba, hiawatha
Arbeitsoberfläche	Fluxbox	Puppy-Wary

	Image HC_Basic (2011)	***Image HC_advanced (2013)***
Dateioperationen	Midnight-Commander	Rox-Dateimanager, Midnight-Commander
Texteditor	Nano	Geany
Im lokalen Linux Netzwerk sichtbar	-	Ja [2)]
Im lokalen Windows Netzwerk sichtbar	-	Ja [2)]
Von Linux Rechner aus index.htm aufrufbar (aktives Netzwerk über SSH)	Ja	Ja [2)]
Von Windows Rechner aus index.htm aufrufbar (aktiv im Workgroup Netzwerk)	-	Ja [2)]
„live“ Modus und Netzwerkaktualisierung Hauscomputer	bedingt 1)	Ja (verkürzt aber die Lebensdauer des Flashspeichers)
CGI – Steuerung des Hauscomputers	-	Ja, per http(s)
Administration von Dateien über das Netzwerk	mittels WinSCP	FTP
Sonstiges		Statistikanalyse, NAS u.a.
Passwörter	Betriebssystem: root: „hcroot“ hcuser: „hcuser“	Betriebssystem: root: „hcroot“ Netzwerkzugriff: admin: „admin“ user: „hcuser“ gast: „“

1) möglich durch Aktivierung der entsprechenden Scriptbefehle in der *hardware.ini*; bei langsamen Rechnern möglicherweise Performanz-probleme, zudem Verringerung der Lebensdauer von CF Cards durch die dann häufigen Zugriffe (alle anderen Prozesse laufen im RAM)

2) Da die internen Perl – Scripte nur über http(s) funktionieren, kann nicht per CGI gesteuert und auch der interne Dateiserver kann nicht genutzt werden.

1.11. Netzwerkfähigkeit unter DOS

Normalerweise ist das Betriebssystem DOS nicht für Netzwerkbetrieb ausgelegt. Unter bestimmten Voraussetzungen kann es aber dennoch funktionieren: die korrekten Treiber für die entsprechende Netzwerkkarte müssen vorhanden sein und eine elend lange Konfiguration von Hand ist notwendig. Das kann man sich aber unter Umständen auch alles sparen. Nachfolgend ein paar Anleitungen für ein GX1LCD Board (Geode Prozessor, Hersteller Kontron) und ein 3/2 IGEL PC mit RTL8100B Netzwerkchip. Beide sind auch als Image auf der Projekt DVD vorhanden. Für das Kopieren auf Datenträger siehe auch Pkt. 1.10.5.

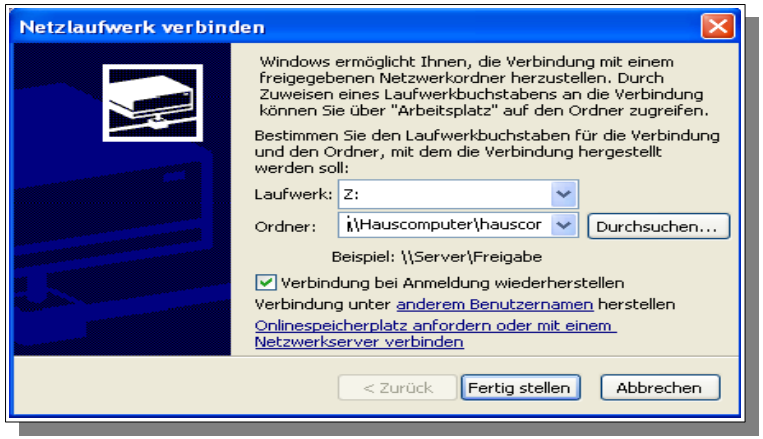
1.11.1. Installation DOS inklusive Netzwerk für GX1LCD (Geode von Kontron) mit Ergänzungen 2014 und 2022

- Notwendig ist ein Compact-Flash-Card (CF- Card) mit mindestens 5 MB Speicher
- Die CF Card bootfähig machen: entweder von der Windows-CD booten und als „Zieldatenträger“ die CF Card angeben oder die FAT-16 formatierte CF Card (vorher primäre aktive DOS-Partition mit FDISK erstellen) unter Win98 mit den DOS-Systemdateien versehen; MS-DOS Eingabeaufforderung und „sys e:“, wenn „e“ das CF- Laufwerk ist, anschließend (wenn sie nicht booten sollte) unter Linux mit Gparted für die Partition neben der Eigenschaft „boot“, „lba“ hinzufügen. Die „boot“ Eigenschaft steht für „Aktive“ Partition, „lba“ ist ein Festplatten-zugriffsmodus, der erst später aufkam. Linux kann man dazu auf einem aktuellen PC auch als live-System von DVD starten. Gparted gehört zur Grundausstattung vieler Distributionen, wie z. B. Ubuntu. Eine bootbare CF-Card lässt sich später mit dem USBImageTool auf Datenträger gleicher Größe einfach ebenfalls bootbar kopieren. Auf der DVD gibt es bereits ein fertiges bootfähiges Image für CF Karten mit 512MB („GX1_Kontron_512MB.img“) das bereits alle notwendigen Dateien enthält inklusive des erweiterten Webinterfaces für DOS – die folgenden Schritte können dann weitestgehend übergangen werden.
- Alle Dateien und Verzeichnisse aus dem Verzeichnis „/Netzwerk/DOS GX1 LCD/“ der Projekt CD außer io.sys, msdos.sys und command.com auf die CF- Card kopieren. Die genannten 3 Dateien dienen nur zum Vergleich, sie lassen sich auch nicht einfach in den Master Boot Record (MBR) kopieren.
- Überprüfen, dass bei allen kopierten Dateien kein Schreibschutz als Attribut gesetzt ist (ist auf der DVD normalerweise so!), ggf. korrigieren.
- Die eigenen Lizenz.rpt (ab Version 6.0.33 nicht mehr notwendig), *hardware.ini* bzw. *hauscomp.ini* und die aktuellste *hauscomp.exe* auf die CF-Card kopieren. *Hinweis: Bei der Benutzung künftiger, größerer Versionen der hauscomp.exe kann es irgendwann zur Fehlermeldung „Runtime Error 203“ kommen. Dann benötigen das Programm und die notwendigen Treiber mehr Arbeitsspeicher als physikalisch vorhanden ist. Lösung: Treiber soweit wie möglich in den oberen Arbeitsspeicher laden, ältere Version der hauscomp.exe benutzen oder ein anderes Betriebssystem (z. B. Linux) verwenden. Die Version 6.0.41 meldete noch mehr als 50 KByte freien Speicher. Eine spezielle Version 6.0.56GX1 wurde 2022 bereitgestellt.*
- Wichtig ist, dass für die Netzwerkfähigkeit in der *hardware.ini* die Scriptzeilen „Netzwerk=300“ und „INT21=on“ enthalten sind. Der Scriptbefehl „PCWatchdog=“ ist ausschließlich für einen GX1 Kontron gedacht und sollte auf diesem auch immer aktiviert sein (auf der DVD ist er noch mit „#“ deaktiviert).
- Bei Bedarf, also wenn ein bislang laufendes System auf Netzwerkfähigkeit „geupdatet“ wird, alle „gestern.*“ und „heute.*“ Dateien vom alten auf das neue System kopieren.

- Die CF- Card in das GX1LCD- Board einstecken und das Board mit einem DHCP- fähigen und eingeschalteten Router verbinden (Freigabe smb1 beachten). Das Board mit einer neuen Knpfzelle CR2032 versehen.
- Am Board eine Spannungsquelle +12V (>0,5A), 5V (>1,5A) am 6-Pin PWR Connector, eine PS2-Tastatur und ein Display (VGA) anschließen.
- Spannungsquelle und damit Board einschalten. Mit <F2> das BIOS starten und folgende Einstellungen vornehmen:
  - mit <F9> die Werkseinstellungen laden
  - CPU Speed auf 200MHz/67MHz
  - Advanced I/O Device Configuration, Seriell Port B auf:
  - Enabled, Mode Normal, Base I/O Adress 2F8
- Die CPU Geschwindigkeit kann auch höher gestellt werden, was aber den Stromverbrauch und die Wärmeentwicklung verschlechtert. Die Einstellungen für den seriellen Port B erlauben die Nutzung des Ports auf der Platine als COM2 neben COM1 im Slotblech.
- geänderte BIOS-Einstellungen abspeichern → Board booten lassen
- Beim Bootvorgang holt sich der entsprechende Treiber eine gültige IP-Adresse vom Router und mountet das Verzeichnis „Hauscomp“ für die Freigabe im Netz. Anschließend startet die *hauscomp.exe*. Es dürfen keine Fehlermeldungen auftreten. Erkennt das Board keinen DHCP Server, weil er z. B. noch ausgeschaltet oder gar nicht angeschlossen ist, so kommt es nach ca. 3 min zu einer Fehlermeldung und ein Neustart ist erforderlich. Das Programm *hauscomp.exe* startet dann auch nicht! Stellt man im BIOS unter dem Punkt INSIDE Utilities/ Watchdog Timeout den Wert „10“ ein, so wird nach ca. 5 min vom BIOS ein Neustart initiiert – bis irgendwann vielleicht sich der Router meldet und der Bootvorgang weitergeht.
- Das Laden der Systemdateien in den unteren und oberen Speicher sind in der „autoexe.bat“, „config.sys“ im Zusammenspiel mit der Datei „himem.sys“ bereits fertig optimiert, sodass für das Programm *hauscomp.exe* maximaler Speicher zur Verfügung steht.
- Alternativ zum DHCP Server kann selbstverständlich auch eine feste IP-Adresse vergeben werden. Der Vorteil: Der PC fährt immer störungsfrei hoch. Die IP-Adresse muss sich aber im Adressraum der angeschlossenen Router bzw. Rechner befinden. Die Adressen dürfen sich also nur im letzten Zahlenwert unterscheiden. Um dauerhaft eine feste IP-Adresse (z. B. 192.168.1.240) zu vergeben, ist im Verzeichnis „/Net/“ die Datei *protocol.ini* folgendermaßen zu korrigieren:

feste IP-Adresse:	automatische IP-Adresse (DHCP):
IPAddress0=192 168 1 240 DisableDHCP=1	IPAddress0=0 0 0 0 DisableDHCP=0

  Bitte beachten, dass die Adressen in der *protocol.ini* nicht durch Punkte, sondern durch Leerzeichen getrennt werden.
- Mit einem anderen Rechner (z. B. mit Win XP), der ebenfalls mit dem Router verbunden ist, im Explorer unter „Extras – Netzlaufwerk verbinden“ anklicken. In dem sich öffnenden Fenster unter „Ordner“ \\Hauscomputer\hauscomp“ eingeben (Groß-/ Kleinschreibung beachten!). Unter Windows 10 versucht Windows zwar einen Verbindungsversuch bei Eingabe von z. B. „\\192.168.1.240\hauscomp“, bricht aber dann mit einer Fehlermeldung ab. Das alte smb-Protokoll wird unter Windows 10 nicht mehr unterstützt (2022).





Ein neues Icon für ein Netzlaufwerk erscheint auf dem Arbeitsplatz. Nach Doppelklick darauf wird man mit dem Board verbunden und befindet sich im Verzeichnis „Hauscomp“. Hier kann man nach Belieben die *ini Dateien verändern, Log- Dateien herauskopieren oder eine neue *hauscomp.exe* einspielen. Nach Doppelklick auf die „index.htm“ startet der Standard-Browser und zeigt die vorbereiteten bzw. sich ständig aktualisierenden auto.bmp, Tagesinf.txt, sps.txt und aktuell.txt an.

Das Netzwerk belastet den Rechner nicht wesentlich. Das regelmäßige Abspeichern der Bildschirminhalte in den 3 autox.bmp ist ab Version 6.0.25 mittels Assemblerkode spürbar beschleunigt und benötigt nur noch eine Sekunde.

Ab der Version 6.0.24-6.0.29 (sowie ab 6.0.34 über den Scriptbefehl „INT21=on“) wird einmal in jedem Zyklus des Programms *hauscomp.exe* der Interrupt 21 aufgerufen. Damit erhalten die Netzwerktreiber (unter DOS sogenannte TSR Programme) die Möglichkeit, das Geschehen im Netz zu aktualisieren. Läuft dagegen anstelle der *hauscomp.exe* ein anderes Anwender-Programm, wie z. B. der Norton Commander, so erfolgt diese Aktualisierung nicht, das Netzwerk „hängt“ sich scheinbar auf, es sei denn, man bewegt sich mit der Tastatur im Commander durch Verzeichnisse und löst dadurch nebenbei den INT21 aus. Auf der DVD wurde der sehr effiziente und freie Volkov Commander („vc“) untergebracht, der ähnlich wie der Norten Commander arbeitet. Das Editieren von Textdateien ist dabei sehr einfach per <F4> möglich.

Die Datei LOGO.SYS enthält das Bild für den Startbildschirm. Sie ist eine umbenannte LOGO.BMP mit 320x400 Pixel und 256 Farben.

Die obige Anleitung für das Kontron Board ist vermutlich auch für andere Boards verwendbar, solange der Netzwerkchip kompatibel ist. Ansonsten bleibt nur experimentieren oder auf DosBox/Dosemu umsteigen, die zugegebenermaßen eleganter aber auch langsamer sind. Das gesamte DOS System benötigt nur ca. 5,5 MB – gegenüber mehreren Gbyte bei Windows oder LINUX und ist zudem äußerst stabil. Nützlich ist auch die Watchdog im BIOS: wird sie aktiviert, kann man den Scriptbefehl „PCWatchdog=“ in der *hardware.ini* einsetzen und erhält einen selbst-startfähigen PC nach Softwareausfall...

Eine spezielle Version 6.0.56GX1 der hauscomp.exe wurde 2022 aus der Version 6.0.56 abgeleitet, um trotz des beschränkten Speicherbereiches im GX1 (inklusive DOS-Netzwerktreiber!) auch Raumsensoren und neuere Funktionen zu nutzen. Dazu mussten allerdings einige andere Funktionen reduziert oder ganz weggelassen werden.. Folgende Einschränkungen gibt es unter 6.0.56GX1:

1) Allgemein

Funktionseinschränkungen	***betrifft Scriptbefehle***
maximal 599 SPS-Module	„z=“
maximal 200 Anzeigen	„Anzeige=“
maximal 200 Texte	„text=“, „textgross=“
kein Heizungslog möglich	„HzgLogxxx=“
keine edbs möglich (damit auch keine Dateien *klima.** möglich)	„EDB=“
kein Datenstream möglich	*taste.txt:* „D …“
Hardware MAX1236 nicht mögl.	„SensorMax1236x=“
Hardware LM26 und DS1631 nicht möglich	„Bussensor=
Hardware AM2320 direkt am Bus nicht möglich	- ohne -
Mechanismus für Neustart unter Linux entfernt	*taste.txt:* „L …“
Datei *_sps.sps* wird nicht geschrieben (nur *sps.txt*)	- ohne -

2) entfernte SPS-Modultypen („z=“)

- 93 Höhenmesser
- 112 Wetterprognose

3) entfernte Anzeigetypen („Anzeige=“)

- 7 Thermometer
- 11 Zeigerinstrument
- 16 Kreis

4) entfernte Anzeigeseiten (im Menü)

- 117 Tagesinfoseite
- 141 Anzeige *hardware.ini*
- 155 Sternhimmel
- 187 Heizungslog
- 189 Sonstiges

5) nicht funktionierende interne Variablen

- f (43) – aktueller UV-Index
- f (44) – Sonnenenergie je m²
- z&f (57) – Temperatur AM2320
- z&f (58) – Feuchte AM2320

6) nicht funktionierende Befehle per *taste.txt*

- „L“, „l“ – Linux Befehle
- „D“, „d“ – Datenstreambefehl

Anmerkung: Im Image GX1_Kontron_512MB6056.img sind zusätzlich in der Datei /net/*protocol.ini* bereits die feste IP-Adresse 192.168.1.240 und in der *hardware.ini* die serielle Schnittstelle COM1 eingestellt. Achtung: Wird der Volkov-Commander für das Editieren der ini-Dateien genutzt (Aufruf per „vc“ unter DOS), so muss er zwingend vor dem Starten von *hauscomp.exe* wieder beendet werden (F10), da sonst der Speicher zu klein ist (→Fehlermeldung „BGI-Error: …“).

1.11.2. Installation DOS inklusive Netzwerk für einen IGEL-3/2 (mit Realtek RTL8100B Netzwerkchip)

- Die Installation erfolgt wie im Pkt. 1.11.1. Alternativ kann der Flash auch direkt mit dem Image IGEL3.2 LAN_DOS.img von der DVD gemäß Pkt. 1.10.5 bespielt werden. Dann benötigt man kein Win98 für die Systemdateien. Das Image ist bereits für die, mit dem IGEL in der Regel mitgelieferte Standard CF Card von 64 MB, ausgelegt. Für die erste Variante befinden sich die relevanten Dateien unter „...\Netzwerk\IGEL3.2 LAN_DOS“ auf der Projekt-DVD. Als RAM Speicher reicht im IGEL natürlich die minimalste Ausstattung – 1 MB reichen bereits.
- Als Standard für den Netzwerkzugriff ist bereits eine feste IP eingestellt (192 168 1 150). Die Angaben in der Datei C:\NET\PROTOCOL.INI lassen sich aber auch leicht für eigene Zwecke anpassen:

```
DefaultGateway0=192 168 1 1
SubNetMask0=255 255 255 0
IPAddress0=192 168 1 150
DisableDHCP=1
```

- Im Testaufbau wurden im BIOS des IGEL (Entf beim Booten) neben der Uhrzeit (MEZ) folgende Veränderungen vorgenommen:

```
Frequency/Voltage Control
    CPU Host Clock (CPU/PCI): 100/33 MHz
    Spread Spectrum: Disabled
```

- Tastatur: um im BIOS Änderungen vornehmen zu können, kann man sowohl eine PS2 als auch eine USB Tastatur anschließen, und zwar unabhängig von der BIOS Einstellung „USB Keyboard enabled/disabled“. Unter DOS kann man mit dieser BIOS Einstellung (enabled) sogar mit einer USB Tastatur arbeiten. Leider wird dabei zusätzlicher Speicherplatz benötigt, sodass das hier behandelte Netzwerk Image mit einer Fehlermeldung beim Hochfahren abbricht. So bleibt also nur der Anschluss einer PS2 Tastatur (sofern man überhaupt eine benötigt).
- Die Startzeit vom Stromanlegen an den IGEL bis zur Aktivität des Hausbusses liegt bei nur ca. 16s (!), wobei allein das Hochfahren des BIOS 10s beansprucht. Die Leistungsaufnahme im Betrieb liegt bei ca. 13,7 W.
- Die USB-Anschlüsse können wegen der fehlenden Treiber zwar nicht genutzt werden, das sollte aber kein Problem sein.
- Insgesamt ist das System mit einer Zykluszeit von etwa 0,15s wahnsinnig schnell. Beim späteren Dauerbetrieb brauchen zudem weder Maus, Tastatur noch Bildschirm angeschlossen sein. Die gesamte Wartung der Software kann über das Netzwerk erfolgen, es muss kein Linux sein!

1.11.2. Installation DOS inklusive Netzwerk für einen IGEL-5/4 (mit VIA VT6102 Netzwerkchip)

- Die Installation erfolgt wie im Pkt. 1.11.2. Im DVD-Ordner „/Netzwerk/IGEL 5–4 LAN DOS“ befinden sich die bereits vorkonfigurierten Dateien. Das Netzwerk lief sowohl auf einem 800 MHz als auch auf einem 1 GHz IGEL erfolgreich.

2. Programmbedienung

2.1. Programmstart und Bedienung

Nachdem der künftige Haus-PC mit dem Interface (siehe [2])verbunden und die Stromversorgung +12V angeschlossen wurden, muss nur noch die Bootdiskette (oder bootbarer USB-Stick oder Flashdisk) eingelegt und der PC gebootet werden. Die Diskette wird wie folgt vorbereitet: Betriebssystem aufspielen (siehe liesmich.txt) und alle Dateien aus Verzeichnis „Software“ auf Diskette kopieren. Danach öffnet man die Datei *hardware.ini* mit einem Texteditor. Darin sollten folgende Programmzeilen zu finden sein:

```
COM=1
Sommerzeit=auto
Seitenwechsel=10
Watchdog= 3 7
Laufschrift=on
```

Wichtig ist, dass der richtige COM-Port und die richtige Adresse bzw. Kanal der Watchdog angegeben sind. Notfalls werden sie geändert.

Nach dem Booten des Rechners sollte das Programm *hauscomp.exe* automatisch ohne Fehlermeldung starten. Ein „RUNTIME ERROR“ bedeutet Syntaxfehler in den *.ini – Dateien. Auf der ersten (Bildschirm-) Seite werden bei korrekt funktionierendem Hausbus die Hausparameter angezeigt. Die Uhrzeit sollte korrekt sein. Da wegen der Vielzahl verschiedener Grafikkarten ohne speziellen Treiber nur 16 Farben möglich sind, ist die gesamte Farbpalette überschaubar. Die Farbe 12 ist als rotes Blinken programmiert und wird bei Bereichs-Überschreitungen in Anzeigen eingesetzt. Mit der Taste <i> kann die aktuelle Farbdarstellung invertiert werden um z. B. ein druckbares Screenshot zu erzeugen. Mit <ALT><F4> kann man jederzeit das Programm beenden. Die Abschaltung dauert ein paar Sekunden, da das Programm noch die Hardware zurücksetzen muss.

Kleinere Korrekturen der Uhrzeit kann man über den Menüpunkt „System/Uhrzeit..“ erzwingen. Für größere Korrekturen (Abweichung größer 3 min) bitte nach Beendigung des Programms an der DOS-Eingabeaufforderung die Befehle „DATE“ bzw. „TIME“ nutzen. Da auf die Datei „Country.sys“ verzichtet wurde, ist die Darstellung auf dem Bildschirm unter DOS, sowie die Tastaturbelegung englisch. Für eine Korrektur des Datums z. B. auf den 14. Mai 2004 ist an der Eingabeaufforderung zu schreiben: „Date“ <ENTER> 5ß14ß2004 <ENTER>. Die Uhrzeit setzt man mittels: „Time“ <ENTER> 22ö13 <ENTER>. Bitte bei der Korrektur der Uhrzeit die Sommerzeit berücksichtigen! Die *hauscomp.exe* sollte normalerweise mit der automatischen Korrektur der Sommerzeit laufen. Dabei muss man die PC-Uhr immer auf MEZ setzen, also im Sommer anstelle von 20:12->19:12 angeben und in der *hardware.ini* den Scriptbefehl „Sommerzeit=auto“ eintragen. Für den erneuten Programmstart gibt es zwei Möglichkeiten: PC aus und wieder einschalten oder von der Eingabeaufforderung „Hauscomp“ und <ENTER>.

2.2. Bedienung/ Tastaturbelegung

- Ziffern 0-9 (48..57)- Bedeutung entsprechend der aktuellen Menüseite
- Raute <#> (35..), sowie Escape <ESC> (27) - Zurückschalten auf die jeweils höhere Ebene im Menü
- Stern <* > (42) - Zwangsumschaltung auf Automatikbetrieb, auch bei abgeschaltetem „AUTO“ (habe ich aber nie gebraucht)
- <i> (105) - Invertierung des aktuellen Farbschemas (sinnvoll bei Screencopys)
- <N> (78) - Speicherschutz einschalten (ab sofort werden keine Daten mehr auf dem Datenträger gespeichert) – dient zum sicheren Beenden bei gewollter späterer Zwangsabschaltung der DOSEMU. Fehlerhafte Sektoren werden so vermieden (**N**ichts speichern).
- <W> (87) - Speicherschutz wieder deaktivieren (**W**ieder erlauben)
- <a> (97) - schnelles Ein- und Ausschalten des Automatikmodus
- <g> (103) - erzeugt (löscht) ein Gitter auf dem Bildschirm zur Bestimmung der Pixel-Koordinaten x und y, unsichtbare Objekte werden sichtbar; mit Kursor-, Tab-, Bildauf- und Bildabtasten können Anzeigen variiert werden, ohne Speicherung
- <l> (108) - speichert die aktuelle Laufschrift in „Laufband.log“ ab, wenn in *hardware.ini* LLog=on gesetzt ist
- <o> (111) - entspricht der Betätigung der programmierten „OK“ Taste am Interface (Reset Hinweistext in der Laufschrift)
- <Druck> - unter DOS: s/w Bildschirmcopy auf Drucker an LPT1 (darf kein GDI-Drucker sein!) unter WINDOWS: kopiert den Bildschirminhalt in Zwischenablage, besser ist hier <Shift>+<Druck>; Funktion ab Windows XP nicht auf allen PC garantiert. Alternative: Taste <p>
- <p> (112) - speichert Screenshot als Bild-Datei <Seitennummer>.bmp (verlustfrei komprimiert mit RLE)
- <r> (114) - erneutes Laden der *.ini Dateien ohne Programmneustart (Restart). Sinnvoll nach Änderungen übers Netzwerk.
- <s> (115) - Speichern der aktuellen SPS Werte in sps.txt. Aktualisierung der Betriebsstundendatei „stunden.txt“.
- <t> (116) - manuelles Einschalten des statischen Bildschirmmodus (Grafikaktualisierung nur bei Seitenwechsel)
- <+> (43) - Umschalten im Menü auf die nächste Seite (+1)
- <-> (45) - Umschalten im Menü auf die vorangehende Seite (-1)
- <ALT>+<F4>; <k> (107) - Programmende
- <Leertaste> (32) - Übergang in Menüauswahlmodus; Aktivierung aktuell ausgewählter Menüpunkte bzw. Rücksprung im Menü
- Kursortasten (72, 80, 75, 77) - im SPS Menü: Auswahl eines SPS-Moduls; bei "Gitter Anzeige": Positionierung der Anzeigen (zusammen mit Tabulator); im Menüauswahlmodus: auf/ab (Auswahl) entsprechender Menüpunkte

- <Entf (Del)> (83) - im SPS Menü: Betriebsstundenzähler auf null setzen
- <Enter> (13) - (im SPS Menü) für den Probebetrieb der SPS gedacht:

  manuelles Umschalten des Zustandes eines SPS- Moduls, das man vorher mit den Kursortasten ausgewählt hat. Der Zustandsausgang dieses Moduls kann dabei dauerhaft ein- bzw. ausgeschalten werden. Es gilt folgende Reihenfolge: Automat → Hand/AUS → Hand/EIN → Automat. Der jeweilige Status ist an der wechselnden Farbe erkennbar. Die Umschaltung wird auch nachts um 0:00 Uhr nicht zurückgesetzt. Damit kann die SPS dauerhaft (bis zum nächsten Stromausfall) beeinflusst werden.

  1. Einschränkung: Diese manuelle Umschaltung funktioniert nicht an Modulen des Typs 42 (astabiler Multivibrator mit zufälligem Schaltverhalten) da diese Module von einer internen Ablaufsteuerung geschaltet werden.

  2. Einschränkung: Bei Modulen, die analoge oder binäre Kanäle auf dem Hausbus ansteuern, wird nur der Ausgangszustand des entsprechenden Moduls beeinflusst, nicht aber der angesteuerte Buskanal! Die betroffenen Kanäle verbleiben während der Handschaltung im vorherigen Zustand. Um diese Kanäle trotzdem zu beeinflussen, muss man den vorangehenden (Quell-) Modul anwählen und entsprechend umschalten.

  - auf den Seiten 8 und 9 (par/anz -Anzeige) sowie 120-129 (Analogsensoren) zum Blättern verwendet

Alle verwendeten Tastaturkodes sind auch in der Tabelle im Punkt 4.7.9 gelistet.

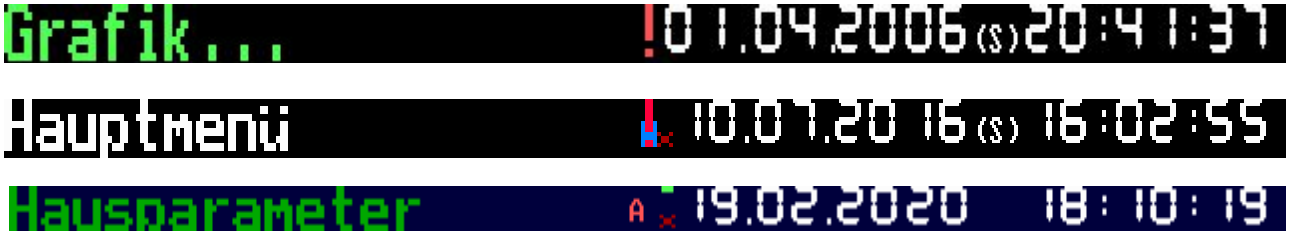
2.3. Die Bildschirmelemente

Nach dem Programmstart wird auf dem Bildschirm immer eine der 256 möglichen Bildschirm- bzw. Menüseiten dargestellt. Der Bildschirm ist einheitlich aufgebaut und besteht aus dem Kopfteil, dem Bereich für die Laufschrift sowie dem restlichen Anzeigefeld. Einzige Ausnahme ist die Seite „255“, sie bewirkt die komplette Abschaltung der Grafik.



2.3.1. Der Kopfteil

Beispiele:



Der Kopfteil besteht von links nach rechts aus der Seitenbezeichnung, dem blinkenden Zeichen „A“ (wenn „AUTO“ abgeschaltet, bzw. kein automatischer Seitenwechsel aktiv ist), dem blinkenden Ausrufezeichen „!“ (beim Auftreten des Signals „HINWEIS!“), dem blauen Zeichen „H“ (wenn mindestens 1 SPS Modul auf HAND geschaltet ist), dem Datum, dem Zeichen „(S)“ bei Sommerzeit, sowie der aktuellen Uhrzeit. Die Doppelpunkte der Uhrzeit blinken bei Zeitinkonsistenz rot.

Zusätzlich gibt es kleine Kreuze/Striche mit folgenden Bedeutungen:



- rotes Kreuz: am Hausbus zu starke Störungen der Signale SDA bzw. SCL, in dieser Zeit ist der Hausbus nur im Testmodus.



- gelbes Kreuz: Die Speicherung von Daten auf dem Datenträger ist untersagt – z. B. weil der Datenträger kopiergeschützt ist oder per „N“ die Speicherung abgeschaltet wurde.



- horizontaler Balken am oberen Rand: Das automatische Lesen aus der Datei **datain.txt** ist erlaubt und erfolgte korrekt (grün) bzw. fehlerhaft (braun).

Auch im statischen Anzeigemodus wird der Kopfteil immer aktualisiert.

2.3.2. Die Laufschrift

In der Zeile darunter ist die Laufschrift angeordnet, die vollautomatisch generiert wird. Sie kann in verschiedenen Modi laufen, die über den Scriptbefehl „Laufschrift=“ in der Datei *hardware.ini* festgelegt werden. Es sind sowohl eine laufende Schrift als auch eine stehend, sich abwechselnde Schrift möglich. Die Schriftfläche wird bei der Programmentwicklung auch zu Testzwecken genutzt. Als Besonderheit gilt der statische Betriebsmodus. Ist er aktiv, so wird der Text:

anstelle der Laufschrift eingeblendet.



2.3.3. die restliche Anzeigefläche

Die restliche Anzeigefläche wird automatisch befüllt – in Abhängigkeit der Seitennummer (Bildschirmseite) und damit auch von den vielfältigen Einstellungsmöglichkeiten in der *hauscomp.ini*. Die Zuordnung der Seiten zu bestimmten Funktionen ist im Abschnitt 2.4. erläutert.

In der unteren, rechten Bildschirm-Ecke erscheint im Handmodus die Nummer der aktuellen Bildschirmseite. Sie verschwindet, wenn das Programm wieder zurück in den Automatikbetrieb schaltet oder der Echtzeitmonitor aufgerufen wurde. Die Farben von Objekten auf dem Bildschirm, sowie des Hintergrunds werden vom Scriptbefehl „Hintergrund=“ bzw. der Taste <i> gesteuert. Die Größe der Menüs wird über den Scriptbefehl „Menüform=“ eingestellt. Wenn das Programm im automatischen Betrieb läuft, sich also 3 Seiten regelmäßig abwechseln, kann durch die SPS gesteuert eine Alarmseite aufgerufen werden. Sie verschwindet erst wieder durch Betätigung der Taste „o“, „OK“, der Leertaste oder einer (mit der Leertaste gekoppelten) Taste am Hausbus, bzw. einem entsprechendem Befehl über das Netzwerk (siehe auch Syntax der Datei „taste.txt“ unter Pkt. 2.11.).

Zur Gewinnung von Rechenzeit, d. h. zur Verringerung der Zykluszeit wurde ab Version 6.0.38 der statische Anzeigemodus eingeführt. Dabei wird die Grafik nur noch beim Seitenwechsel und bei Eingaben aktualisiert, was beim Netzwerkbetrieb nicht weiter tragisch ist, da sowieso kein Bildschirm zu sehen ist. Die Folge ist eine Reduzierung der Zykluszeit auf weniger als ein Drittel. Folgende Bedingungen führen zum Einschalten des statischen Anzeigemodus:

- Betriebssystem=Dosemu (Linux) UND mindestens eine Eingabe über die Datei „taste.txt“ (Netzwerkzugriff) UND langsamer PC (zk<3000)
- Bei allen Betriebssystemen: 40 Sekunden nach letzter Tastenbetätigung, wenn die interne Zeit-Abweichung (Zeitkern: Dalles/Timer) > 1,5 % beträgt UND langsamer PC (zk<3000). Das heißt bei einer Abweichung der internen Uhren voneinander von mehr als 20 Minuten am Tag.
- Eine nicht autorisierte Eingabe über die Datei „taste.txt“ (Netzwerkzugriff nicht als Admin auf gesperrte Funktionen)
- Manuelles Auslösen über die Taste „t“
- Automatisch 40 Sekunden nach letzter Tastenbetätigung, wenn der Scriptbefehl „Laufschrift=statisch“ in *hardware.ini* vorhanden ist UND langsamer PC (zk<3000)

Abgeschaltet wird der Modus, sobald wieder eine Taste auf der Tastatur (vor Ort) gedrückt, eine Bustaste betätigt wurde oder bei langsamen Rechnern, wenn die interne Zeitabweichung wieder 1 % unterschreitet

2.4. Die Menü-Struktur

Die Bedienung des Programms im Handbetrieb orientiert sich an einer Menüstruktur, die ausgehend vom Hauptmenü (Bildschirmseite 1) alle anderen Bildschirmseiten erreichen lässt. Durch Betätigung der linksseitig auf dem Bildschirm sichtbaren Ziffer auf der Tastatur wird die entsprechende Aktion ausgelöst. Durch Betätigung der Raute- Taste „#“ gelangt man in der Regel wieder zum vorangegangenen Menü. Wird die Tastatur längere Zeit nicht betätigt (45 s), so schaltet sich der automatische Bildschirm-Wechsel-Modus ein, sofern er über das Menü „System“ bzw. der Taste „a“ nicht explizit deaktiviert wurde. Mittels der Taste „*“ kann man den automatischen Modus sofort erzwingen. Damit genügen zur Steuerung des Programms 12 Tasten auf der PC Tastatur. Diese Art der Programmbedienung hat ihren Ursprung im Wunsch, den Hauscomputer einfach über das Haustelefon zu steuern. Die Telefon-Ankopplung ist allerdings hardwareseitig nicht implementiert. Egal in welcher Betriebsart das Programm läuft, die SPS im Hintergrund schaltet und regelt ununterbrochen weiter.



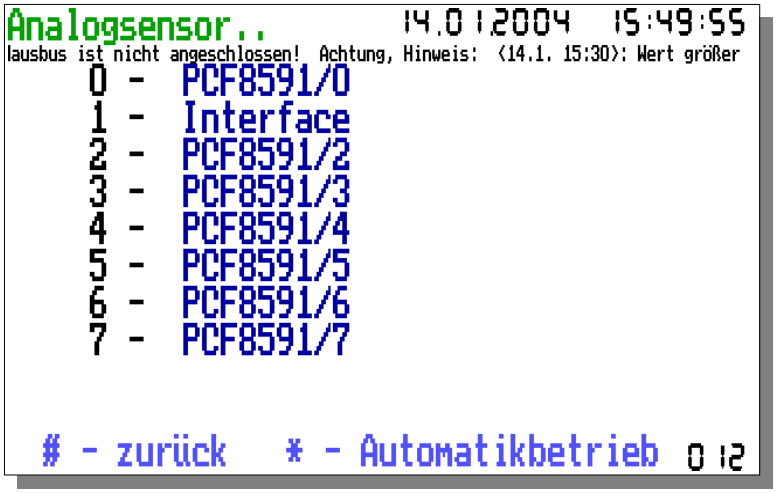
2.4.1. Der Menüauswahlmodus

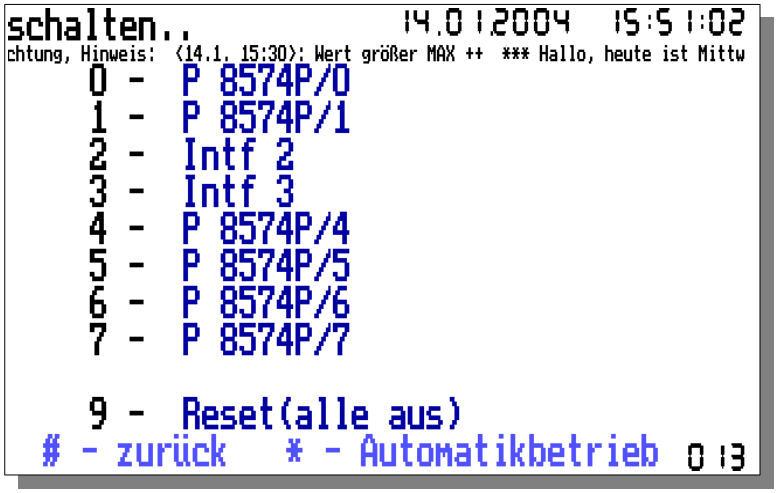


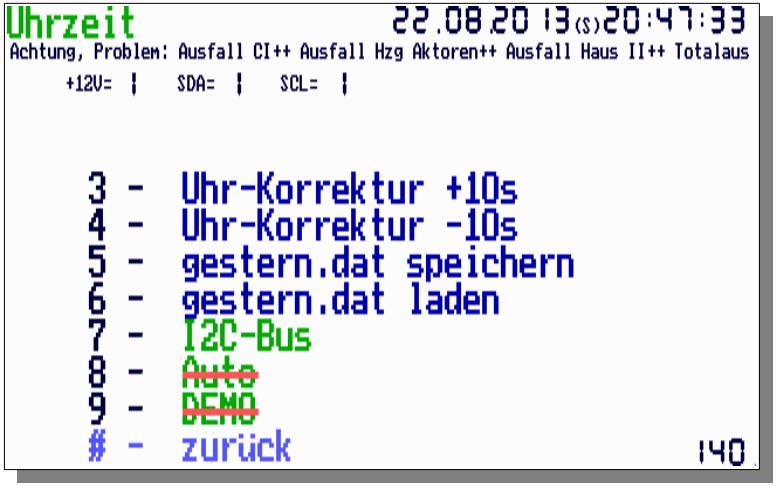
Eine weitere Möglichkeit der Steuerung ist der Übergang in den Menüauswahlmodus – dabei kann man mit zwei Kursortasten auf der PC-Tastatur oder vom Control Interface (CI) aus, den entsprechenden Menüpunkt ansteuern und mit Betätigung der Leertaste auch auslösen. Die entsprechenden Menüpunkte sind farblich gelb markiert. Der Menüauswahlmodus aktiviert sich automatisch nach erstmaligem Betätigen der Leertaste und endet mit dem Übergang in den Automatikmodus. In Verbindung mit der Kopplung von Tasten am Hausbus zu entsprechenden Tasten auf der PC Tastatur (Scriptbefehl „Binär=..“) lässt sich das Programm damit auch abgesetzt vom Rechner mit minimal 2 Tasten steuern (Auslösen und Kursor nach unten). Das speziell für diesen Zweck entwickelte CI [2] erlaubt eine Programmsteuerung ohne PC-Tastatur.

2.4.2. Untermenüs

Die meisten Menüs sind selbsterklärend. Man hangelt sich mit den entsprechenden Tasten durch die Seiten.







Eine Besonderheit bietet die Bildschirmseite „Uhrzeit“. Hier stehen die grün dargestellten Aktionen „I2C-Bus, Auto, DEMO“ für Steuervariable. Die Betätigung der Ziffer 8 auf der Tastatur schaltet z. B. „Auto“ aus bzw. wieder ein. Im Zustand AUS wird ein roter Balken über die Variable gelegt. Das Laden der gestern.dai/.dar/.dab führt zum Überschreiben der aktuellen Grafik mit den Daten vom Vortag. Sollte das Datum der Dateien älter als ein Tag sein, ist zusätzlich vorher „DEMO“ zu aktivieren. So können beliebig andere „gestern.dat“ dargestellt werden. Nach einer ununterbrochenen Laufzeit von mehr als 7 Tagen wird zusätzlich eine Information zur Zeitkorrektion (Wochenkorrektion=..) eingeblendet.

2.4.3. Analogsensoren-Untermenü

Über das Menü „Analogsensoren.“ erreicht man jeden der 40 möglichen analogen Sensoren. Dargestellt werden neben einigen Parametern, wie Min- und Max- Werte, die Eichgrafik und, so sie programmiert, die grafische Tagesübersicht mit Masstabslineal. Auf alle angezeigten Parameter kann man in der Programmierung der SPS zugreifen. Der Modultyp 3 gestattet z. B. über den Parameter „Typ“ nicht nur auf den aktuellen oder geglätteten Wert zuzugreifen, sondern auch die Änderungsgeschwindigkeit bzw. Wertänderung einer Temperatur zu analysieren.

Der nebenstehende Screenshot verdeutlicht die möglichen Bildelemente, er entstand während eines Testlaufes mit virtueller Hardware. Es war also nicht wirklich eine Kühltruhe angeschlossen.

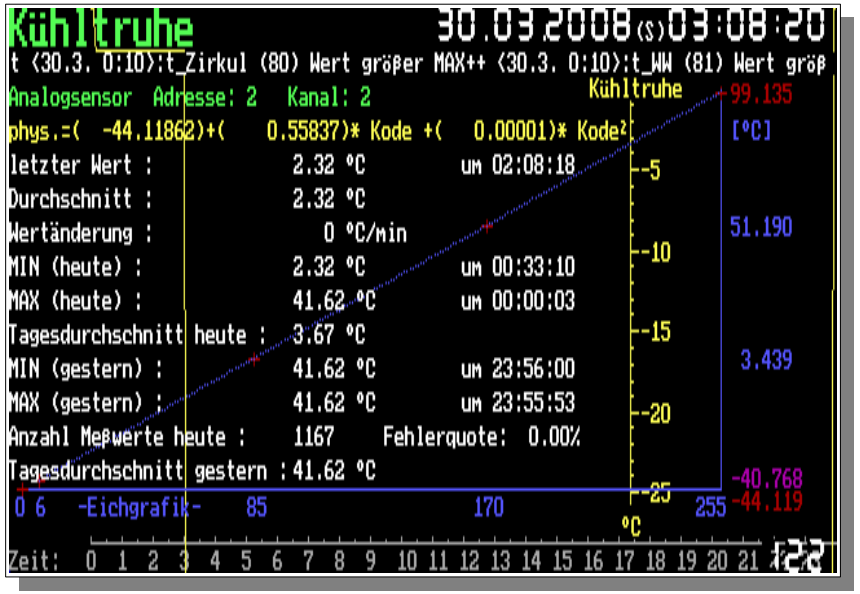
Die Eichgrafik ist die Zuordnung der physikalischen Parameter zu den Kodewerten des A/D Wandlers bzw. des Hausbusses. Die rot dargestellten Werte sind die automatisch an Hand „Sensor=..“ ermittelten physikalischen Grenzwerte.

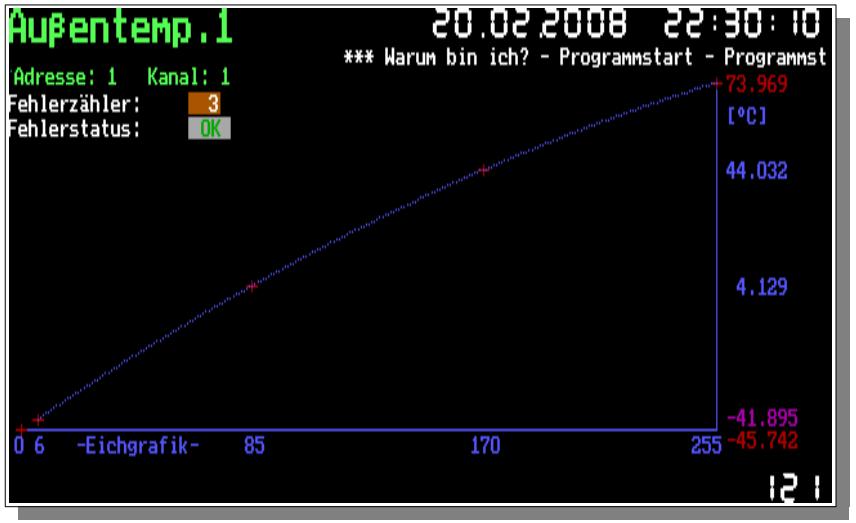
Zusätzlich wird der physikalische Wert bei 6 Kodeeinheiten dargestellt. Dieser Wert entspricht in etwa der niedrigsten Ausgangsspannung beim Einsatz von OPV's. In diesem Grenzbereich ist die Elektronik sozusagen am Anschlag. Die in der *hauscomp.ini* tatsächlich programmierten Kalibrierwerte sind nicht dargestellt. Sie dienen beim Programmstart lediglich zur Ermittlung der Berechnungsformel:

$$physikalischer\ Wert = k0 + k1 \cdot Kodewert + k2 \cdot Kodewert^2$$

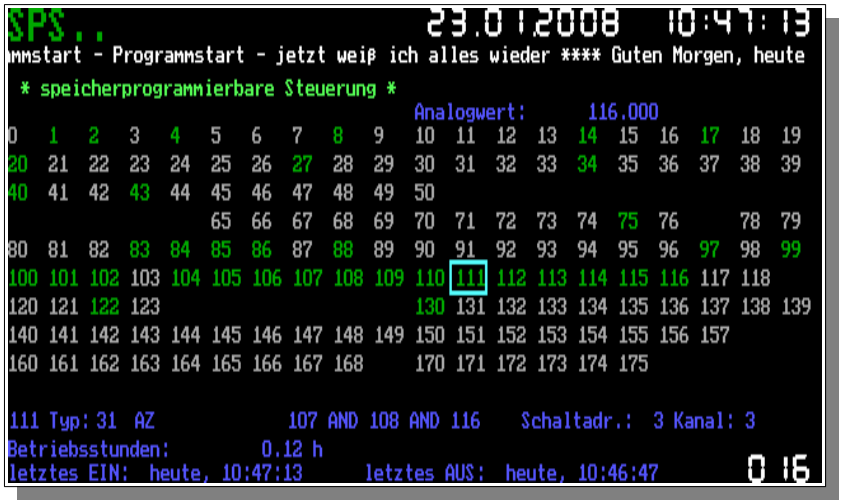
k0 ist die Verschiebung, k1 der (mittlere) Anstieg und k2 der Grad der Krümmung. Bei k2=0 ist die Konvertierung linear.

Sollte jemand noch mehr Informationen zur Kalibrierung oder Fehlersuche an einem Analogkanal benötigen, so braucht er nur mehrfach die Enter- Taste betätigen. Auf der letzten entsprechenden Seite wird der Fehlerzähler und der Fehlerstatus eingeblendet. Der Fehlerzähler kann max. 10 erreichen. Jeder richtige Zugriff zum Sensor verringert den Zähler um 1, jeder falsche erhöht ihn um 1. Der Fehlerstatus wird aus dem Fehlerzähler berechnet.





2.4.4. SPS-Untermenü



Die Bildschirmseite „SPS..“ bietet eine optische Übersicht über alle SPS-Module. Mit den Kursortasten erreicht man jeden der 1199 Module. Der jeweilige SPS-Ausgangszustand (digitaler Ausgang) ist farbkodiert: grau: Automat-AUS; grün: Automat-EIN; blau: Hand-AUS; ws/sw: Hand-EIN. Nicht programmierte Module sind ausgeblendet. Der analoge Ausgang der SPS (real-Wert) ist oberhalb der Matrix zu sehen (Analogwert:..). Unterhalb der Matrix werden typgebunden die SPS-Parameter angezeigt. Hier findet man auch die entsprechenden Betriebsstundenanzeigen, sowie Zeit-Angaben zum Schaltverhalten. Mit der Enter-Taste lässt sich der digitale Modulzustand zwischen Automat-Hand/EIN-Hand/AUS umschalten. Dieser Status bleibt dauerhaft erhalten. Wird ein Modul z. B. auf Hand/AUS geschaltet, so sind die Verknüpfungen, die das Modul durch die Programmierung einnehmen würde, deaktiviert. Diese Funktion dient der Erleichterung von Simulationen in der Testphase eigener Anwendungen.

2.4.5. Die Tagesinfoseite

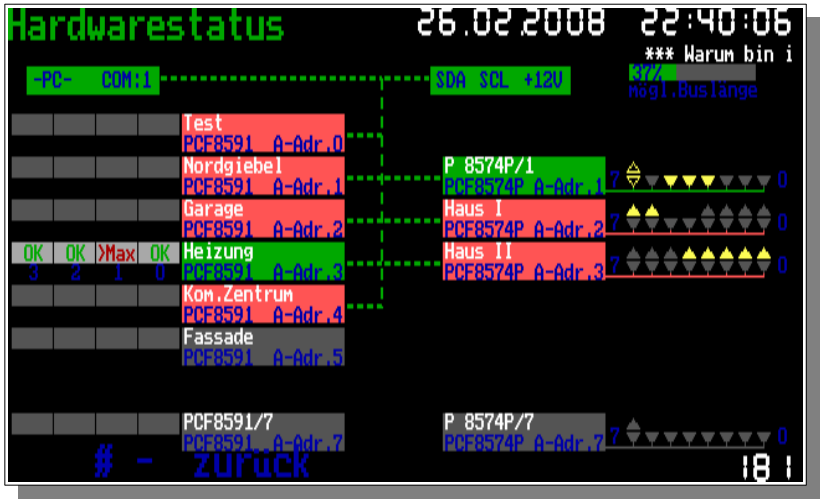


Die Tagesinfoseite gibt neben der berechneten Sonnendaten zusätzlich geschätzte Informationen zur Ultraviolettstrahlung der Sonne. Dieser UV-Index wird in Abhängigkeit von der Sonnenhöhe errechnet und bezieht sich auf eine durchschnittliche europäische Luftmasse bei wolkenlosem Himmel, auf einen Standort nicht höher als ca. 500 m über NN. Der tatsächliche UV-Index hängt u. a. von der Konzentration des Ozons in den höheren Schichten der Atmosphäre ab und unterliegt Schwankungen bis ca. 30 %. Der UVI steht als laufend berechneter Wert im Modul 43 zur Verfügung. Die maximale Sonnenscheindauer bis zu einem Sonnenbrand beziehen sich auf ungeschützte, nicht sonnen gewöhnte Haut entsprechend der 4 Hauttypen.

Die Tageseigenschaft ergibt sich aus der Datum- Bereichsprogrammierung wie z. B. Urlaub oder abwesend. Da diese Eigenschaft als Zahlenwert im (internen) SPS-Modul Nr. 16 bereitgestellt wird, lassen sich z. B. Heizungsabsenkungen über längere Zeiträume durch einfache Verknüpfungen in der SPS realisieren.

2.4.6. Funktionen-Untermenüs

Ab Version 6.0.19 wurde der bisherige Hausbus-Menüpunkt völlig überarbeitet. Er gliedert sich jetzt in mehrere Untermenüs. Das Untermenü „Hardwarestatus“ zeigt grafisch den Einsatzzustand aller Hardwarekomponenten an. Die Philosophie ist einfach: Ist alles „OK“ wird die entsprechende Baugruppe grün dargestellt, Ausfälle blinken rot und „keine Aussage möglich“ ist grau. Die Hierarchie geht dabei vom PC über den Bus, den Interface-ICs zu den Sensoren. Natürlich ist die Detektierung von Ausfällen nicht im I^2C Protokoll verankert. Das Programm wertet deshalb bei der Analyse alle zur Verfügung stehenden Daten in Echtzeit aus. Dabei spielen sowohl die am Bus gemessen Signale, als auch die in der *hauscomp.ini* programmierten PCFs und Sensoren eine Rolle. Die gemeinsamen Ein-/Ausgänge der PCF8574 werden getrennt nach „gemessenen“ Signalen am Eingang und vom Programm „gewünschten“ Signalen am Ausgang als Dreiecke stilisiert angezeigt. Diese Seite kann sehr gut bei Inbetriebnahme und Fehlersuche an der Hardware verwendet werden.





Der Anzeigebalken oben rechts („mögl. Hausbuslänge“ ab 6.0.48: „Taktzeit DT“) misst in Echtzeit die Impulslänge am Hausbus (Bustakt) bezogen auf den Sollwert von 180µs=100%.

Das Untermenü „A/D Wandler“ zeigt übersichtlich die Datenströme der analogen ICs auf dem Hausbus. Dargestellt sind die jeweils 4 Kanäle der 8 möglichen PCF8591 sowie der digital/analog-Ausgang bzw. die Kanäle der speziellen Bussensoren. Ganz rechts ist der Sensortyp ersichtlich:

0,1 – n/a 1 – PCF8574P 2 – LM75 3 – DS1631 4..7 – MAX1236

8,9 – ATtiny26

Die angezeigten Werte der analogen Kanäle sind sogenannte „Kodewerte“ – das über den Bus übermittelte digitale Ergebnis der Digital-Analog Wandlung im Schaltkreis. Dieser Wert wird für die Kalibrierung von Sensoren benötigt. Ab Version 6.0.40 werden nicht initialisierte Adressen (Typ 0) weder abgefragt noch angezeigt.

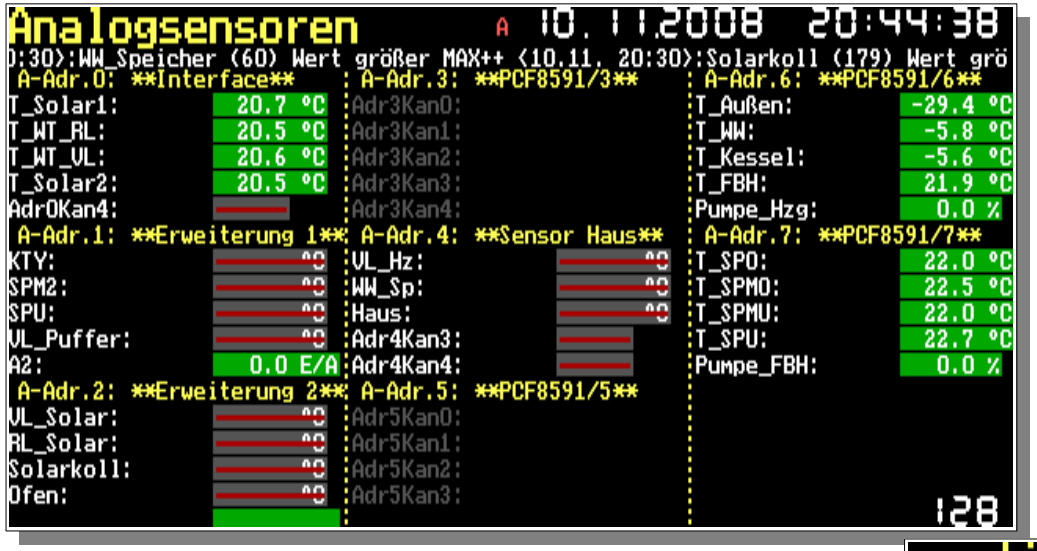
Das Untermenü „Festlegungen“ enthält Informationen über festgelegte spezielle Ein-/Ausgänge wie Watchdogs, Bustasten usw.

Im unteren Bereich befindet sich eine Übersicht der an die ATtiny 26 übertragenden Startwerte, sowie der Erfolg der Übertragung. Die übertragung erfolgt einmalig beim Programmstart, als auch täglich zum Tageswechsel um 0:00 Uhr. Diese Werte werden aber erst aktiv, sobald die Watchdog der betroffenen ICs anspricht – in der Regel beim Ausfall des PCs oder des Hausbusses. Im linken Beispiel bedeutet das: Der binäre Kanal 7 der Adresse 7 ist dann „ein“ und der PWM Kanal 1 der analogen Adresse 1 taktet in dem Fall des Ausfalls des übergeordneten PCs mit 25 %.



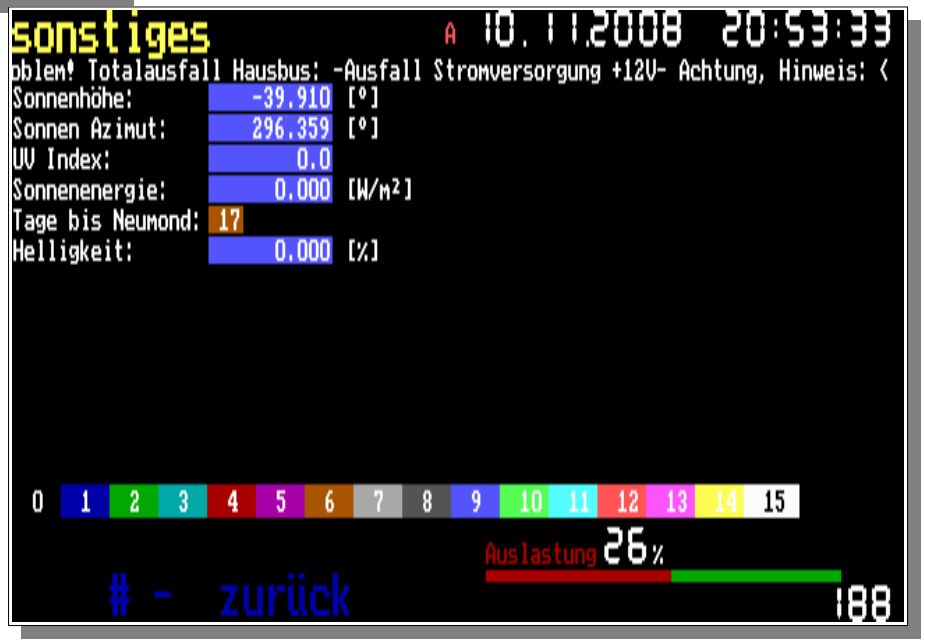
Das Untermenü „I/O Kanäle“ listet alle 64 möglichen digitalen Ein-/Ausgänge der PCF8574. Physikalischer „low-Pegel“ bedeutet „EIN“. Angezeigt wird der Status am Pin der ICs, egal ob das Programm ein „EIN“ erzwingt oder ein externes Signal anliegt. Schaltkreise, die sich nicht melden, sind grau dargestellt.



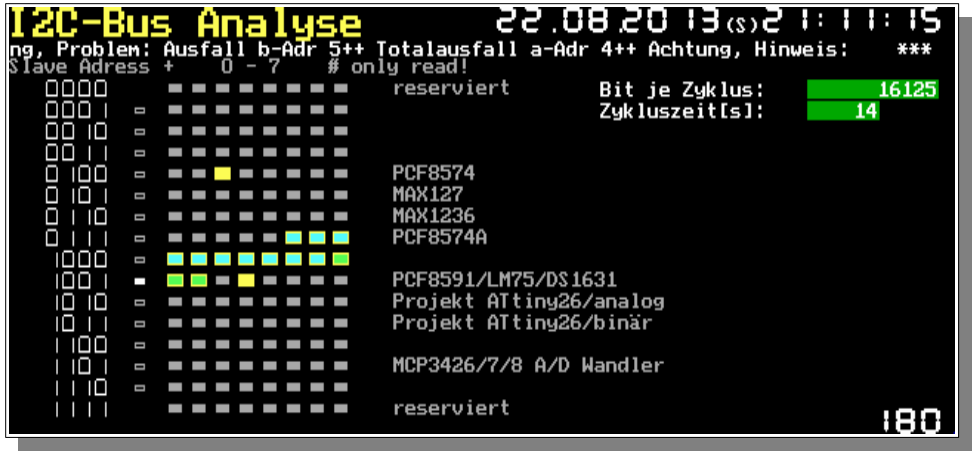


Ähnlich aufgebaut ist der Menüpunkt „Analogsensoren“. Auf einen Blick sind alle 32 Sensoren und 8 D/A Wandler nach Adressen geordnet zu sehen. Nicht definierte Adressen werden ausgeblendet. Definierte, aber sich nicht meldende (damit offenbar ausgefallene) Sensoren sind mit einem roten Balken versehen.

Das Untermenü „sonstiges“ ist eine Sammlung nützlicher Details, die nicht anderweitig zugeordnet werden konnten. Am unteren Bildschirmrand wird die aktuelle Farbpalette dargestellt. Die entsprechenden Zahlen dienen als Referenz zur Farbkodierung in der *hauscomp.ini*. Das Element „Auslastung“ zeigt an, wie viel Prozent der Rechenzeit aktuell für Aktivitäten auf dem Hausbus verwendet werden.



Das Untermenü „I2C-Busanalyse“ hat 2 Funktionen. Zunächst werden mit der Aktivierung dieser Seite vom Programm zusätzlich zum normalen Busverkehr zyklisch alle möglichen I2C Adressen angesprochen und dabei registriert, ob sie sich melden. Das Ergebnis wird grafisch dargestellt. Die zweite Funktion ist, dass der I2C-Bus diese Adressen mit der doppelten Taktfrequenz und damit unter erschwerten Übertragungsbedingungen bearbeitet. Damit ist eine qualitative Aussage zur Stabilität des Hausbusses bei laufendem Betrieb und bei gegebener Leitungslänge möglich. Die Abfrage der I2C Adressen erfolgt dabei völlig unabhängig von der normalen Busaktivität des Programms, vom zusätzlichen Zeitbedarf mal abgesehen.



Im nebenstehenden Beispiel meldeten sich 2 I2C Schaltkreise - einmal mit der Adresse 0100010 und ein anderer mit der Adresse 1001011. Zur besseren Übersicht sind die 3 niederwertigen Bits nicht bezeichnet worden, sondern als 8 Karos nebeneinander dargestellt. Das heißt, das linke Karo entspricht „000“ und das rechte „111“ der niederwertigen Bits. Zugleich entspricht diese Darstellung der im Programm verwendeten Form der „Adressen“ von 0 bis 7. Somit haben sich ein binärer IC mit der Adresse 2 und ein analoger IC mit der Adresse 3 gemeldet.

Ist die Busverbindung nicht stabil oder geht die Verbindung zu einem IC verloren, so wechselt die Farbe der Karos. Erst wenn der entsprechende IC sich 4 mal in Folge korrekt gemeldet hat, wird das Karo wieder gelb. Wurde eine Adresse auch nur einmal detektiert, so bleibt das zugehörige Karo immer optisch erkennbar. Für das obige Bild wurde der Bus kurz elektrisch getrennt.

Ergänzend wird die Zykluszeit eingeblendet. Gemeint ist damit die Zeit, die benötigt wird, um alle analogen Kanäle abzufragen. Sie sollte 13-15s betragen (256 * 55 ms = 14,08 s). Bei größeren Abweichungen passen Betriebssystem und Bustakt nicht zusammen. Ab Version 6.0.40 sind die zeitlichen Reihenfolgen auf dem Bus wesentlich komplexer geworden, sodass nun nur noch entscheidend ist, ob es Datenausfälle gibt oder nicht.

Sind ATtiny26 am Bus vorhanden, so kann es bei zu schnell eingestelltem Bustakt (siehe auch Scriptbefehl „Bustakt=“) zu Störungen der Kommunikation kommen, da die Firmware der ATtinys nicht ganz so flott ist. Das gleiche trifft natürlich auch auf zu lange Bus-Leitungen zu.

Beim Verlassen des Menüs wird die Adressabfrage eingestellt und der Bus arbeitet wieder mit normaler Geschwindigkeit.

Im Untermenü „Raumsensoren“ werden automatisch alle am Bus aktiven Multisensoren gelistet (ab Version 6.0.46). Der Aufbau der Seite kann sich etwas verzögern, da der Umfang der Informationen aus Gründen der Performance nur alle 3 s berechnet wird. Auch müssen bei der erstmaligen Aktualisierung erst die Sensordaten validiert werden.



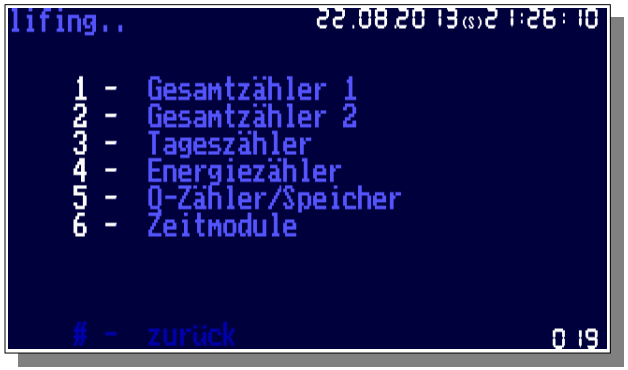
Letztlich hat man aber alle relevanten Informationen im Blick: Neben den Zustandsinformationen (Signal „LAMP“, Relais, Hinweis, Bewegungsmelder und Alarm) werden die wichtigsten Parameter direkt angezeigt. Die absolute Feuchte und der Taupunkt werden dabei zusätzlich vom Programm berechnet.

Für den Sollwert bzw. die Helligkeit wird der vom Raumsensor gemessene ADC-Wert in Kodeeinheiten dargestellt, also das 6. vom Sensor übertragenem Datenbyte. Bei vorhandenen Wärmemengenmessern wird in der Spalte CO2/Wind die aktuelle Durchflussmenge [l/h] angezeigt.

Am Ende erfolgt noch die Anzeige des Abfrageintervalls aller Sensoren. Dieses Intervall ergibt sich aus dem Umfang des Projektes und ist deshalb ein individueller Wert, der von 3 s bis zu 2 min reichen kann. Liegt die letzte gültige Abfrage eines Multisensors länger als 3 min zurück, so wird er nicht angezeigt.

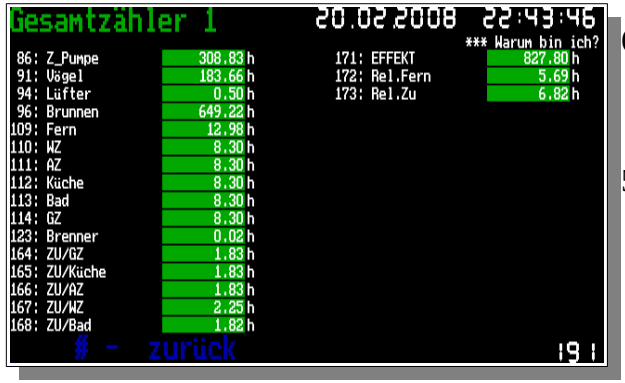
Wurde ein MAX1236 im System initialisiert, so wird der, sich auf der gleichen Adresse befindende Multisensor, gesperrt dargestellt.

2.4.7. Betriebsstunden, Energiezähler oder auch „lifing“



Das Programm bietet in der SPS eine Reihe von Module, die Informationen über die Zeit sammeln. Neben normalen Zählern für Betriebsstunden gibt es auch solche für Energie, Tagesbetriebsminuten, Tagesbetriebsminuten des Vortages und Leistungsmesser. Das Untermenü „lifing“ durchforstet die SPS nach entsprechenden Modulen und listet die Informationen übersichtlich auf. Äquivalent dazu findet man die Daten in der Datei „stunden.txt“ (ab Version 6.0.39e).

Das Untermenü „Gesamtzähler“ listet die Betriebsstunden aller Module vom Typ 9, 17,30,31,33,59. Damit können praktisch alle Schaltkanäle vollautomatisch ohne zusätzlichen Betriebsstundenzähler erfasst werden. Die ersten gefundenen 32 Module werden unter „Gesamtzähler 1“ und die folgenden unter „Gesamtzähler 2“ dargestellt.



Das Untermenü „Tageszähler“ listet alle Module vom Typ 66 (Betriebsminuten Vortag) und 68 (Tagesbetriebsminuten). Das Untermenü „Energiezähler“ erfasst die Module 95 (Wärmeleistungsmesser), 96 (Gesamtertrag) und 97 (Tagesertrag), das Untermenü „Q-Zähler/Speicher“ die Module 56 (Variablenspeicher), 57 (Quantumszähler), 58 (Langzeitspeicher) und das Untermenü „Zeitmodule“ ist für die Darstellung der Typen 4 und 5 bzw. 37/38 (Weckzeit/Zeitbereich) zuständig - entsprechende Programmierung der SPS in der *hauscomp.ini* natürlich vorausgesetzt.

Die Menüs haben reinen informativen Charakter, eine Änderung von Werten ist nicht möglich.

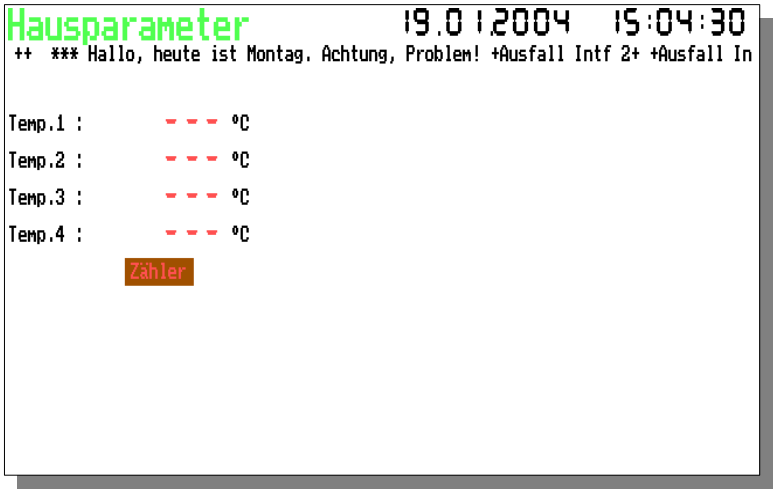


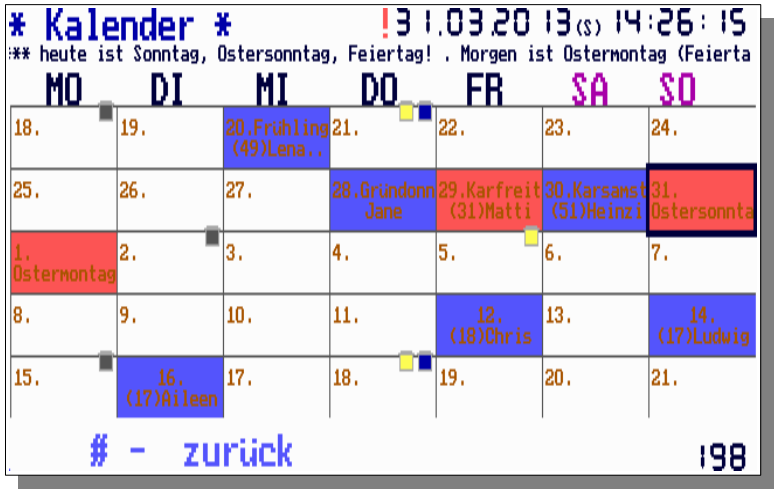




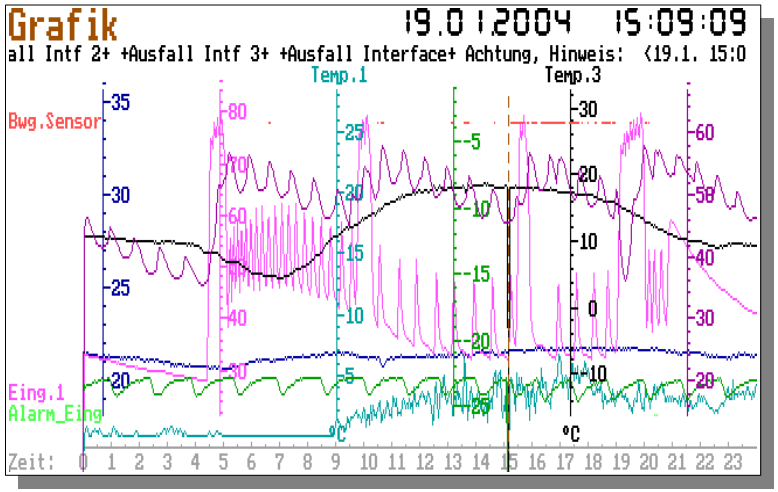
2.4.8. Die Automatikseiten

Im Automatikbetrieb sind standardmäßig abwechselnd die Seiten 095, 096 und 097 zu sehen. Sowohl die Zeitdauer des Bildwechsels als auch die Seiten selbst sind konfigurierbar über die *hardware.ini* bzw. *hauscomp.ini*.





Seite 95 mit den Hausparametern ist weitgehend frei gestaltbar. Wenn nichts vorgegeben wird, ist die Seite unterhalb der Laufschrift leer. Der Kalender zeigt immer in der zweiten Woche den aktuellen Wochentag mit einer dicken Umrandung an. Die Inhalte sind programmierbar. In der Grafik darunter (Seite 97) sind bis zu 6 analoge, sowie bis zu 16 binäre Grafen darstellbar. Sie zeigen den Verlauf der entsprechenden Parameter in den letzten 24 Stunden. Die Maßstabslineale werden automatisch erzeugt. Nach Stromausfall und erneutem Programmstart lädt das Programm aus der Datei „gestern.dat“ die grafischen Daten vom Vortag – sodass die entsprechenden Graphen rechts von der aktuellen Uhrzeit wieder sichtbar werden. Die Daten von 0 Uhr bis zum erneuten Systemstart sind bei einem Stromausfall dann allerdings futsch. Wird das Programm dagegen von Hand beendet (<k> bzw. <ALT-F4>) so werden nach erneuten Programmstart am selben Tag weitgehend alle Daten wiederhergestellt.



2.4.9. Übersicht der Bildschirmseiten

Hinweis: Ab Version 6.0.19 bzw. 6.0.38 gab es einzelne Änderungen bzw. Erweiterungen in den Seitenzuordnungen

0 - Rücksprungseite zum Automatikbetrieb
1 - Hauptmenü
2 - Kalender
3 - Anzeige Array „c“
4 - Anzeige Inhalt *hardware.ini*
5 - Test Bus-Geschwindigkeit
6 - Anzeigetest für Softwareentwicklung
7 - "b"-Array (geladene I/O Kanäle)
8 - "par"-Array (geladene Analogkanäle)
9 - "anz"-Array (geladene Anzeigen)
10 - Parameter eines SPS Moduls
11..19 – Untermenüs
20,21 - Entwicklungsseiten für Dimmerfunktion
22 - Anzeige Zykluszeit
23..94 – vorbehalten für spätere Funktionen
95 - Automatikbetrieb: Hausparameter
96 - Automatikbetrieb: Kalender
97 - Automatikbetrieb: Grafik
98..109 - vorbehalten für spätere Funktionen
110..119 – „Anwendungen..“
120..129 – „Analogsensoren..“
130..139 – „schalten..“
140..149 – „System..“
150..159 – „Grafik..“
160 – „SPS..“
161..179 – vorbehalten für spätere Funktionen
180..189 – „Funktionen..“
190..199 – „lifing..“
200..250 – frei für eigene Untermenüs
251..254 – Reserviert für Erweiterungen
255 - Abschaltung der kompletten Grafik

11 - „Anwendungen..“ (freie Gestaltung Seiten 11, sowie 110..118 durch Nutzer)
0 - Menü110
1 - Menü111
2 - Menü112
3 - Menü113
4 - Menü114
5 - Menü115
6 - Menü116
7 - Menü117
8 - Menü118
9 - Menü119

12 - „Analogsensoren..“ (120-129)
0 - 7 Adresse 0-7
(120-127 jeweils Kanal 0-3)
8 – alle Sensoren

13 - „schalten..“ (130-137) /PCF8574P schalten
0 - "Name P8574P/0)"
...
7 - "Name P8574P/7)"
9 – Reset (alle aus)

14- „System..“ (140-149)
0 - Uhrkorrektur
1 - Anzeige Inhalt *hardware.ini*
2 - "b"-Array (geladene I/O Kanäle)
3 - "par"-Array (geladene Analogkanäle)
4 - "anz"-Array (geladene Anzeigen)
5 -
6 - Anzeige Zykluszeit
7 – Fontanzeige (bis 6.0.39)
8,9 - Zeitkern

15 - „Grafik..“
151 - Grafik1_analog+binär
152 - Grafik1_analog
153 - Grafik2_analog+binär
154 - Grafik2_analog
155 - Laufschrift
156 - Sternenhimmel
157 - Hausparameter
158 - Echtzeitmonitor
159 - Kalender

16 - „SPS..“

17 - Tagesinfoseite

18 - „Funktionen..“
180 – I2C-Busanalyse
181 – Hardwarestatus
182 – A/D Wandler
183 – Festlegungen
184 – I/O Kanäle
185 – Analogsensoren
186 –
187 – HzgLog-Parameter
189 – Sonstiges

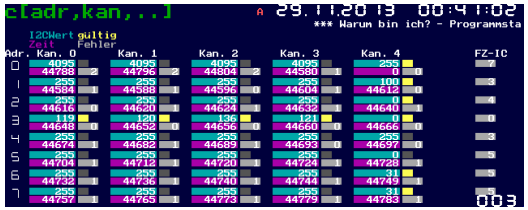
19 - „lifing..“
191/2 – Gesamtzähler
193 – Tageszähler
194 – Energiezähler
195 – Q-Zähler/Speicher
196 – Zeitmodule

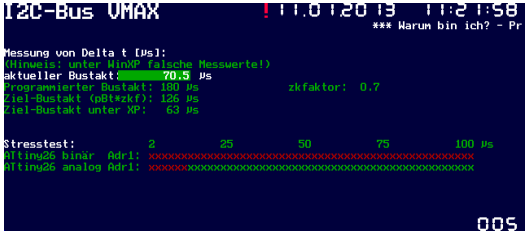
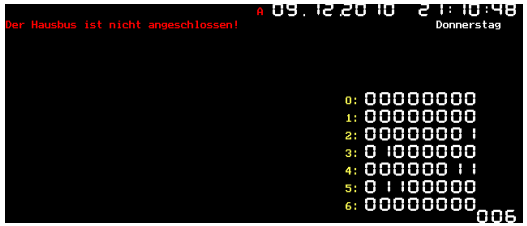
2.4.10. Menüsteuerung

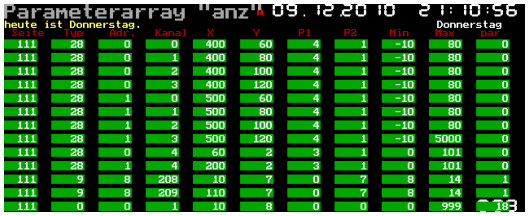
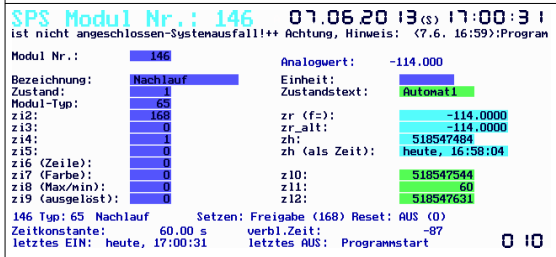
1. Auf der Tastatur jeweils durch Anwählen einer Ziffer entsprechend der Menüdarstellung;
2. Auf der Tastatur durch Wählen einer 3-stelligen Seitenzahl innerhalb von 4 Sekunden (ab Version 5 leider nicht mehr möglich);
3. „#“ auf der Tastatur - zurück zur letzten Seite,
4. „*“ auf der Tastatur – Einschalten des Automatikbetriebes (selten gebraucht),
5. „+“ bzw. “-“ auf der Tastatur: aktuelle Seite +/-1 (bei engl. Tastatur Ziffernwahlblock bzw. <ß>/<`>)
6. Auf der Tastatur mit Leertaste, sowie Kursor auf/ab, alternativ, abgesetzt vom PC, über das Control Interface (CI) oder andere Bustasten
7. Über das Netzwerk durch Schreiben von Text in die Datei „taste.txt“

2.4.11. „versteckte“ Seiten

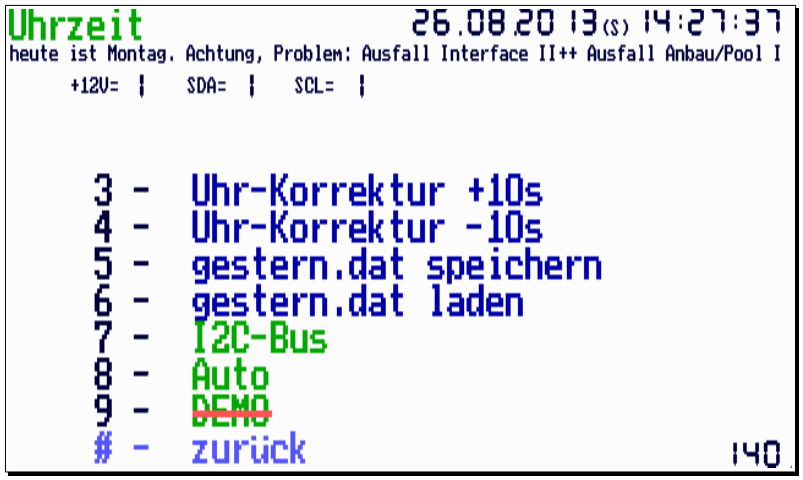
Die Bildschirmseiten 2 .. 9 sind über die Menüführung nicht direkt erreichbar, deshalb hier ein paar Infos zu den scheinbar „versteckten“ Seiten, die sich aber von Seite 1 bzw. 11 ausgehend ganz normal mit den Tasten „+“ bzw. „-“ anzeigen lassen. Sie dienen in erster Linie der Programmentwicklung. Je nach Programmversion können **einige dieser Seiten auch deaktiviert sein**. Die Seite 10 ist vom Hauptmenü direkt über die Taste „0“ erreichbar. Die wichtigsten hier genannten Seiten sind ab Version 6.0.38 auch über den Menüpunkt „System..“ aufrufbar.

Bildschirmseite	Bedeutung	Bemerkung
	Kalender	Entspricht Seite 159.
	Anzeige Array „c“ -deaktiviert-	Das Array „c“ wird von den I2C Bus Treibern mit dem I2C-Wert und der Bezugszeit gefüllt und dient der Übergabe an die SPS. Zusätzlich sind die Gültigkeit der Werte und der Fehlerzähler gelistet.

Bildschirmseite	Bedeutung	Bemerkung
	*hardware.ini*	Anzeige des Inhaltes der *hardware.ini*, und zwar so, wie das Programm die Datei einliest. Wird z. B. unter Linux ein falscher Zeilenumbruch generiert, so werden die Scriptbefehle in einer einzigen Zeile gelesen und können damit nicht mehr ausgeführt werden. Es lassen sich bei Problemen so die Syntax kontrollieren.
	Hardware – Stresstest -deaktiviert-	Diese Seite startet einen Geschwindigkeitstest des Hausbusses. Dabei wird der normale Datenverkehr unterbrochen zugunsten einer gezielten Abfrage der analogen Adresse 1 und der binären Adresse 1. An beiden Adressen wird ein ATtiny26 mit der Standard-Firmware erwartet. Durch die periodische Übertragung mit unterschiedlichen Impulslängen erfolgt eine geschwindigkeitsabhängige Auswertung der Daten, die Rückschlüsse auf die Qualität des Hausbusses zulassen. Achtung, der bis dahin normale Datenverkehr auf dem Hausbus ist in dieser Zeit unterbrochen!
	Testanzeige -deaktiviert-	Nur für Softwareentwicklung. Derzeit werden die Register des Schnittstellenbausteins für die aktuelle serielle Schnittstelle sowie die Umgebungsvariablen angezeigt.
	Parameterarray “b“	Anzeige aller Daten des internen Schnittstellenarrays für binäre Daten vom und zum Hausbus.

Bildschirmseite	Bedeutung	Bemerkung
	Parameterarray „par“	Anzeige aller Daten des internen Schnittstellenarrays „par“ für analoge Daten vom und zum Hausbus. Die einzelnen Adressen lassen sich mit „Enter“ durchschalten. Gelistet sind u.a. auch die Transformationskoeffizienten für die Ermittlung der Sensorwerte. Wert = a0 + a1*Kode + a2*Kode²
	Anzeigearray „anz“	Anzeige aller Parameter der von der *hauscomp.ini* gelesenen Anzeigen (Scriptbefehl: „Anzeige=“). Mit „Enter“ blättern möglich.
	Detaillierte Auflistung aller internen Parameter eines SPS Moduls	Ab Version 6.0.38 Dient der weitergehenden Fehlersuche in Steuer- und Regelungen. Bedienung mit gleichen Tastenbefehlen wie im SPS Menü, hier mit erweiterten Anzeigen.

2.5. Die Korrektur der Uhrzeit des PCs



Die Uhrzeit ist für ein Steuerungssystem extrem wichtig. Da aber die gewählte Systemkonfiguration des PCs ohne DCF-77 (Funkuhr) auskommen muss, wird die Uhrzeit ausschließlich durch die PC- Zeit generiert. Zeitabweichungen im Sekundenbereich sind deshalb nach längerem Betrieb die Folge. Eine Korrektur ist von Hand über das Menü „System/Uhrzeit“ auf 10 Sekunden ausreichend genau möglich. Eine eingebaute Automatik sorgt einmal in der Woche für eine automatische Korrektur der Uhr entsprechend dem Eintrag in der *hauscomp.ini* (z. B. Wochenkorrektur=+53). Um diesen Wert zu ermitteln, muss man allerdings die Zeitabweichung je Woche sekundengenau manuell ermitteln. Zur Erleichterung dieser mühseligen Rechnerei ist in der Software eine Routine eingebettet, die sich ca. 1 Woche nach Programmstart aktiviert. Auf der Menüseite „Uhrzeit“ wird ab dann automatisch ein Korrekturwert als Vorschlag angezeigt – errechnet aus der tatsächlichen Programmlaufzeit unter Berücksichtigung der Zeitkorrekturen von Hand. Nicht berücksichtigt werden die ersten 5 min nach Programmstart, wo man in der Regel die Uhr erstmalig genau stellt. Wenn also nach Programmstart die Uhr genau gestellt wird (Vergleich mit Funkuhr!) und noch einmal nach ein paar Wochen verglichen und korrigiert wird, so kann man den Vorschlag der Software getrost in die *hauscomp.ini* hineinschreiben. Die Software ändert aus Sicherheitsgründen nichts selbständig in der *hauscomp.ini*! Die Umschaltung auf Sommerzeit wird mittels des Scriptbefehls „Sommerzeit=“ gesteuert. Unter Linux muss man die Korrektur der Uhrzeit dem Betriebssystem überlassen, da weder Dosemu noch DosBox einen echten Zugriff auf die interne PC-Uhr gestatten. Eine Verbindung zum Internet inklusive automatischer Zeitkorrektur über Linux macht also Sinn. Wird die Zeit zurückgestellt, fällt für diesen Zeitraum die SPS aus und der Hinweis „negative Zeit“ wird gesetzt.

2.6. Sprachausgabe mit RN – SPEAK (http://www.robotikhardware.de/)

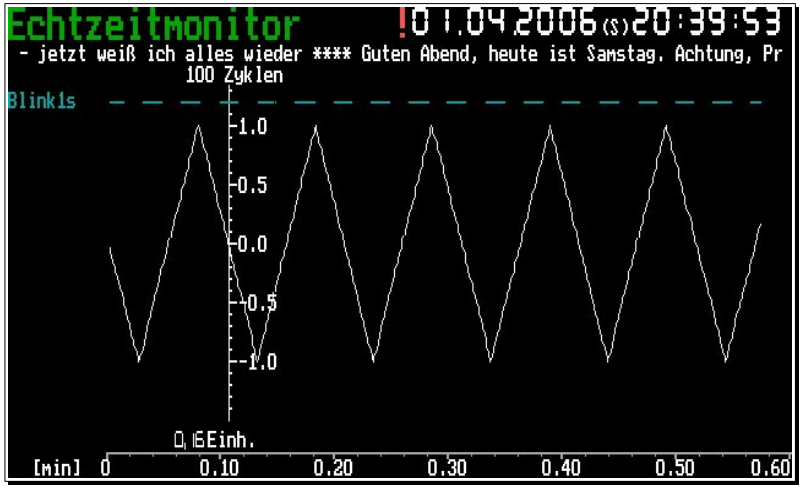
Das Programm unterstützte in den Versionen 6.0.7 und 6.0.8 den Anschluss des RN-Speak Boards über den Hausbus. Diese Idee wird nicht mehr verfolgt.

2.7. Der Echtzeitmonitor

Beim Aufbau von Steuer- oder Regelungssystemen ist oftmals eine Identifizierung der Regelstrecke mit den erforderlichen Zeitkonstanten notwendig. Zudem ist eine Kontrolle der SPS-Schaltsequenzen in der Entwicklungsphase unerlässlich. Zu diesem Zweck wurde der Echtzeitmonitor entworfen. Der Echtzeitmonitor ist über <Hauptmenü>, <Grafik>, <Echtzeitmonitor> erreichbar. Alle analogen und binären Parameter, die für die normale Grafik programmiert wurden, sind hier ebenfalls dargestellt. Die zeitliche Auflösung ist mit dem Scriptbefehl

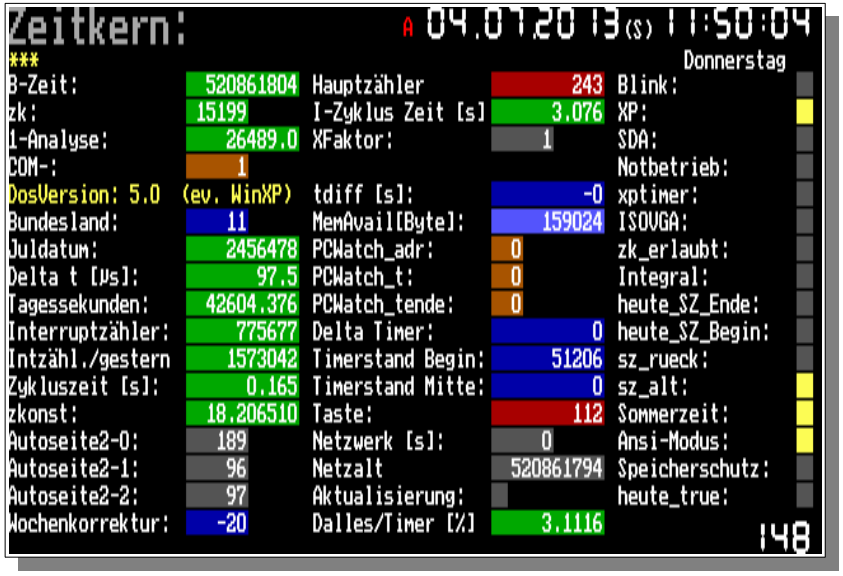
„Echtzeitmonitor=“ einstellbar im Bereich 0,6 min bis zu 60 Tagen. Das Ganze funktioniert ähnlich einem Oszilloskop.

Neben der Bezeichnung und der Grafen der entsprechenden Signale wird auch der aktuelle physikalische Wert der Analogsignale angezeigt. Im nebenstehenden Beispiel wird als Binärsignal der Ausgang des internen SPS-Moduls Nr. 12 (Blink1s) verwendet. Sehr gut zu sehen ist der Signalwechsel nach jeweils einer Sekunde. Das interne Modul 18 liefert das analoge Signal. Es läuft ständig im Bereich -1 bis +1. Der Monitor ist hervorragend verwendbar um z. B. Heizungsregelungen in der Anheizphase oder im stationären Betrieb zu analysieren. Mit der Taste „p“ wird ein Screenshot erzeugt (im Beispiel wäre das dann die Datei „158.bmp“).



Da die grafischen Werte im Gegensatz zur 24h Grafik nicht gespeichert werden, verhindert eine Sperre das versehentliche Zurückschalten in den Automatikbetrieb - bis von Hand die Seite gewechselt wird.

2.8. Der Zeitkern



Für den normalen Betrieb des Programms ist es unerlässlich, die vielen zeitlichen Abhängigkeiten präzise zu steuern. Da die Arbeitsgrundlage ein PC mit verschiedenen Betriebssystemen ist, mussten bei der Programmierung etliche Kompromisse eingegangen werden um eine Echtzeitfähigkeit zu erreichen. Letztlich dienen dazu als Uhren der CMOS- Uhrenbaustein auf dem Motherboard sowie der Timerkanal „0“ mit seinem eigenen Quarz. Dinge, die sich an Stunde und Minute orientieren, verwenden die Zeit des Uhrenbausteins, alles andere hat die gezählten Interrupts seit Tagesbeginn als Grundlage. Die daraus ermittelten Tagessekunden sind für Regelungsaufgaben linear und präzise. Aus der Differenz beider Uhren ergibt sich für den nächsten Tag der Korrekturfaktor zkonst. Überschreitet die Differenz zwischen beiden Uhren 25 min (Zeitinkonsistenz) so blinken ab Version 6.0.38 in der Uhrzeit die Doppelpunkte rot. Das Zeitverhältnis seit Programmstart zwischen beiden Uhren ist der Wert „Dalles/Timer“ in Prozent. Alles, was im Minuten- oder Tagesbereich abläuft, ist damit stabil. Leider wird seit Windows 2000/XP der Timertakt vom Betriebssystem laufend umprogrammiert, sodass damit die Zykluszeit, die sowohl den I2C-Bus als auch den Echtzeitmonitor mit Zeitinformationen im µs Bereich versorgt, fehlerhaft sein kann. Ab Version 6.0.18. ist die Programmstabilität durch eine interne gleitende

Timerkorrektur unter XP verbessert worden. Die Zykluszeit ist die Zeit, die ein kompletter Echtzeit-Programmdurchlauf inklusive der Grafikausgabe dieser (Zeit-) Seite benötigt. Sie ist von der Hardware abhängig und sollte für einen zügigen Bildaufbau 1s nicht wesentlich überschreiten. Die Busansteuerung ist davon unabhängig. Erst wenn der PC es nur noch schafft, den Hausbus zu bedienen, aber länger als 3,5s keine Zeit mehr für die Grafik oder die SPS hat, so geht das Programm für genau einen Zyklus in den Notbetrieb, um diese Aufgaben einmal abzuarbeiten. Die beiden Uhren driften dann natürlich auseinander. Das Screenshot zeigt die Werte zum Vergleich (Thinkpad T30 mit XP-System). Wegen der Fülle der Daten gibt es mittlerweile auch einen Zeitkern2.

2.9. Die Ausfall- und Ereignisüberwachung

Dieser Abschnitt umreißt die Komponenten, die Programm intern die Zuverlässigkeit des Gesamtsystems garantieren und weitgehend automatisiert ablaufen. Dazu gehören:

- Nach Programmstart: Analyse des übergeordneten Betriebssystems, Ausmessen der Geschwindigkeit der PC Hardware und Einstellen optimaler Zeitkonstanten für den Hausbus, Überprüfung der ini-Dateien auf Syntaxfehler, Übertragung der Startwerte in den EEPROM angeschlossener ATtiny26, Überprüfung des Datenträgers auf Schreibschutz, Programmierung der „Autokill“ Funktion.
- Ständige Eigendiagnose des PCs und der Software (CBIT) für die Überwachung des Zeitkerns auf Plausibilität (notfalls Notbetrieb bei Rechenzeitproblemen oder Abschaltung von Aktoren bei negativer Zeit), Kontrolle des noch freien Speicherplatzes auf dem Datenträger, Überwachung der Funktion des Hausbusses, angeschlossener Interface und Sensoren, Feststellung falsch gesetzter Modulparameter, Überwachung der Ereignissteuerung. Generierung von Fehlern in der Laufschrift als „Problem“.
- Ständige Überwachung interner oder Hausbus bezogener Ereignisse wie: Über- oder Unterschreitung zulässiger Werte (z. B. MAX- oder MIN-Werte in Anzeigen), Sensorwerte im Grenzbereich, Auslösung von Alarmseiten bei Erreichen programmierter Zustände, Auftreten von Kalenderereignissen, Betätigung von Menütasten. Diese Ereignisse werden in der Laufschrift geordnet und mit Zeitangabe des ersten Auftretens bei Notwendigkeit als „HINWEIS!“ generiert. Zusammenfassung gleicher Ereignisse mit der Formulierung „...und später mehrfach...“.
- Programmabbruch nur bei grundsätzlichen Problemen in der Kommunikation mit dem Betriebssystem und wenn weniger als 50 Hauptprogramm-Zyklen je Minute im automatischen Modus gemessen werden (→ „Linuxbefehl 4/1“); ansonsten weitgehende Fehlertoleranz (z. B. beim Lesen vom Uhrenbaustein, der Grafikausgabe, der Steuerung der seriellen Schnittstelle). Ausgabe einer Fehlermeldung bei unvermeidbarem Programmabbruch.
- Anzeige des dynamischen Laufbandes mit dem Scriptbefehl „Laufschrift=on“.
- Bei Bedarf tägliche Speicherung von Ereignissen in der Datei „Laufband.log“ über den Scriptbefehl „LLog=on“. Hier ein Auszug:
  - 13.05.2011 00:00:00 (Freitag)Ich bin ohkei... Achtung, Hinweis: Alarmseite ausgelöst
  - 14.05.2011 00:00:01 (Samstag)Ich bin ohkei... Achtung, Hinweis: Alarmseite ausgelöst <13.5. 5:02>:Flammwächter (196) <1 min
- Bei Bedarf detailliertere Speicherung von Ereignissen in der Datei „Ereignis.log“ über den Scriptbefehl „ELog=on“. Hier ein Auszug:

- 17.08.2012 19:27:52 (Freitag) Kinderzimmer Wert größer MAX
- 17.08.2012 19:27:52 (Freitag) T_SPO (344) Wert größer MAX
- 17.08.2012 19:28:06 (Freitag) 247/8
- 17.08.2012 19:28:06 (Freitag) Programmstart!
- 17.08.2012 19:33:39 (Freitag) Windchill (76) Wert kleiner MIN

• Bei Bedarf die Erzeugung der Datei „alarm.txt“ über den Scriptbefehl „Alarm=on“. Abgespeichert werden alle Ereignisse, die auch in der Datei „Ereignis.log“ erfasst werden können, mit Ausnahme von Über- bzw. Unterschreitung von Werten sowie die Auslösedauer von SPS Modulen („Wert kleiner MIN“/“Wert größer Max“). Praktisch sind enthalten: der Programmstart, Ausfälle von Interface, Ausfälle von Sensoren, Aktivierung von Alarmseiten.

2.10. Die Hardwarekopplung des Signals „HINWEIS!“

Wie bereits im vorangegangenen Abschnitt beschrieben, wird im Rahmen der automatischen Überwachung softwaremäßig das Signal „Hinweis!“ generiert, vergleichbar mit einer „Master Caution“ Warnung. Es wird damit eine Abweichung im automatischen Betrieb des Hauscomputers signalisiert, die natürlich nicht nur auf den IT Komponenten sichtbar sein sollte. Deshalb kann dieses Signal auch auf einen Binärkanal über den Hausbus gesendet werden, der mit dem Befehl „Hinweis=“ in der *hauscomp.ini* programmiert ist.

Mit der Vorstellung des Control-Interfaces (CI) [2], von denen mehrere gleichzeitig an der Binäradresse 0 angeschlossen werden können, hat sich der Kanal 3 dieser Adresse als Standard für das Signal „HINWEIS!“ etabliert. Mit Einführung der Binärinterface auf der Basis des ATtiny26 wurde das berücksichtigt: die ATtinys „hören“ bereits vollautomatisch auf genannten Kanal und lassen bei Bedarf die grüne Status-LED blinken. Leider konnte diese Funktion in Ermangelung eines Ausgangs-pins am ATtiny26 nicht auf die Status LED des analogen Universalinterfaces übernommen werden. Werden am Bus ausschließlich Binärinterface mit ATtiny26 betrieben (d. h. es ist auch kein CI Interface vorhanden), so darf die Binäradresse 0 hardwaremäßig trotzdem nicht belegt werden, um die volle Funktionalität zu erhalten. In diesem Fall erkennt das Programm *hauscomp.exe* ab Version 6.0.36 diesen Hardware-Zustand und sendet auf der Binäradresse 0, Kanal 3 automatisch für die angeschlossenen ATtinys den Zustand des Signals „HINWEIS!“. In der Hardwareübersicht wird diese Binäradresse als „System“ gekennzeichnet. Der Scriptbefehl „Hinweis=“ funktioniert völlig unabhängig davon weiterhin.

Beim alten A+D Interface ([2] Abschnitt 7) gibt es nun ein Problem: Einerseits ist dieses Interface zwar mit einer Brücke für die Hinweisfunktion ausgestattet, anderseits kann das Interface aber ausschließlich auf die mit Jumpern eingestellte Doppeladresse „hören“. Die Adresse 0 Kanal 3 ist in der Regel also nicht verfügbar. Somit muss bei gemischter Verwendung von CIs, ATtinys und A+D Interface auf die Brücke und auf die Hinweis-Blinkfunktion am Interface verzichtet werden.

2.11. Die Syntax der Datei „taste.txt“

Im „live“ Modus, also in der interaktiven Betriebsart über ein lokales oder entferntes Netzwerk, erfolgt die Steuerung des Programms

hauscomp.exe über die Datei „taste.txt“ sozusagen als digitaler Eingabekanal. Voraussetzung ist dabei der Scriptbefehl „live=on“ in der Datei *hardware.ini*. Diese Steuerungsmöglichkeit kann vom entfernten Rechner sowohl von Hand (über einen Dateieditor), per copy einer vorbereiteten Datei „taste.txt“ (z. B. als Teil einer Batch-Programmierung) oder aber durch einen übergeordneten Server (siehe auch [6]) eingesetzt werden.

Wird in die Datei „taste.txt“ ein Text geschrieben, so wird dieser anschließend vom Programm interpretiert und ausgeführt, der Inhalt der Datei wieder gelöscht und die Datei „live.bmp“ aktualisiert. Die Datei „taste.txt“ wird aller 15s gelesen und dürfte in der Regel leer sein. Nachdem ein Text als Inhalt erkannt wird, erfolgt das Auslesen für die nächsten 30s beschleunigt aller 2s.

Für die erstmalige Anmeldung über das Netzwerk ist der Kode „108“ vorgesehen. Er erzeugt zunächst eine Aktualisierung der Datei „live.bmp“, verbunden mit der Umschaltung in den statischen Anzeigemodus (nur unter Dosemu) und schreibt in die Datei Laufband.log einen Hinweis zum Onlinezugriff.

Ab Version 6.0.39 zusätzliche Rechtevergabe: Am Ende des Textes wird ab jetzt ein User-Name erwartet, wie z. B. „admin“, „user“, oder „gast“. Wird zum Abschluss „admin“ als User-Name eingegeben, so wird der Befehl immer ausgeführt, allen anderen Namen dürfen nur Anzeigen einschalten, sowie Steuerzeichen senden für: „o“, „i“, „p“, „+“, „-“, „#“ und auf den Bildschirmseiten 0..119 zusätzlich die Tasten „0“ .. „9“ betätigen. Damit erfolgt eine Unterscheidung zwischen „normalem Nutzer“ und „Techniker/Spezialist“. Die Überprüfung des entsprechend notwendigen Passwortes erfolgt nicht durch dieses DOS-Programm, sondern im übergeordneten Server (siehe [6]).

> ***Hinweis:*** unter Linux wird u.U. beim Schreiben in die Datei taste.txt am Ende ein „.“ angehängt, sodass der HC fälschlicherweise „admin.“ liest und den Befehl daher nicht ausführt. Ein einfaches zusätzliches Zeichen mit Leerzeichen getrennt schafft Abhilfe. Hier ein Auszug aus einem Linux-Script:

```
Strng="P 700 12  0 admin k"          # Befehl SPS-Modul 700 auf „Automat“ setzen
echo  $Strng >> /flash/live/tmp/taste.txt     # Befehl in taste.txt speichern
```

Es gibt folgende Möglichkeiten der Befehlseingabe (sowohl Groß- als auch Kleinbuchstaben werden akzeptiert):

- Eine Zahl (ohne Lehrzeichen) im Bereich 0..255 → Interpretation als Steuerkode, wie im Abschnitt 2.2. beschrieben; zusätzlich werden die Zahlen 0..9 direkt als Tasten „0“.. „9“ interpretiert (sonst Steuerkode 48..57).

```
114 admin      - „r“ wird gelesen, daraufhin ein Reset durchgeführt und die hauscomp.ini neu eingelesen
```

- Ein vorangestelltes „S“ mit folgendem Leerzeichen und einer Zahl im Bereich 0..255 → Einschalten der angegebenen Bildschirmseite

```
S 123  - auf „Bildschirmseite“ 123 wird umgeschaltet
```

- Ein vorangestelltes „A“ mit folgendem Leerzeichen und zwei Zahlen im Bereich 0..7 → Anzeigen des angegebenen Analogsensors (Adresse, Kanal)

A 2 3 - auf „Bildschirmseite“ 122 - Analogsensor Adresse 2 / Kanal 3 wird umgeschaltet

- Ein vorangestelltes „M“ mit folgendem Leerzeichen und einer Zahl im Bereich 0..zmax → Anzeigen des angegebenen SPS Moduls auf Seite 10. Bei der Angabe von „0“ wird der letzte eingestellte SPS Modul angezeigt.

M 76 - auf „Bildschirmseite“ 10 mit den Parametern des SPS Moduls 76 wird umgeschaltet

- Ein „#“ (Raute), → Interpretation als Steuerkode 39; entspricht auch der „#“ Taste

 # - auf die vorhergehende/übergeordnete „Bildschirmseite“ wird umgeschaltet

- Ein vorangestelltes „T“ mit folgendem Leerzeichen und zwei Zahlen (im Bereich 0..255 bzw. 0..15) → Auslösen einer Tastenaktion auf der angegebenen Seite. Für die Seiten 0..119 sind alle user zugelassen, die restlichen Seiten funktionieren nur mit „admin“.

T 113 5 user - Auf Seite 113 wird die (hoffentlich auch programmierte) Taste 5 ausgelöst

- Ein vorangestelltes „P“ mit folgendem Leerzeichen und 3 Zahlen → Speichern des angegebenen Parameters in der SPS (nur für Experten!)

  Erste Zahl – SPS Modulnummer (0..1199)
  Zweite Zahl (0..7) – SPS Parameternummer
  8 – SPS Modul auf HAND AUS schalten ohne Parameter f setzen
  9 – SPS Modul auf HAND EIN schalten ohne Parameter f setzen
  10 – SPS Modul auf HAND AUS schalten mit Parameter f setzen
  11 – SPS Modul auf HAND EIN schalten mit Parameter f setzen
  12 – SPS Modul auf AUTO AUS schalten mit Parameter f setzen (der dann beim nächsten SPS-Durchlauf ggf. neu berechnet und entsprechend wieder überschrieben wird.
  13 – SPS Parameter zr_alt[z] (ab 6.0.59B3)
  14 – SPS Parameter zh[z] (ab 6.0.59B3)
  Dritte Zahl – Parameterwert (Real Gleitkommazahl, wie z. B. „18.3821“)
  Nach der dritten Zahl ist immer „admin“ anzugeben

P 51 2 78 admin - Im SPS Modul 51 wird der Parameter 2 (zi2) auf 78 gesetzt.

- Ein vorangestelltes „L“ oder „l“ mit folgendem Leerzeichen, 1 Zahl und nachfolgendem „admin“ → Ausgabe eines Linux-Scriptfiles

/flash/live/scripte/script mit Befehlen für die Wartung des Linux-Betriebssystems (nur für Experten!)

Zahl:
- 1 – „reboot“ kompletter Neustart Linux → mit Speicherung der pers. Speicherdatei (slackosave-user.4fs). Die Ausführungszeit ist > 30 s länger im Vergleich zur reinen Startzeit.
- 2 – „/flash/live/scripte/backup“ und „restartwm“ → Backup Datenbanken und Neustart des X-Servers mit *hauscomp.exe*
- 3 – „yaf-splash … „Test“ Ausgabe Schriftfeld „Test“
- 4 – „hard_reboot“ schneller Neustart mit HC-Datenrettung ohne Speicherung pers. Speicherdatei (slackosave-user.4fs)
- 5 – „killall hiawatha“ und „hiawatha.run restart“ → Neustart des Hiawatha-Servers
- 6 – „/usr/sbin/save2flash“ persönliche Speicherdatei sichern → Ausführungszeit ca. 7 min (HC_V2, S400 FUTRO)
- 7 – „/flash/live/scripte/serverlogs.sh“ Serverneustart Hiawatha&ftpd → Ausführungszeit ca. 1-2 min
- 8 – „/flash/live/scripte/cron_ntp“ Zeitsynchronisation → Ausführungszeit ca. 1 min
- 9 – „/flash/live/scripte/s/proc.sh“ Prozessprotokoll → ../scripte/s/proc.txt

Bei 1 – 4 werden von *hauscomp.exe* automatisch die Betriebsstunden und grafische Daten gespeichert und danach der Schreibschutz gesetzt (gelbes Kreuz), sodass ein folgender Programmneustart ohne Probleme möglich wird. Die Ausführungszeit kann bis zu 5 min betragen. Durchläuft das Hauptprogramm (*hauscomp.exe*) weniger als 50 mal je Minute alle Programmteile (Hauptdurchlauf), so wird intern automatisiert ein Linux-Neustart erzwungen (Entspricht Zahl=4). Nach der Zahl ist immer „admin“ anzugeben

L 6 admin - Der aktuelle Stand des Betriebssystems wird in die persönliche Speicherdatei geschrieben

Die Steuerung des Programms über das Netzwerk kann manuell von der Tastatur vor Ort unterbunden werden in dem man im Menü über „System/Zugriff-Pfade“ mit dem Menüpunkt „0“ den Livemodus aktiviert/deaktiviert.

- Ein vorangestelltes „D“ oder „d“ mit folgendem Leerzeichen → Aktivierung einer regelmäßig automatisch aktualisierten „D“atei mit ausgewählten SPS-Parametern. Dient als Service für Linux-Scripte, z. B. für weitere grafische Auswertungen. Die folgenden Elemente sind in der Steuerdatei „taste.txt“ durch Leerzeichen zu trennen:

Erstes Element:	[String]	– Dateiname (maximal 8 Zeichen + „ .“ + 3 Zeichen, keine Sonderzeichen erlaubt)
Zweites Element:	[Integer]	– Zeitintervall für die Aktualisierung der Datei (1..32767 [s])
Drittes Element:	[LongInt]	– Zeitdauer für den gesamten Zeitraum der automatischen Aktualisierung der Datei (10..2000000000s [s])
Ab viertem Element:	[Integer]	– Nummer des SPS-Moduls. Positive Zahl: analoger SPS-Ausgangswert, negative Zahl: SPS-Zustand

Es können bis zu 33 SPS-Module vorgegeben werden. Jeder neue Befehl vom Typ „D“ überschreibt die bisherigen Steuervariablen und erzeugt die nächste Datei – bis der vorgegebene Zeitraum abgelaufen ist. Es wird also gleichzeitig jeweils nur eine Datei automatisch aktualisiert. In der erzeugten Datei werden alle 33 Werte in einer Zeile, durch Leerzeichen getrennt abgebildet. Nicht genutzte Elemente erzeugen eine „0“ in der Ziel-Datei. Bei Zeitintervallen von <60s wird das aktuelle Intervall der Rechner-Performance angepasst (verlängert) um die reguläre Arbeit der Bus-Treiber zu gewährleisten.

D _klima.txt 30 600 142 -147 163 701 - aller 30 s wird die Datei „_klima.txt“ für 600s/60=10min mit den Werten einiger SPS-Module geschrieben

Der Inhalt von „klima.txt“ sieht dann beispielsweise so aus:

```
0.001 0.278 7.000 1 0 0 1.000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Voraussetzungen für die „D“-Funktion sind in der *hardware.ini*: "live=on" UND "PfadTasteD=" gesetzt UND "Netzwerk=" mit dem Zusatz "RAM" versehen. Um Zugriffskonflikte auf die automatisch erzeugte Datei zwischen Linux-Scripten und dem Programm *hauscomp.exe* zu vermeiden, sollte bei Zeitintervallen > 1 min der Dateiname mit einem Tiefstrich beginnen. Die Datei wird dann minütlich von einem cron-Script (HC_V2A) automatisch in eine ohne Tiefstrich kopiert, auf die unter Linux uneingeschränkt zugegriffen werden kann. Zugleich Bei kürzeren Intervallen müssen vom bearbeitenden Linux-Script die (Schreib/Ausführungs-) Eigenschaften der Datei ausgewertet werden: *hauscomp.exe* setzt nach der Aktualisierung diese Eigenschaften auf r-- ??? ??? zurück. Das Linux-Script muss also seinerseits nach seinem Lesen die Eigenschaften wieder auf rwx ??? ??? (also z. B. 755) setzen, sonst erfolgt keine Aktualisierung der Datei.

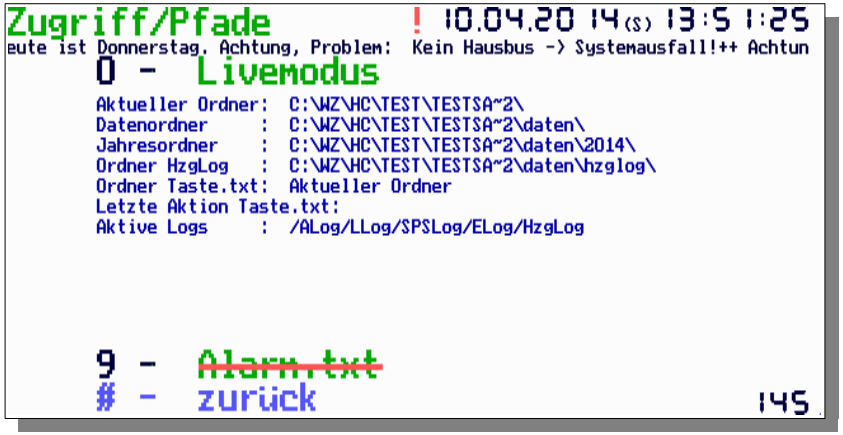
2.12. Die Ordner- bzw. Verzeichnisstruktur

Mit der Version 6.0.38 wurden automatisch erzeugte Unterordner eingeführt, um die Datenflut besser zu beherrschen. Im Hauptverzeichnis, das nach wie vor frei wählbar ist, finden sich nun nur noch die Programmdateien und die Logdateien. Im Unterordner „\daten\“ wird die Datei SPS2.csv geschrieben. Dort gibt es auch weitere Unterordner für die einzelnen Jahre. Diese Ordner enthalten alle täglich aktualisierten Sensordaten sowie die Datei SPS.csv. Durch die unterschiedliche Speicherung der beiden *.csv Dateien kann eine bessere statistische Auswertung erfolgen. So enthält jetzt jede SPS.csv tatsächlich nur noch die Daten eines Jahres, während in der SPS2.csv alle programmierten Daten fortlaufend gesammelt werden.

Die Heizungslogdateien vom Typ „mixxxxxx.txt“ werden im Unterordner „daten\hzglog\“ wochenweise abgelegt.

Im Jahresordner gibt es eine automatisch um Mitternacht aktualisierte Datei „senslist.txt“. Sie dient der Kommunikation zwischen dem Dos-Programm und den Linux-Scripten.

Eine Übersicht über die Pfade bzw. Ordner findet sich unter „System/Zugriff..“:



Hier kann man auch den Livemodus (Beeinflussung des Programms über das Netzwerk in Echtzeit) und die automatische Generierung von Alarmmails abschalten.

Wird beim Programmstart eine Datei mit Namen „version.txt“ im Ordner „Taste.txt“ gefunden, so wird ein alternatives Speicherkonzept aktiviert (vorangestellter Tiefstrich bei temporären Dateien).

Der Pfad der Steuerungsdatei „taste.txt“ liegt normalerweise im Startverzeichnis des Programms *hauscomp.exe*. Da unter dem Linuxderivat „wary“ die Schreibrechte an dieser Datei nur schwer im Netzwerkbetrieb freizugeben waren, musste eine Lösung her. Mit dem Scriptbefehl „PfadTaste=“ in der *hardware.ini* kann nun ein völlig anderer Pfad zu der Datei incl. eines anderen Laufwerks vorgegeben werden. So lautet die korrekte Angabe unter Linux-Wary, sowie den folgenden HC_V2 ff.: „PfadTaste=D:“ Das aktiviert zugleich eine zusätzliche Datei „_sps.sps“, die minütlich die Werte der gesamten SPS für die weitere Verarbeitung unter Linux abspeichert.

Mit der Nutzung des Programms *hauscomp.exe* unter Linux/Dosemu direkt im RAM wurden neue Möglichkeiten der Datengewinnung erschlossen. Die Rechenkapazität reicht nun auch für die regelmäßige Aktualisierung von 10 Datenbanken und der Gewinnung von Klimadaten der Raumsensoren (klima.txt). Die dazu erforderlichen Dateien werden unter Linux automatisch in dem temporären Ordner **/flash/live/tmp/** abgelegt.

2.13. Zusammenarbeit hauscomp.exe mit anderen Programmen

Das Programm hauscomp.exe erzeugt eine Reihe von Daten/Dateien, die u.a. für den Webserver bereitgestellt werden. Daneben gibt es spezialisiert für den Datenaustausch und für Steuerungsmöglichkeiten durch externe Programme weitere Möglichkeiten. Der Zugriff der externen Programme erfolgt über das Betriebssystem (DOS mit Netzwerktreibern, Linux HC_V2A) per Netzwerk auf die entsprechende Dateien in Echtzeit:

Datenexport aus hauscomp.exe heraus

- Automatisch per Datei ***sps.sps*** (maschinenlesbar) bzw. ***sps.txt*** (Augen optimiert) aktualisiert aller paar Minuten in der Form:

```
AUS 0 0.000 -
EIN 1 1.000 -
Testmodus 0 2.000 -
AUTOMAT 1 0.500 -
WERKTAG 0 0.001 -
...
```

```
 0 AUS       [ ]      0 -
 1 EIN       [X]      1 -
 2 Testmodus [ ]      2 -
 3 AUTOMAT   [X]  0.500 -
 4 WERKTAG   [ ]  0.001 -
 ...
```

  Die Datei ***sps.sps*** listet die Werte aller 1199 SPS-Module, die Datei ***sps.txt*** dagegen nur die Werte programmierter SPS-Module.
- Beschleunigt per Datei ***hcdata.txt*** für bis zu 39 SPS-Werte (aktualisiert sofort nach Änderung der entsprechenden Werte) in der Form:

```
1 2 0 0 0 0 10 100 1000 3 6 20615 3 0 0 0 0 0 0 0 0 0 0 0 0 0 2023 1016 6958 23 0 625 1 0 0 0 0 0 0 32387
```

  Die Programmierung der SPS-Module für die Werte erfolgt dabei mit den Scriptbefehlen „HCdata=“, „HCdata2=“, „HCdata3=“ in der hauscomp.ini
- In frei wählbarer Datei in einstellbarem Zeitintervall mit bis zu 33 SPS-Werte per „D“- Befehl in Steuer-Datei ***taste.txt*** (siehe Abschnitt 2.11). Aus Gründen der Performance ist bei kurzen Zeitintervallen (<30 s) die Variante mit der Datei ***hcdata.txt*** vorzuziehen.

Datenimport/Steuerung hauscomp.exe

- In (fast) Echtzeit per Datei ***taste.txt*** mit eigener Syntax (siehe Abschnitt 2.11)
- In (fast) Echtzeit per Datei ***datain.txt***, wenn diese Funktion mit dem Scriptbefehl ***DataIN=on*** (hauscomp.ini) freigeschaltet wurde. So können an bis zu 20 SPS-Module der Zustand (Ein/Aus) oder auch analoge Werte übergeben werden:

```
150 0 1 151 6 99.3 0 0 407
```

Des Weiteren gibt es zahlreiche statistische und Log- Dateien (z.B. ***sps.csv***), die für Auswertungen herangezogen werden können.

3. Syntax der hardware.ini – Datei

3.1. Allgemeines:

Die *hardware.ini* Datei dient der Vorgabe vom Rechner abhängigen Einstellungen. Sie wird nur beim Programmstart berücksichtigt. Somit kann ein beliebiger Arbeits-PC (als Test-Rechner!) unabhängig vom Hauscomputer konfiguriert werden. Wenn Schlüsselwörter nicht in der *hardware.ini* auftauchen, werden vom Programm die entsprechenden Standardwerte gesetzt. Konvertierungsprobleme ANSI / ASCII können wegen der festgelegten Syntax nicht auftreten. Die *hardware.ini* kann auch „leer“ sein.

3.2. Schlüsselwörter (Bei Nichtangabe werden intern die Standardwerte verwendet):

- Zuweisung der seriellen PC-Schnittstelle für den Hausbus (notwendige spezielle Hardware). Dabei gilt folgende Zuordnung:

COM1: 03F8-03FF	COM2: 02F8-02FF	COM3: 03E8-03EF	COM4: 02E8-02EF

Fehlt dieser Scriptbefehl in der *hardware.ini* oder wird COM=0 angegeben, so wird kein COM-Port bzw. keine serielle Schnittstelle angesprochen. COM=5 dient zur automatischen COM Suche unter Verwendung eines „SDA-Korrektur- Adapters“. Der Adapter vergrößert die zulässige I2C-Bus Leitungslänge.

Bei der automatischen Initialisierung werden dabei nacheinander alle Ports auf Funktion abgefragt. Wird keine Hardware gefunden, so bleibt das Programm auf COM 4 aktiv. Besonderheit unter Windows XP: Das Betriebssystem bemängelt den (nicht funktionierenden) angesprochenen COM-Port 4, → ignorieren. Wird bei laufendem Programm im Systemmenü der Bus abgeschaltet, so werden die Laufschrift und die blinkende rote Farbe ebenfalls abgeschaltet.

```
COM= 1 {0,1,2,3,4,5}
COM= 1                          {Verwendung der seriellen COM-Schnittstelle 1}
```

- Festlegen der Impulslänge auf dem I2C Bus. Standardwert= 180 (µs). Verwendung bei kurzen Busleitungen um Rechenzeit zu gewinnen. Kann auch genutzt werden, um die maximal mögliche Buslänge abzuschätzen (z. B. testweise mit „45“ laufen lassen – bei Fehlerfreiheit kann der Bus ca. 4x länger sein). Vergrößerung der Impulslänge führt zu stabileren Busabfragen, reduziert aber die Geschwindigkeit des Grafikaufbaus. Ab V.6.0.32 wird ein derart modifizierter Bustakt nicht mehr zusätzlich um einen Faktor (wegen vorhandener spezieller I2C-ICs) korrigiert. Bei langsamen Rechnern kann die tatsächlich gemessene Taktzeit (Seiten Hardwarestatus, Zeitkern) vom eingestellten Bustakt deutlich abweichen. Unter Windows XP beträgt sie aus technischen Gründen nur die Hälfte des eingestellten Wertes.

```
Bustakt=    50 {20..255}              Standardwert=180
```

- Manuelles Festlegen des übergeordneten Betriebssystems – wird im „Zeitkern2“ angezeigt

```
BS=    Linux {Linux,DOS,XP,98}              Standardwert=wird automatisch ermittelt
```

- Scriptbefehl zur Herstellung der Kompatibilität der Bussequenz zu Versionen, die älter als 6.0.38 sind. Siehe auch 1.3.3.

```
Bussequenz=      alt {neu}            Standardwert=neu
```

- Festlegen der maximalen Anzahl der binären ICs (mittlerweile nicht nur PCF8574P). Vom Programm werden dann nur noch die Adressen 0 bis (Anzahl_PCF8574P – 1) abgefragt. Dieser Scriptbefehl sollte nur dann verwendet werden, wenn die zyklische Abfragezeit für die binären Schaltkreise zu lang sein sollte. Die normale Abfragezeit beträgt bei 8x PCF8574P: (8-1)*4/18.2= 1.5 s. Unter Nutzung dieses Scriptbefehls kann man diese Zeit auf 1*4/18.2= 0.23 s verkürzen. Die hohe Abfragefrequenz des PCF8574P(0) beträgt immer 1/18.2= 0,055 s. Die Abfragefrequenz der PCF8574P(1..7) beträgt: (Anzahl der IS-1)*4/18.2 [s]. Standardwert: 8. Bei Steuerungsanwendungen ist neben der Ein- und Ausgabezeiten zusätzlich noch die Zykluszeit der SPS zu berücksichtigen. Ab V6.0.41 gibt es diesen Scriptbefehl nicht mehr.

```
Anzahl_PCF8574P=      1 {0,2,3,4,5,6,7,8}
```

- Festlegen der maximalen Anzahl der analogen PCF8591. Vom Programm werden dann nur noch die Adressen 0 bis (Anzahl_PCF8591 – 1) abgefragt. Allerdings werden dann die höheren Adressen generell nicht mehr abgefragt! Standardwert: 8. Ab Version 6.0.49 werden nicht initialisierte Kanäle automatisch übersprungen (schnellere Abarbeitung), was diesen Befehl mittlerweile überflüssig macht.

```
Anzahl_PCF8591= 1 {0,2,3,4,5,6,7,8}
```

- Zuschaltung des Interrupt 21. Der Interrupt 21 (hier: Funktion 19h) dient dem Programm *hauscomp.exe* unter DOS für die zuverlässige Arbeit der Netzwerkanbindung über TSR Programme und kann mit diesem Scriptbefehl eingeschaltet werden. Läuft das Programm dagegen unter anderen, bereits von Haus aus netzwerkfähigen Betriebssystemen, ist der Interrupt nicht notwendig. Die Abschaltung ist zwingend notwendig, wenn *hauscomp.exe* unter der Dos – Emulationssoftware „Dosemu“ unter Linux gestartet wird.

```
INT21=      on {off}        Standardwert=off
```

- Einstellung der automatischen Erkennung des SDA-Korrektur Adapters bzw. des 230V Schaltinterface. Notwendig, wenn *hauscomp.exe* unter der Dos – Emulationssoftware „DosBox“ gestartet wird. Das TxD – Senderegister wird dort anders behandelt. Unter „Dosemu“ muss „on“ angegeben werden, wenn in der Hardwareübersicht der Busadapter „Deluxe“ angezeigt werden soll (wenn vorhanden).

```
SDAKorrektur=    on {off, auto}       Standardwert=auto
```

- automatische Korrektur im Programm in MEZ bzw. MESZ {auto}, Rechner-BIOS läuft dagegen ganzjährig mit MEZ und braucht nicht

umgestellt zu werden! Standardwert=off. Für den Hauscomputer sollte generell auto eingestellt werden. Besonderheiten bei der Zeitumstellung sind: Zeitangaben, die sich in der Nacht der Umstellung nach 2 bzw. 3 Uhr auf Analogsensoren beziehen, sind um eine Stunde verkehrt; bei der Umstellung auf Sommerzeit werden für 1 h alle Binärausgänge abgeschaltet (Verhinderung von Fehlauslösungen wegen „negativer Zeit“).

```
Sommerzeit=     auto {off}
```

- Einstellung der Bildschirmhintergrundfarbe (z. B. für Dokumentationen/ Bildschirmausdrucke o.ä.) Standardwert=normal. Zusätzlich können bei laufendem Programm über die Taste „i“ die Farben invertiert werden, wobei es dabei naturgemäß zu Farbänderungen kommt. Sollte das unerwünscht sein, kann „hellgrau“ verwendet werden. Dabei werden nur die Vorder- und Hintergrundfarbe getauscht.

```
Hintergrund=    normal {hell, blau, gruen, cyan, rot, grau, hellgrau}
```

- Festlegung des Watchdog-Ausganges (Blinksignal am entsprechenden E/A-Kanal). Für jeden der 8 möglichen PCF-8574P kann ein Watchdog-Ausgang definiert werden. Die Blinkfrequenz beträgt bei Adresse 0 etwa 4Hz, bei den Adressen 1-7 etwa 0,3Hz. Ein längeres Ausbleiben des Pegelwechsels wird durch die Hardware (Interface) als Programmausfall gewertet. Voraussetzung für die Erkennung ist die entsprechende Brücke zwischen dem Ausgang des programmierten Kanals und dem Watchdog Eingang. Diese Sicherheitseinrichtung sollte immer verwendet werden. In der Voreinstellung ist keine Watchdog definiert.

```
Watchdog=    [Adresse]  [Kanal]
             0-7        0-7
```

- Einstellung der Zeit zwischen den Bildschirmwechseln im automatischen Modus. Standardwert=18 Sekunden. Dieser Scriptbefehl korrespondiert mit dem Befehl „Anzeigeseiten=“ im nächsten Kapitel. Der erste (Zeit-) Wert bezieht sich auf den Wechsel von der ersten Seite zur zweiten Seite, der zweite Wert auf den Wechsel von der zweiten Seite zur dritten Seite und so weiter. Wird nur ein (Zeit-) Wert angegeben, so wird dieser Wert für alle Wechsel gleichermaßen verwendet.

```
Seitenwechsel= [Sekunden] ([Sekunden]) ([Sekunden])([Sekunden])([Sekunden])([Sekunden])
                 2-59         2-59          2-59         2-59         2-59         2-59
```

- Automatische Erzeugung einer Laufschrift

  In der Laufschrift werden fortlaufend dargestellt:

  1. Begrüßung und Wochentag; 2. Ergebnis des ständig laufenden Eigentest (Hard- und Software); 3. aufgetretene Über- bzw. Unterschreitungen bzw. blinkende rote Farbe in Bildschirmanzeigen (Ereignisrecorder); von Hand betätigte SPS-Module, die zur Auslösung führen werden dabei ignoriert; 4. Mitteilung über kalendarische Ereignisse des laufenden und des nächsten Tages. 5. Lizenzinformationen.

  Über-/ bzw. Unterschreitungen können durch die Taste OK gelöscht werden. Nach der Löschung sind die entsprechenden Angaben noch für ca. 5 min sichtbar. Wenn kein COM-Anschluss angegeben wird, schaltet sich die Laufschrift automatisch ab. Mit der Angabe „statisch“ wird 40s

nach jeder Tastenbetätigung der statische Anzeigemodus bei langsamen PCs automatisch aktiviert. Die Angabe „Test“ führt zur Anzeige von speziellen Variablen im Bereich der Laufschrift, die Angabe „noInfo“ deaktiviert die Laufschrift und das Ausrufezeichen „Hinweis“. Standardwert=off.

```
Laufschrift=    on  {off, statisch, noInfo, Test}
```

- Aktivierung der Log- Datei „Laufband.log“. Es wird täglich um Mitternacht eine Kopie des Laufbandes ausschließlich der Kalendererinnerungen an diese Datei angehängt. Damit sind alle Probleme und Hinweise, die zu diesem Zeitpunkt mit der Taste „O“ noch nicht bestätigt wurden registriert. Programmstart und -Ende werden immer in der Datei aufgezeichnet.

```
LLog= on {off}       Standardwert=off
```

Gedacht ist diese Funktion für die Langzeitüberwachung der Hardware z. B. bei Abwesenheit. Jeder Programmstart und jedes manuelle Programmende wird ebenfalls aufgezeichnet. Die Betätigung der Taste „L“ bewirkt ein Abspeichern des aktuellen Inhaltes des Laufbandes.

- Aktivierung der Log-Datei „Ereignis.log“. Bei Auftreten eines neuen Hinweises wird dieser sofort an diese Log- Datei angehängt. Registriert werden ausschließlich Hinweise wie z. B. „WW_Speicher (60) Wert größer MAX“ oder „Wst Ausfall“, nicht aber Probleme wie z. B. „Der Hausbus ist nicht angeschlossen!“. Im Unterschied zum Laufband werden mehrfach auftretende Hinweise nicht gesammelt, sondern einzeln mit Datum und Uhrzeit ausgewiesen. Die regelmäßige Speicherung der Daten geht zulasten der Rechenzeit, sodass diese Funktion auf langsamen Rechnern nur zur Fehlersuche und nicht dauerhaft eingesetzt werden sollte.

```
ELog= on {off}       Standardwert=off
```

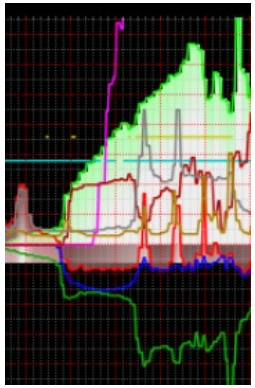
- Vorgabe der wichtigsten Zeitkonstante des Zeitkerns – sie beschreibt, wie oft die den Hausbus steuernde Interruptroutine je Sekunde aufgerufen wird und hat damit grundlegenden Einfluss auf alle internen Zeitmesser, insbesondere der Betriebsstundenzähler und 24h-Grafikausgaben. Die Aufrufe selbst werden vom Betriebssystem organisiert und erfolgen unter DOS 18,206 mal je Sekunde. Unter anderen Betriebssystemen wie z. B. Linux können Abweichungen auftreten. Das Programm findet durch Vergleich mit der PC-Uhrzeit automatisch nach mehreren Tagen den korrekten Wert für zkonst und zeigt ihn im Zeitkern an (ab 6.0.39e stündliche Anpassung). Nach Programmstart wird aber zunächst vom DOS Wert (18.2…) ausgegangen. Diesen Startwert kann man mit dem Scriptbefehl auf andere Werte setzen, sodass der Rechner bereits unmittelbar nach einem Neustart (z. B. nach Stromausfall) mit den richtigen Werten läuft. Auf verschiedenen Testrechnern wurde unter Linux nach 3 Tagen z. B. Werte zwischen 17,574 und 18,2 je nach Umfang zusätzlich laufender Programme ermittelt (abgelesen im Zeitkern unter „zkonst“).

```
zkonst=    10.000...25.000          Standardwert=18.20651
```

- Zuweisung des Abfrageintervalls der S0-Zählern/Energiezählern im Busadapter „Deluxe“.

S0Intervall= [Zeitintervall (s)] Standardwert: 540s (9min)

Dieser Befehl legt das minimale Abfrageintervall für die vier S0-Zähler (Busadapter „Deluxe“) fest. Dieses Zeitintervall ist entscheidend für die Berechnung von Leistung bzw. Verbrauch in den Modulen 50..53. Ein deutlich kürzeres Intervall als 540s verschlechtert die Genauigkeit der Berechnung und verlangsamt zusätzlich die Abfrage aller binären Interfaces. Eine Vergrößerung des Intervalls verlangsamt die Aktualisierungsrate der Leistungsermittlung und verbessert die Genauigkeit bei geringen Zählraten. Liefert eine Solaranlage z.B. nur 50W (→ 0,02kWh/Stunde) und der zugehörige Zähler (2000 Imp./kWh) ist an einem S0-Eingang angeschlossen, so werden in 540s 6 Impulse gezählt (0,02*2000/3600*540=6). Die Genauigkeit beträgt somit 6±1 = ca. ±20 %. Ab Version 6.0.59B2 wurde der methodische Messfehler verringert (neue Formel: Impulsdifferenz +0.5).



540 s: grüne Fläche mit Quatisierungen

Der aktuell eingestellte Wert ist in der Übersicht „Zeitkern2“ dargestellt. Ist die Leistungsberechnung nur für die grafische Darstellung (z.B. in der EDB) vorgesehen, so wären 300 s (5 min) optimal, weil die EDBs aller 5 min aktualisiert werden (keine Quatisierungseffekte, siehe Bilder). Bei 30 Impulsen im Abfrageintervall beträgt der Messfehler (Version 6.0.59B2 unter Linux) maximal ±1,7 % (z.B. Zähler 1000 Imp./kWh, 300 s Abfrageintervall, 30 Impulse → 360 W ±6 W).

- Direkte Vorgabe des Zählumfanges der Warteschleife in der Bussteuerung. Sinnvoll beim Einsatz immer gleicher Hardware.

zk= 0 {0..Longintmax} **Standardwert=0 (Funktion abgeschaltet)**

- Automatisches Programmende nach bestimmter Zeit [s] vorgeben. Zusammen mit einer Batchdatei (z. B. in der autoexec.bat) dient es zyklischen Test bei der Programmentwicklung. Beim Wert „0“ ist die Funktion abgeschaltet.

Autokill= 0 {0..Longintmax} Standardwert=0

- Automatisches Erzeugen einer Datei „alarm.txt“ bei wichtigen Ereignissen wird erlaubt (siehe auch Abschnitt 2.9.). Dient zur Auslösung von automatischen Mails unter Dosemu/Linux. Die Datei wird nach dem erfolgreichen Senden vom Mail-Clienten wieder gelöscht.

Alarm= on {off} Standardwert=off

- Deaktivierung der automatischen Font-Darstellung gemäß ISO

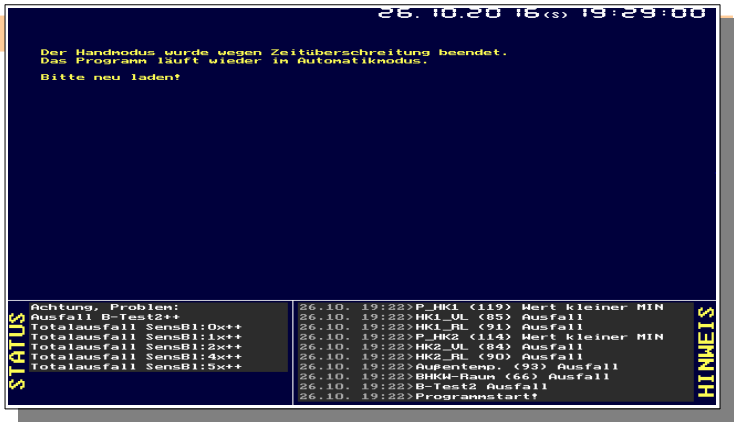
ISOVGA= off {on} Standardwert=on

Wird dieser Schalter auf „off“ gesetzt, so werden Textdarstellungen vom Programm immer per Datei „font.abc“ konvertiert, auch wenn das Betriebssystem lt. ISO korrekte (ANSI) Fonts darstellen würde. Nur so ist es möglich, anstelle von „m^{n}“ korrekt „m³“ anzuzeigen. Im originalen FONT-Satz von DOS bzw. Dosemu gibt es „³“ nicht. Des Weiteren ändert sich ein wenig das Layout einiger Sonderzeichen. Nachteil dieser Betriebsart: ca. 40 % aller Zeichen müssen konvertiert, das heißt „gemalt“ werden. Das kostet, je nach Umfang der darzustellenden Texte, etwa 5–10 % der Rechenzeit und wird für langsame Rechner (z. B. KONTRON GX-1) nicht empfohlen.

- Aktivierung der Grafikauflösung 640x480

VGA= on {off} Standardwert=off

Wird dieser Schalter auf „on“ gesetzt, so wird das Programm im VGA-Grafikmodus 640x480 mit 16 Farben dargestellt (Standard ist sonst 640x200 bei 16 Farben). Neben einer vergrößerten Darstellungsfläche werden im unteren Bereich 2 Fenster mit dem aktuellen Hardware- bzw. Programmstatus und den Hinweisen dargestellt, die sonst nur in der Laufschrift sichtbar waren. Die Laufschrift kann somit mit dem Scriptbefehl „Laufschrift=noinfo“ deaktiviert werden. Die höhere Auflösung in Verbindung mit den zusätzlichen Fenstern bedeutet natürlich auch mehr Rechenzeit, weshalb ein leistungsfähiger PC vorausgesetzt wird.

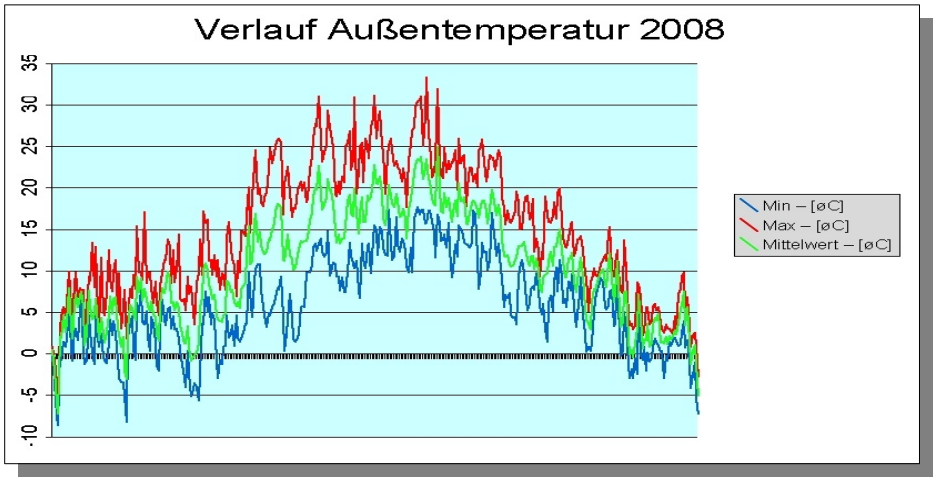


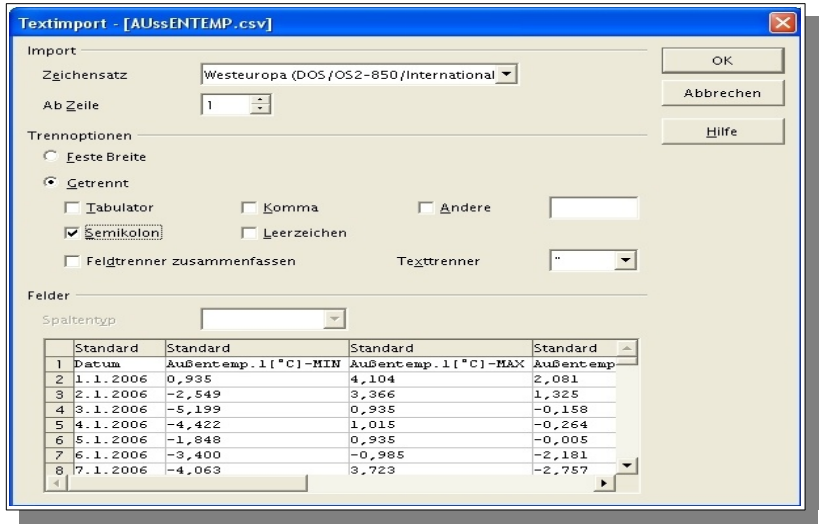
- Aktivierung von Log-Dateien für jeden angeschlossenen Analog-Kanal und ab V. 6.0.40 auch für ausgewählte SPS Module. Es wird täglich um Mitternacht automatisch ein Datensatz mit dem Min-wert, dem Max-wert, dem Tagesdurchschnittswert und Fehlerinformationen an diese Dateien angehängt. Der Dateiname setzt sich zusammen aus der Kanalbezeichnung, dem Jahr, der Adresse und der Kanalnummer. SPS Module vom Typ 66 (Tagesbetriebsminuten), 97 (Tagesertragszähler), 102 (Übernahmemodul), sowie Module, in deren Maßeinheit „/d“ enthalten ist, werden ebenso in der Form „<Modulnummer>_<Jahr>.sps“ automatisch aufgezeichnet. Die Dateien können mit einem Tabellenkalkulationsprogramm ausgewertet oder auch automatisch mit dem Analysetool gnuplot unter Linux (siehe Abschnitt 6.10 oder auch [6]) in eine png-Grafik gewandelt werden. Damit lassen sich z. B. Langzeitklimadiagramme oder Ölverbrauchswerte erstellen. Als Trennzeichen wird in den Dateien das Semikolon verwendet. Eine Datei für einen Analogkanal verbraucht ca. 16 kByte je Jahr. Weiterhin wird die Betriebsstundendatei „stunden.txt“ aktualisiert.

ALog= on {off} Standardwert=off

Ein Beispiel für die weitere Datenanalyse einer Log- Datei:

Die zu analysierende Log- Datei wird zunächst von Hand umbenannt in *.csv und dann unter OpenOffice Datei/Öffnen als Textdatei importiert. Dabei sind die entsprechenden Optionen wie nebenstehend einzustellen. In der entstandenen Tabelle löscht man die Spalte „...Anzahl Messwerte“. Danach fügt man ein Diagramm unter Verwendung der ersten Zeile und Spalte für die Beschriftung ein und folgt dem entsprechenden Assistenten.





Als Ergebnis erhält man eine übersichtliche Grafik mit dem Temperaturverlauf (in diesem Beispiel) über einen längeren Zeitraum. Selbstverständlich kann man über die Kopier- und Einfüge-Funktionen auch verschiedene Sensordaten kombinieren und entsprechend auswerten. Weiterführende Infos siehe Abschnitt 6.10.

- Aktivierung einer/zweier Log-Datei(n) für maximal 16 ausgewählte Module der SPS. Nach vorgegebener Zeit oder bei Zustandswechsel wird ein entsprechender Datensatz mit dem aktuellen Zustand (z) und dem analogen Wert des Moduls (f) an diese Datei (SPS.csv bzw. SPS2.csv) angehängt. Die Datei kann natürlich wie im vorangegangenem Abschnitt mit einem Tabellenkalkulationsprogramm ausgewertet werden. Die Umbenennung in *.csv entfällt dabei. Der große Vorteil des Befehls ist die freie Wahl der SPS-Module und des Zeitintervalls. Für jeden Eintrag werden neben Datum und Uhrzeit die Unix-Zeit (Sekunden nach dem 1.1.1970 MEZ) und die Sekunden seit Programmstart geschrieben. Das Log beginnt mit der 16. Sekunde ab Programmstart gerechnet. Wird als Intervall Zeit „0“ angegeben, so erfolgt ausschließlich täglich um Mitternacht eine Aufzeichnung. Die Zeichenanzahl je Zeile ist auf 255 begrenzt: darüber hinaus gehende Zeichen werden abgeschnitten. Bei der Angabe „SPS“ erfolgt die Aufzeichnung im Zeitintervall nur dann, wenn der zusätzlich angegebene SPS Modul „ein“ ist. Damit kann eine bedingungsabhängige Aufzeichnung erfolgen (z. B. Aufzeichnung aller 10s, wenn Heizungsbrenner EIN). Soll aufgezeichnet werden, wenn der SPS Modul dagegen „aus“ ist, so ist das Modul mit einem vorangestelltem Minus einzutragen. Die Aufzeichnung ausschließlich bei Zustandswechsel des SPS Moduls (Eventlog) wird mit „-1“ als Zeitintervall programmiert.

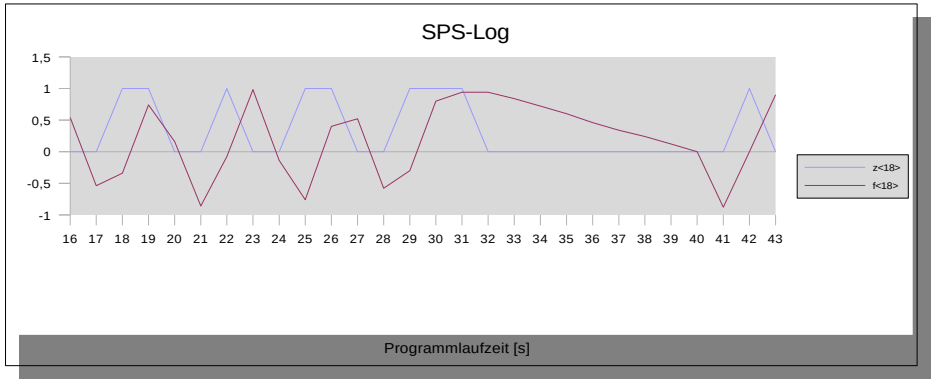
```
SPSLog=     on [Intervall] [Modul] ([Modul]) ... maximal 16 SPS Module
            0..20000000s  0..1199   0..1199   ...
SPSLog=     SPS [SPS Modul] [Intervall] [Modul] ([Modul]) ... maximal 16 SPS Module
                 -1199..1199    0..20000000s  0..1199   0..1199  ...
SPSLog2=    on [Intervall] [Modul] ([Modul]) ... maximal 16 SPS Module
                 0..20000000s  0..1199   0..1199   ...
SPSLog2=    SPS [SPS Modul] [Intervall] [Modul] ([Modul]) ... maximal 16 SPS Module
                 -1199..1199    0..20000000s  0..1199   0..1199  ...
```

SPSLog=on 1 18 39 41 /jede Sekunde werden die Inhalte der Module 18, 19 und 41 in der Datei „sps.csv“ registriert
SPSLog2=SPS 25 -1 /ändert sich der Zustand von „Binär-Adr. 1 OK“ (24) so wird eine neue Zeile an die Datei „sps2.csv“ angehängt

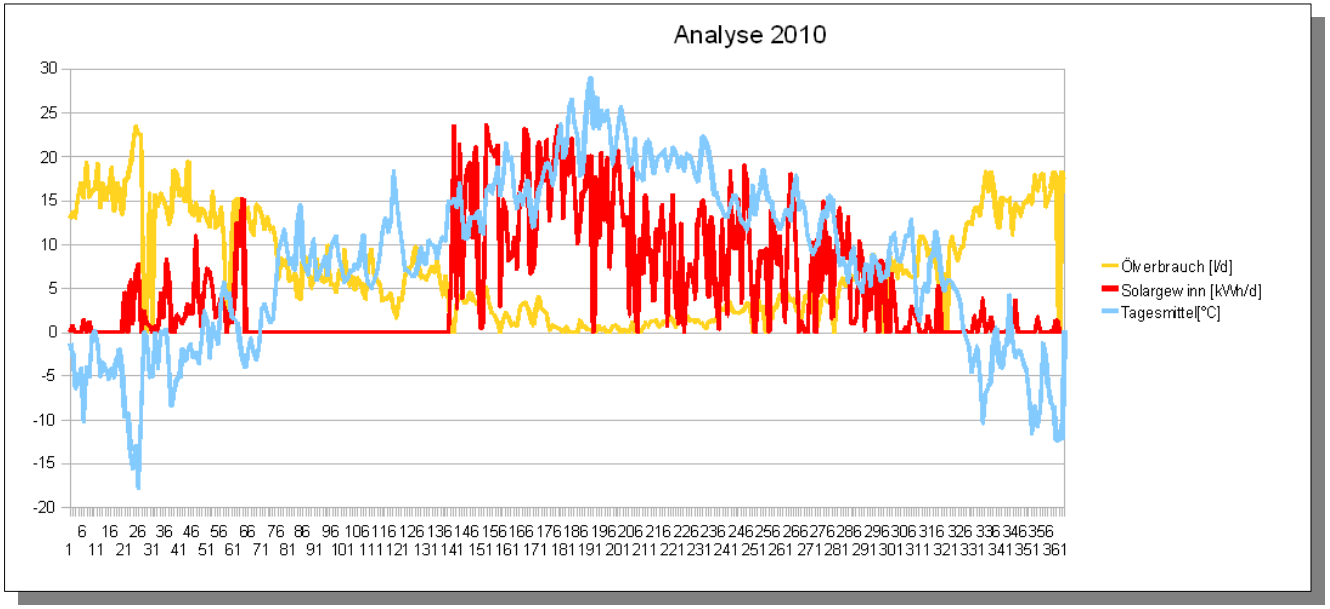
Das Ergebnis der ersten Zeile im Beispiel sieht folgendermaßen aus:

SPS-Log	aller 1s	Hauscomputer	****	(100 Zyklen)	(c%)	(PCF8591 (7) OK)	()	(Dunkel)	(ø)
Datum	Uhrzeit	Unix-Zeit[s]	Zeit[s]	z<18>	f<18>	z<39>	f<39>	z<41>	f<41>
26.09.06	02:32:20	1159320740	16	0	0,54	0	16	1	-35,97
26.09.06	02:32:21	1159320741	17	0	-0,54	0	17	1	-35,97
26.09.06	02:32:22	1159320742	18	1	-0,34	0	18	1	-35,97
...									

Ein Beispiel für die weitere Datenanalyse mit der Tabellenkalkulation von Open Office:



Hier noch ein Beispiel für eine weitergehende Aufbereitung der täglich gesammelten Verbrauchs-Daten in einer konkreten Anwendung:



- Aktivierung einer Log- Datei für die Aufzeichnung von Rohdaten analoger Sensoren. Es werden jeweils alle Sensoren der angegebenen Adressen aufgezeichnet, maximal also 8 x 4 = 32 Sensoren. Nach vorgegebener Zeit wird ein entsprechender Datensatz an diese Datei (Sensoren.csv) angehängt. Aufbau der Datei und Auswertung siehe Scriptbefehle „SPSLog“ bzw. „ALog“. Für jeden Eintrag werden neben Datum und Uhrzeit die Unix-Zeit (Sekunden nach dem 1.1.1970 MEZ) und die Sekunden seit Programmstart geschrieben. Das Log beginnt mit der 16. Sekunde ab Programmstart gerechnet. Wird als Intervall Zeit „0“ angegeben, so erfolgt ausschließlich täglich um Mitternacht eine Aufzeichnung. Die anfallende Datenmenge beträgt ca. 300 kByte je Stunde bei 2 Adressen á 4 Sensoren bei einem Zeitintervall von 1 s . Die Speicherung der Daten geht zulasten der Rechenzeit, sodass diese Funktion auf langsamen Rechnern nur in größeren Zeitintervallen eingesetzt werden sollte. Bei der Angabe „SPS“ erfolgt die Aufzeichnung nur dann, wenn der zusätzlich angegebene SPS Modul „ein“ ist. Damit kann eine bedingungsabhängige Aufzeichnung erfolgen.

```
SensLog=    on [Intervall] [Analog-Adresse] ([Analog-Adresse]) ... maximal 8 analoge Adressen
                 0..20000000s       0..7            0..7          ...
SensLog=    SPS [SPS Modul] [Intervall] [Analog-Adresse] ([Analog-Adresse]) ... max.8 anal.Adr.
                 -1199..1199  0..20000000s       0..7            0..7          ...
```

SensLog=on 10 1 2 /aller 10 Sekunde werden die Sensorwerte der analogen Adressen 1 und 2 der Datei Sensoren.csv hinzugefügt

zweites Beispiel, eine „menügesteuerte Aufzeichnung“:

in *hardware.ini*:

SensLog=sps 99 1 0 /jede Sekunde alle Kanäle der Adresse 0 aufzeichnen, wenn SPS Modul 99 „ein“ ist

in *hauscomp.ini*:

z=99 91 Aufzeichnung /Bezeichnung „Aufzeichnung“ dem Modul 99 zuweisen
Zeile= 111 0 1 99 /Bildschirm-Taste auf Seite 111 programmieren, die den Modul 99 umschalten kann

- Aktivierung einer speziellen Log- Datei für Heizungs- und Solarsteuerungen.

```
HzgLog=     on {off}              Standardwert= off
```

Bei Aktivierung wird ein Datensatz mit 46 Parametern der SPS in einer Datei „mixxxxxx.txt“ minütig abgespeichert. Die Datei beinhaltet die Daten einer Woche (Montag 00:00 bis Sonntag 23:59). Die Parameter werden über Scriptbefehle „HzgLogXXX“ in der *hauscomp.ini* festgelegt. Weitergehende Informationen finden sich im Beispiel 6.9.

- Festlegung des geografischen Standortes des Computers. Die Daten werden für die korrekte Berechnung des Sternenhimmels und des Sonnenauf- bzw. Unterganges und damit des virtuellen Helligkeitssensors benötigt (Astrofunktionen).

```
Geo=   Länge(grd) Länge(min) Breite(grd) Breite(min)          Standardwert= 11 12 53 56
Geo= 11 22 47 15        /geografische Koordinaten für Innsbruck (Österreich)
```

- Festlegung der Höhe des Standortes des Computers über Normal-null (NN). Die Daten werden für die korrekte Berechnung der Sonnenintensität und des UVI Indexes benötigt (Astrofunktionen).

```
NN=    Höhe(m)          Standardwert= 40
```

- Festlegung des Zeitintervalls der Aktualisierung von Dateien, auf die die Datei „Index.htm“ zugreift. Bei einem Betrieb des Programms in einem Netzwerk ergibt sich damit ein „Webinterface“. Bei einem Zeitintervall=0 ist die Aktualisierung abgeschaltet. Während der Aktualisierung bleibt die Laufschrift stehen und der Hausbus ist inaktiv. Es erfolgt ein Zugriff auf den Datenträger (Flash empfohlen!). Sinnvoll ist ein Zeitintervall von 10 min (=600s). Siehe auch Anwendungsbeispiele Punkt 6.6. Mit der zusätzlichen Angabe „RAM“ werden die Daten unter Linux in einem Ordner im RAM abgelegt, der Flashspeicher wird nicht beschrieben → schneller, geringere Abnutzung Flashspeicher.

```
Netzwerk=   Zeitintervall (10.. 65000s) {RAM}          Standardwert= 0
Netzwerk= 600       /aller 10 min werden Dateien im momentanen Verzeichnis aktualisiert
```

- Vorgabe des „live“ Modus für Netzwerkbetrieb. Dabei wird aller 15 Sekunden die Datei „taste.txt“ auf eventuellen Inhalt abgefragt und anschließend wieder geleert. Der Inhalt wird als Tastenkode interpretiert und ausgeführt. Wird ein Kode detektiert, so wird 30 s lang das Abfrageintervall von 15 s auf 3 s reduziert. Wird z. B. in die Datei über das Netzwerk eine „114“ geschrieben, so führt das Programm einen „Restart“ durch. 2 s nach Ausführung des Inhaltes der Datei „taste.txt“ wird der aktuelle Bildschirminhalt in der Datei „live.bmp“ abgespeichert, die über das Netzwerk angezeigt werden kann.

```
live= on {off}       Standardwert=off
live= on       /Einschalten „live“ Modus
```

- Festlegung eines alternativen Laufwerkes/Pfades zu zahlreichen Dateien, wie z.B. „taste.txt“. Siehe auch 2.12.

```
PfadTaste=    [Pfad]
              String[79]
PfadTaste=     c:\wz\HC\test
```

- Zusammenarbeit mit der Watchdog eines Industrie-PCs (PC-Watchdog). Automatischer Systemstart bei Softwarefehlern.

```
PCWatchdog=    Port-Adresse(Dezimal)   Wert(Dezimal) Endwert(Dezimal)     /Standardwert= 0   0   0
                 0..255                    0..255       0..255
```

PCWatchdog=250 2 100 /für einen Industrie-Prozessor vom Typ GX1 – in Adresse 0FAh wird 2 geschrieben == 45s Timeout bei lfd. hauscomp.exe; 50 min Timeout nach regulärem Programmende

Voraussetzung: Das BIOS unterstützt diese Funktion. Ist die Watchdog im BIOS des Ziel-PCs mit z. B. 2 min aktiviert, so zählt ein PC interner Timer bis auf null herunter und initialisiert nach spätestens 2 min einen Hardware-Reset – sofern nicht ein Anwendungsprogramm (z. B. *hauscomp.exe*) regelmäßig diesen Timer auf einen bestimmten Wert wieder anhebt. Der beschrieben Scriptbefehl lässt das Programm aller 14s in die angegebene Port-Adresse (z. B. 250d bzw.0AFh) den angegebenen Wert (im obigen Beispiel „2“) schreiben. Der Wert errechnet sich bei einem GX1 Prozessor wie folgt: 1 = 15s, jede weitere Zahl jeweils 30s mehr. Bei einem Wert von 2 wird der PC beim Hängenbleiben des Programms bzw. Programmabsturz spätestens nach 15+30+14=59 [s] einen Programmneustart herbeiführen. Sollte die *hauscomp.exe* für Laufwerksaktionen mehr Zeit benötigen als für die Watchdog vorgesehen, so kann ein vorzeitiger automatischer Neustart ausgelöst werden. Wird der Scriptbefehl nicht benutzt oder die Port-Adresse 0 verwendet, so ist diese Funktion im Programm *hauscomp.exe* deaktiviert. Der dritte Dezimalwert wird bei regulärem Programmende (<ALT>+<F4>/<k>) in die Watchdog geschrieben und bewirkt eine lange Phase ohne lästigem Neustart, da die regulären DOS-/Windowsprogramme, die dann aktiv werden, eine Watchdog leider nicht unterstützen.

- Festlegung der A/D Wandlerzeit für ATtiny26 (als Analogsensor). Wandlerzeit=Faktor*100µs. Für die Firmware I2C_A1.0 ist als Faktor 7 einzustellen, ab Firmware I2C_A1.1 ist dieser Scriptbefehl nicht mehr notwendig.

```
XFaktor=    [Faktor]          Standardwert= 1
```

Beispiel für eine *hardware.ini* Datei:

```
COM=1
Hintergrund= hell
Sommerzeit= auto
Geo=  11 22 47 15        /geografische Koordinaten für Innsbruck (Österreich)
NN= 574                  /Höhe von Innsbruck über Normal null
Laufschrift=  on
LLog=on                  /Log vom Laufband erstellen
ALog=on                  /Log vom allen Analogkanälen
Watchdog=  3  7
Watchdog=  5  7
Seitenwechsel= 13 10 17   /Hausparameterseite: 13s, Kalenderseite: 10s, Grafikseite: 17s sichtbar
```

- Aktivierung von Datenpaketen für externe Datenbank-Programme mit regelmäßiger Aktualisierung (→Streaming im RAM).

EDB=[Nr][Name Datenbank] [YMin] [YMax] [SPS-Adresse]([SPS-Adresse]) ... maximal 10 SPS-Adressen
0-9 String real real -1199..+1199 -1199..+1199

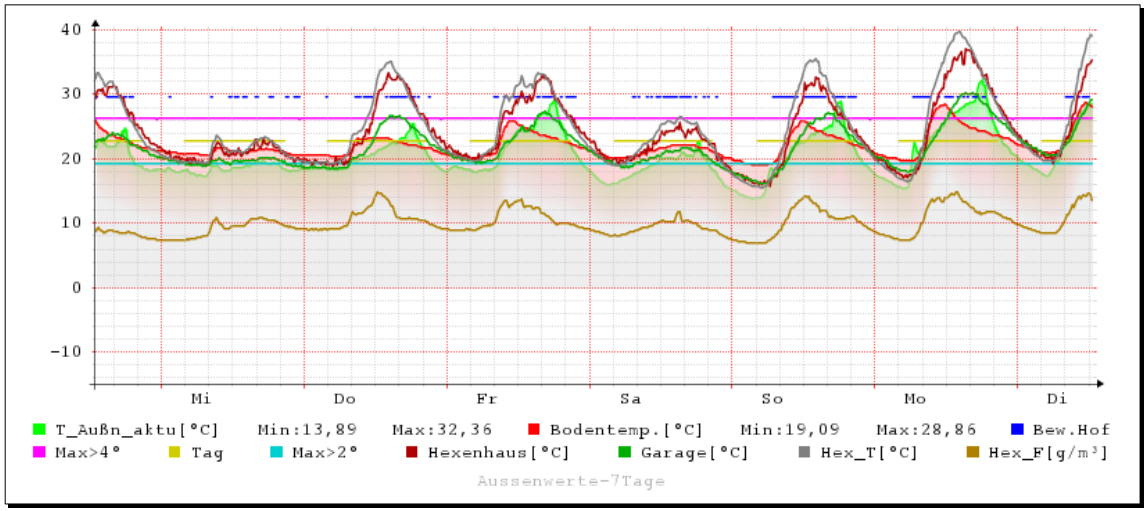
Bei Aktivierung werden bis zu 10 Dateipakete mit frei wählbaren SPS-Daten regelmäßig im Intervall von 5 min aktualisiert, genauer: alle 30s eines von 10 Datenpaketen. Diese Daten können von externen Datenbanken verwendet werden (z. B. RRDtool unter Linux). Jedes Dateipaket besteht jeweils aus 2 Dateien, die bis zu 10 unterschiedliche SPS-Daten (SPS-Nr, Bezeichnung, Einheit, Wert) enthalten. Die Dateibezeichnungen sind: edbx.bez und edbx.dat, wobei x die Nummer der eingelesenen Datenbank ist. Der angegebene Name erscheint später in den Grafiken. YMin und YMax legen den vertikalen Wertebereich in der späteren Grafik fest. Insgesamt werden so bis zu 100 Daten regelmäßig aktualisiert zur weiteren Auswertung zur Verfügung gestellt. Der Datenbankname darf keine Umlaute oder „ß“ beinhalten.
Die SPS-Werte sind in Abhängigkeit von der angegebenen SPS-Adresse:

- bei positiver SPS-Adresse → der anaolge SPS-Ausgang (f)
- bei negativer SPS-Adresse → der Zustand des SPS-Ausganges (Wert „-999“ – entspricht „aus“; „ypos“ – Wert in der Grafik für „ein“) „ypos“ ist ein automatisch ermittelter Wert im Y-Wertebereich – je nachdem, an wievielter Stelle die SPS-Adresse im Scriptbefehl steht. So werden Zustände in der Grafik optisch sinnvoll von oben beginnend gleichmäßig angeordnet.
- „0“ wird ignoriert, daher als vorläufiger Platzhalter für spätere Graphen nicht geeignet (besser: „4“ verwenden)

EDB=0 Systemintern -10 +10 19 22 40 -47 /Abspeicherung folgender Werte in „edb0.bez“ bzw. „edb0.dat“: aktuelle Zykluszeit, aktuelle I-Zykluszeit, aktuelle berechnete Helligkeit, Notbetrieb (ein/aus). Die Darstellung erfolgt im Y-Bereich -10.. +10

Bei der späteren Verwendung der erzeugten Dateien (z. B. in der Round-Robin Datenbank „RRDtool“ unter Linux) können die *.dat Dateien nach dem Auslesen auch regelmäßig wieder gelöscht werden.

Rechtes Bild: Beispiel für die automatische Generierung einer Grafik unter Linux unter Verwendung der EDB-Daten (HC_V2). Da die Daten dort ausschließlich im RAM übergeben werden, werden Flash-Speicher und Prozessor nur wenig zusätzlich belastet.



4. Syntax der hauscomp.ini - Datei

4.1. Allgemeines:

Die *hauscomp.ini* Datei dient der Vorgabe aller notwendigen Einstellungen. Sie wird nicht nur beim Programmstart, sondern jeden Tag jeweils um 0:00 Uhr (Mitternacht) berücksichtigt bzw. nach Auslösung eines Resets (Taste 'r'). Normalerweise wird die Datei mithilfe von Programmen unter WINDOWS™ erstellt und folgt damit dem ASCII-Standard. Sie kann aber auch von DOS-Programmen im ANSI-Format geschrieben werden. Das Programm *hauscomp.exe* läuft auch ohne einen einzigen Eintrag in der *hauscomp.ini*!

4.2. allgemeine Befehle [GLOBAL]:

- Automatische Korrektur der folgenden Zeichen entsprechend dem ASCII bzw. ANSI Standard. Praktisch sind alle deutschen Umlaute und Sonderzeichen betroffen. Dieser Befehl sollte immer in der ersten Zeile stehen und auch immer definiert sein. Programm intern wird immer ASCII verwendet, bei Dateiausgaben erfolgt eine Konvertierung in ANSI. Eine Veränderung der Datei *hauscomp.ini* findet nicht statt.

Norm= ASCII {ANSI} Standardwert: ANSI

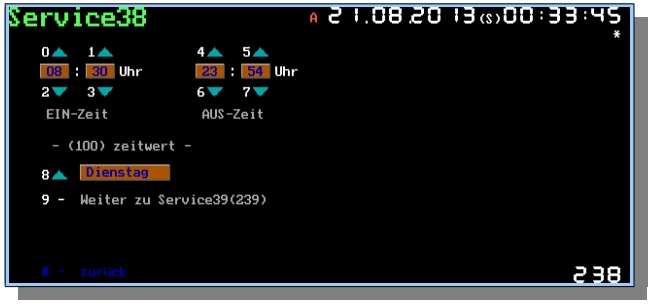
Norm= ASCII /Die folgenden Zeichen werden nicht konvertiert übernommen!

- Auswahl der Menüdarstellung. Es gibt 2 Möglichkeiten: große Schrift oder kleine. Ab Version 6.0.38. In der weiteren Beschreibung in diesem Handbuch werden alle Beispiele mit großer Schrift demonstriert.

Menüform= klein {groß} Standardwert: groß

Menüform=klein /durch die kleine Schrift wird zusätzlicher Platz für eigene Anzeigen geschaffen





- Einschalten des Testmodus. Nach Programmstart wird die automatische Seitenumschaltung ausgeschaltet und die angegebene Seite angezeigt. Bei Angabe „Test=255“ wird die Grafik dauerhaft abgeschaltet und nur noch die internen Steuer- und Regelungsfunktionen sind aktiv.

```
Test= [AnzeigeSeite]
         0..255
```

Test=18 /nach Programmstart wird dauerhaft die Hausbusseite angezeigt – sinnvoll beim Aufbau der Hardware!

- Festlegung der Reihenfolge angezeigter Seiten im Automatikbetrieb auf dem Bildschirm. Das Ganze wiederholt sich sequentiell, es kann aber auch dreimal die gleiche Seite (z. B. die Kalenderseite 96) angezeigt werden. Wird dieser Schlüsselbefehl nicht angegeben, so werden die voreingestellten Seiten 95 (Hausparameter), 96 (Kalender) und 97 (Grafik) angezeigt. Werden als Seiten 255 angegeben, so wird die Grafik komplett abgeschaltet (Sinnvoll für reine Steuerungsrechner ohne Display zur Leistungssteigerung). Es sind maximal 6 Seiten möglich.

```
Anzeigeseiten= [1.Seite]_[2.Seite]_[3.Seite]_{[4.Seite]}_{[5.Seite]}_{[6.Seite]}
                0..255      2..255    2..255     2..255       2..255      2..255
```

Anzeigeseiten= 95 96 97

- Festlegung einer Seite, die beim Übergang vom Automatik- zum Handbetrieb automatisch angezeigt wird, sozusagen als Startseite. Wird im Automatikbetrieb eine Taste betätigt, so wechselt das Programm nicht nur in den Handmodus, sondern zugleich auf die programmierte Seite → z. B. in ein Bedienmenü oder in das Hauptmenü

```
Handstartseite=       [AnzeigeSeite]      Standardwert: keiner
                         1..254
```

Handstartseite= 1 /bei Tastenbetätigung Hauptmenü anzeigen

- automatische Zeitkorrektur PC-Uhr jeden Sonnabend 02:30. Standardwert=0. Die PC-Uhr wird über ein BIOS-Interrupt um die entsprechenden Sekunden automatisch einmal in der Woche vor- bzw. zurückgestellt. Die Alternative wäre ein DCF-Modul gewesen - mit dem bekannten Treiberproblem.

```
Wochenkorrektur=      [Sekunden]
```

-1999 ... +1999 „-“ Uhr nachstellen; „+“ Uhr vorstellen

Wochenkorrektur= -14

- Zuweisung der E/A Signalleitung für das Signal „HINWEIS!“ (führt zum Blinken der grünen Betriebs-LEDs, wenn wichtige Informationen auftreten). Standardwert: Binäradresse 0, Kanal 3. Zusätzlich muss am A+D Interface eine Brücke zwischen dem programmierten Kanal und der Blinkschaltung gesetzt werden. In Fall der Adresse 2 Kanal 3 also zwischen den Kontakten _11 und P3. Das Signal „Hinweis!“ kann nur an einem Kanal ausgegeben werden. Siehe auch Abschnitt 2.10.

```
Hinweis!= [Adresse]_[Kanal]
            0-7      0-7
```

Hinweis!=2 3

- Zuweisung der E/A Signalleitung für die Bestätigungstaste „OK“ (dient zum Rücksetzen des Signals „HINWEIS!“ und nach kurzer Zeit zum automatischen Löschen des Ereignisrecorders im Laufband). Standardwert: Adresse 2, Kanal 4. Hardwaremäßig muss an dem entsprechenden Binärkanal eine Taste angeschlossen werden. Die Bestätigungstaste kann nur an einer Signalleitung verwendet werden. Alternativ kann die Taste <o> auf der Tastatur oder die Leertaste oder eine gekoppelte Taste am Bus genutzt werden.

```
TasteOK= [Adresse]_[Kanal]
           0-7      0-7
```

TasteOK=2 4

- Programmierung einer automatischen Zahlen-Ausgabe in Textform von vorgegebenen Werten der SPS in die Datei **hcdata.txt**. ***Achtung, dieser Scriptbefehl wurde ab Version 6.0.58B1 komplett überarbeitet!*** Dabei wird von den Ausgängen der SPS-Module (f) im Interesse einer beschleunigten Ausgabe, nur der ganzzahlige Wert (***im begrenzten Bereich -32768…+32767!***) unter Abschneidung der Kommastellen, gespeichert. Wird ein SPS-Modul mit einem „Minus“ vorgegeben, so wird anstelle von f der Zustand („0“ oder „1“) abgespeichert. Bis zu 39 SPS-Module können vorgegeben werden. Da in einer Scriptzeile ab dem Gleichheitszeichen nur maximal 80 Zeichen möglich sind, gibt es 3 Scriptbefehle mit jeweils 13 Module. Diese Anzahl muss nicht ausgeschöpft werden. Die Ausgabe in die Datei **hcdata.txt** erfolgt in einer Zeile mit Leerzeichen als Trennelement. Es werden immer alle 39 Werte geschrieben (nicht programmierte SPS-Module als Null). Als letzter Wert wird eine Quersumme in diese Zeile geschrieben (einfache vorzeichenrichtige Addition aller voriger Werte plus eins). Die Ausgabe erfolgt vollautomatisch 4 Zyklen nach Änderung der entsprechenden SPS-Werte (interne Überwachung der Quersumme). Ändert sich die Quersumme während dieser 4 Zyklen erneut, so wird solange mit dem Schreiben gewartet, bis keine Änderung mehr auftritt oder maximal 50 Zyklen vergangen sind. Das verhindert eine Blockierung des Programms durch ständige Dateizugriffe. Nach der letzten Aktualisierung der Datei erfolgt die nächste Speicherung bei unveränderten Werten automatisch nach 5000 Zyklen.

  *Hinweis: Die Datei **hcdata.txt** wird mit gesetztem „ReadOnly“ und „Archive“ Flags geschrieben. Das „abholende“ Script unter Linux/Windows muss die Datei aktiv löschen oder zumindest das „ReadOnly“-Flag zurücksetzen, bevor die Datei vom Programm*

hauscomp.exe erneut geschrieben werden kann. Das erhöht die Sicherheit bei der Datenübergabe und verhindert nicht notwendige Schreibzugriffe. Allerdings können die Daten beim „Abholen“ veraltet sein, da mit dem Schreiben entsprechend gewartet wird. Ist im gleichen Ordner eine Datei „version.txt“ vorhanden, so wird die Datei ***hcdata.txt*** *mit vorangestelltem Tiefstrich als* ***_hcdata.txt*** *geschrieben. Unter Linux HC_V2 werden dann automatisch von einem minütlichen Cronjob die Flags korrigiert und eine weiterverwendbare* ***hcdata.txt*** *gleichen Inhalts aktualisiert. Damit werden die Speicherproblematik des Puppy-Linux umgangen und die Schreib/Lesevorgänge entkoppelt.*

```
HCdata= [SPS Modul 1]_[SPS Modul 2] … {[SPS Modul 13]}
HCdata2= [SPS Modul 14]_[SPS Modul 15] … {[SPS Modul 26]}
HCdata3= [SPS Modul 27]_[SPS Modul 28] … {[SPS Modul 39]}
     jeweils:    -1199...+1199
```

HCdata= 543 547 682 -241 /Vorgabe von 4 SPS-Modulen (3x f, 1x Zustand)

Die Scriptbefehle

HCdata= 1 2 3 4 5 6 7 8 9 10 11 12 13
HCdata3=36 37 38 39 43 44

… erzeugten die Datei hcdata.txt:

1 2 0 0 0 0 10 100 1000 3 6 20615 3 0 0 0 0 0 0 0 0 0 0 0 0 2023 1016 6958 23 0 625 1 0 0 0 0 0 0 32387

(1)	1	f=1		(36)	2023	f=aktuelles Jahr
(2)	2	f=2		(37)	1016	f=COM-Adresse 3F8
…				…		
(9)	1000	f=1000		(43)	0	f=aktueller UV-Index
(10)	3	f=3,14159		(44)	625	f=Sonnenenergie je m² (ohne Bewölkung)
(11)	6	f=6,28318		(-10)	1	heute ist Donnerstag
(12)	20615	f=Echtzeit				
(13)	3	f=3,6			32387	Quersumme+1

- Eine, von einem externen Programm generierte Datei **datain.txt,** wird automatisch regelmäßig gelesen, wenn diese Funktion per folgendem Scriptbefehl freigeschaltet ist. Mit dieser Eingabemethode können laufend Werte direkt in die SPS geschrieben/aktualisiert werden.

```
DataIN=     on {off}        Standardwert=off
```

DataIN= on   /Datei datain.txt darf gelesen und ausgeführt werden

Das Lesen erfolgt, wenn auch der „Livemodus“ aktiv ist, aller 64 Zyklen (> 6s) per Polling. Der Lesestatus wird in der Kopfzeile auf dem Bildschirm angezeigt. Die Datei datain.txt muss sich gemäß Scriptbefehl „PfadTaste=“ im entsprechenden Ordner befinden und dabei folgenden Konventionen entsprechen:

- In einer Zeile (max. 255 Zeichen) befinden sich fortlaufend, durch Leerzeichen getrennt, ganze Zahlen (Integer) in Textform - mehrere Dreierkombinationen in der Form: <SPS-Modulnummer> <zi-Nummer> <Wert>. Werte können, bei zi-Nummer=6, auch als Gleitkommazahlen (mit Punkt) angegeben werden. Sie werden dabei intern auf 2 Stellen nach dem Komma ausgewertet. Werte werden beim Einlesen auf den Bereich -21474836 … +21474836 begrenzt. Gleitkommazahlen (zi-Nummer=6) kleiner als -1E7 werden auf -1E7 begrenzt (Ausfallkennzeichnung).
- Bis zu 20 SPS-Modulwerte bzw. Dreierkombinationen werden abgeschlossen durch die folgende Kombination: <**0**> <**0**> <Quersumme>
- Die Quersumme ist die Summe aller vorangegangenen Zahlen in Form eines vorzeichenrichtigen Integerwertes unter Abschneidung eventueller Nachkommastellen bei Werten, sie wird beim Einlesen auf den Bereich -2147483647 … +2147483647 begrenzt.
- Als <SPS-Modulnummer> werden Werte im Bereich 60 … 1199 akzeptiert, als <zi-Nummer> sind Ziffern gemäß der Konvention im Abschnitt 4.7.1 im Bereich {0 … 7, 9} zulässig.

Angenommen, die SPS-Module 150 und 151 wurden als Leermodule in der SPS in der Form „z=150 91 Vorgabe_extern %“ vorgegeben. Wenn dann später im laufenden Betrieb ein externes Script/Programm folgende Zeile in die Datei **datain.txt** schreibt:

150 0 1 151 6 99.3 0 0 407

… So wird nach dem Lesen der SPS-Modul 150 auf „autom. EIN“ gesetzt und der SPS-Modul 151 erhält f=99,3 – da die Quersumme (407) korrekt angegeben ist. Anwendungsbeispiel: Übergabe von Zustandsparametern eines externen Systems mit eigenem Controller (z. B. Rollladensteuerung / Wetterstation / Heizungssteuerung / Lichtsteuerung) an den Hauscomputer.

4.3. Befehle zur Verwaltung der Hardware [HAUSBUS]:

Binär - E/A:

- Zuweisung von Namen zu den 8 möglichen E/A Schaltkreisen PCF8574P/ATtiny26 (die Namen werden in den Bildschirm-Menüs verwendet):

```
PCF8574P= [Name IS0]_[Name IS1]_ ... [Name IS7]
           8 x String [10]
```

PCF8574P=MFW P_8574P/1 Haus_I Haus_II P_8574P/4 P_8574P/5 P_8574P/6 P_8574P/7

- Zuweisung von Bezeichnungen und Funktionen zu den maximal 64 möglichen E/A Signalleitungen bzw. Bustasten (die Namen werden u.a. in den Bildschirm-Menüs verwendet). Je nach Hardware - PCF8574 oder ATtiny26 ist der entsprechende Scriptbefehl zu verwenden.

```
Binär=   [Bezeichnung]_[Adresse]_[Kanal]_[Maske]_{[Tastaturkode]}
Binär26= [Bezeichnung]_[Adresse]_[Kanal]_[Maske]_{[Tastaturkode]}_{Startwert}
           String[13]      0-7      0-7    0/1/3        0..255         0/1

   Maske:   0 - nur Eingang, kein Ausgang
            1 - Ein- und Ausgabe (low-Pegel am I/O Pin = ein)
            3 - inverse Ein- und Ausgabe (high-Pegel am I/O Pin = ein)   (ab Version 6.0.40)
   Tastaturkode:
      <Kursor hoch> -  72          <Kursor runter>-  80     <Escape>       -    27
      <Leertaste>   -  32          < i >          - 105
      < o >         - 111          < + >          -  43
      < - >         -  45          < g >          - 103
      < 0 >...< 9 > -  48..57      <Enter>        -  13
      < * >         -  42          < # >          -  35
      <Tabulator>   -   9          < r >          - 114
      < s >         - 115          < p >          - 112
      < k >         - 107          keine Reaktion -   0

   Startwert:    0 - Ausgang bei Ausfall „aus"
            1 - Ausgang bei Ausfall „ein"
```

Binär=Blumenlicht 3 5 1
Binär=Lüfter_I 3 6 1
Binär=Lüfter_II 3 7 1
Binär=runter 0 5 0 80 /Signal löst gleiche Reaktion wie „Kursor runter“ aus
Binär=Leertaste 0 6 0 32 /Signal löst gleiche Reaktion wie „Leertaste“ aus
Binär26=Adr1_K5 1 5 1 0 1 /Adresse 1 Kanal 5 eines binären ATtiny26 Interfaces heißt „Adr1_K5“ und kann als Ein- und Ausgang eingesetzt werden. Dieser Kanal schaltet nach Ansprechen der ATtiny-internen Watchdog „ein“

Über den Tastaturkode kann die entsprechende Signalleitung logisch mit einer Taste auf der PC Tastatur gekoppelt werden. Damit ist es möglich, mittels „Hardware-“ Tasten (sogenannte Bustasten), abgesetzt vom PC das Programm zu bedienen (Menüführung). Sinnvollerweise sollte der dabei verwendete PCF 8574P auf die Adresse „0“ gelegt werden, um eine schnellere Reaktion auf einen Tastendruck zu erreichen. Theoretisch denkbar ist sogar eine Parallelschaltung mehrerer ICs mit der Adresse „0“ mit gleicher Tastenbelegung in verschiedenen Räumen per Bus oder Funk (über entsprechende Adapter).

Die Kopplung mit der Leertaste ersetzt dabei vollständig die bisher verwendete „OK“ Taste. Wegen der Abwärts-Kompatibilität des Programms muss die „OK“ Taste aber weiterhin in der *hauscomp.ini* („TasteOK=“) angegeben werden.

Die Angabe eines Startwertes führt beim binären ATtiny26 zur Programmierung eines im EEPROM hinterlegten Wertes. Der betroffene Kanal schaltet dann bei Ausfall des PCs oder des Hausbusses entsprechend. Diese Funktion ist erst ab Firmware Version I2C_B2.0 gegeben. Bis dahin ist die Angabe des Startwertes wirkungslos. Eine Übersicht der programmierten Werte kann unter „Funktionen/Festlegungen“ abgerufen werden.

Alle verwendbaren Tastaturkodes sind unter 4.7.9 gelistet.

Analog/Digital Wandler (Sensoren, Eingänge), Digital/Analog Wandler (Ausgang):

- Zuweisung von Namen zu maximal 8 analogen Schaltkreisen PCF8591 bzw. LM75/ DS1631 (die Namen werden in den Menüs verwendet):

```
PCF8591= [Name IS0]_[Name IS1]_ ... [Name IS7]
           8 x String [10]..
```

PCF8591=Test Nordgiebel Garage Heizung Kom.Zentrum Fassade PCF8591/6 PCF8591/7

Genau genommen ist die Syntax nicht ganz korrekt, denn der Befehl bezieht sich mittlerweile nicht nur auf PCF8591, sondern auch auf die I2C Bussensoren. Wegen der Abwärts-Kompatibilität muss die Syntax allerdings so beibehalten werden.

- Zuweisung von Bezeichnungen und Skalierungsdaten zu den maximal 32 analogen Eingängen der PCF 8591 (die Namen werden in den Menüs verwendet):

```
Sensor=[Bezeichnung]_[Einheit]_[Adresse]_[Kanal]_[Z.DIFF]_[Y0]_[X0]_[Y1]_[X1]_{[Y2]}_{[X2]}
       String [20]  String[5]    0-7         0-3     Real    Y0, Y1, Y2: physik. Wert (real)
                                 X0, X1, X2: Kodewerte (0-255)
```

Durch Vorgabe von Wertepaaren wird vom Programm die Übertragungskennlinie errechnet. Die Interpolation erfolgt intern nach Newton mit einer Approximation 2. Ordnung. Wird eine lineare Kennlinie mit zwei Wertepaaren gefordert (z. B. Spannungsmessung), so ist als drittes Wertepaar das zweite nochmal einzugeben. Die Zeitdifferenz z.diff dient der Vorgabe eines sinnvollen Intervalls zur automatischen Berechnung von Wertänderungen am analogen Eingang. Beim Erreichen des maximalen (Kodewert=255) oder minimalen Analog-Wertes (zugehöriger Kodewert=0) wird vom Programm der entsprechende Sensor als ausgefallen markiert (Parallele zum Industriestandard 4-20mA).

Bei z.diff=0 wird die automatische Sensorüberwachung abgeschaltet. Somit kann auch eine Ausgangsspannung des Sensors von 0 V (zugehöriger Kodewert=0) bearbeitet werden. Sinnvoll ist das bei der Nutzung eines A/D Einganges als zusätzlichen Binär- Eingang zur Zustandsüberwachung von Schaltsignalen. Voraussichtlich langsame Änderungen am Sensoreingang, wie z. B. Temperaturmessungen sollten mit einem großen z.diff (z. B. 90s) versehen werden. Die Analog-Adresse darf nicht bereits im Befehl „Bussensor=“ verwendet werden. Ein interner digitaler Sprungfilter sortiert fehlerhafte Daten aus. Bei der Angabe von „0“ oder „1“ für Z.Diff wird der interne digitale Sprungfilter abgeschaltet.

```
Sensor=Kellertemp.  °C  2 2  90  -24.0  38    4.0  87   20.6  120
Sensor=Kühltruhe    °C  2 3  90  -24.0  36    4.0  86   45.0  159
Sensor=Zirkulation  °C  5 0  90  -24.0  33    4.2  84   44.4  155
```

- Zuweisung von Bezeichnungen und Skalierungsdaten zu den maximal 32 analogen Eingängen der PCF 8591, Kalibrierung mit einem Wertepaar:

```
Sensor2=[Bezeichnung]_[Einheit]_[Adresse]_[Kanal]_[Z.DIFF]_[Y0]_[X0]_[Anstieg]
        String [20]  String[5]   0-7          0-3     Real   Y0: physik. Wert (real)
                                                        X0: Kodewerte (0-255)
```

Im Unterschied zum Befehl „Sensor=“ wird nur ein Wertepaar benötigt, dafür ist aber der Anstieg der Übertragungskennlinie in [Einheit/Kodewert] anzugeben. Verwendung bei Sensoren, deren Kennlinienanstieg bekannt ist – Kalibrierung bei nur einer Vergleichsprüfung möglich.

```
Sensor2= Luftdruck kPa  1  0  900  100 105  -0.0477    /bei 100kPa, bzw. 2,05V am A/D Wandler ergeben sich 105 Kodewerte
```

- Zuweisung der Wassermenge je Niederschlagsimpuls am Außensensor (Regensensor) in Liter je m²:

```
Regenloeffel= [Wert] Standardwert: 0.353
                Real
```

Wird für die automatische Berechnung der Niederschlagsmengen benötigt. Er kann z.B. wie folgt ermittelt werden: z.B. 0,5 l Wasser langsam in den Regensensor gießen und die Impulse zählen. $Wert = \frac{0{,}5\,l}{Impulse \times Auffangfläche\,[m^2]}$ [l/m²]

```
Regenloeffel= 0.0456          /kalibrierte Wertvorgabe für einen modifizierten Regensensor
```

- Zuweisung von Bezeichnungen und Skalierungsdaten zu den maximal 16 Multisensoren/Raumsensoren/Wärmemengenzähler

```
Raumsensor=      [Subadresse]_[Bezeichnung]__{[dT]}_{[Feuchte:Soll]}_{[Feuchte:Sensor]}
Raumsensor2000=  [Subadresse]_[Bezeichnung]__{[dT]}_{[Feuchte:Soll]}_{[Feuchte:Sensor]}
                   0..15      String [20]  -5.0..+5.0    0..65.0             0..65.0
RaumsensorWMM=   [Subadresse]_[Bezeichnung]___[dT1]___[Durchflusskonstante]
                   0..15      String [20]  -5.0..+5.0    0..32768
```

Die Bezeichnung ist ein aussagekräftiger Name für den Multisensor/Raumsensor, wie z. B. „RS_Flur“. Der Wert „dT“ dient zur Kalibrierung der Temperaturmessung: Die angezeigte Temperatur = gemessene Temperatur + dT.

Die beiden Angaben „Feuchte:..“ dienen zur Korrektur der relativen Luftfeuchte für Raumsensoren. Hier kann ein Wertepaar von einer Vergleichsmessung eingetragen werden. Diese Messung sollte möglichst bei rF< 50 % erfolgen. Das, für die spätere automatische Korrektur, notwendige zweite Wertepaar, wird mit 100 %/100 % angenommen. „Raumsensor2000=“ dient der Abfrage von CO2-Sensoren mit max. 2000 ppm, während „Raumsensor=“ CO2-Sensoren mit 5000 ppm voraussetzt (Standardtyp).

„RaumsensorWMM“ ist Wärmemengenmessern mit spezieller Firmware vorbehalten (WMMx.x) – Dabei werden zwei Temperaturen von SHT3x Sensoren, sowie die Impulse eines Durchflusssensors übertragen (ab HC Version 6.0.60B7). Die Durchflusskonstante ist für die Berechnung der Wärmeleistung notwendig, sie beschreibt die Zahl der Impulse je Liter.

```
Raumsensor=  1  RS_Flur    0.0     /Der Multisensor mit der Subadresse 1 heißt „RS_Flur“. Die Temperatur wird nicht korrigiert.
Raumsensor=  1  RS_Flur    0.0     33.5   43.2   /wie oben, zusätzliche Feuchte-Kalibrierung mit MgCl2 bei 33,5%
RaumsensorWMM=  12  WMM_Hzg  -0.1  396    /Wärmemengenmessung, 1. Temperatur-Sensor wird um 0,1°C nach unten korrigiert,
                                            Impulszähler mit 396 Impulsen je l
```

Hinweis: Wird im System eine Initialisierung mit dem Scriptbefehl „SensorMax1236..=“ erkannt, so wird der Raumsensor mit der entsprechend (gleichen) I^2C-dresse nicht abgefragt.

- Treiberanweisung (ab Version 6.0.51): analoge Adressen 0 und 1, sowie alle Raumsensoren doppelt so schnell abfragen, die analogen Adressen 2..7 entsprechend nur halb so schnell. Die Anweisung ist gültig, wenn der angegebene SPS-Modul EIN ist. So ist auch eine dynamische Treiber-Steuerung zur Laufzeit möglich. Siehe auch Abschnitt 1.3.3.

```
Vorrang01=     [SPS-Modul]       Standardwert= 0
              -1199..1199
Vorrang01=     1          /Der Treiber wird angewiesen immer die analogen Adressen 0 und 1 bevorzugt auszulesen
Vorrang01=     -40        /Der Treiber wird angewiesen die analogen Adressen 0 und 1 bevorzugt auszulesen, wenn kein Tag ist
```

- Zuweisung von Bezeichnungen und Skalierungsdaten zu den 4 analogen Eingängen des MAX1236 (12Bit A/D Wandler):

```
SensorMax1236K=[Bezeichnung]_[Einheit]_[Adresse]_[Kanal]_[Z.DIFF]_[Y0]_[X0]_[Y1]_[X1]_[Y2]_[X2]
SensorMax1236L=
SensorMAX1236=
SensorMAX1236M=
               String[20] String[5]   0-7      0-3    Real    Y0-2: physikalischer. Wert (real)
                                                               X0-2: zugehöriger „Kodewert“
```

In den Fällen, wo es auf eine hohe Genauigkeit der Messung von analogen Signalen ankommt, ist der Max1236 eine Alternative zum PCF8591, der nur 8Bit Wandlerauflösung besitzt. Da der Max1236xxx eine feste Slaveadresse besitzt, kann nur ein Schaltkreis an den Bus angeschlossen werden. Es gibt aber 4 verschiedene Bauformen mit unterschiedlichen Slave Adressen (Kennbuchstaben K, L, -, M), sodass maximal 16 analoge Kanäle mit 12Bit Wandlung möglich sind. Jeder der 4 möglichen A/D Kanäle eines MAX1236 wird mit jeweils einer Scriptzeile initialisiert. Der MAX1236 wird vom Programm folgendermaßen initialisiert: Single Ended Mode, internal clock, unipolar Input, Internal Reference. Die, nur in diesem Programm verwendete Adresse (Anzeigen, SPS usw.), kann in Abhängigkeit der restlichen Busbausteine frei vergeben werden, die Kanäle entsprechen den Input Pins AIN0-AIN3. Ansonsten entspricht die Programmierung der Kennlinien dem Script Befehl „Sensor=“.

Hier eine reale Einbindung eines MAX1236EUA:

```
SensorMAX1236=Test1  V  2   0   30    0   0  4.096 4096      /Kanal AIN0  0..+4,096V
SensorMAX1236=Test2  V  2   1   30    0   0  5.000  771      /Kanal AIN1  0..+26,8V (über Spannungsteiler 1,2M/220k am Eingang)
SensorMAX1236=Test3  V  2   2   30    0   0  4.096 4096      /Kanal AIN2  0..+4,096V
SensorMAX1236=Test4  V  2   3   30    0   0  4.096 4096      /Kanal AIN3  0..+4,096V
```

- Zuweisung von Bezeichnung und Skalierungsdaten zu dem analogen Ausgang der maximal möglichen 8 möglichen PCF8591 oder auch ATtiny26 mit Software „I2C_A1.0 ff.“:

```
Analog=[Bezeichnung]_[Einheit]_[Adresse]_[Z.DIFF]_[Startwert]_[Y0]_[X0]_[Y1]_[X1]_{[Y2]}_{[X2]}
         String [20]  String[5]   0-7      Real       Real       Y0, Y1, Y2: physik. Wert (real)
                                                                 X0, X1, X2: Kodewerte (0-255)
```

Der Startwert ist der physikalische (skalierte) Wert, auf den der entsprechende D/A Ausgang nach Programmstart gesetzt wird. Im Normalfall wird man ihn auf 0 setzen. Die Angabe ist notwendig, um undefinierte Werte nach Programmstart zu verhindern. Im weiteren Programmverlauf wird der Ausgang von der SPS gesteuert (SPS-Modultypen 76, 32, 33, 89). Bei analogen Interfaces mit ATtiny26 wird ab Firmware Version „I2C_A2.0“ der Startwert zusätzlich an das Interface übertragen und im EEPROM abgelegt, und zwar sowohl einmalig nach Programmstart als auch regelmäßig zum Tageswechsel. Eine Übersicht der programmierten Werte kann unter „Funktionen/Festlegungen“ abgerufen werden. Sinnvoll ist diese Funktion z. B. für die Vorgabe einer Pumpendrehzahl bei Ausfall des steuernden PCs.

Die Zeitdifferenz z.diff dient der Vorgabe eines sinnvollen Intervalls [s] zur automatischen Berechnung von Wertänderungen am analogen Ausgang. Lineare Ausgänge benötigen natürlich nur zwei Wertepaare. Als drittes Wertepaar wird in einem solchen Fall das zweite nochmal geschrieben. Ist der analoge Ausgang nicht linear, so kann dies ab Version 6.0.37 durch alle 3 Wertepaare entsprechend angegeben werden. Die Interpolation erfolgt intern nach Newton mit einer Approximation 2. Ordnung für den SPS-Modultyp 76. Für alle anderen SPS-Modultypen (bei denen der D/A Wert direkt gesetzt wird) wird nur linear interpoliert. Die interpolierten Werte werden in den Anzeigen und in der Grafik verwendet. Bei nicht linearen Ausgängen unter Verwendung der SPS-Modultypen 32, 33 und 89 kann es deshalb zu Abweichungen in den Anzeigen kommen.

```
Analog=Zirkulation  E/A  3    120   0     0   0    1   255   1  255  /analoger Ausgang Adresse 3
Analog=Adresse_5    %    5     30   75    0   0   50   131 100 191  /ATtiny26 an Adresse5
```

Je nachdem, welche analogen Sensoren zu diesem analogen Ausgang gehören, wird der zugehörige programminterne Treiber automatisch ausgewählt. Wird z. B. ein Sensor über den Scriptbefehl „SensorAt26=..“ der gleichen Adresse zugewiesen, so erfolgt die analoge Ausgabe über den ATtiny26. Andernfalls wird der entsprechende analoge Kanal des PCF8591 angesprochen. Der Bereichsumfang des analogen Ausgangs am PCF8591beträgt 0..255 Kodewerte und der Umfang am ATtiny26 0..191 Kodewerte.

- Zuweisung von Bezeichnungen und Skalierungsdaten zu den maximal 32 analogen Eingängen des Controllers ATtiny26 (8 Controller mit je 4 analogen Kanälen) mit Firmware „I2C_Ax.x“:

```
SensorAt26=[Bezeichnung]_[Einheit]_[Adresse]_[Kanal]_[Z.DIFF]_[Y0]_[X0]_[Y1]_[X1]_{[Y2]}_{[X2]}
           String [20]  String[5]    0-7        0-3    Real Y0, Y1, Y2: physik. Wert (real)
                                                          X0, X1, X2: Kodewerte (0-1024)
```

Dieser Scriptbefehl entspricht weitgehend dem Befehl „Sensor=“, Programm intern wird lediglich ein anderer Treiber für den Datenaustausch auf dem Bus verwendet. Die A/D Wandlung erfolgt mit 10 Bit und die Ausfallerkennung ist an die übertragende Prüfsumme gebunden, sodass alle Kodewerte genutzt werden können. Bei der Angabe von „1“ für Z.Diff wird nicht nur der Sprungfilter abgeschaltet, sondern zusätzlich noch eine Ausfallerkennung im Bereich <5 und >1020 aktiviert → sinnvoll beim Anschluss von Widerstandsketten für Schaltkontakte. Bei der Angabe von „2“ für Z.Diff wird eine spezielle Kalibrierkurve für einen Stromsensor aktiviert. Die folgenden Beispiele illustrieren das Verhalten für verschiedene Sensoren (hier als Beispiel nur für Adresse 1, Kanal 2):

```
SensorAt26= Spannung  V   1   2   0   2.51   1023  /Ausgabe einer gemessenen Spannung 0..2,51V, keine Extremwertüberwachung, kein
   Sprungfilter
SensorAt26= Temperatur °C 1   2  60  +22.6   477   +58.1   579   +101.1   722  /Temperatursensor angeschlossen, Ausfallüberwachung,
   Sprungfilter aktiv
SensorAt26= Strom       A   1   2   2   +0.0      0   +2.51   1023  /Ausgabe des Stromes (mit besonderer Kennlinie), obwohl mit „+2.51“ die
   Referenzspannung am Pin 17 des ATtiny26/261 angegeben wird!
SensorAt26 = GruppeA Bit   1   2   1    +42    106  +546  688  +991   969  /Widerstandskette an Adr. 1 Kanal 2 (3k3), Ausfallerkennung <5
   bzw. >1020 Kodewerte aktiv, Sprungfilter abgeschaltet
```

Die besonderen Kalibrierwerte für Z.Diff= „2“ sind:

Spannungsbereich U [V]	Formel für Strom I [A]
< 0,001	I= 0
> 0,001 … 0,105	I= 0,054 + 3,037 * U – 4,2 * U²
> 0,105 … 0,186	I= 0,713 – 9,205 * U + 52,8 * U²
> 0,186 … 0,413	I= –0,413 + 5,93 * U + 17,6 * U²
> 0,413 … 2,5	I= –3,95 + 20,7 * U – 0,11 * U²

Besonderheit: [Y1] ist kein Stromwert, sondern die Referenzspannung am Pin 17 am ATtiny26/261. Das Wertepaar [Y2] / [X2] wird nicht verwendet.

- Festlegung der Nordrichtung eines Windrichtungssensors am Raumsensor-Subadresse 15 (Wetterstation). Der interne Treiber übergibt dann später den Rohdatenwert abzüglich des angegebenen Nordwertes korrekt an die SPS.

```
WindNord=   [Winkel Nord]                  Standardwert: 0
            ganzahlig im Bereich 0 … 359
```
WindNord= 38 /Die Berechnung der Windrichtung (aus der der Wind weht) wird um 38° verringert bez. der gemessenen angezeigt

- Zuweisung von Kode-Bereichen zu Winkeln eines (analogen) Windrichtungssensors am Raumsensor-Subadresse 15 (Wetterstation). Die Kodegrenzen müssen in aufsteigender Folge angegeben sein, damit der Treiber in Echtzeit die richtigen Winkel zuordnen kann. Maximal 17 Werte für Kode und 16 Werte für Winkel können angegeben werden. Mit der Angabe von mindestens 2 Kodegrenzen wird zugleich der Treiber für die Berechnung der Windrichtung am analogen Eingang des Raumsensors 15 aktiviert und die Berechnung von Helligkeit bzw. Sollwert entsprechend deaktiviert. Die Werte sind unter den (internen) SPS-Modulen 42ff. /3/4 gelistet.

```
Windkode= [Kodegrenze 1]_[Kodegrenze 2]_ … _[Kodegrenze 17]
          alle Werte: ganzahlig im Bereich 0 … 255
Windgrd= [Winkel 1]_[Winkel 2]_ … _[Winkel 16]
           alle Werte: ganzahlig im Bereich 0 … 359 (Sensor mit Wertzuordnungen)
           erster Wert:   +360 – Windrichtungssensor mit linearer Spannungsausgabe
           erster Wert:   -360 –         -"-             , umgekehrte Drehrichtung
```

Im folgenden Beispiel wird einem Windrichtungssensor mit 8 Reedkontakten dem Kodebereich 37 … 43 ein Winkel von 68° zugeordnet. Kodewerte kleiner 28 und größer 251 werden als Ausfall interpretiert. Somit sind bis zu 16 Windrichtungen anzeigbar.

Windkode= 28 37 43 52 70 92 109 133 156 178 195 204 217 227 234 243 251 /Kodebereiche der Reedkontakte im Windrichtungssensor
Windgrd= 113 68 90 158 135 203 180 23 45 248 225 338 0 293 315 270 /zugeordnete Winkelwerte Windrichtungssensor

Ein Windrichtungssensor mit bereits normalisierter Spannungsausgabe wird wie folgt initialisiert:

Windkode= 0 154 /Kodebereich für analogen Windrichtungssensor mit Spannungsausgabe 0°/0V-Kode=0; 360°/3V-Kode=154
Windgrd= 360 /Kennzeichnung Windrichtungssensor mit Spannungsausgabe

Die Berechnung des Windwinkels erfolgt bei dieser Art Windrichtungs-Sensor (0…3V) auf 2,3° genau.

Ein digitaler Windrichtungssensors (AS560x) wird vom Treiber automatisch erkannt, die Angabe von „Windkode“ und „Windgrd“ entfällt. Einzige Besonderheit: Ist eine Spiegelung der Windrichtung in Ost-Westrichtung (umgekehrte Drehrichtung) notwendig, so ist die Angabe…

Windgrd= -360 /digitaler oder analoger (Spannung) Windrichtungssensor mit umgekehrter Drehrichtung

…erforderlich. Die Berechnung des Windwinkels erfolgt beim digitalen Windrichtungssensor auf 1,4° genau.

- Kalibrierung eines Anemometers (Windgeschwindigkeitsmesser) am Raumsensor-Subadresse 15 (Wetterstation). Dieser Wert wird für die interne Umrechnung der gemessenen Frequenz in die Windgeschwindigkeit benötigt.

```
Anemometer= [Frequenz (Hz)]     [Geschwindigkeit (km/h)]      Standardwerte: 21.2      100.0
                 Real                  Real
```

Anemometer= 37.67 85.3 /Der Windmesser erzeugt bei 85,3km/h 37,67 Hz (bzw. 226 Impulse in 6s)

- Zuweisung von Namen und Skalierungsdaten zu den maximal 8 möglichen I2C Temperatursensoren (LM75 und/ oder DS1631):

```
Bussensor=[Bezeichnung]_[Einheit]_[Adresse]_[Typ ]_[Z.DIFF]_[Korrekturwert]
          String [20]    String[5]   0-7    2-LM75   Real      Real
                                            3-DS1631
```

Die Temperatursensoren LM75 (9bit) und DS1631 (12bit) können mit diesem Befehl in das Programm eingebunden werden. Außer der Bezeichnung und der Einheit muss die Adresse (0-7) und der Typ angegeben werden. Die Adresse darf nicht bereits von einem PCF8591 verwendet werden. Die Zeitdifferenz z.diff dient der Vorgabe eines sinnvollen Intervalls [s] zur automatischen Berechnung von Trend Werten. Der Korrekturwert verschiebt die gemessene Temperatur um den entsprechenden Wert. Damit ist eine Kalibrierung bei Bedarf möglich. Bei der Verwendung des DS1631 ist zu beachten, dass der Bustakt wegen der umfangreicheren I^2C Protokolle automatisch für alle angeschlossenen ICs verdoppelt wird. Die maximal mögliche Busreichweite wird damit halbiert. Der DS1631 benötigt außerdem eine Initialisierung nach Spannungsabfall am Sensor. Sie erfolgt automatisch bei fehlerhaften Daten im Wechsel mit einem Auswertezyklus - bis der Sensor wieder läuft. Zusätzlich wird die zyklische Abfrage des Sensors auf minimal 750 ms begrenzt.

Bussensor=Teewasser °C 1 3 90 -0.25 /DS1631 an analoger Adresse 1; Temp. wird um 0,25°C nach unten korrigiert

- Zuweisung von Parametern zu den maximal 4 möglichen S0-Zählern/Energiezählern am Busadapter „Deluxe“:

```
S0x=       [Zähldifferenz]_[Teiler]_[Einheit (Zähler)]_[Einheit (Leistung/Verbrauch)]
 x:0..3          Real      Integer        String                  String
```

Dieser Befehl konfiguriert die Auswertung der vier S0-Zähler (Busadapter „Deluxe“) im Programm. Beim Auslesen der Zähler aus dem ATtiny des Busadapters wird automatisch auch jeweils die angegebene Zähldifferenz hinzuaddiert und das Ergebnis in den internen SPS-Modulen 46..49 bereitgestellt. Die Zähldifferenz wird bei Bedarf einmalig „händisch“ folgendermaßen ermittelt:

1. Programm zunächst mit Zähldifferenz „0“ starten.
2. Zählerstand des ATtiny auslesen (SPS-Module 46 … 49)
3. tatsächliche korrekte Zähldifferenz ermitteln: Zähldifferenz = Zählerstand des Energiezählers – Zählerstand des Attinys
4. ermittelte Zähldifferenz in *hauscomp.ini* eintragen. Nach einem Neustart sollte der angezeigte Zählerstand des Hauscomputers mit dem realen Zählers übereinstimmen.

Der Teiler ist vom Energiezähler abhängig – z. B. bei 1000 Impulsen je kWh beträgt er 1000. Die beiden Einheiten werden direkt für die Anzeige übernommen. Die Leistung bzw. der Verbrauch wird automatisch aller 9 min ermittelt und in den Modulen 50..53 bereitgestellt.

```
S00=   0        400     kWh     kW     /Energiezähler
S01=   16850    1000    m³      m³/h   /Wasserzähler
S02=   0        1000    l       l/h    /Durchflusszähler
S03=   0        1000    kWh     kW     /Energiezähler
```

- Zuweisung von Bezeichnung und Skalierungsdaten zu dem Frequenzmesseingang K3U des ATtiny26 mit Software „I2C_A1.0“ (die Bezeichnung wird u.a. im Menü verwendet):

```
Frequenz26=[Bezeichnung]_[Einheit]_[Adresse]_[Z.DIFF]_[Y0]_[X0]_[Y1]_[X1]_{[Y2]}_{[X2]}
           String [20]  String[5]    0-7      Real     Y0, Y1, Y2: physik. Wert (real)
                                                       X0, X1, X2: Kodewerte (0-65536)
```

Bei der Frequenzmessung wird regelmäßig nach mindestens 10 s der interne Ereigniszähler des Kanals K3U im ATtiny26 ausgelesen und nach der Formel $f = \frac{Zähler_{Ende} - Zähler_{Anfang}}{10 * Delta\,t} * a1 + a0$ berechnet, wobei a0 und a1 die Skalierung-Koeffizienten sind. Der Faktor 10 ist wegen der internen Datenformate notwendig und muss in der Skalierung berücksichtigt werden. Es sind damit maximal 4 Frequenzkanäle am Bus möglich. Der 16 Bit Ereigniszähler hat einen Zählumfang von 65536 und kann daher bei einer Abtastrate von 10 s max. 6,55 kHz bei einer Auflösung von 0,1 Hz messen. Da die Abfrage nicht mit den regelmäßigen Busprotokollen synchronisiert ist, kann die Zeitdifferenz im Bereich 10..20 s liegen. Die maximale messbare Frequenz reduziert sich daher auf ca. 3 kHz. Die Genauigkeit der Messung ist ausschließlich vom Betriebssystem abhängig (siehe Pkt. 1.9.). Im Menü „A/D Wandler“ wird der Stand des Ereigniszählers angezeigt. Folgende Scriptzeile organisiert eine reine Frequenzmessung, die für Anzeigen oder in der SPS über die übliche analoge Adresse und Kanal weiter verwertbar ist:

```
Frequenz26=test  Hz  1   30   0    0  +100  1000 +200   2000 /Frequenzzähler an analoger Adresse 1, Kanal 3
```

- Zuweisung von Parametern für die spezielle Heizungs-Log Datei. Mehr Informationen dazu im Beispiel 6.9.

```
HzgLogS01=  [S01] [S02] [S03] [S04] [S05] [S06] [S07] [S08] [S09] [S10] [S11] [S12]
HzgLogS13=  [S13] [S14] [S15] [S16] [S17] [S18] [S19] [S20] [S21] [S22] [S23] [S24]
           alle Parameter sind Bezug auf ein SPS Modul: 0..1199
HzgLogA01=  [A01] [A02] [A03] [A04] [A05] [A06] [A07] [A08] [A09] [A10] [A11]
HzgLogA12=  [A12] [A13] [A14] [A15] [A16] [A17] [A18] [A19] [A20] [A21] [A22]
           alle Parameter sind Bezug auf ein SPS Modul: -1199..1199
```

Die Parameter S01 bis S24, sowie A01 bis A22 legen fest, von welchen SPS Modulen Werte in das Log übertragen werden. Sxx bezieht sich dabei immer auf den analogen Ausgangswert des Moduls. Axx kann dagegen sowohl analoge Ausgangswerte als auch den Schaltzustand bezeichnen. Kriterien für diese automatische Zuweisung sind die Einheit und der Modultyp. Bei einer Einheit von „-“ oder keiner Einheit des SPS-Moduls bzw. bei Modultypen 30, 31 und 33 wird der Schaltzustand abgespeichert, genauso, wenn der Parameter mit einem Minus geschrieben wird. Im Log bedeuten aus: 0; ein: 100. Bei allen anderen Angaben wird der analoge SPS-Wert abgespeichert. Eine Übersicht der gespeicherten Parameter wird auf der Menüseite 145 (System.., HzgLog..) dargestellt.

4.4. Befehle zur Programmierung eigener Bildschirmanzeigen [Anzeigen]:

4.4.1. Darstellen von (beliebigen) Text auf (beliebiger) Bildschirm-Seite unter (beliebigen) SPS-Bedingungen:

```
text=      [Seite]_[  X  ]_[  Y  ]_[Farbe]_[SPS-Modul]_[  Text   ]
textgross= [Seite]_[  X  ]_[  Y  ]_[Farbe]_[SPS-Modul]_[  Text   ]
            0..255  1..160  0..352  0..15   -1199..1199   String[30]
```

„text=“ ist der Scriptbefehl für Text in normaler Größe, „textgross=“ für Text in doppelter Größe. Die Seite bezeichnet die Bildschirmseite, auf der der Text ausgegeben werden soll, X ist die Position in horizontaler Richtung (1..119) mit folgender Besonderheit: X=1 – linke Spalte, X=2 – rechte Spalte. Y ist die vertikale Position, wobei „0“ oben und „199“ („352“ bei VGA=on) unten ist. Zur Vereinfachung der Programmierung werden vom Programm Zahlenwerte von 0 – 19 als Bildschirmzeile und Werte von 20 – 199 als Pixel interpretiert. Der Parameter „SPS-Modul“ verweist auf den Zustand eines Softwaremoduls in der SPS. Der Text erscheint nur, wenn das entsprechende Modul „EIN“ ist. Wird als Modul eine „1“ vorgegeben, so erscheint der Text immer auf der entsprechenden Seite. Wird als Farbe 12 gesetzt (rot, blinkend), so wird mit dem Sichtbarwerden auf dem Bildschirm ein entsprechender Hinweis im Laufband/Textfeld gesetzt.

Einschränkungen:

- Es sind maximal 255 Einträge vom Typ „text=“ bzw. „textgross=“ möglich. Der Farbwert entspricht den Angaben unter 1.4 „Typografie“. Wird ein Text mit der Farbe 12 (rot blinken) versehen, so wird sein erstmaliges Auftreten im Laufband registriert und das Signal „HINWEIS!“ gesetzt.

text= 96 1 10 15 1 Maria:0173654321 /schreibt immer „Maria:0173654321“ ganz unten auf die automatische Kalenderseite
textgross= 95 2 9 12 229 EINBRUCH! / „EINBRUCH!“ blinkt rot rechts unten in doppelter Größe auf der automatischen Seite 95, wenn das Modul 229 EIN ist

4.4.2. Gestaltung von eigenen Bildschirmmenüs (ausschließlich reserviert für Bildschirmseiten 110 – 119 sowie 200 – 254)

- Zuweisung von Menünamen zu den selbst gestalteten Bildschirm-Menüs:

```
Menü11x= [Menü]_[Benennung]
         110..119 String[20]
     bzw. 200..254
```

Menü11x= 110 Heizung

4.4.3. Programmierung von Bildschirm- „Tasten“ in den selbst gestalteten Menüs (Softkey): (Erweiterung ab 6.0.38)

```
Zeile= oder
Taste= [Menü]___[Zeile]___[Art]___[Modul]
      110..119   0..9     0..15  51..1199
      200..254
```

Mit diesem Befehl kann man durch Betätigung einer Ziffer in dem entsprechenden Menü den Zustand von Modulen in der SPS verändern bzw. modifizieren und Untermenüs erreichen. Anwendungen sind z. B. die Sollwertvorgabe für eine Heizung oder das zu/ab/umschalten von Steuerungsabläufen.

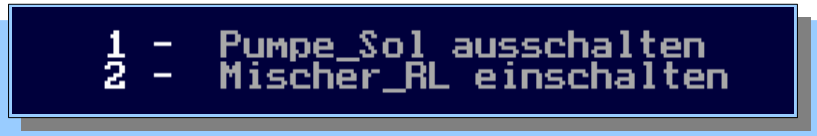
Die Zeile steht für die auslösende Ziffer (entspricht zugleich der Zeile auf der entsprechenden Menüseite) und Modul bezeichnet das Softwaremodul in der SPS, das geschaltet wird bzw. die an zuspringende Untermenüseite. Unter Art wird der Typ der Taste vorgegeben:

```
 0- EIN-AUS-EIN..   (Zustand im SPS Modul umschalten; geht, wenn Modul im Zustand „HAND“ oder „Auto“ ist)
 1- EIN-AUTO-AUS-EIN..   (Zustand im SPS Modul umschalten – die Standardtaste schlechthin!)
 2- AUS                  (Zustand im SPS Modul ausschalten + „HAND“ einschalten)
 3- EIN                  (Zustand im SPS Modul einschalten + „HAND“ einschalten)
 4- Verriegelung auslösen (nur in Verbindung mit Tastensätzen/ SPS Modultyp 90)
 5- um tasten (im autom. Betrieb für einen Zyklus Zustand negieren – z. B. zum starten von Monoflops)
 6- +/-Schrittweite      (interne Verwendung für Tastengruppe siehe 4.4.4.)
 7- Standardwert setzen  (interne Verwendung für Tastengruppe siehe 4.4.4.)
 9- weiter verzweigen in ein Untermenü (Modul bezeichnet in diesem Fall die Untermenüseite)
 10- Reset der Betriebsstunden des SPS Moduls (entspricht Taste „Entf“ im SPS Menü)
 11- Zeitmenü 1          (interne Verwendung Tastengruppe siehe 4.4.4.)
 12- Zeitmenü 2, Bereich (interne Verwendung Tastengruppe siehe 4.4.4.)
 13- Zeitmenü 3, Taste 8 (interne Verwendung Tastengruppe siehe 4.4.4.)
 14- Zeitmenü 4, Taste 9 (interne Verwendung Tastengruppe siehe 4.4.4.)
 15- weiter schalten in ein anderes Menü, Rücksprung wird nicht verändert (Modul bezeichnet die Untermenüseite)
```

Damit Veränderungen an SPS Modulen auch nach Stromausfall erhalten bleiben, wird 18 Sekunden nach der letzten Betätigung von Tasten des Typs 0, 1, 4, 6, 7, 10…14 ein Speicher-Event ausgelöst. In den Dateien „gesternx.daz“ werden dann die in ihrem Zustand veränderten SPS-Module vom Typ 6, 10, 91 und die im Wert veränderten Module der SPS vom Typ 5, 38, 56…58 abgespeichert. Nach Programm-Neustart werden die (veränderten) Betriebsarten und Variablen entsprechend wiederhergestellt.

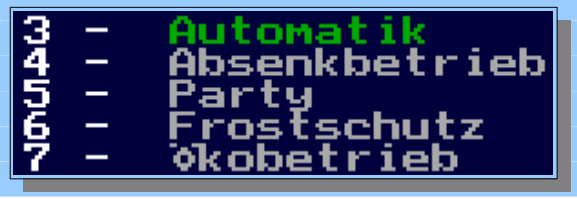


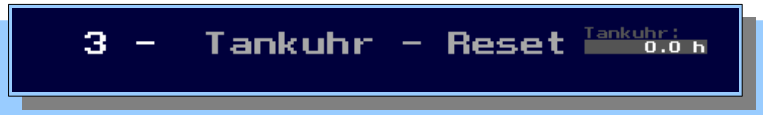
Taste= 114 1 1 205 /Pumpe_Sol Betriebsart: schaltet SPS-Modul 205 bei Betätigung von Taste „1“ auf der Anwendungsseite 114 zyklisch um: EIN-AUTO-AUS-EIN



Taste= 115 1 2 265 /schaltet SPS-Modul 265 (Pumpe_Sol) bei Betätigung von Taste „1“ aus
Taste= 115 2 3 318 /schaltet SPS-Modul 318 (Mischer_RL) bei Betätigung von Taste „2“ ein

Taste= 114 3 4 181 /erzeugt auf Seite 114 5 voneinander abhängige Tasten mit den Bezeichnungen der SPS-Module 181-185
Taste= 114 4 4 182
Taste= 114 5 4 183
Taste= 114 6 4 184
Taste= 114 7 4 185
/ein SPS Modul vom Typ 90 verknüpft die Tasten:
z=300 90 Betriebsart 181 182 183 184 185






```
Taste=    200   3  10  190    /Taste 3 auf Seite 200 "Resettet" den Betriebsstundenzähler von Modul 190 „Tankuhr"
Anzeige=  200  30   8  190  400   3  6   1  -1  32767 8  /zusätzliche Betriebsstundenanzeige der Tankuhr rechts
                                                          neben der Taste.
```

Beispiel für die Erzeugung von Untermenüs (Tastenart „9“):

Die Ausgangsseiten für die Anwendermenüs sind die Seiten 110..119. Für entsprechende Untermenüs sind die Seiten 200..254 vorgesehen. Das Zurückschalten erfolgt wie gewohnt über „#“ bzw. über die Menüsteuerung.

1. Vom Hauptmenü (1) über „Anwendungen“ (11) erreicht man die Seite „Extras“ (119) mittels folgender Scriptzeilen:

```
Menü11x=  119 Extras        /Überschrift Seite 119 vorgeben
Taste=    119   5   9  210  /Taste 5 führt zur Seite 210
```



2. Ausgangspunkt für ein weiteres Untermenü sei die Seite 210. Von hier können ebenso mehrere Untermenüseiten angesprungen werden. Der Rücksprung von diesen Untermenüseiten über die Rautetaste „#“ erfolgt bei Verwendung von Tasten der Art „9“ immer zurück auf die Seite 210. Beispiel für die Gestaltung der Seite 210:

```
Menü11x=  210 Service13       /Überschrift Seite 210 vorgeben
Taste=    210   0   9  240    /Taste 0 führt zu Seite 240
Taste=    210   1   9  241    /Taste 1 führt zu Seite 241
```



3. Ein Beispiel für diese (letzte) Untermenüseite 240 mit mehreren Funktionstasten:

```
Menü11x=  240 ParametersetA   /Überschrift Seite 240 vorgeben
Taste=    240   0   1  149    /Taste 0 beeinflusst SPS Modul  149
Tastengruppe=240 123 313 1 0  /Tasten 1..3 beeinflussen SPS Modul 313
…
z=149  12 Effekt    131 134    0   1   /Verknüpfung
…
z=313  56 Delta-Wert   K     0     /Speicher: Delta-Wert für Heizungssteuerung
…
```



Beispiel für eine fortlaufende Menüführung (Tastenart „15“):

Im Gegensatz zur Untermenüverzweigung wird bei der fortlaufenden Menüführung zwar ebenso zu einer weiteren Seite geschaltet, der Rücksprung erfolgt aber immer zu dem ursprünglichen Ausgangspunkt und nicht zu der vorangegangenen Seite. Damit lassen sich vorteilhaft z. B. Parametersets in ein Menü einbinden.



Zunächst wird die Startseite 118 definiert:

```
Menü11x= 118 Service    /Überschrift "Service" auf Seite 118
Taste=       118 3 9 250    /Sprung zu Untermenü 250
```



Weiter geht es zur Seite 250:

```
Menü11x=   250  Param_1ff   /Überschrift "Param_1ff" auf Seite 250
Taste=       250   0   15   251  /Weiterschaltung zu Seite 251
```

Danach zur Seite 251:

```
Menü11x=   251 Param_2   /Überschrift "Param_2 auf Seite 251
Taste=       251  0   15   252 /Weiterschaltung zu Seite 252
```



Weiter zur Seite 252:

```
Menü11x=   252 Param_3  /Überschrift "Param_3" auf Seite 252
```

Von den Seiten 250, 251 sowie 252 erfolgt der Rücksprung durch die Rautetaste immer zurück zur Seite 118.

4.4.4. Programmierung einer Bildschirm- „Tastengruppe“ in den selbst gestalteten Menüs (Softkeys): (Erweiterung ab 6.0.38)

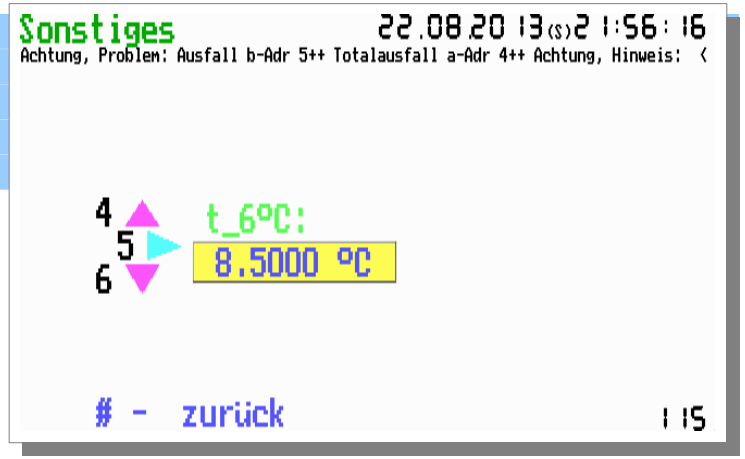
```
Analogwert= oder
Tastengruppe=    [Menü]___[Typ]__[Modul]__[Schrittweite]_[Standardwert]
                110..119   xxx   51..1199          real         real
                200..254
                   Typ xxx:
                       123 - Variation f von SPS Modul mit Tasten 1-3
                       456 - Variation f von SPS Modul mit Tasten 4-6
                       789 - Variation f von SPS Modul mit Tasten 7-9
                       z03 - Variation SPS Zeit-Modultyp 4/37 mit Tasten 0-3 *)
                       z47 - Variation SPS Zeit-Modultyp 4/37 mit Tasten 4-7 *)
                       z07 - Variation SPS Zeit-Modultyp 5/38 mit Tasten 0-7 *)
                     z89 - Variation SPS Zeit-Modultyp 37/38 mit Tasten 8/9 *)
```

*) Die Angabe von Schrittweite und Standardwert werden hier vom Programm nicht ausgewertet bzw. verwendet.

Dieser komplexe Befehl gruppiert gleichzeitig mehrere Tasten im Anwendermenü. Diese Tasten sind jeweils für die Erhöhung, Verringerung sowie das Setzen eines Standardwertes vorgesehen. Schrittweite bezeichnet den Betrag, um den der Analogwert des betroffenen Moduls bei jeder „Tasten“ Betätigung erhöht bzw. verringert wird, Standardwert ist der physikalische Wert, auf den der Analogwert mit der mittleren Taste gesetzt wird. Es gibt mehrere Möglichkeiten für die Auswahl der Tasten – als Typ muss entsprechend **123**, **456, z03** usw. angegeben werden. Die drei Zeilen

```
Menü11x=   115  Sonstiges
Tastengruppe=115   456    82   0.5   8
z=82   70   t_6°C   grd  6       /Differenztemperatur vorgeben; oder auch:
z=82   56   t_6°C   °C   6.0     /ebenfalls Differenztemperatur vorgeben,
                    aber mit Langzeitspeicherung des (geänderten) Wertes
```

erzeugen folgendes Bild auf der Bildschirmseite 115:



Die erste Zeile benennt die Menüseite als „Sonstiges“. Die Programmierung der Bildschirmanzeige und der Tastenbelegung wird durch die Zeile „Tastengruppe=...“ erzeugt. Angezeigt wird der Name des SPS-Moduls Nr. 82 „t_6°C“ und der zugehörige aktuelle Analogwert (8,5 °C). Mit der Taste „4“ kann man den Analogwert des SPS-Moduls jeweils um 0,5 °C erhöhen, mit der Taste „6“ entsprechend verringern und mit der Taste „5“ auf den Wert 8 °C setzen.

Beispiele für die Variation der SPS Zeitmodule (SPS- Modultpen 4, 5, 37, 38):

Zunächst werden die Anwendungsseite 118 und dann die Untermenüs 235-239 eingerichtet:

Menü11x= 118 Service /Überschrift "Service" auf Seite 118

Taste= 118 3 9 235 /Sprung zu Untermenü 235
Taste= 118 4 9 236 /Sprung zu Untermenü 236
Taste= 118 5 9 237 /Sprung zu Untermenü 237
Taste= 118 6 9 238 /Sprung zu Untermenü 238
Taste= 118 7 9 239 /Sprung zu Untermenü 239

Natürlich müssen für die Veränderung von Zeiten auch solche in der SPS vorhanden sein. Es seien also 5 SPS Module gegeben, 2 vom SPS-Typ 4 mit jeweils einer Zeit, ein Zeitbereichsmodul vom SPS-Typ 5 und jeweils noch 2 Zeitmodule mit Freigabeeingang:

z=63 5 Heizungsparameter 4 28 23 58 /04:28 – 23:58
z=64 4 Parameter_12345 08 13 /08:13
z=65 4 Parameter_12346 20 55 /20:55
z=66 37 zeitwert_123456 09 24 8 /09:24 nur mittwochs (8)
z=67 38 zeitwert_987654 07 30 13 54 -4 11 /7:30 – 13:54 Freigabemodul: -4 (kein Werktag), Speicherung aktiviert (11)

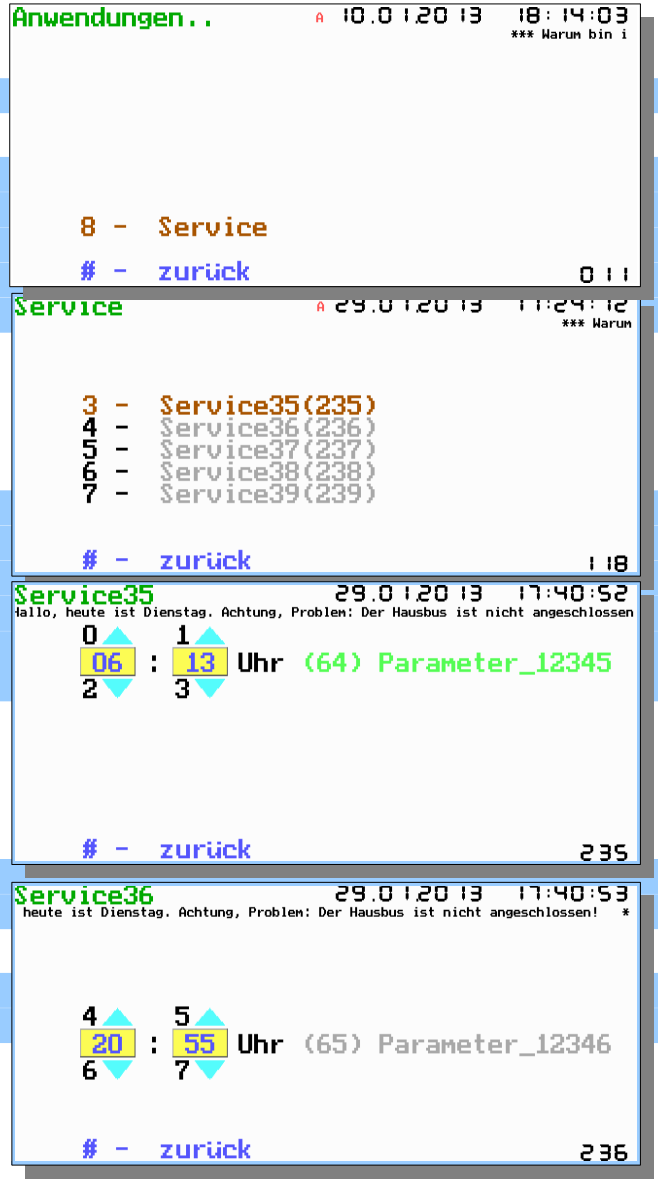
Danach wird für jede der geplanten Serviceseiten jeweils die Überschrift und eine Tastengruppe für die Änderung der Zeitangaben in den entsprechenden SPS Modulen eingerichtet. Zur Anwendung kommen hier alle 4 Typen („z03“, „z47“, „z07“, „z89“):

Menü11x= 235 Service35 /Überschrift "Service35" auf Seite 235
Tastengruppe= 235 z03 64 /Tastengruppe für Zeitparameter SPS-Modul 64

Menü11x= 236 Service36 /Überschrift "Service36" auf Seite 236
Tastengruppe= 236 z47 65 /Tastengruppe für Zeitparameter SPS-Modul 65

Die beiden Typen „z03“ und „z47“ können natürlich auch auf einer Seite kombiniert werden.

Die Variation des Zeitbereiches benötigt die Tasten von 0 bis 7 und setzt einen



Zeitbereichsmodul vom SPS Typ 5 oder 38 als Bezugsmodul voraus:

Menü11x= 237 Service37 /Überschrift "Service37" auf Seite 237
Tastengruppe= 237 z07 63 /Tastengruppe für Zeitparameter SPS-Modul 63

Die Tasten 8 und 9 sind noch frei und können anderweitig belegt werden.

Menü11x= 238 Service38 /Überschrift "Service38" auf Seite 238
Tastengruppe= 238 z47 66 /Tastengruppe für Zeitparameter SPS-Modul 66
Tastengruppe= 238 z89 66 /Tastengruppe für Freigabemodul am SPS-Modul 66

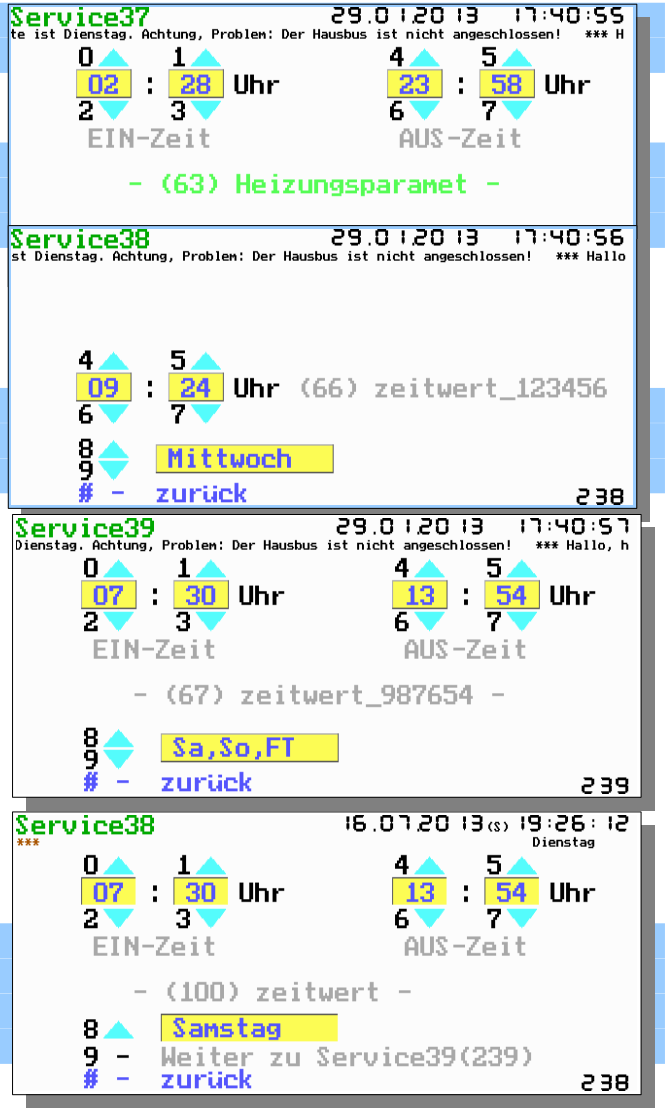
Im Menü „Service38“ kann zusätzlich zum Zeitmenü der Freigabebezug des SPS Moduls 66 mit den Tasten 8 und 9 geändert werden. Aktuell steht der Freigabebezug intern auf „8“, das bedeutet „Mittwoch“.

Menü11x= 239 Service39 /Überschrift "Service37" auf Seite 239
Tastengruppe= 239 z07 67 /Tastengruppe für Zeitparameter SPS-Modul 67
Tastengruppe= 239 z89 67 /Tastengruppe für Freigabemodul am SPS-Modul 67

Ebenfalls zum Zeitmenü ist im Menü „Service39“ der Freigabeeingang modifizierbar. Auf dem rechten Bild (Sa, So, FT) ist intern „-4“ eingestellt. Verwendbar ist die Tastengruppe 8/9 in Verbindung mit den SPS Modultypen 37/38. Der Umschaltbereich beträgt: „Sa,So,FT“ (-4) - „inaktiv“ (0) - „täglich“ (1) - „Testmodus“ (2) - „AUTOMAT“ (3) - „WERKTAG“ (4) - „Sonntag“ (5) - „Montag“ (6) - … - „Samstag“ (11). Ist die entsprechende Angabe aktuell zutreffend, so wird der Text farblich gekennzeichnet.

Jetzt noch eine praktische Variation: Die vorangegangene Bereichs-Anzeige wird auf die Seite 238 übertragen und zusätzlich soll es möglich sein, von der Seite 238 auf die Seite 239 mit einer Menütaste „weiterzuschalten“. Wenn man in Kauf nehmen kann, dass die Taste 9 für die Wochentagsumschaltung nicht mehr zur Verfügung steht, so ist folgende Lösung möglich:

Menü11x= 238 Service38 /Überschrift
Tastengruppe= 238 z07 100
Taste= 238 8 13 100 /Taste 8 für Wochentagsumschaltung
Taste= 238 9 15 239 /Taste 9 für Weiterschaltung zur Seite 239



Automatisches Abspeichern geänderter Werte

Werden mittels der Tastengruppen auf SPS Module vom Typ 70 oder 4/5/37/38 (ohne aktiviertem Speicherflag) zugegriffen, so gehen die geänderten Werte nach Programmende und Neustart verloren.

Bei Verwendung der SPS Modultypen 56, 4/5/37/38 (mit aktiviertem Speicherflag) erfolgt 18 Sekunden nach der letzten Betätigung einer Taste zusätzlich die Speicherung aller geänderten Werte inklusive aller aktueller Betriebsstunden in den „gesternx.daz“. Diese Werte werden nach Neustart des Programms erneut geladen und überschreiben im laufenden Programm die ursprünglichen aus der *hauscomp.ini*. Das „Zurücksetzen“ auf die alten Werte ist bei den Zeitmodulen (Typ 4/5/37/38) ist nur durch die Löschung der Dateien „gesternx.daz“ (und damit dem Verlust auch aller Betriebsstunden) bzw. „von Hand“ über das entsprechende Menü möglich.

4.4.5. Gestaltung von Bildschirmanzeigen (möglich für alle Bildschirmseiten) :

- Darstellung von analogen Parametern auf beliebigen Bildschirmseiten:

```
Anzeige= [Anz.seite]_[Darstellungstyp]_[Adresse]_[Kanal]_[X]_[Y]_[P1]_[P2]_[Min]_[Max]_[welcher par?]
           0..255         0..34        0..8  -1199..1199     alle anderen Parameter:Integer
X=1: 1.Spalte  x=2: 2.Spalte, andere x entsprechend Bildschirm- Pixel von links (3..639)
        Y<25: Y bezeichnet ein Zeile    Y>=25: Y bezeichnet die Pixel von oben (25..199)
wenn Adresse=8:     Anzeige des Analogwertes f aus Zustandsmatrix
                der Kanal entspricht dabei der Modulnummer
```

Anzeige= 95 6 3 2 2 7 1 2 +20 +85 0 ;Bargraph(6) für Analogsensor Adresse 3 Kanal 2, linke Bildschirmspalte, 7. Zeile auf Seite 95 (Hausparameter); Wert ist blau eingefärbt, wenn +20 unterschritten werden; blinkt rot, wenn Wert über +85

Dieser Befehl ist der umfangreichste im Programm. Für die verschiedenen Möglichkeiten werden z. T. nur wenige Parameter benötigt. Trotzdem müssen zur Einhaltung der Syntax immer alle 11 Parameter angegeben werden. Die nicht benötigten werden als „0“ eingetragen.
Einschränkung: In der *hauscomp.ini* dürfen maximal 400 dieser Befehle verwendet werden, dann ist der entsprechende Speicher-Puffer voll.

Hinweis: Wenn mehr als 80 Zeichen in der Script-Zeile stehen, werden möglicherweise die letzten Parameter vom Programm auf null gesetzt!

Anz.seite: Nummer der Bildschirmseite, auf der die Anzeige erscheinen soll. z. B. ist die Seite 95 die Seite „Hausparameter“ im automatischen Modus.

Darstellungstyp: siehe nachfolgende Tabelle

Adresse, Kanal: Adresse, bzw. Kanal des Analogwertes vom Hausbus bzw. Adresse=8 und Kanal=Modulnummer von der SPS;

X: Horizontale Positionierung der Anzeige. Bei x- Werten von 1 und 2 wird die Anzeige auf der linken bzw. rechten Bildschirmhälfte positioniert. Alle anderen x- Werte sind Pixelwerte von links beginnend. Es ist damit eine freie Positionierung im Bereich 3 (links) - 639(rechter Rand) möglich.

Y: Vertikale Positionierung der Anzeige. Bei y- Werten kleiner 25 wird die Anzeige auf ein Zeilenraster gezogen. Es sind die Zeilen 1 bis 9 möglich. 7 entspricht z. B. Zeile 7. Bei y- Werten größer 25 wird die Anzeige entsprechende Pixel vom oberen Bildschirmrand positioniert. Damit ist eine freie Positionierung im Bereich 25(oben) – 199 (unterer Rand) möglich.

P1, P2: verschiedene Bedeutungen, Zuordnung siehe Tabelle

Welcher par?:

0 - **letzter gültiger Wert**
1 - Bezugszeit
2 - Zeitdifferenz
3 - Wertänderung je Sekunde
4 - **Durchschnitt/Mittelwert letzte 10 Werte (Glättung)**
5,6,7 - Konvertierungskonstante a0, a1, a2
8 - letzter Durchschnitt
9 - Bezugszeit letzte Änderungsberechnung
18 - heute: Anzahl Messwerte
19 - heute: laufender Tagesmittelwert
10 - heute: minimaler Wert (vom Mittelwert)
11 - Zeit MIN (heute)
12 - heute: maximaler Wert (vom Mittelwert)
13 - Zeit MAX (heute)
14 - gestern: minimaler Wert (vom Mittelwert)
15 - Zeit MIN (gestern)
16 - gestern: maximaler Wert (vom Mittelwert)
17 - Zeit MAX (gestern)
20 - gestern: Tagesmittelwert

`Mittelwertberechnung:` $\overline{x}_i = (1/i)((i-1)\overline{x}_{i-1} + x_i)$

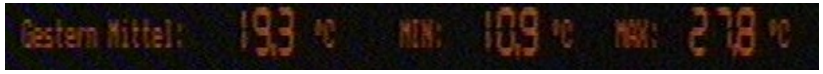
Skalenanfang, Skalenende:

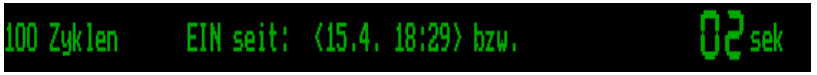
Bei grafischen Anzeigen wird durch die Software automatisch der Beginn und das Ende der anzuzeigenden Skala berechnet. Sie brauchen daher nirgendwo angegeben zu werden, es reicht die jeweiligen MIN und MAX Werte vorzugeben! Bei Anzeigen, die sich direkt auf Sensoren beziehen (Adressen 0-7), wird der theoretisch mögliche, nur durch den jeweiligen AD Wandler begrenzte, physikalische Messbereich für den Skalenbereich verwendet. Bei Anzeigen, die sich auf den Ausgang von SPS Module beziehen, kann zur Berechnung der Skalenenden nur auf die angegebenen MIN bzw. MAX Werte zurückgegriffen werden, da ein SPS Modul am Ausgang beliebige Werte annehmen könnte. Intern wird dabei der Bereich von MIN zu MAX als 100% gesetzt. Der Skalenanfang wird dann 25% unter MIN gelegt und das Skalenende 25% über MAX, oder anders herum, will man den Bereich von 0 bis 100 anzeigen lassen, muss man als MIN 17 und als MAX 83 angeben!

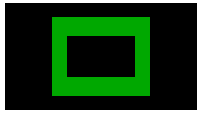
MIN: Unterer Grenzwert. Wird dieser minimale Wert unterschritten, so wird die Anzeige blau eingefärbt und intern ein Ereignis registriert. Es sind nur ganze Zahlen im Integer- Bereich zulässig. Diese Funktion und die Ausfallmeldung sind deaktiviert bei MIN= -32768.

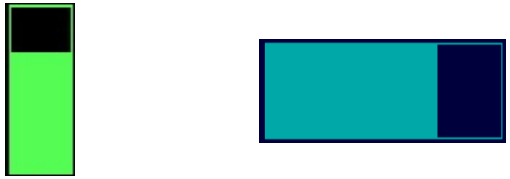
MAX: Oberer Grenzwert. Wird dieser maximale Wert überschritten, so blinkt die Anzeige rot. Intern wird ein Ereignis registriert. Es sind nur ganze Zahlen im Integer- Bereich zulässig. Diese Funktion ist deaktiviert bei MAX= 32767

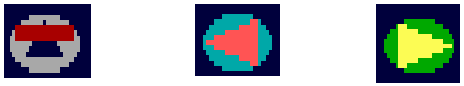
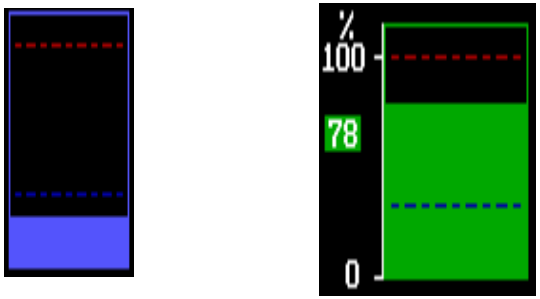
7-Segmentanzeigen: Die Darstellung von Werten in Form einer 7-Segmentanzeige (z. B. Anzeigetypen 0..4) ist im Zahlenbereich begrenzt. Die Angabe „-----“ bedeutet 5 Stellen, die Angabe „---“ 3 Stellen. Den vollen Zahlenbereich erreicht man mit den Anzeigetypen 25 und 28.

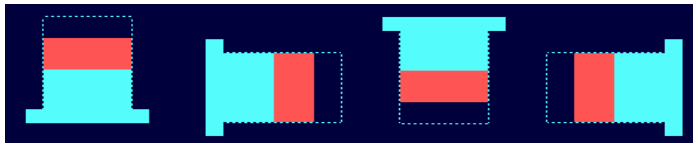
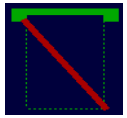
Darstellungs-typ	*Grafik*	*Beschreibung*
0		[Bezeichnung Analog-Parameter] ----- [Einheit] [Parametertyp]
0		[Bezeichnung SPS-Parameter] ----- [Einheit] +„letzter Wert“
1		Programmstart [Datum, Uhrzeit]
2		[Bezeichnung Analog-Parameter] --- [Einheit] [Bezeichnung SPS-Parameter] --- [Einheit] Wertanzeige ist 3-stellig, bei SPS-Parametern auch 4-stellig
3		[Bez. Analog-Parameter] --- [Einheit] heute MIN: --- [Einheit] MAX:--- [Einheit]
4		[Bez. Analog-Parameter] gestern Mittel:[Einheit] heute MIN: --- [Einheit] MAX:--- [Einheit]
5		[welcher par?][Bezeichnung Analog-Parameter][Wert] [Einheit] [Uhrzeit]
6		[Bezeichnung Analog-Parameter] -horizontaler Balken mit aktuellem Wert und Tendenz- Pfeil. Die heutigen Maximal- und Minimalwerte, sowie der Durchschnittswert sind weiß gekennzeichnet.

Darstellungs-typ	*Grafik*	*Beschreibung*
7		Thermometer mit automatischer Skalierung. Der grafische Anzeigebereich wird bei realen Sensoren automatisch gesetzt, bei der Anzeige von analogen Werten der SPS dagegen aus den anzugebenden „Min“ und „Max“ Werten errechnet. Ausfallanzeige mit rotem Diagonalstrich.
8		Windstärke nach Beaufort für Analog-Werte in m/s
9		[Bezeichnung Zustand SPS-Modul] als Leuchtfeld in der entsprechenden programmierten Farbe. Die Feldfarbe wird automatisch blau bzw. weiß, wenn das SPS-Modul auf „Hand EIN“ bzw. „Hand AUS“ geschaltet wurde. Überlagerungstechnik ist möglich 1).
10		Stoppuhr: [Bezeichnung Binär-Parameter] „EIN“ bzw. „AUS“ „seit“ [Datum/Uhrzeit] [Stoppuhr]. Die Anzeige von Stunden und Minuten werden automatisch generiert.
11		Zeigerinstrument mit Anzeige der Bezeichnung und der Einheit. Min/Max Werte werden farblich berücksichtigt. Ausfallanzeige mit rotem Diagonalstrich.

Darstellungs-typ	***Grafik***	***Beschreibung***
12		Balkenanzeige mit Bezeichnung, Wert und Einheit. Min/Max Werte werden farblich berücksichtigt. Ausfallanzeige mit rotem Diagonalstrich.
13, 21, 35		Reine 7-Segmentanzeige ohne Extras, Größe und Anzahl Nachkommastellen einstellbar. Die Gesamtzahl der Stellen beträgt immer 2 (Typ 35), 4 (Typ 21), 5 (Typ 13). Führende nullen werden weggelassen.
14		Zeichnet einen Rechteckrahmen. Größe, Linienstärke und Farbe einstellbar. Überlagerungstechnik ist möglich 1).
15		Zeichnet eine Linie. Farbe und Linienart einstellbar. Überlagerungstechnik ist möglich 1).
16		Zeichnet einen Kreis. Radius und Farbe einstellbar. Überlagerungstechnik ist möglich 1).
17		Zeichnet einen Vollkreis in entsprechender Farbe in Abhängigkeit vom Zustand eines SPS Moduls. Überlagerungstechnik ist möglich 1).
18		7-Segmentanzeige, 5- stellig mit [Bezeichnung], [Einheit] und Bargraf

Darstellungs-typ	*Grafik*	*Beschreibung*
19		Zeichnet einen Kessel ähnlichen Rahmen. Überlagerungs-technik ist möglich 1).
20 (vertikal) 36 (horizontal)		Bargraf ohne Beschriftung. Der Max/Min-Bereich (rot bzw. blau) wird bei realen Sensoren automatisch gesetzt, bei der Anzeige von analogen Werten der SPS dagegen aus den anzugebenden „Min“ und „Max“ Werten errechnet (+/- 25 %). Ausfallanzeige mit rotem Diagonalstrich.
22		Zeichnet ein Heizregister (z. B. in einem Kessel). Überlagerungstechnik ist möglich 1).
23		Zeichnet ein Dreieck (z. B. als Pumpensymbol) in Abhängigkeit von Zustands-Signalen. Überlagerungstechnik ist möglich 1).
24, 34 (Rippen=0)		Zeichnet ein Rechteck/-Rahmen in Abhängigkeit von Zuständen. Überlagerungstechnik ist möglich 1).
26		Zeichnet eine Linie Abhängigkeit von Zuständen. Farbe und Linienart einstellbar.
25, 28, 38, 39		Parameteranzeige mit Einheit. Stellenanzahl programmierbar, Min/Max Werte und Ausfall werden farblich berücksichtigt. Bei den Typ 28, 39 wird oberhalb der Anzeige zusätzlich die Bezeichnung dargestellt. Die Typen 38, 39 zeigen die Restzeit von Zeit-SPS-Modulen (Typ: 61, 62, 64, 65) an. Wird als

Darstellungs-typ	*Grafik*	*Beschreibung*
(Zeitformat ab 6.0.34)		Anzahl der Vor- und Nachkommastellen eine „0“ eingetragen, so erfolgt die Anzeige automatisch als Zeit in der Form hh:mm:ss oder hhhhh:mm. 2) 3)
27		Symbol für eine (Heizungs-/Wasser-) Pumpe in Abhängigkeit von Zustands-Signalen. Überlagerungstechnik ist möglich 1).
29, 30 (ab 6.0.31 bzw. 6.0.34)		Typ 29 und 30 zeigen im Unterschied zu 25 und 28 die Betriebsstunden (sofern möglich) des betreffenden SPS Moduls an (Ausgang t). Deshalb ist als Bezug nur der Zugriff auf entsprechend ausgestattete SPS Module möglich. Wird als Anzahl der Vor- und Nachkommastellen eine „0“ eingetragen, so erfolgt die Anzeige automatisch als Zeit in der Form hh:mm:ss oder hhhhh:mm. 2) 3)
31, 32 (ab 6.0.31)		Vertikaler Bargraf wie Anzeigetyp 20. Physikalische Grenzen besser einstellbar. 100 % Grenze und minimal Warnung (low level) Grenze werden eingeblendet. Breite einstellbar. Typ 32 zusätzlich mit automatischer Prozent-Anzeige. Optimiert für Füllstandsanzeigen.

Darstellungs-typ	***Grafik***	***Beschreibung***
33, 34		Zeichnet einen Heizkörper, Zahl der Rippen und Größe wählbar. Überlagerungstechnik ist möglich 1).
37	 Orientierung: 0 1 2 3  Ausfall:  Die Farbe des Rollladenkopfes ist bei Angabe eines Moduls für den Fensterkontakt abhängig von f des Fensterkontaktmoduls.	Grafik für Rollläden, Soll- und Istwerte werden dargestellt, Breite, Höhe sowie Grundfarbe des Rollladens sind einstellbar. Weicht der Istwert vom Sollwert mehr als 4 % ab, so wird die Differenz als rote Fläche angezeigt. Der (Soll-) Wert kommt vom **f** des Moduls, der (Ist-) Wert von **zh** des gleichen Moduls. Der SPS-Modultyp 111 ist für diesen Zweck bereits optimiert. Bei **zh=0** wird der Rollladen einfarbig dargestellt, bei **zh<>0** zweifarbig. Wird der Istwert **zh** von einem externen Programm regelmäßig in **zh** aktualisiert, so ist die zweite Farbe (rot) nur während des Fahrens des Rollladens zu sehen. Ist **zh<-5**, so wird eine Ausfallanzeige generiert.

1) Überlagerungstechnik: Mehrere verschiedene bedingungsabhängige grafische Objekte können an der gleichen Stelle platziert werden, um besondere Effekte (wie z. B. die Einblendung von unterschiedlichen Fehlermeldungen) zu erreichen. Dazu müssen die „AUS“-Farben auf „0“ oder „-1“ gesetzt sein (siehe folgende Tabellen). Aktivierte Objekte (Zustand „EIN“) werden gezeichnet. Ein einmal gesetztes Element kann allerdings nur durch einen Bild- bzw. Seitenwechsel wieder gelöscht (wenn die Bedingung, die zum Setzen führte, nicht mehr aktiv ist) oder von einem anderen Element überschrieben werden …

2) Zeitformat: Der darzustellende Wert wird als Anzahl von Sekunden interpretiert und in das Format Stunden:Minuten:Sekunden umgerechnet. Überschreitet die Zahl der Stunden 99, so werden nur Stunden:Minuten angezeigt. Bei den Darstellungstypen 25 und 28 (die sich auf den Ausgangsparameter f des entsprechenden SPS Moduls beziehen) erfolgt zusätzlich eine Analyse der Maßeinheit. Wird „min“, „h“oder „min/d“, wie sie SPS Module der Typen 68, 59, 66, 17, 9 30 oder 33 liefern, detektiert, so erfolgt automatisch die Umrechnung in das richtige Zeitformat. Das ist bei eigenen Scripten gegebenenfalls zu berücksichtigen. Bei den Darstellungstypen 29 und 30 erfolgt immer eine korrekte Anzeige, da sie sich ausschließlich auf das interne Betriebsstundenformat beziehen und durch den Anwender nicht beeinflusst werden können





3) Der Parameter „Farbe für den Normalbereich“ (eine Vorgabe ist bei der Anzeige von SPS-Werten möglich) bezieht sich bei Farben von „0..15“ auf die Farbe für das „Hintergrund“-Rechteck. Die Schriftfarbe ist bei Farbwerten von „0..9“ immer weiß und von „10..15“ immer schwarz. Hier ist beispielhaft der Farbwert „14“ (gelb) angegeben:



Werden als Farbwert allerdings Werte im Bereich „16.. 31“ angegeben, so wird das „Hintergrund“-Rechteck in der aktuellen Hintergrundfarbe (also unsichtbar) gezeichnet, während die Schrift dem Farbwert Minus 16 entspricht. Hier beispielhaft der Farbwert 30, d. h. Farbe 30–16=14 (gelb):



In beiden Fällen wird bei Wertänderungen der gesamte Bereich gelöscht und neu überschrieben. Deshalb ist auch keine Überlagerungstechnik möglich.

a) Möglichkeiten der Darstellung für analoge Werte vom Hausbus:

[Typ]	[Adresse]	[Kanal]	[x] [y]	[P1]	[P2]	[Min]	[Max]	[welcher par?]
0,2,3,4,5,6,7,8,11,12, 18	0..7	0..4	x, y	0	0	Min. Wert	Max. Wert	0..20
0,2,3,4,5,6,8,18	0..7	0..4	x, y	Farbe Text	Farbe Wert	Min. Wert	Max. Wert	0..20
13, 21, 35	0..7	0..4	x, y	Pixelhöhe	Anzahl Nachkomma-stellen	Min. Wert	Max. Wert	0..20
20	0..7	0..4	x, y	Pixelbreite	Pixelhöhe	Min. Wert	Max. Wert	0..20
36	0..7	0..4	x, y	Pixelhöhe	Pixelbreite	Min. Wert	Max. Wert	0..20
25, 28	0..7	0..4	x, y	Anzahl Vorkomma-stellen	Anzahl Nachkomma-stellen	Min. Wert	Max. Wert	0..20

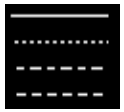
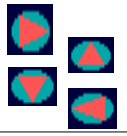
```
[welcher par?]:     0  - letzter gültiger Wert
                    3  - Wertänderung je Sekunde
                    4  - Durchschnitt letzte 10 Werte (Glättung)
                    10 - heute: minimaler Wert
                    12 - heute: maximaler Wert
                    14 - gestern: minimaler Wert
                    16 - gestern: maximaler Wert
                    18 - heute: Anzahl Messwerte
                    19 - heute: laufender Tagesmittelwert
                    20 - gestern: Tagesmittelwert
```

b) Möglichkeiten der Darstellung für analoge Werte der SPS:

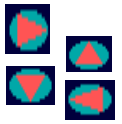
[Typ]	[Adresse]	[Kanal]	[x] [y]	[P1]	[P2]	[Min]	[Max]	[welcher par?]
0,2,3,4,6,8,11,12,18	8	Modul Nr.	x, y	0	0	Min. Wert	Max. Wert	0
0,2,3,4,5,6,8,18	8	Modul Nr.	x, y	Farbe Text	Farbe Wert	Min. Wert	Max. Wert	0
2	8	Modul Nr.	x, y	Farbe Text (0..15)	Farbe Wert (0..15)	Min. Wert	Max. Wert	-1:3-stellig, keine Nachkommastelle 1: 3-stellig, eine Nachkommestelle 2: 4-stellig, eine Nachkommastelle 3: 4-stellig, keine Nachkommastelle
2	8	Modul Nr.	x, y	Modul Nr. Freigabe (>15)	Farbe Text und Wert (0..15)	Min. Wert	Max. Wert	-1:3-stellig, keine Nachkommastelle 1: 3-stellig, eine Nachkommestelle 2: 4-stellig, eine Nachkommastelle 3: 4-stellig, keine Nachkommastelle
7	8	Modul Nr.	x, y	Farbe Rahmen 0: kein Rahmen	Farbe Beschriftung	Min. Wert	Max. Wert	Farbe für Normalbereich und Anzeigewert
13, 21, 35	8	Modul Nr.	x, y	Höhe	Anzahl Nachkomma-stellen	Min. Wert	Max. Wert	Farbe Wert
20	8	Modul Nr.	x, y	Pixelbreite	Pixelhöhe	Min. Wert	Max. Wert	Farbe für Normalbereich
36	8	Modul Nr.	x, y	Pixelhöhe	Pixelbreite	Min. Wert	Max. Wert	Farbe für Normalbereich
23 (Dreieck)[4)] 24 (Rechteck)[4)] 34 (Heizkörper)[4)] 9 (Leuchtfeld)[4)] 17 (Kreisfläche)[4)]	8	Modul Nr.	x, y	dx (23,24,34) (Pixelbreite), Farbe Text, wenn aus (9) 0 (17)	dy (23,24,34) (Pixelhöhe), Farbe Text, wenn ein (9) 0 (17)	0	0	Dreiecktyp (23) 0 (24) Zahl der Rippen (34) Zeichenhöhe 1od. 2 (9) Radius (17)
25, 28, 29, 30, 38, 39	8	Modul Nr.	x, y	Anzahl Vorkommastellen	Anzahl Nachkomma-stellen	Min. Wert	Max. Wert	Farbe für Normalbereich
31,32	8	Modul Nr.	x, y	Physikalisch 100%	Physikalisch abs. Min	Physikalisch Min Warnung	Pixelbreite	Farbe für Normalbereich
37	8	Modul Nr.	x, y	Pixelbreite	Pixelhöhe	Farbe Rollladen	0 oder Modul Nr. Fensterkontakt [4)]	Orientierung (0-3)

[4)] Die Farbe dieser Elemente wird vom analogen Wert des SPS Moduls bestimmt (+0..+15). Es ist keine Überlagerungstechnik möglich.

c) Möglichkeiten der Darstellung des Zustandes von E/A- Kanälen (Binärkanäle des Hausbusses):

[Typ]	[Adresse]	[Kanal]	[x] [y]	[P1]	[P2]	[Min]	[Max]	[welcher par?]
26 (Linie)	0..7	0..7	X1, Y1	Linienfarbe, wenn „EIN“	Linienart, wenn „AUS“: 0 1 2 3 	X2	Y2	Linienfarbe, wenn „AUS“ [5]
9	0..7	0..7	x, y	Farbe Text, wenn „AUS“	Farbe Text, wenn „EIN“	Farbe Feld, wenn „AUS“ [5]	Farbe Feld, wenn „EIN“	1- einfache Zeichenhöhe 2- doppelte Zeichenhöhe
17 (Kreisfläche)	0..7	0..7	x, y	0	0	Farbe Fläche, wenn „AUS“ [5]	Farbe Fläche, wenn „EIN“	Radius (Pixel)
23 (Dreieck)	0..7	0..7	x, y	dx (Pixelbreite)	dy (Pixelhöhe)	Farbe Dreieck, wenn „AUS“ [5]	Farbe Dreieck, wenn „EIN“	Dreieckstyp: 
24 (Rechteck) 34 (Heizkörper)	0..7	0..7	x, y	dx (Pixelbreite)	dy (Pixelhöhe)	Farbe Rechteck, wenn „AUS“ [5, 6]	Farbe Rechteck, wenn „EIN“ [6]	0 (24) Zahl der Rippen (34)
27 (Pumpe)	0..7	0..7	x, y	Farbe Kreis „AUS“	Farbe Kreis „EIN“	Farbe Dreieck „AUS“ [5]	Farbe Dreieck „EIN“	0 - 1 - 2 - 3 - 

d) Möglichkeiten der Darstellung des Zustandes von Modulen der SPS (werden im Gegensatz zu „c)“ auch bei Änderungen aktualisiert):

[Typ]	[Adresse]	[Kanal]	[x] [y]	[P1]	[P2]	[Min]	[Max]	[welcher par?]
9 (Leuchtfeld)	8	+/- Modul Nr.	x, y	Farbe Text, wenn „AUS“	Farbe Text, wenn „EIN“ [7]	Farbe Feld, wenn „AUS“ [5]	Farbe Feld, wenn „EIN“	1- einfache Zeichenhöhe, 2- doppelte Zeichenhöhe
10 (Stoppuhr)	8	Modul Nr.	x, y	0 - ohne Stoppuhr 1 - mit Stoppuhr	0	0	0	0
17 (Kreisfläche)	8	+/- Modul Nr.	x, y	0	0	Farbe Fläche, wenn „AUS“ [5]	Farbe Fläche, wenn „EIN“	Radius (Pixel)
23 (Dreieck)	8	+/- Modul Nr.	x, y	dx (Pixelbreite)	dy (Pixelhöhe)	Farbe Dreieck, wenn „AUS“ [5]	Farbe Dreieck, wenn „EIN“	Dreieckstyp: 
24 (Rechteck) 34 (Heizkörper)	8	+/- Modul Nr.	x, y	dx (Pixelbreite)	dy (Pixelhöhe)	Farbe Rechteck, wenn „AUS“ [5], [6]	Farbe Rechteck, wenn „EIN“ [6]	0 Zahl der Rippen (34)
26 (Linie)	8	+/- Modul Nr.	X1, Y1	Linienfarbe, wenn „EIN“	Linienart, wenn „AUS“: 0 1 2 3 	X2	Y2	Linienfarbe, wenn „AUS“ [5] 0: Linie wird gelöscht, wenn als Linienart „0“ angegeben und der Zustand des Moduls „Aus“ ist.
27 (Pumpe)	8	+/- Modul Nr.	x, y	Farbe Kreis „AUS“	Farbe Kreis „EIN“	Farbe Dreieck „AUS“[5]	Farbe Dreieck „EIN“	0 - 1 - 2 - 3 - 

[5] Ist als Wert „-1“ angegeben, so wird das Bildelement nur dann gezeichnet, wenn der SPS Modul „ein“ ist (bzw. unter Verwendung des Minuszeichens entsprechend „aus“ ist). Damit kann auch die Überlagerungstechnik von mehreren Elementen eingesetzt werden. Die geometrischen Abmessungen dieser Objekte sollten aber ebenso gleich sein, um Überlappungen zu vermeiden. Einmal gezeichnete Elemente werden erst bei einem Bild- bzw. Seitenwechsel gelöscht oder eben durch ein (überschreibendes) anderes Bildelement. Sinnvolle Anwendungen

sind z. B.: verschiedene Zustandsdarstellungen mit Leuchtfeldern an der gleichen Stelle in einer Grafik, das bedingte „Ausblenden“ von Anzeigen mit einem Rechteck in der Hintergrundfarbe oder wechselnde Richtungen des Pumpensymbols.

[6] Achtung, Besonderheit: Die Angabe „0“ beim Typ 24 zeichnet ein Rechteck in der Hintergrundfarbe mit einem hellgrauen Rahmen, die Angabe „16“ zeichnet nur ein Rechteck in der Hintergrundfarbe. Die Werte „1..15“ zeichnen wie gewohnt in der entsprechenden Farbe.

[7] Achtung, Besonderheit: Die Angabe „12“ bei [P2], Typ 9 (rot, blinkend) führt bei Aktivierung/Deaktivierung des Leuchtfeldes automatisch zu einem Hinweis (Signal … Set/Reset), der auch per alarm.txt als email/SMS versandt wird.

e) Statische Anzeigen bzw. grafische Elemente (werden nur einmalig nach Bildwechsel aktualisiert):

[Typ]	[Adresse]	[Kanal]	[x] [y]	[P1]	[P2]	[Min]	[Max]	[welcher par?]
1 (Programmstart)	0	0	x, y	0	0	0	0	0
14 (Rechteckrahmen)	0 (8)	0 (+/-Modul Nr.)	x, y	Linienfarbe	Linienstärke	Länge X	Länge Y	0
15 (Linie)	0 (8)	0 (+/-Modul Nr.)	X1, Y1	Linienfarbe	Linienart: 0 1 2 3 	X2	Y2	0
16 (Kreisring)	0 (8)	0 (+/-Modul Nr.)	x, y	Linienfarbe	0	Radius (Pixel)	0	0
19 (Kessel)	0 (8)	0 (+/-Modul Nr.)	x, y	Linienfarbe	0	Länge X	Länge Y	0
22 (Heizregister)	0 (8)	0 (+/-Modul Nr.)	x, y	Linienfarbe	Anschlüsse: +1: links -1: rechts	0 oder dx	0 oder dy	0
33 (Heizkörper)	0 (8)	0 (+/-Modul Nr.)	x, y	Linienfarbe	Zahl der Rippen	Länge X	Länge Y	0

Diese Anzeigen können sowohl als statische Bildelemente eingesetzt, aber auch als dynamische, vom Zustand eines SPS Moduls beeinflusste verwendet werden. Im zweiten Fall wird das Bildelement nur dann gezeichnet, wenn der SPS Modul „ein“ ist (bzw. unter Verwendung des Minuszeichens entsprechend „aus“). Das Element wird nicht neu gezeichnet, wenn der SPS Modul „aus“ ist (bzw. unter Verwendung des Minuszeichens entsprechend „ein“). Damit ist die Überlagerungstechnik einsetzbar, vorausgesetzt, es findet ein regelmäßiger Bildwechsel statt.

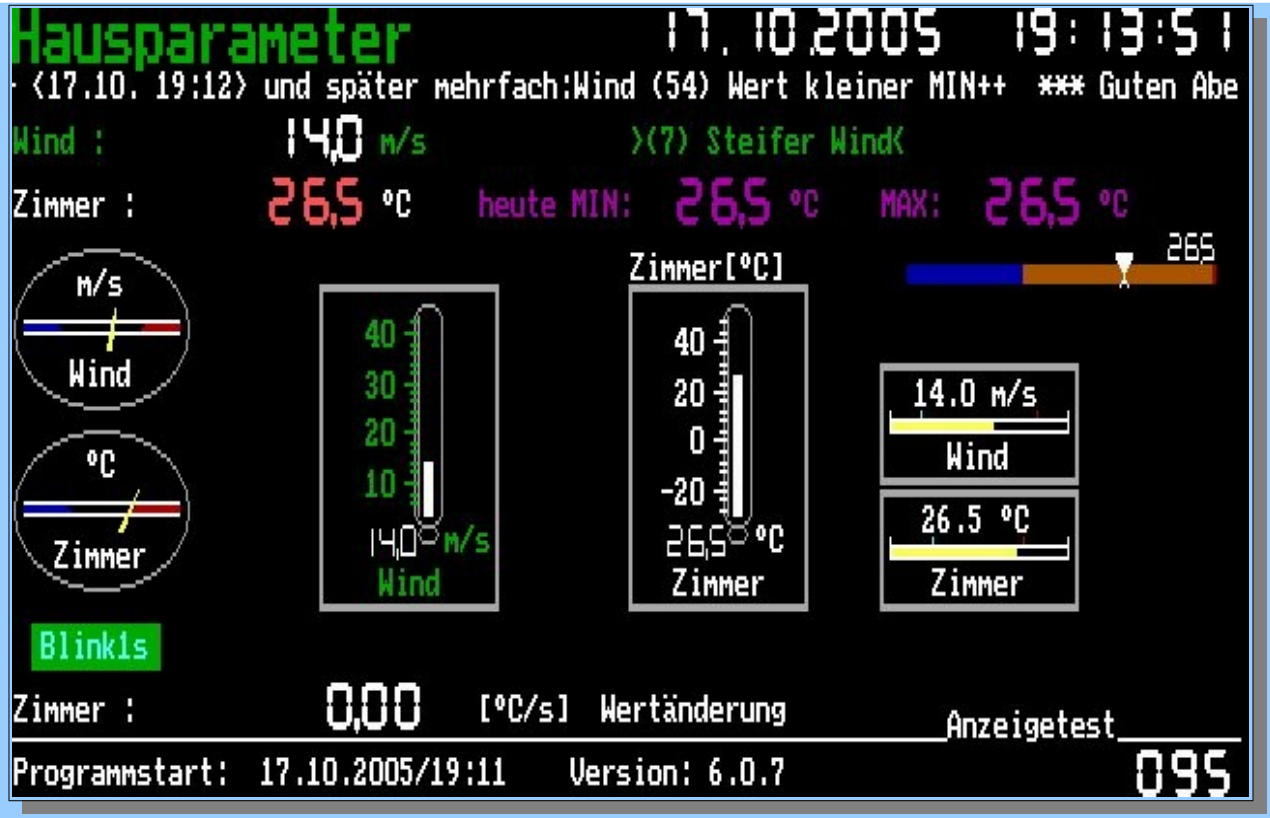
f) Möglichkeiten der direkten Darstellung von Raumsensor-Parametern:

Die verschiedenen Parameter der Raumsensoren werden natürlich nur dann angezeigt, wenn die entsprechenden Sensoren auch vorhanden und detektiert werden. Zusätzlich kann die Anzeige mit einem beliebigen SPS-Modulen verknüpft werden, sodass die grafische Darstellung nur bei Bedarf erfolgt. Das Datenformat (Anzahl Vor-/Nachkommastellen) wird automatisch in Abhängigkeit des ausgewählten Parameters gesetzt.

[Typ]	[Adresse]	[Kanal]	[x] [y]	[P1]	[P2]	[Min]	[Max]	[welcher par?]
25, 28	9	0..15 (Subadresse)	x, y	Farbe für Normalbereich	0 – keine Anzeige 1 – normale Anzeige +/-SPS-Modul Nr. - Anzeige nur, wenn SPS-Modul „ein“	Min. Wert	Max. Wert	0 – Raumsensorart 1 – Temperatur [°C] 2 – Temperatur [°F] 3 – Temperatur [K] 4 – relative Feuchte [%] 5 – absolute Feuchte [g/m³] 6 – CO2-Gehalt [%] 7 – CO2-Gehalt [ppm] 8 – +/-Sollwert [grd] (-6,1..+6,1 grd) 9 – Helligkeit [Lux] 10 – Sättigungsdampfdruck [hPa] 11 – aktueller Taupunkt [°C] 12 – Frequenz [Hz] 13 – Luftdruck [hPa] 14..23 – Bus-Data 1..10 [KE] 24 – Spannung an SOLL [V] 200 ff. siehe SPS-Modultyp 105 (Regenmengen, Windrichtung usw.

Hinweis: Bei dieser Art der direkten Anzeige von Raumsensor-Parametern erfolgen keine automatischen Überwachungsmeldungen bei Über- oder Unterschreitungen der eingestellten minimalen bzw. maximalen Werte. Die jeweiligen Farben blau (Unterschreitung Min) bzw. rot blinkend (Überschreitung Max) sind dann nur in der Grafik sichtbar und nicht wie bei SPS-Parametern auch in der Laufschrift (oder entsprechenden Signalisationen). Raumsensor-Parameter können aber selbstverständlich auch in SPS-Modulen (SPS-Modultyp 105) gespiegelt und über diese auch in Anzeigen mit den entsprechenden Überschreitungsmeldungen wiedergegeben werden (siehe auch b), d)).

Beispiele für die praktische Realisierung von Anzeigen und ihre Programmierung in der hauscomp.ini:




```
Sensor=Zimmer °C 1 0  30    -30     0   +50  255 ;Def. Temp. Sensor analoge Adresse 1 Kanal 0
z=54 80 Wind m/s 18 8 2                          ;Bereitstellung Sägezahn -50..+50 [m/s]

    /Seite/Typ/Adr/Kan/x/ Y/ P1/P2/Min/Max/ par?
Anzeige= 95 2  8  54   1   0  0 0   0  32000  1  ;Wertanzeige (Kurzform) von Modul 54 in der SPS
Anzeige= 95 8  8  54   2   0  0 0 -20 +35     0  ;Beaufort - Anzeige
Anzeige= 95 3  1   0   1   1  0 0  +5 +29     4  ;Mittelwertanzeige und Min/Max von Zimmersensor
Anzeige= 95 6  1   0   2   2  0 0   0  50     0  ;Zimmersensor als Balkenanzeige
Anzeige= 95 7  8  54   160 7  0 0  10  36     0  ;Thermometer Modul 54 SPS
```

```
Anzeige= 95 7  1   0   2   7  0 0  12  36     0  ;Thermometer Zimmersensor
Anzeige= 95 11 8  54   1  89  0 0 -11  30     0  ;Zeigerinstrument Modul 54 SPS
Anzeige= 95 11 1   0   1 135  0 0 -11  30     0  ;Zeigerinstrument Zimmersensor
Anzeige= 95 12 8  54  450  5  0 0 -11  30     0  ;Bargraf Modul 54 SPS
Anzeige= 95 12 1   0  450  7  0 0 -11  30     0  ;Bargraf Modul Zimmersensor
Anzeige= 95 9  8  12  10   8  0 11  7   2     1  ;Zustand Modul 12 als Leuchtfeld
Anzeige= 95 0  1   0   1   9  0 0  +1 +30     3  ;Wertanzeige Zimmer, Langform- Temp. Änderung
Anzeige= 95 1  0   0   1  10                     ;Programmstart- Anzeige

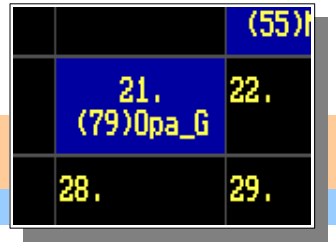
text=    95 1   185 15 1 ____________________________________
text=    95 60  185 15 1 ____________________________________
text=    95 122 185 15 1 Anzeigetest_____________________
```




```
Anzeige=110 19  0   0   1 158  11   1  90  70   0 /Kessel
z=99  70 k       - 50                             /Faktor 50
z=100 72 mal100 °C 18 23 99                       /Testsignal erzeugen
Anzeige=110 20  8 100  25 162  30  56  35  80  10 /vertikaler Bargraf
textgross=110 15 120  11 1 °C                     / „°C"
Anzeige=110 21  8  100  81 92  10   1  35  80  11 /7Segment Anz. Kessel
text=110 24 97  11 1 _____ °C                     / „____°C"
Anzeige=110 21  8  100 141 90   5   1  35  95   2 /7Segment oberer Sensor
text=110 24 117 11 1 _____ °C                     / „____°C"
Anzeige=110 21  8   99 141 111  5   1  30  90   2 /7Segment mittlerer oben Sensor
text=110 24 137 11 1 _____ °C                     / „____°C"
Anzeige=110 21  8  100 141 131  5   1  20  80   2 /7Segment mittlerer unten Sensor
text=110 24 157 11 1 _____ °C                     / „____°C"
Anzeige=110 21  8   18 141 151  5   1  10  70   2 /7Segment unterer Sensor
text=110 5  175 11 1 SPEICHER                     / „SPEICHER"
```

4.5. Programmierung von Kalendereinträgen [KALENDER]:

- Zuweisung von Texten zu bestimmten Kalendertagen. Die Angabe des Jahres ist optional möglich. Es können beliebig viele dieser Datumseinträge in der *hauscomp.ini* vorhanden sein. Das Kalenderfeld wird automatisch blau hinterlegt.




```
Datum= [Tag]_[Monat]_ [Text]_ ([Jahr])
       0..31  0..12  String[40](JJJJ)
```

Datum=21 1 Opa_G 1924 /Diese Zeile erzeugt im Januar des Jahres 2004:

Zwei Eintragungen je Tag sind möglich. Gibt es mehr als 2 Eintragungen am selben Tag, so werden nur die ersten 2 Eintragungen im Kalender mit angehängtem „..“ angezeigt. In der Laufschrift erfolgt dagegen der Hinweis auf alle Ereignisse am heutigen und morgigen Tag.





- Vorgabe des Bundeslandes. Damit werden die Feiertage automatisch ermittelt und rot hinterlegt.

sonstige	- 0	Rheinland-Pfalz	- 6	Österreich*	- 12
Baden-Württemberg	- 1	Saarland	- 7	Berlin	- 13
Bayern	- 2	Sachsen	- 8	Schleswig-Holstein	- 14
Sachsen-Anhalt	- 3	Thüringen	- 9	Hamburg	- 15
Hessen	- 4	Brandenburg	- 10	Bremen	- 16
Nordrhein-Westfalen	- 5	Mecklenburg-Vorpommern	- 11	Niedersachsen	- 17

* Österreich ist natürlich kein Bundesland Deutschlands, es war aber einfacher es so zu programmieren, Entschuldigung!

```
Bundesland= [Land]
             0..17
```

Bundesland=11 /stellt das Programm auf die Feiertage in Mecklenburg – Vorpommern ein

- Vorgabe einer farbigen Gestaltung/Legende für die Kalender-Anzeige, insbesondere für den Befehl „DatBereich“. Für alle betroffenen Tage wird ein entsprechender Farbwert als Tageseigenschaft registriert. Dieser Wert ist täglich über die internen Konstante (SPS-Modul) 16 verfügbar. Damit lassen sich z. B. eine automatische Heizungsabsenkung bei längerer Abwesenheit oder ein verändertes Verhalten der Rollläden in der Ferienzeit programmieren.

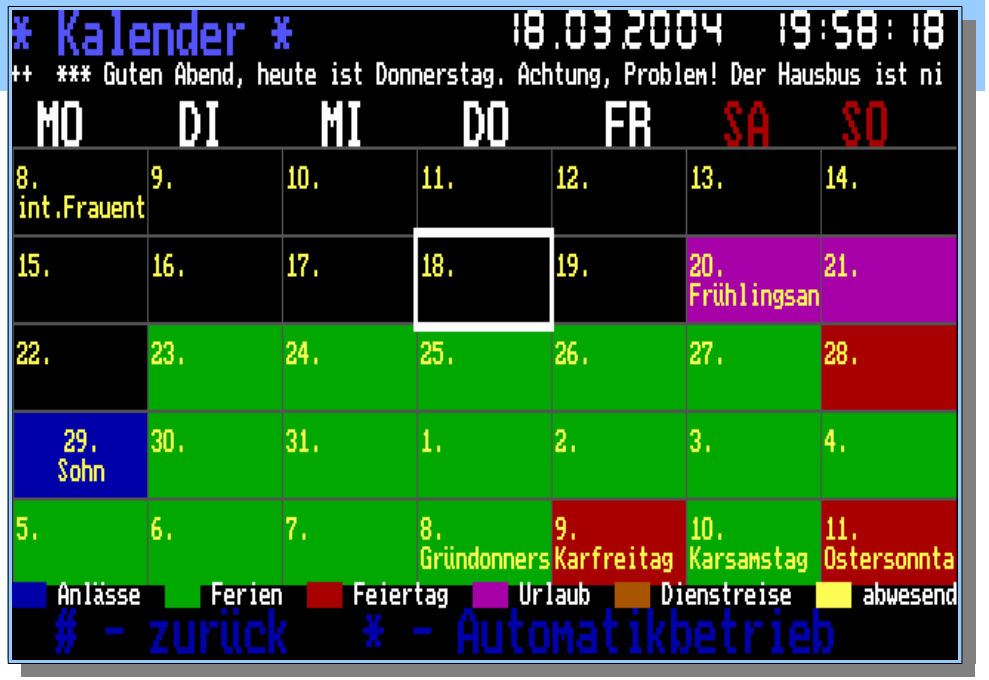
  Der Befehl „DatBereich“ weißt einem Datumsbereich eine bestimmte Farbe/Tageseigenschaft zu. Die Farbe 4 ist für Feiertage reserviert. In der *hauscomp.ini* können beliebig viele Bereiche eingetragen werden, auch auf Jahre voraus.

```
BezDatBereich= [Farbe1] [Bezeichnung1] [Farbe2] [Bezeichnung2]...
                 0..15     String(10)
DatBereich= [Tag-Beginn] [Monat-Beginn] [Jahr-Beginn] [Tag-Ende] [Monat-Ende][Jahr-Ende][Farbe]
```

Folgendes praktisches Beispiel:

```
BezDatBereich=1 Anlässe 2 Ferien 4 Feiertag 6 Dienstreise 14 abwesend 5 Urlaub
DatBereich=20 03 2004  21 03  2004  5  /=Urlaub
DatBereich=23 03 04    29 04    04  2  /=Ferien
DatBereich=28 03 04    29 03  2004  4  /=zusätzlicher Feiertag
```

..und hier das Ergebnis:



- Programmierung von wöchentlich regelmäßig wiederkehrenden Daten (Mülltonnenfunktion) *ab Version 6.0.37*

  Der Befehl „Müll=“ weist im Kalenderbereich den entsprechenden Tagen bis zu 16 verschiedenfarbige Mülltonnen zu. Sie werden im entsprechenden Kalendertag nebeneinander aufgereiht. Mit „Tag“, „Monat“, „Jahr“ wird das Datum des ersten Termins vergeben und mit „Periode“ das Wochen-Intervall. Die entsprechenden Berechnungen für den anzuzeigenden aktuellen Kalender erfolgen jeweils zu Tagesbeginn. Fällt eine Eintragung auf einen Feiertag, so wird ein Fragezeichen eingeblendet. Das Problem von Änderungen der Müllabfuhr wegen Feiertagen wurde wie folgt gelöst: Zunächst wird einmalig für jede Mülltonne die Periodizität programmiert. Dann wird in ***danach folgenden*** Zeilen die (einmalige) Sonderleerung festgelegt. Vor dem automatischen Eintragen der Sonderleerung wird im Kalender vom Programm im Bereich +/- 5 Tage um die Sonderleerung eine eventuelle planmäßige Leerung der entsprechenden Tonne gelöscht.




```
Müll= [Tag]  [Monat]  [Jahr]  [Periode]  [Farbe]   /regelmäßige Eintragung
      1..31   1..12   0..2100   1..100    0..15
                              Periode: Anzahl der Wochen
Müll= [Tag]  [Monat]  [Jahr]  [   0   ]  [Farbe]   /einmalige Sondereintragung
      1..31   1..12   0..2100     0       0..15
```

Praktisches Beispiel: Hier das Ergebnis:

```
Müll=21 01 2013   4 02          /grüne Tonne, montags aller 4 Wochen
Müll=21 01 2013   2 08          /schwarze Tonne, montags aller 14 Tage
```

Der danach folgende zusätzliche Eintrag..

```
Müll=02 04 2013   0 08          /schwarze Tonne Sonderleerung am 2.4.2013
```

..löscht den Eintrag vom 1.4. und erzeugt einen neuen am 2.4.2013:



Beispiele für Farbcodierungen:

```
          Grün        - 2
          Braun       - 6
          Grau        - 7
          Schwarz     - 8
          Blau        - 9
          Gelb        - 14
```

4.6. Befehle zur Programmierung der Bildschirmseiten „Grafik..“ [GRAFIK]: *(letzte Änderungen: V. 6.0.46)*

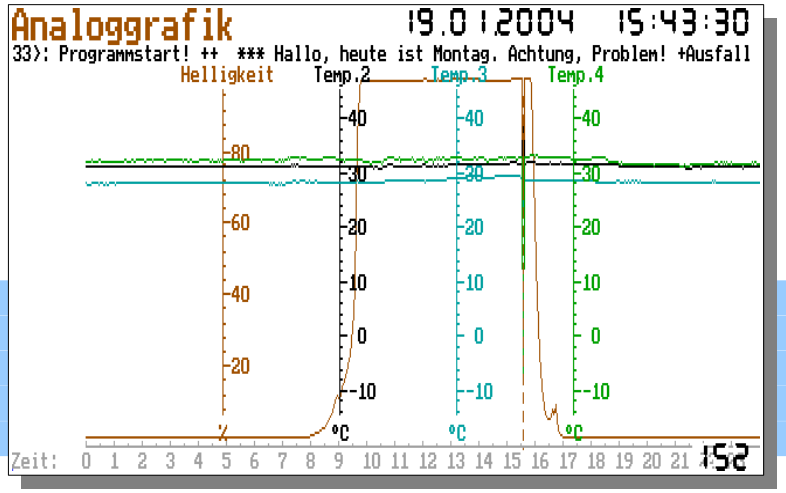
Anzeige von Analogwerten in der GRAFIK:

- Aufnahme von analogen Messwerten des Hausbusses, Raumsensoren oder Analogwerten von SPS-Modulen in die 24h-Grafik. Es können bis zu 2x6 Graphen aufgezeichnet und dargestellt werden. Die Darstellung erfolgt jeweils in der Analoggrafik (Seite 97 und 151 – 154) und in der Sensorübersicht. Die Graphen werden täglich in der gestern.dab/.dai/.dar gesichert (nicht „graph_a2=“!). Sie lassen sich über erneutes Laden (gestern.dat laden im Systemmenü) wieder anzeigen. Unterscheidet sich das Datum der gestern.da* vom Datum des Vortages, muss zusätzlich „DEMO“ vor dem Laden aktiviert werden. Diese erneute Anzeige ist nur sinnvoll, wenn die *hauscomp.ini* zwischenzeitlich nicht verändert wurde.

```
graph_a= [Graph Nr.]_[Adresse]_[Kanal]_[Farbe]_[physikalisch MIN]_[physikalisch MAX]
graph_a2=    0-5         0-7        0-3     0-15       real      real- Analogwert vom Hausbus
             0-5         8          0-1199  0-15       real      real- Modul-Analogwert
             0-5       -15..-1      0-22    0-15       real      real- Wert vom Raumsensor
```

Die Adresse und der Kanal beziehen sich auf entsprechende Analogwerte des Hausbusses. Der Kanal 4 (Analogausgang) wird indirekt aus den Sollausgaben der steuernden SPS Module ermittelt. Bei Adresse=8 werden Analogwerte aus der SPS übernommen und grafisch dargestellt. Wird eine negative Adresse angegeben, so wird der entsprechende Kanal vom Raumsensor ermittelt, siehe auch Pkt. *4.4.5 f)*. Die Subadresse „0“ kann nicht direkt zur Anzeige gebracht werden, sondern nur über den Umweg als SPS-Wert.

Phys. Min bzw. Max sind zur Skalierung auf dem Bildschirm notwendig. Sie entsprechen dem oberen bzw. unteren Bildschirmrand und können völlig frei gewählt werden. Eine Abhängigkeit von den tatsächlich minimal/maximal erreichbaren Sensordaten gibt es nicht. Ist der „z-diff“-Wert in der Sensorinitialisierung > 500 s oder die SPS-Einheit „hPa“, so wird der Graph automatisch punktiert dargestellt.



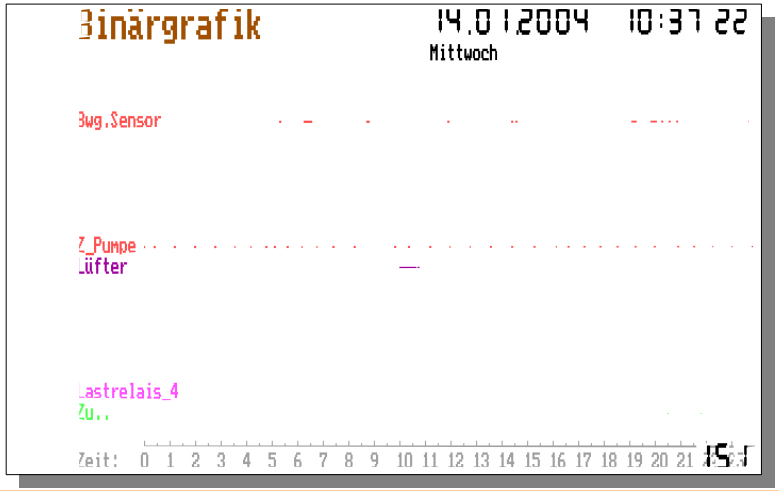

```
graph_a=   1      1   0   14  -10  120  /Helligkeit
graph_a=   2      1   1   15  -25  60   /Temp.2
graph_a=   3      1   2   11  -25  60   /Temp.3
graph_a=   4      1   3   10  -25  60   /Temp.4
graph_a=   5     -2   7   13 -700  8000 /CO2-Gehalt von Raumsensor-2
```

Anzeige von Binärwerten in der GRAFIK:

- Aufnahme von binären Zuständen des Hausbusses / SPS- Modulen in die 24h-Grafik. Es sind max. 2x16 Graphen darstellbar. Anzeige auf den Seiten 97, 151 (graph_b) sowie 153 (graph_b2). Einschränkung: Ein Binärkanal kann maximal einmal zur Anzeige gebracht werden (also nicht gleichzeitig mittels „graph_b" und „graph2_b" und nicht mehrfach auf einer Seite!).

```
graph_b=    [Graph Nr.]_[Adresse]_[Kanal]_[Farbe]
graph_b2=      1-16         0-7       0-7      0-15     - binäre Zustände vom Hausbus
               1-16          8    0-{-}1199    0-15     - binäre Zustände von SPS-Modulen
```

Graph Nr. bezieht sich auf die Bildschirmzeile, auf welcher der Graph gezeichnet wird. 1 ist oben, 16 ganz unten. Die Namen der Signale werden automatisch aus der Bus-Def. bzw. der SPS entnommen. Wird der SPS Kanal mit einem Minus programmiert, so erfolgt die Darstellung invers. Ein SPS-Kanal kann ebenfalls nur einmal programmiert werden.



Beispiele:

```
graph_b=   2   2  0   4   ;Bewegungssensor
graph_b=   8   8  86  12  ;Zirkulationspumpe
graph_b=   9   8  94  13  ;Lüfter_1 anzeigen
graph_b=  15   2  1   5   ;Lastrelais 4
graph_b=  16   2  5   2   ;Prüfen Relais 'ZU'
```

Einstellung des Zeit- Maßstabes im Echtzeitmonitor:

- Festlegung Zeitmaßstab im Echtzeitmonitor

```
Echtzeitmonitor=      [Zeit in min]
                      0.6 ... 86000
```

Hinweis: Im Echtzeitmonitor werden die mit „graph_a", sowie „graph_b" programmierten Signale dargestellt.

4.7. Befehle zur Verwaltung der Modulmatrix (Programmierung Softwaremodule) [SPS]:

In der Modulmatrix oder auch speicherprogrammierbare Steuerung (SPS) genannt, gibt es 59 interne und 740 für den Nutzer frei programmierbare Softwareblöcke, die regelmäßig nacheinander berechnet werden. Sie haben verschiedene Eingänge und mehrere unterschiedliche Ausgänge: Der EIN/AUS Ausgang heißt ***Zustand,*** der zweite Ausgang ist der ***Analogwert (f)***. Der Zeit-Ausgang ***(t)*** wird nur in der SPS-Übersichtsseite angezeigt. Die Module 0 - 50 sind interne Module und sind bereits vordefiniert. Sie stellen an ihren Ausgängen häufig benötigte Konstante, sowie Programmparameter zur Verfügung. Die Module 60 – 1199 können durch den Anwender beliebig verwendet werden. Damit kann die Steuerung externer Geräte wie Pumpen, Rollläden, Lampen, Heizungen o. ä. automatisiert werden. Eine Vielzahl von Modultypen stehen zur Verfügung.

4.7.0. Speicherung von Betriebsstunden:

Ab Programmversion 6.0.13 ist die Betriebsstundenerfassung einheitlich geregelt: Um Mitternacht und bei absichtlichem Programmende werden die erfassten Betriebsstunden bzw. Energiemengen in der Datei „gestern.daz“ gespeichert. Ab Version 6.0.32 zusätzlich 18 s nach Betätigung einer entsprechenden Menütaste und ab Version 6.0.38 unter Linux bzw. Dosemu erfolgt die Speicherung stündlich. Ab Version 6.0.37 und 6.0.54 gibt es wegen der auf 1199 Module vergrößerten SPS zusätzlich die Datei „gestern2/3.daz“. Beim nächsten Programmstart werden die Daten als Start-Werte in den entsprechenden Modulen (z. B. 9, 30, 31, 33, 59 u.a.) wieder übernommen. Damit ist die Fortschreibung der Betriebsstunden/ Energiemengen über lange Zeit gesichert. Ein „Resetten“ dieser Daten ist über zwei Wege möglich: 1. Löschen der Datei(en) „gestern(2/3).daz“ vor dem Programmstart – dabei werden alle Betriebsstundenzähler auf null gesetzt; 2. In der SPS-Übersichtsseite mit dem Kursor auf das entsprechende Modul gehen und die Taste „Entf (Del)“ drücken – es werden die Betriebsstunden des angewählten Moduls auf null gesetzt. Wird ein SPS Modul, das Betriebsstunden zählt, in den Zustand „Hand EIN“ versetzt, so werden keine Betriebsstunden hinzugezählt. Neben der Langzeit-Erfassung der Betriebsstunden gibt es den Begriff „Einschaltzeit“. Damit ist nur die Zeit gemeint, während welcher der Zustand des entsprechenden Moduls seit Programmstart „EIN“ ist. Zusätzlich kann der Modultyp 66 „Tagesbetriebsminuten“ eingesetzt werden. Er liefert die Betriebsminuten eines Moduls für den laufenden und den vorangegangenen Tag. Die Genauigkeit der Erfassung der Betriebsstunden unter DOS ist besser als 0,2 %. Unter Windows XP kann die Abweichung wegen der Timer-Problematik bis zu 10 % betragen. Eine detaillierte Übersicht über Betriebsstunden - relevante Modultypen gibt es im Abschnitt 4.7.8. Ab Version 6.0.39e werden die Betriebsstunden automatisch auch lesbar in der Datei „stunden.txt“ gespeichert.

4.7.1. Allgemeine Befehlsstruktur für die Programmierung der SPS-Module in der Datei *hauscomp.ini*:

Die Programmierung der Softwaremodule erfolgt einheitlich über den Schlüsselbefehl „**z=**“. Für jedes Softwaremodul kann es genau einen solchen Befehl in der *hauscomp.ini* geben. Darin wird das Modul eindeutig programmiert. Nicht programmierte Module sind automatisch als Leermodule gesetzt und damit inaktiv. Änderungen von Parametern an Modulen in der *hauscomp.ini* bei laufendem Programm führen nicht zum Rücksetzen temporärer Daten, d.h. fortgeschriebene Daten, wie z. B. Betriebsstunden bleiben erhalten. Sollte jedoch ein Modul einen neuen Typ zugewiesen bekommen, so wird das entsprechende Modul nach dem Laden der *hauscomp.ini* neu parametrisiert.

Das Befehlsformat sieht wie folgt aus:

```
z= [Modul-Nr.]_ [Modultyp]_[Bezeichnung]_{[Parameter1]_[Parameter2]...[Parameter n]}
    51 - 1199    -128..128    Text[15]        - hängt vom Typ ab -
```

Die verschiedenen Parameter hängen natürlich vom Typ ab und können sowohl der Verweis auf einen anderen Modul sein, als auch direkt einen bestimmten benötigten Zahlenwert oder eine physikalische Einheit darstellen. Im ersten Fall wird der Parameter in der folgenden Befehlsliste mit eckigen Klammern versehen, was auf die indirekte Adressierung hinweisen soll. Diese Klammern bitte nicht in der *.ini Datei verwenden! Modulnummer und Modulbezeichnung werden im weiteren durch .. angedeutet und müssen immer vorhanden sein.

Die Verkopplung von Modulen erfolgt rückwirkend über direkte Adressierung. Alle Analogmodule beziehen sich dabei auf den Ausgang f bzw. 6, alle anderen auf den binären Zustand, sofern nicht gesondert angegeben. Die Analogsignalverarbeitung erfolgt im Gleitkommaformat auf ca. 7 Stellen genau. Die Maßeinheit wird in den Bildschirmanzeigen verwendet. Steht ein Minus vor einer Modulnummer bedeutet das bei einem binären Verknüpfungsmodul (z. B. UND) die Negierung dieses Einganges vor der Verknüpfung. Das Minus bei einer Modulnummer, von dem der analoge Ausgang bezogen wird, bedeutet dagegen die mathematische Negierung x= -x.

Anmerkungen:

- Der Bezeichnungstext (Bezeichnung) dient der Benennung von Modulen und muss ohne Leerzeichen und Anführungszeichen geschrieben sein. Maximal 15 Stellen werden gespeichert.
- 2000000000s sind etwa 63 Jahre. Das ist die längste Zeitdauer, die programmiert werden kann.
- (-Modul) bedeutet: der Zustand des adressierten Moduls wird negiert verwendet.
- Ein Zustand ist immer dann AUS, wenn die EIN-Bedingung nicht erfüllt wird, sofern nicht gesondert angegeben.

Parameterzuordnung:

formal:	0	1	2	3	4	5	6	7	8	9
programmintern:	zi0	zi1	zi2	zi3	zi4	zi5	zr	zl1	ze[]	zh[z]
Kennzeichnung:	Zustand	typ	Adr.	Kan.	a2	k2	f	t	Einheit	-
Datentyp:	Integer						real	longint	String[7]	longint

Die formale Parameterzuordnung wird in der folgenden Befehlsauflistung mit angegeben. Das bedeutet, wenn z. B. unter einem Parameter eine 2 steht, ist dieser vom Typ Integer und stellt somit eine ganze Zahl im Bereich -32768 bis 32767 dar. Der SPS Modultyp 28 verwendet übrigens die formale Zuweisung direkt. Longint bedeutet -2147483648…2147483647, bei Zeiten sind nur positive Werte gültig.

4.7.2. Ausgänge der SPS-Module

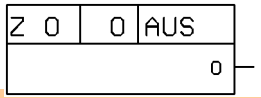
Jeder programmierte SPS-Modul stellt seine Werte auch anderen Modulen zur Verfügung bzw. diese Ausgänge können mit Grafikelementen verkoppelt oder auch in spezialisierte Dateien ausgegeben werden:

- ***Analogwert f (zr)*** – Gleitkommazahl im Bereich -1e7 … +1e7. Der Wert -1e7 bedeutet „Ausfall“.
- ***Zustand zi0*** – logisch „Ein bzw. Aus“. Programmintern werden allerdings mehrere „Zustände“ klassifiziert:
  - 0 - aus auto - grau - normales „AUS“ nach automatischer Berechnung des Moduls.
  - 1 - ein auto - grün - normales „EIN“ nach automatischer Berechnung des Moduls.
  - 2 - AUS Hand - dunkelblau - „AUS“, bei dem das Modul im Handbetrieb, ohne automatische Berechnung verbleibt.
  - 3 - EIN Hand - weiß - „EIN“, bei dem das Modul im Handbetrieb, ohne automatische Berechnung verbleibt.
  - 4 - - gelb
  - 5 - aus, auto, ausgelöst - violett - wird für korrekte Funktion von Tastensätzen benötigt (Verriegelung, Tastentyp 4).
  - 6 - ein, auto, ausgelöst - violett - wird für korrekte Funktion von Tastensätzen benötigt (Verriegelung, Tastentyp 4).
  - 7 - aus, auto, umtasten - violett - wird für korrekte Funktion von Tasten benötigt (Tastentyp 5).
  - 8 - ein, auto, umtasten - violett - wird für korrekte Funktion von Tasten benötigt (Tastentyp 5).
  - 9 - - rot blinken
- ***Zustand zi1 –*** SPS-Modultyp-Nummer, kennzeichnet die Art des SPS-Moduls. Einige Grafiken nutzen diesen Wert.
- ***(Betriebs-) Zeit t*** – wird automatisch von einigen SPS-Modultypen übernommen bzw. in den Betriebsstundenübersichten dargestellt.
- ***Einheit ze*** – Maßeinheit von Analogwert f, wird zum Teil auch automatisch gesetzt oder ausgewertet.

4.7.2. Binäre Verknüpfungs- Module, Zähler, Speicher

Leermodul.

Ausgänge sind immer AUS bzw. null. Das interne Modul Nr. 0, sowie alle nicht benutzten Module sind automatisch von diesem Typ. Modul Nr.0 wird verwendet, um nicht benutzte OR-Eingänge mit einer logischen „0“ abzuschließen.



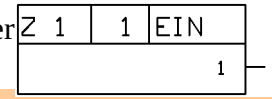

```
Z=.. 0 ..
```

kein Beispiel verfügbar, da dieser Modul praktisch nicht vom Anwender programmiert wird

Im Gegensatz zu Modul 91 erfolgt im automatischen Betrieb eine ständige Zuweisung des Zustandes.

Einsmodul.

Ausgang ist immer EIN. Modul Nr. 1 ist von diesem Typ. Es wird verwendet, um nicht benutzte AND-Eingänge mit einer logischen „1“ abzuschließen.

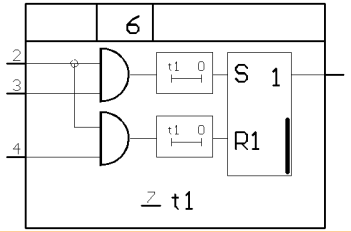



```
Z=.. 1 ..
```

kein Beispiel verfügbar, da dieser Modul praktisch nicht vom Anwender programmiert wird

Speicher/ Flip-Flop, bistabil, UND-Verknüpfung mit Zeitbedingung.

```
Zustand  = EIN, wenn (Zustand x) UND (Zustand y)für min. delta t vorhanden sind,
         = AUS, wenn (Zustand x) UND (Zustand z)für min. delta t vorhanden sind.
Hinweis: Bei Programmende wird der aktuelle Zustand abgespeichert und bei Programmneustart
wieder hergestellt.
```

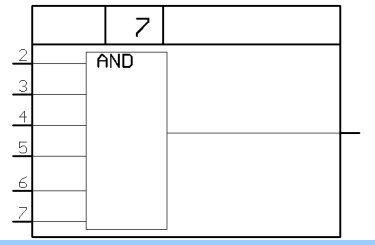



```
Z=.. 6 .._[{-}Modul x]_[{-}Modul y]_[{-}Modul z]_[delta t]
       -1199..1199   -1199..1199  -1199..1199  0..2000000000s
            2             3           4           7
```

z=116 6 verz 1 108 -108 3 /das Ausgangssignal von Modul 108 wird um 3 s durch Modul 116 zeitverzögert weitergegeben
z=59 6 Multivbr 1 -59 59 2 /am Ausgang von Modul 59 wechselt das Signal alle 2 Sekunden (astabiler Multivibrator).

Verknüpfungsmodul (UND).

```
Zustand  = Z1 UND Z2 UND Z3 UND Z4 UND Z5 UND Z6
```

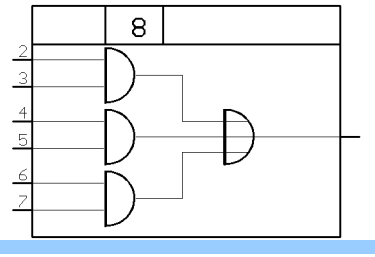



```
z=.. 7 .._[{-}Z1]_[{-}Z2]_[{-}Z3]_[{-}Z4]_[{-}Z5]_[{-}Z6]
            2          3       4         5        6        7
Anmerkung:  2-7 nur im Bereich -1199..1199 zulässig
```

z=51 7 PCFsOK 25 26 27 28 1 1 / der Ausgangszustand vom Modul 51 ist EIN, wenn PCF8574P(1-4) korrekt arbeiten

Verknüpfungsmodul (ODER).

```
Zustand       = (Z1 UND Z2) ODER (Z3 UND Z4) ODER (Z5 UND Z6)
```

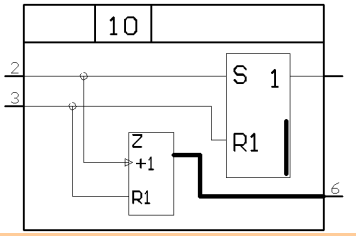



```
z=.. 8 .._[{-}Z1]_[{-}Z2]_[{-}Z3]_[{-}Z4]_[{-}Z5]_[{-}Z6]
            2          3       4         5        6        7
Anmerkung:  2-7 nur im Bereich -1199..1199 zulässig
```

z=160 8 Z8 151 143 152 -143 1 150 /der Ausgangszustand von Modul160 ist EIN, wenn die Zustände von 151 UND 143 EIN sind, oder 152 EIN ist UND 143 AUS ist, aber auch, wenn 150 EIN ist.

Speicher/ Flip- Flop, bistabil.

```
Zustand       = EIN, wenn (Zustand x) mind. einmal EIN war, die Anzahl der Einschaltungen
          (x)wird in Zustandsübersicht dargestellt;
        = AUS, wenn (Zustand y) = EIN (Reset), die Anzahl der Einschaltungen wird
          auf null gesetzt.
Hinweis: Bei Programmende wird der aktuelle Zustand abgespeichert und bei
Programmneustart wieder hergestellt.
```



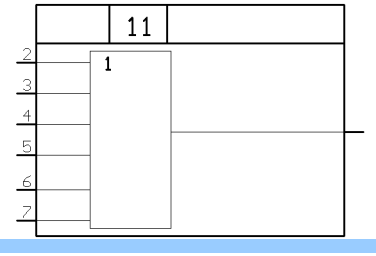

```
z=.. 10 .._[{-}Modul x]_[{-}Modul y]
              -1199..1199    -1199..1199
               2             3
```

z= 51 10 Merker1 12 21 /in Zustandsübersicht zählt Modul 51 wie oft je Minute die Blinkvariable gesetzt wird

Verknüpfungsmodul (ODER).

```
Zustand       = Z1 ODER Z2 ODER Z3 ODER Z4 ODER Z5 ODER Z6
z=.. 11 .._[{-}Z1]_[{-}Z2]_[{-}Z3]_[{-}Z4]_[{-}Z5]_[{-}Z6]
              2        3     4      5         6       7
Anmerkung:  2-7 nur im Bereich -1199..1199 zulässig
```

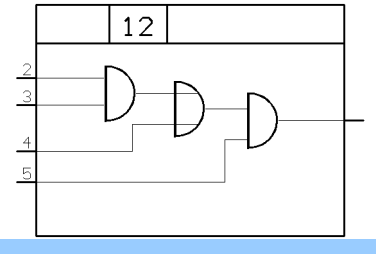
z= 51 11 Wochenende 11 5 0 0 0 0 / der Ausgangszustand von Modul 51 ist sonnabends und sonntags EIN



Verknüpfungsmodul (UND/ODER).

```
Zustand       = ((Z1 UND Z2)ODER Z3) UND Z4
z=.. 12 .._[{-}Z1]_[{-}Z2]_[{-}Z3]_[{-}Z4]
              2        3      4        5
Anmerkung:  2-5 nur im Bereich -1199..1199 zulässig
```

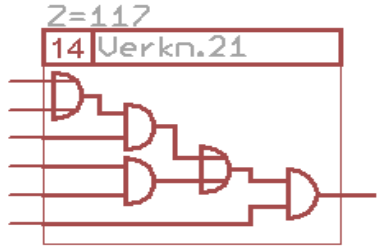
z= 51 12 allesOK 1 12 4 -15 /Ausg. von Modul 51 ist EIN an Werktagen, blinkt an anderen Tagen und ist AUS b. Fehler SPS



Verknüpfungsmodul (UND/ODER).

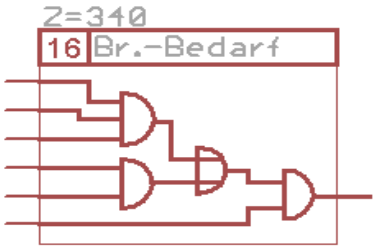
```
Zustand       = (((Z1 ODER Z2)UND Z3)ODER(Z4 UND Z5)) UND Z6
z=.. 14 .._[{-}Z1]_[{-}Z2]_[{-}Z3]_[{-}Z4]_[{-}Z5]_[{-}Z6]
              2              4        5      6       7
Anmerkung:  2-7 nur im Bereich -1199..1199 zulässig
```

z= 52 14 allesOK 1 12 4 -15



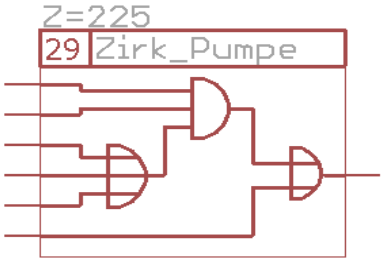
Verknüpfungsmodul (UND/ODER).

```
Zustand       = ((Z1 UND Z2 UND Z3)ODER(Z4 UND Z5)) UND Z6
z=.. 16 .._[{-}Z1]_[{-}Z2]_[{-}Z3]_[{-}Z4]_[{-}Z5]_[{-}Z6]
               2        3        4       5        6        7
Anmerkung:  2-7 nur im Bereich -1199..1199 zulässig
z=340  16 Br.-Bedarf   -302 -337  316   335  337 -332  /Umschaltung  WW/Heizung in Heizungsprojekt
```

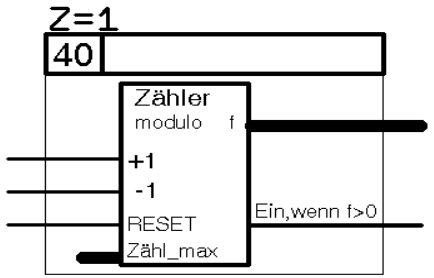


Verknüpfungsmodul (ODER/UND/ODER).

```
Zustand       = ((Z1 UND Z2 UND (Z3 ODER Z4 ODER Z5))ODER Z6
z=.. 29 .._[{-}Z1]_[{-}Z2]_[{-}Z3]_[{-}Z4]_[{-}Z5]_[{-}Z6]
               2        3        4       5        6        7
Anmerkung:  2-7 nur im Bereich -1199..1199 zulässig
z=225  29 Zirk_Pumpe   223  224  398  709  228  222  /Freigabesignal für Zirkulationspumpe
```



Zähler, zyklisch/Modulo zählend.



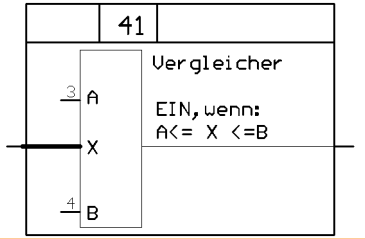

```
Zustand = EIN, wenn Zähler (f) ungleich 0 ist.
f       = aktueller Zählerstand.
Zähler  = +1, wenn „Modul+1" von AUS auf EIN geht (vorwärts zählend bis
          Zählende, dann wieder von 0 beginnend),
Zähler  = -1, wenn „Modul-1" von AUS auf EIN geht (rückwärts zählend bis 0,
          dann auf Zählende springend),
Zähler  = 0, wenn „Modul Reset" von AUS auf EIN geht (RESET).
```

```
z=.. 39 .._[Modul+1]_[Modul-1]_[Modul Reset]_[Zählende]
         -1199..1199 -1199..1199 -1199..1199  0..1199
z=.. 40 .._[Modul+1]_[Modul-1]_[Modul Reset]_Zählende
         -1199..1199 -1199..1199 -1199..1199 0..2000000000
               2           3           4             7
```

```
z=51  2  Taster_St1    4  3  1   /Signal EIN, wenn Lichttaster 1 in der Stube betätigt wird
z=52 40  Zähler   51   0  54 1   /zählt die Impulse von Modul 51: abwechselnd EIN bzw. AUS (entspricht Stromstoßrelais)
z=53 30  Lampe_St2   52  5  2  /steuert die Deckenleuchte der Stube an
z=54  2  Taster_Res    4  5  2   /Signal EIN, wenn Licht- Reset- Taster für 2s betätigt wird: alle Lampen werden ausgeschaltet ...
```

Beim Modul vom Typ 39 erfolgt die Zuweisung des Zählumfanges indirekt über ein anderes Modul, beim Modul vom Typ 40 direkt in der Scriptzeile. Typ 39 ist also ideal als Zähler, dessen Zählumfang erst zur Laufzeit bestimmt wird.

Vergleicher, Fensterkomparator, für ganze Zahlen, ein Kanal.

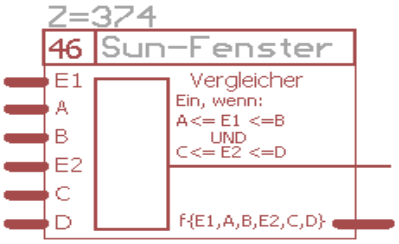


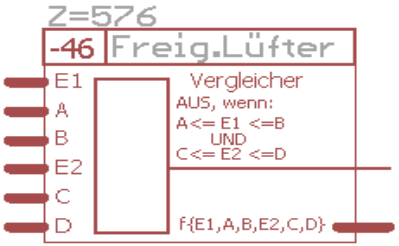

```
Zustand      = EIN, wenn A <= f(x)<= B
Zusammen mit dem Zählertyp 40 lassen sich z. B. Abläufe bzw. Sequenzen steuern.
```

```
z=.. 41 .._[Modul x]_ A_ B
         -1199..1199  A,B: Integer (ganze Zahlen)
               2       3  4
```

```
z=61  41  Prüf6+7   60   6   7  /Modul 61 ist EIN, wenn der Zähler im Modul 60 am Analogausgang eine 6 oder 7 hat (siehe auch obiges
   Beispiel zum Zähler)
```

Vergleicher, Fensterkomparator / Fensterdiskriminator, indirekt, zwei Kanäle, mit Wichtung.

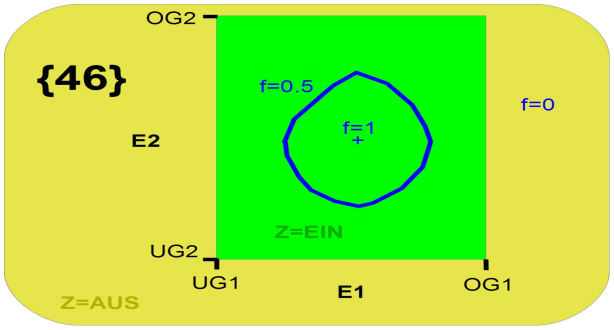


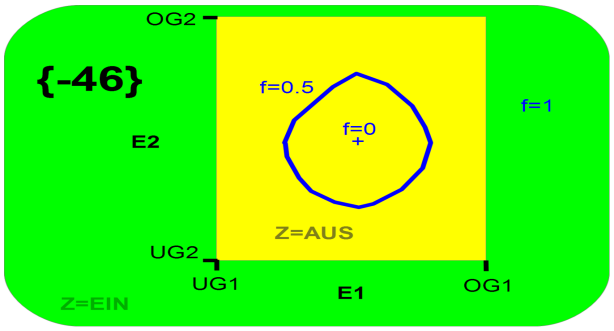


{46} Zustand = EIN, wenn (UG1) < (E1)< (OG1) UND (UG2) < (E2)< (OG2), sonst AUS
{-46}Zustand = AUS, wenn (UG1) < (E1)< (OG1) UND (UG2) < (E2)< (OG2), sonst EIN
Dient der Analyse, ob ein/zwei physikalische Parameter im vorgegebenen Bereich (zwischen unterer und oberer Grenze) liegen (2 Komparatoren mit UND-Verknüpfung) oder NICHT liegen (2 Diskriminatoren mit UND-Verknüpfung). Es wird dabei nur auf SPS-Module mit den Eingängen und Grenzen verwiesen. Sinnvoll zur Auswertung von Bereichen, wie z. B. Sonnenstand (Höhe/Azimut) oder Klima (Temperatur/Luftfeuchte).
{46 (-46)} **f** = **0** (**1**), wenn E1 UND E2 außerhalb des Fensters,
= **0..1** (**1..0**), wenn E1 UND E2 innerhalb des Fensters,
= **1** (**0**), wenn E1 UND E2 genau in der Mitte des Fensters.
Die Wichtung (**f**) beschreibt den Abstand {E1,E2} zum Zentrum des Fensters. Die Funktion setzt UGx < OGx voraus. Wenn als [Mod.E2] „0" angegeben ist, erfolgt nur der Vergleich mit dem ersten Fenster, das zweite Fenster wird ignoriert.

```
z=..  46  .._[Mod.E1]_[Mod.UG1]_[Mod.OG1]_[Mod.E2]_[Mod.UG2]_[Mod.OG2]
z=..  -46 .._[Mod.E1]_[Mod.UG1]_[Mod.OG1]_[Mod.E2]_[Mod.UG2]_[Mod.OG2]
        jeweils -1199..1199
                    2          3          4          5          9          7
```

z=370 70 Hmin ° 17.4 /Vorgabe untere Grenze Sonnenhöhe
z=371 70 Hmax ° 65.0 /Vorgabe obere Grenze Sonnenhöhe
z=372 70 Azimut_min ° 165.5 /Vorgabe untere Grenze Sonnenazimut
z=373 70 Azimut_max ° 223.3 /Vorgabe obere Grenze Sonnenazimut
z=374 46 Sun-Fenster 41 370 371 42 372 373 /Modul 374 ist EIN, wenn der Sonnenstand im Bereich H&A der Grenzen 370 – 373 ist





Überwachung mehrerer SPS- Module auf Änderung.



43:

```
f       = signifikante Quersumme nach Freigabe (Summe der Modulnummern, deren
          Zustand EIN ist)
Zustand = EIN, wenn nach Zuschalten von Modul [z] sich Ausgangszustände von Modulen
          im Bereich X bis Y ändern (Änderung der Quersumme)
Zustand = AUS, wenn die Änderung zurückgenommen wird oder Modul [z] AUS ist.
```

45:

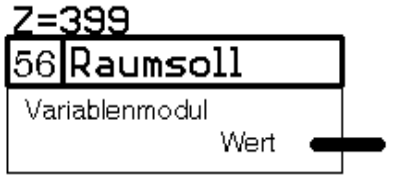
```
f       = Anzahl der Module im Bereich X bis Y mit Zustand=EIN, wenn Freigabe Z
Zustand = EIN, wenn Anzahl > 0 und wenn Freigabe Z
f,Zustand = „0“ bzw. AUS, wenn Freigabe Modul [z] AUS ist.

Anwendung: Überwachung von Zuständen (z. B. nach Scharfschaltung einer Alarmanlage), die Überwachung erfasst dabei
zusätzlich den IST-Zustand. Jede spätere Änderung wird angezeigt.
```

```
z=.. 43 .._[Freigabe z]_[Modul x]_[Modul y]
z=.. 45 .._[Freigabe z]_[Modul x]_[Modul y]
          -1199..1199   0..1199   0..1199
                 4          2          3
```

z=54 43 Control 5 55 70 /der Ausgangszustand von Modul 54 ist EIN, wenn mindestens ein SPS- Modul im Bereich 55- 70 sonntags von EIN auf AUS oder AUS auf EIN wechselt.

Variablen- Speicher mit automatischer Langzeitspeicherung des Variableninhaltes



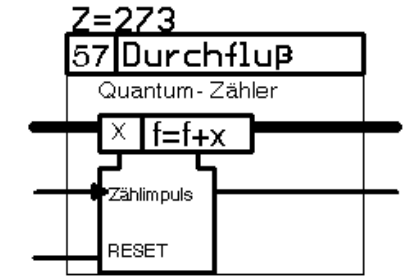

```
f       = f (Wert bleibt immer erhalten), kann aber durch Menütasten „Analogwert="
          manuell während der Laufzeit verändert werden
        = Startwert, einmalig nach Neuprogrammierung SPS oder Löschung
          gestern(2/3).daz
Zustand = EIN, wenn Wert verändert wurde, sonst und nach Programmstart: AUS
```

Der analoge Wert f kann mit Entf/del in SPS Übersicht von Hand auf den Startwert „gesetzt" werden. Hinweis: Wird dieses Modul nachträglich in eine bestehende SPS eingefügt, so können falsche Werteinhalte angezeigt und verarbeitet werden! In diesem Fall sollte der Startwert mit der Taste Entf/del von Hand zugewiesen werden. Rundungsfehler: max. ±0,0005 (durch Speicherung in gestern(2/3).daz)

```
z=.. 56 .._Einheit_Startwert
         String[10]   real
               8          7
```

z=399 56 Raumsoll °C 20.25 /Raum-Solltemperatur vorgeben (der Wert kann von anderen SPS Modulen verwendet werden)
Analogwert=112 456 399 0.25 20.0 /Solltemperatur wird mit 3 Menütasten auf der Anwenderseite 112 modifiziert

Quantum- Zähler mit automatischer Langzeitspeicherung des Zählerinhaltes

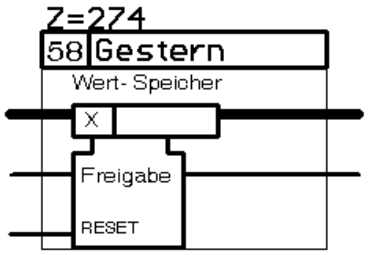



```
f       = f + f(Modul_x), wenn Zählmodul EIN -> AUS, d.h. Hinzufügen des
          „Quantums" von X zu f bei fallender Zählflanke des Zählermoduls
        = 0, wenn Reset= EIN
Zustand = EIN, wenn Zählmodul EIN -> AUS  sonst AUS
t       = Speicherung von f als f*1024; damit erfolgt automatisches Backup über
          „gestern(2/3).daz", daher auch Rundungsfehler: max. ±0,001
e       = Einheit wird automatisch von Modul x übernommen
Der analoge Wert f kann mit Entf/del in SPS Übersicht auf  0 „resettet" werden.
```

```
z=.. 57 .._[Modul x]_[Zählermodul]_[Reset]
           0..1199  -1199..1199  -1199..1199
                2          3          4
```

z=200 70 l 1.5 /Durchflußmenge 1,5 liter je Impuls vorgeben
z=201 86 Durchfluß 0 6 200 /Impulsgeber/Durchflußmengenmesser an Adresse 0, Kanal 6
z=273 57 Durchfluß 200 201 0 /Inkrementelle Addition von 1,5 liter (Modul 200) bei jedem Binär- Impuls (Modul 201)

Speicher mit automatischer Langzeitspeicherung des Speicherinhaltes



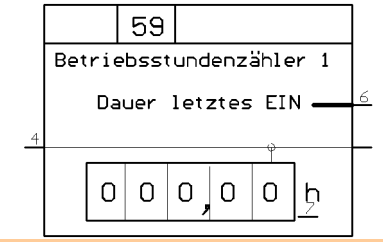

```
f       = f(Modul_x), wenn Freigabemodul aktiv; danach bleibt Wert erhalten
        = 0, wenn Reset EIN
Zustand = Zustand(Freigabemodul)
t       = Speicherung von f als f*1024; damit erfolgt automatisches Backup über
          „gestern(2/3).daz“, daher auch Rundungsfehler: max. ±0,001
e       = Einheit wird automatisch von Modul x übernommen

Der analoge Wert f kann mit Entf/del in SPS Übersicht auf 0 „resettet“ werden.
```

```
z=.. 58 .._[Modul x]_[Freigabemodul]_[Reset]
          0..1199   -1199..1199     -1199..1199
               2            3            4
```

z=xxx yy irgendeinModul ... /dieses Modul liefert einen analogen Ausgangswert
z=273 58 vorgestern 274 42 0 /Übernahme des Wertes zu Tagesbeginn von Modul 274
z=274 58 gestern xxx 42 0 /Übernahme des Wertes zu Tagesbeginn von Modul xxx

Betriebsstundenzähler.



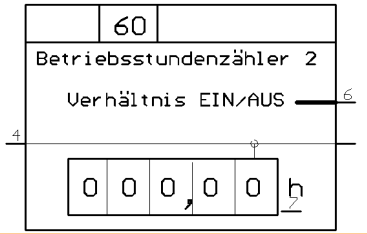

```
Zustand = (Zustand x)  t: Betriebsstundenzähler  (nur in Zustandsübersicht sichtbar)
f       = Zeitdauer letztes EIN (kann für Anzeigen verwendet werden)
e       = „s“ (Einheit wird automatisch gesetzt)
```

```
z=.. 59 .._[Modul x]
          -1199..1199
               4
```

z=100 59 Prgm 1 / in der Zustandsübersicht kann man am Modul 100 die Laufzeit des Programms [h] nach Programmstart sehen

Tastverhältnis.

```
Zustand = (Zustand x)
t       = Einschaltzeit  (nur in Zustandsübersicht sichtbar)
f       = Tastverhältnis über gesamte ununterbrochene Programmlaufzeit
```

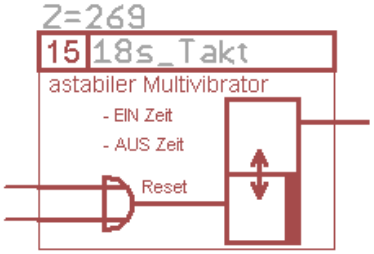



```
z=..  60  .._[Modul x]
           -1199..1199
                 4
```

z=100 60 Tast 12 / in der Zustandsübersicht kann man am Modul 100 das Tastverhältnis des Blinksignals sehen: es liegt bei 0,5

astabiler Multivibrator/ Generator / Taktgeber, resetbar.

```
Zustand = abwechselnd für EIN-Zeit [s]: EIN, für AUS-Zeit [s]: AUS
        = AUS, wenn Modul X ODER Modul Y EIN ist (RESET).
f       = EIN-Zeit + AUS-Zeit
```

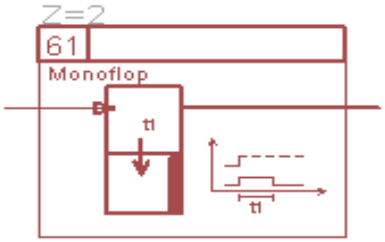



```
z=..  15  .._EIN-Zeit_AUS-Zeit_[Modul x]_[Modul y]
              0..32767          -1199..1199
               2        3          4          5
```

z=269 15 18s_Takt   3 15 0 0   /Ausgangssignal ist jeweils abwechselnd 3s EIN und 15s AUS

Differenzierglied/ Monoflop/ Signalverkürzung.

```
Zustand = EIN für delta t [s], wenn Zustand von Modul x von AUS auf EIN schaltet
f       = verbleibende Restzeit in Sekunden
```

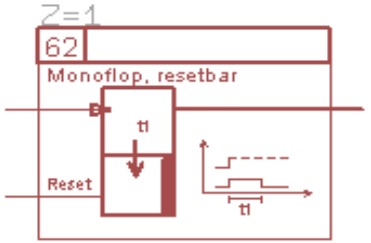



```
z=..  61  .._[Modul x]_delta t
           -1199..1199   0..2000000000
                 4         7
```

z=100 61   Signal1   21   5   /der Ausgangszustand von Modul 100 ist für 5 Sekunden je Minute EIN, d.h. der Minutenimpuls vom Modul 21 wird auf 5s gestreckt.

Differenzierglied/ Monoflop/ Signalverkürzung, resetbar.

```
Zustand = EIN für delta t [s], wenn Zustand von Modul x von AUS auf EIN schalt
        = immer AUS, wenn Zustand von Modul y EIN ist.
f       = f von Modul x
```

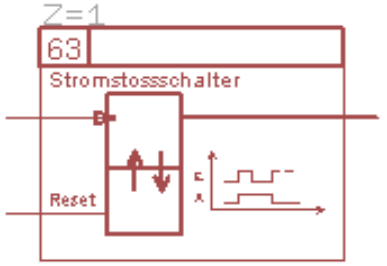



```
z=..  62  .._[Modul x]_[Modul y]_delta t
                  2          3        7
     -1199..1199 -1199..1199   0..2000000000
```

z=230 62 3min   229 -202 180   /Signaldehnung von Modul 229 auf 3min, AUS, wenn Modul 202 AUS ist

Bistabiles Relais/Teiler 2:1/Stromstoßschalter, resetbar.

```
Zustand = abwechselnd umschaltend auf EIN bzw. AUS, wenn Zustand von Modul x
          von AUS auf EIN schaltet. Die Einschaltzeit ist begrenzt auf delta t.
        = immer AUS, wenn Zustand von Modul y EIN ist.
f       = f von Modul x
```

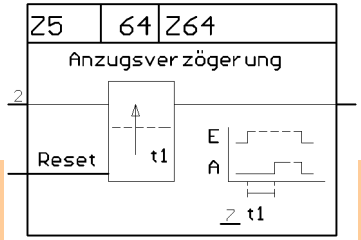



```
z=..  63  .._[Modul x]_[Modul y]_delta t
                  2          3        7
      -1199..1199 -1199..1199   0..2000000000
```

z=60 2 Taste_SL   1   5   0   /Taste_SL vom Bus lesen (Adr.1 Kanal 5)
z=61 63 Stehlampe 60 19 28800   /Signal Stehlampe: bei Tastenbetätigung abwechselnd EIN/AUS, spätestens nach 8h wieder AUS, aber auch AUS, wenn Taste OK (Modul 19) betätigt wird
z=62 30 Relais_SL 61 1   6   /Relais über Hausbus schalten (Adr.1 Kanal 6)

Anprechverzögerung Anzugsverzögerung, resetbar.

```
Zustand = nach delta t EIN, wenn RESET AUS und Modul x EIN ist; sonst AUS.
        = nach delta t EIN, wenn RESET von EIN auf AUS und Modul x EIN ist.
        = immer AUS, wenn Zustand von Modul RESET EIN ist.
f       = f von Modul x
```

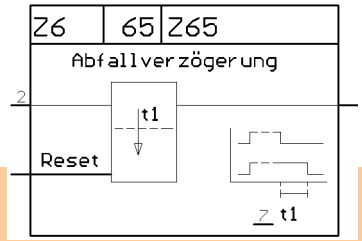



```
z=..  64  .._[Modul x]_[Modul RESET]_delta t
                 2              3          7
      -1199..1199   -1199..1199   0..2000000000s
```

z=74 64 Lüfter 73 0   180   /Signal Lüfter EIN - 3min nachdem Modul 73 auf EIN gewechselt hat „Treppenlichtautomat“:
z=112 2 TasterPoll 2 6 1 /Taster von Bin Adresse 2 Kanal 6 lesen
z=113 10 einschalten 112 114 /Speicher Flipflop Ein: Taste gedrückt, Aus: wenn 114 schaltet
z=114 64 3minAnzug 113 0 180 /Ein nach 3 min, schaltet danach aber gleich wieder ab, da 113 ausgeht

Abfallverzögerung, resetbar.

```
Zustand = EIN wie Modul x, zusätzlich für delta t EIN, wenn Modul x von EIN auf AUS
          wechselt.
        = immer AUS, wenn Zustand von Modul y EIN ist.
f       = f von Modul x
```



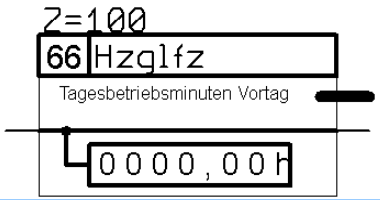

```
z=..  65  .._[Modul x]_[Modul RESET]_delta t
                 2              3          7
      -1199..1199    -1199..1199   0..2000000000s
```

z=82 65 Lüfter 73 0   240   /Signal Lüfter EIN mit Modul 73 dann 4min nachlaufend

Betriebsminuten des Vortages

```
Zustand = (Zustand x)
t       = Betriebsminuten des Moduls x am lfd. Tag (nur in Zustandsübersicht)
f       = Betriebsminuten des Moduls x vom Vortag (für Anzeigen verwendbar)
```

```
e       = „min/d“ (Einheit „Minuten je Tag“ wird automatisch gesetzt)
```

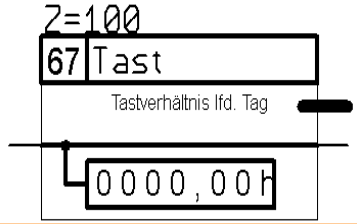



```
z=..  66  .._[Modul x]
          -1199..1199
                4
```

z=99 3 Brenner 3 3 0 2 1 /Heizungssensor (Spannungsmessung) länger als 1s größer 2 Volt: Signal, ob Brenner EIN oder AUS
z=100 66 Hzglfz 99 / der Ausgang des Moduls 100 liefert die Brennerlaufzeit [Minuten] der Heizung am Vortag
z=101 70 Verbrauch l/min 0.0276 /Heizölverbrauch Heizung vorgeben
z=102 80 Verbrauch l/d 100 101 1 /Heizölverbrauch des Vortages [l/Tag] berechnen

Tastverhältnis lfd. Tag.

```
Zustand = (Zustand x)
t       = Einschaltzeit für den lfd. Tag (nur in Zustandsübersicht)
f       = Tastverhältnis Modul x des laufenden Tages
```

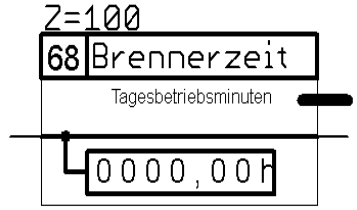



```
z=..  67  .._[Modul x]
          -1199..1199
                4
```

z=100 67 Tast 12 / in der Zustandsübersicht kann man am Modul 100 das Tastverhältnis des Blinksignals sehen: es liegt bei 0,5

Tagesbetriebsminuten

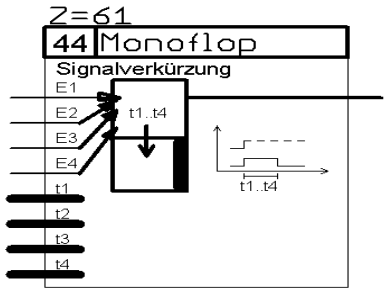
```
Zustand = (Zustand x)
f       = Betriebsminuten des lfd. Tages (für Anzeigen verwendbar)
e       = „min/d“ (Einheit „Minuten je Tag“ wird automatisch gesetzt)
```




```
z=..  68  .._[Modul x]
          -1199..1199
                4
```

z=99 3 Brenner 3 3 0 2 1 /Heizungssensor (Spannungsmessung) länger als 1s größer 2 Volt: Signal, ob Brenner EIN oder AUS
z=100 68 Brennerzeit 99 / der Ausgang des Moduls 100 liefert die Brennerlaufzeit [Minuten] der Heizung des lfd. Tages

Differenzierglied/ Monoflop/ Signalverkürzung, multiple Zeiten (ab Version 6.0.30)




```
Zustand = EIN für delta t1 (t2, t3, t4) [s], wenn Zustand von Modul x1 (x2, x3,
          x4) von AUS auf EIN schaltet.
Dieser Modul ist speziell auf Rollladensteuerungen zugeschnitten. Je nach
auslösendem Modul, können bis zu 4 verschiedene Monoflopzeiten getriggert werden.
Die zugehörigen Zeiten t1-t4 werden in einem Block aufeinander folgender SPS Module
ab 'Modul t' erwartet.
Besonderheiten:
Geht der Zustand des auslösenden Moduls x bereits vor Ablauf der Zeit t auf AUS, so
schaltet das Modul nach Ablauf von t1 auf AUS. Sind mehrere Module gleichzeitig aktiv, so schaltet der Monoflop
nach der zugehörigen längsten Zeit ab. Ein Nachtriggern erfolgt nicht.
```

```
z=.. 44 .._[Modul x1]_[Modul x2]__[Modul x3]_[Modul x4]_[Modul t]
             2              3             4         5          7
                2..5,7: -1199..1199
```

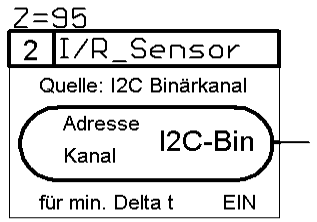
```
Anzeige=110 9  8  61  0 50  0  11 0 7 1      /Anzeige Leuchtfeld „Monoflop“ auf Seite 110
Zeile=110 3  4  55                           /Realisierung Tastensatz auf Seite 110...
Zeile=110 4  4  56
Zeile=110 5  4  57
Zeile=110 6  4  58
Zeile=110 7  4  59

z=51 70 t1 s 9                 /Vorgabe 9s für halbzu
z=52 70 t2 s 15                /Vorgabe 15s für 3/4Zu
z=53 70 t3 s 18                /Vorgabe 18s für Lüftungsstellung
z=54 70 t4 s 22                /Vorgabe 22s für GanzZu
z=55 91 AUS    -               /Bezeichnung der Tasten...
z=56 91 halbzu  -
z=57 91 3/4Zu   -
z=58 91 Lüftungsstellung -
z=59 91 GanzZu -
z=60 90 Test 55 56 57 58 59                  /Notwendig für Tastensatz (realisiert die gegenseitige Veriegelung)
z=61 44 Monoflop 56 57 58 59 51              /Das eigentliche Monoflop - darauf könnte sich ein Rollladenmotor beziehen
```

4.7.3. Verknüpfung der SPS mit Hardware-Eingängen oder der Uhrzeit

- **Binärkanal vom Hausbus lesen.**

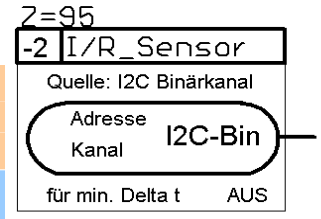
```
Zustand       = EIN, wenn Binärkanal für min. delta t [s] ein
z=..  2  .._Adresse_  Kanal_    t
            0..7       0..7   0..2000000000s
              2          3       7
```



z=95 2 I/R_Sensor 5 4 2 /der Ausgangszustand von Modul 95 ist EIN, wenn der E/A Kanal Nr. 4 am PCF8574P(5) für mindestens 2 Sekunden eingeschaltet wurde.

- **Binärkanal vom Hausbus lesen.**

```
Zustand       = EIN, wenn Binärkanal für min. delta t [s] aus
z=..  -2  .._Adresse_   Kanal_   t
            0..7     0..7    0..2000000000s
               2          3       7
```



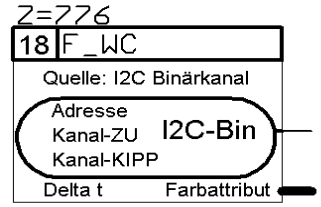
z=95 -2 I/R_Sensor 5 4 2 /der Ausgangszustand von Modul 95 ist EIN, wenn der E/A Kanal Nr. 4 am PCF8574P(5) für mindestens 2 Sekunden ausgeschaltet wurde (nachdem er eingeschaltet war).

- **2 Binärkanäle vom Hausbus (oder auch SPS Kanäle) lesen und entsprechendes Farbattribut generieren.**

```
Zustand = EIN, wenn Binärkanal ZU für min. delta t [s] ein und Farbe=grün ist.
f       = 0(sw)-Ausfall SPS; 12 (rt,blink)-Ausfall Binäradresse bzw. lt Tabellen:
```

SPS Module 13, 18, 27	ZU-Kontakt (Reed oben) oder (Gestänge links)	Kipp-Kontakt (Gestänge rechts)	f (Farbe)
Fenster geschlossen	geschlossen (EIN)	offen (AUS)	2 (grün)
Fenster gekippt	offen (AUS)	geschlossen (EIN)	14 (gelb)
Fenster offen	offen (AUS)	offen (AUS)	4 (rot)
undef. /Sabotage	geschlossen (EIN)	geschlossen (EIN)	12 (rot blink)

SPS Module -18, -27	ZU-Kontakt (Reed oben) oder (Gestänge links)	Kipp-Kontakt (Reed unten)	f (Farbe)
Fenster geschlossen	geschlossen (EIN)	geschlossen (EIN)	2 (grün)
Fenster gekippt	offen (AUS)	geschlossen (EIN)	14 (gelb)
Fenster offen	offen (AUS)	offen (AUS)	4 (rot)
undef. /Sabotage	geschlossen (EIN)	offen (AUS)	12 (rot blink)

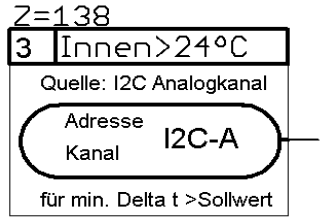



```
Wird als Adresse „8" angegeben, so werden beide Kanäle als SPS-Modulzustände interpretiert.
Bezieht sich dabei ein Kanal indirekt auf den Modultyp 2 (Binärkanal lesen), so wird auch dessen Ausfall erkannt.
z=..  18  .._Adresse_  Kanal-ZU_  Kanal-KIPP_   t        (linke Kontakt-Tabelle gültig)
z=..-18  .._Adresse_  Kanal-ZU_  Kanal-KIPP_   t        (rechte Kontakt-Tabelle gültig)
            0..7 (8)     je 0..7 (-1199..+1199)   0..2000000000s
               2           3           4           7
```

Binär= F_WC_____kipp 7 4 3 0 0 /Kontakt WC_kipp an Binärkanal Adr. 7 Kanal 4 anschließen, negiert (Maske=3)
Binär= F_WC_____zu 7 5 0 0 0 /Kontakt WC_zu an Binärkanal Adr. 7 Kanal 5 anschließen, normal
z=776 18 F_WC 7 5 4 10 /der Ausgangszustand von Modul 776 ist EIN, wenn der E/A Kanal Nr. 5 am PCF8574P(7) für mindestens 10 Sekunden ein ist, während der Kanal 4 an der gleichen Adresse aus ist.
Anzeige= 114 24 8 776 15 105 5 10 0 0 0 /farbiges Rechteck WC auf Seite 114

Dieser Modul dient zur gleichzeitigen Detektion zweier Kontakte. Beispiel: ein Kontakt signalisiert auf/zu an einem Fenster, der zweite Kontakt gekippt/nicht gekippt. Die 4 daraus sich ergebenen verschiedenen Zustände werden als Farbattribut zur Verfügung gestellt und können mit entsprechenden Anzeigen (Tabelle b), Anmerkung [4]) visualisiert werden. Siehe auch Pkt.5 in [4].

Analogkanal größer Festwert.




```
Zustand = EIN, wenn Analogkanal für min. delta t [s] größer als Festwert
f       = vorgegebener Festwert; die Maßeinheit wird automatisch gesetzt.
          Die Funktion wird erst nach dem gültigen Lesen von 10 Messwerten aktiviert.

z=.. 3 .._Adresse_Kanal_Art_Festwert_delta t
            0..7   0..7  0..20   real    0..2000000000s
              2      3     4      6         7
```

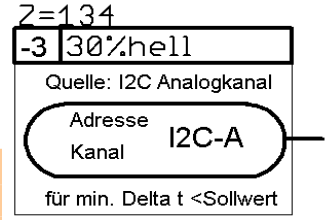
```
 Art:  0  - letzter gültiger Wert
       3  - Wertänderung je Sekunde
       4  - Durchschnitt letzte 10 Werte (Glättung)
       10 - heute: minimaler Wert
       12 - heute: maximaler Wert
       14 - gestern: minimaler Wert
       16 - gestern: maximaler Wert
       18 - heute: Anzahl Messwerte
       19 - heute: laufender Tagesmittelwert    Mittelwertberechnung:
```

$$\overline{x}_i = (1/i)((i-1)\overline{x}_{i-1} + x_i)$$

```
       20 - gestern: Tagesmittelwert
```

z=138 3 innen>24° 4 2 4 24 60 /der Ausgangszustand von Modul 138 ist EIN, wenn die Innentemperatur für min 1 min >24 °C ist; vorausgesetzt, an Analogadresse 4 Kanal 2 hängt der Innentemperaturfühler.
z=137 3 Gestrn>20° 1 2 20 19 0 /der Ausgangszustand von Modul 137 ist EIN, wenn die Tagemittelaußentemperatur gestern >19 °C war; vorausgesetzt, an Analogadresse 1 Kanal 2 hängt der Außentemperaturfühler.

Analogkanal kleiner Festwert.



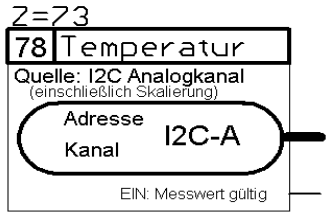

```
Zustand = EIN, wenn Analogkanal für min. delta t [s] kleiner als Festwert
f       = vorgegebener Festwert; die Maßeinheit wird automatisch gesetzt.
          Die Funktion wird erst nach dem gültigen Lesen von 10 Messwerten aktiviert.
```

```
z=.. -3 .._Adresse_Kanal_Art_Festwert_delta t
            0..7     0..7   0..20   real    0..2000000000s
              2        3      4      6        7
```

Art ist wie bei Modultyp 3.

z=134 -3 30%hell 0 1 4 30 120 /der Ausgangszustand von Modul 134 ist EIN, wenn das Tageslicht für mindestens 2 min dunkler als 30% ist, vorausgesetzt, der Hellichkeitssensor ist an der Analogadresse 0 und Kanal 1 angeschlossen.

Analogkanal lesen, einschließlich Skalierung.



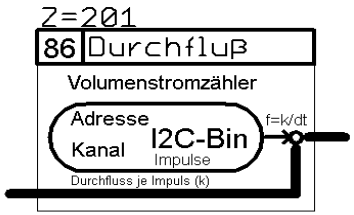

```
f            = Analogmesswert (Art=0: lfd. Wert; Art=1-20: letzter, gültiger Wert)
Zustand      = EIN, wenn der Messwert vom Analogkanal gültig ist (Art=0-20)
               AUS, wenn der Messwert vom Analogkanal ungültig ist (Art=0-20) oder der
               Analogkanal ausgefallen ist (Art=0). Der Ausfall wird gesetzt, wenn der
               Fehlerintegrator des Kanals > 3 ist. Die Funktion wird erst nach dem
               Lesen von 10 Messwerten aktiviert.
```

```
z=.. 78 .._Adresse_Kanal_Art
            0..7    0..4  0..20
              2       3      4      Art: wie Modultyp 3
```

z=73 78 Temperatur 1 2 0 /Außentemperaturmessung [°C]
z=74 78 Wind 1 0 0 /Messung Windstärke [m/s]
z=75 71 Wind km/h 74 13 0 0 /Umrechnung m/s in km/h
z=76 92 Windchill 75 73

Anmerkung: Die Maßeinheit des Messwertes wird automatisch aus der Definition des Sensors übernommen.

Impulsmessung, Volumenstromberechnung, Durchflussmessung mit einem Binärkanal



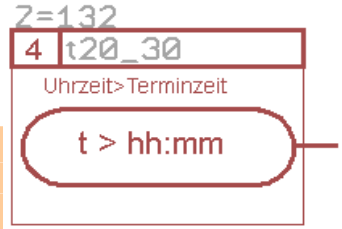
f = k/Δt [Einheiten/h] Zustand = Zustand Binärkanal
k = Konstante, z. B. 1,5 Liter (je Impuls)
Δt = Zeit zwischen 2 Impulsen an einem Binäreingang (z. B. Reedkontakt an Volumenzähler)
Um das Software-Jitter zu minimieren, wird der Volumenstrom gleitend über Mittelwertbildung berechnet. Die Problematik des letzten Impulses wurde wie folgt gelöst: nach $2*\Delta t_{alt}$ wird die Mittlung abgeschalten, nach $4*\Delta t_{alt}$ wird f=0 gesetzt.

```
z=..  86  .._Adresse_Kanal_[Modul k]
            0..7    0..7    0..1199
             2       3        4
```

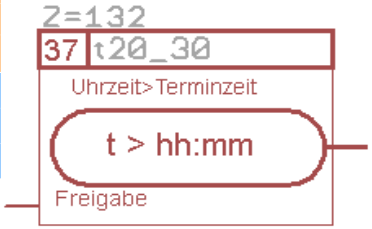
/Durchflußmessung an einer Heizung
z=200 70 l 1.5 /Durchflußmenge 1,5 liter je Impuls vorgeben
z=201 86 Durchfluß 0 6 200 /Impulsgeber/Durchflußmengenmesser an Adresse 0, Kanal 6

Zeitmodul/ Wecker (1).

Zustand = EIN, wenn aktuelle Tageszeit größer als Terminzeit
UND Freigabemodul=ein (nur Typ 37)

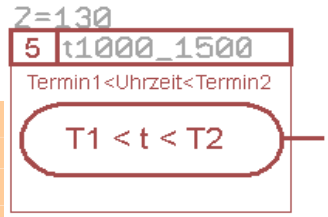



```
z=..  4  .._Stunden_Minuten_(Speicherflag*)
            0..23    0..59        11
             2        3           6
z=..  37.._Stunden_Minuten_[Freigabemodul]_(Speicherflag*)
            0..23    0..59    -1199..+1199          11
             2        3          7                   6
```



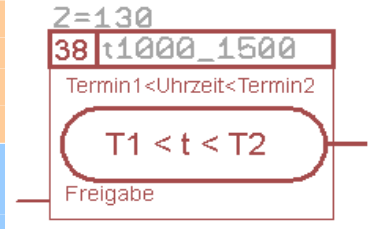
z=132 4 t2030 20 30 /der Ausgangszustand von Modul 132 ist EIN, wenn die Uhrzeit > 20:30 ist.
z=132 4 t2030 20 30 11 /der gleiche Modul mit gesetztem Speicherflag
z=132 37 t2030 20 30 8 11 /der gleiche Modul mit gesetztem Speicherflag aber nur Mittwochs aktiv.

Zeitmodul/ Wecker (2).



Zustand = EIN, wenn aktuelle Tageszeit größer als Terminzeit1 aber kleiner Terminzeit2 ist.

```
z=..  5  .._Stunden_Minuten_Stunden_Minuten_(Speicherflag*)
            0..23    0..59     0..23    0..59      11
             2        3         4        5         6
         Terminzeit1       Terminzeit2
z=..  38  .._Stunden_Minuten_Stunden_Minuten_[Freigabemodul]_(Speicherflag*)
            0..23    0..59     0..23      0..59      -1199..+1199        11
             2        3         4          5             7                6
         Terminzeit1       Terminzeit2.
```



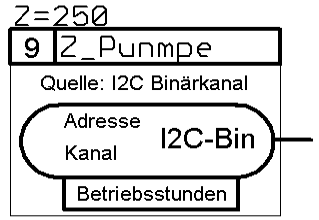
z=130 5 t1000_1500 10 0 15 0 /der Ausgangszustand von Modul 130 ist EIN, wenn die Uhrzeit zwischen 10:00 und 15:00 liegt.

z=130 5 t1000_1500 10 0 15 0 11 /der gleiche Modul mit gesetztem Speicherflag

z=130 38 t1000_1500 10 0 15 0 4 11 /der gleiche Modul mit gesetztem Speicherflag aber nur an Werktagen (Mo-Fr).

Das Speicherflag bewirkt, so es gesetzt ist, dass eine Zeitänderung durch eine Menütaste dauerhaft in der Betriebsstundendatei „gestern(2/3).daz" gespeichert wird und auch nach einem Neustart nur von dort (und nicht mehr aus der *hauscomp.ini*) übernommen wird. Die Zeitangaben in diesem Befehl sind dann nur noch die erstmaligen Startwerte. Selbst spätere Änderungen in der *hauscomp.ini* haben dann keine Wirkung mehr. Funktion erst ab Version 6.0.37.

Betriebsstundenzähler für Binärkanal.



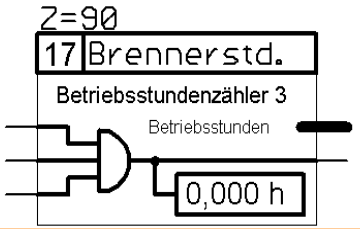
Zustand = Binärkanal,
t = Betriebsstundenzähler [s] (nur in Zustandsübersicht sichtbar)
f = Betriebsstundenzähler Binärkanal [h] (kann für Anzeigen verwendet werden)
Einheit = [h] wird automatisch gesetzt

```
z=..  9  .._Adresse_Kanal
            0..7    0..7
             2       3
```

z=250 9 Z_Pumpe 4 1 /der Ausgangszustand von Modul 250 ist EIN, wenn der Binärkanal 1 an Adresse 4 eingeschaltet wurde (von wem auch immer) und in der Zustandsübersicht werden die Betriebsstunden aufsummiert dargestellt.

Betriebsstundenzähler für SPS Modul. (ab Version 6.0.31)

```
Zustand = EIN, wenn (Modul X)AND(Modul Y)AND(Modul Z)= EIN, gleichzeitig werden
          Betriebsstunden aufsummiert
t       = Betriebsstundenzähler [s] (nur in Zustandsübersicht sichtbar)
f       = Betriebsstundenzähler SPS-Modul [h] (kann für Anzeigen und andere
          SPS Module verwendet werden)
Einheit = [h] - wird automatisch gesetzt
```

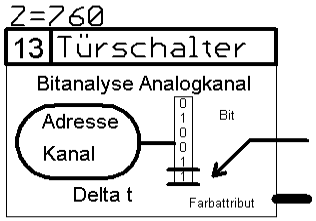


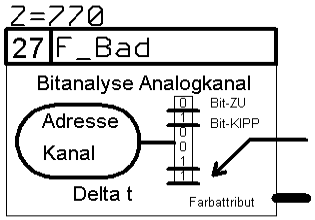
z=.. 17 .._[Modul X]_[Modul Y]_[Modul Z]
-1199..1199 -1199..1199 -1199..1199
2 3 4

z=90 17 Brennerstunden 88 89 1 /der Ausgangszustand von Modul 90 ist EIN, wenn die Module 88 und 90 EIN sind.

Bitanalyse Analogkanal mit Setzen Farbattribut.

Wird verwendet, um mit spezieller Hardware [3] mehrere Schalter in einer Widerstands-Alarmschleife eindeutig zu detektieren. Typ 13 detektiert einen Kanal, die Typen 27/-27 dagegen zwei Kanäle.
Zustand = EIN, wenn Binärkanal ZU für min. delta t [s] ein und Farbe=grün ist.
f = 0(sw)-Ausfall SPS; **12** (rt,blink)-Ausfall Binäradresse bzw. **lt Tabellen:**





SPS Module 13, 18, 27	ZU-Kontakt (Reed oben) oder (Gestänge links)	Kipp-Kontakt (Gestänge rechts)	f (Farbe)
Fenster geschlossen	geschlossen (EIN)	offen (AUS)	2 (grün)
Fenster gekippt	offen (AUS)	geschlossen (EIN)	14 (gelb)
Fenster offen	offen (AUS)	offen (AUS)	4 (rot)
undef. /Sabotage	geschlossen (EIN)	geschlossen (EIN)	12 (rot blink)

SPS Module -18, -27	ZU-Kontakt (Reed oben) oder (Gestänge links)	Kipp-Kontakt (Reed unten)	f (Farbe)
Fenster geschlossen	geschlossen (EIN)	geschlossen (EIN)	2 (grün)
Fenster gekippt	offen (AUS)	geschlossen (EIN)	14 (gelb)
Fenster offen	offen (AUS)	offen (AUS)	4 (rot)
undef. /Sabotage	geschlossen (EIN)	offen (AUS)	12 (rot blink)

Bit-Zuordnung:

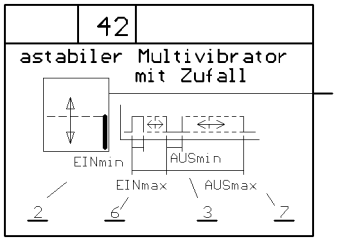
Bit	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Wertigkeit (PCF8591)	1	2	4	8	16	32	64	128
Wertigkeit (ATtiny26)	4	8	16	32	64	128	256	512
SPS	4	8	16	32	64	128	256	512

Z. B. sind bei einem Analogwert von 13 (PCF8591) auf dem Hausbus die Bits 0, 2 und 3 gesetzt (8+4+1=13). Die daraus sich ergebenen verschiedenen Zustände werden als Farbattribut zur Verfügung gestellt und können mit entsprechenden Anzeigen (Tabelle b, Anmerkung [4]) visualisiert werden. Siehe auch Pkt.5 in [4]. Praktisch nutzbar sind dabei mit einem analog-binär Adapter [3] nur die höherwertigen Bit 3-7. Die Scriptzeile unterstützt auch die einzelne Negierung der beiden Bitstellen, sodass alle denkbaren Varianten abgedeckt sein sollten. Ab Version 6.0.47 kann die Maskierung auch mit einem SPS-Modul erfolgen, sodass auch Widerstandsketten per Linearisierung mit ATtiny26 [1] möglich sind.

```
z=.. 13 .._Adresse_Kanal_{-}BitZU_            delta t        (Bit-KIPP wird als aus angenommen)
z=.. 27 .._Adresse_Kanal_{-}BitZU_{-}BitKIPP_delta t        (linke Kontakt-Tabelle gültig)
z=..-27 .._Adresse_Kanal_{-}BitZU_{-}BitKIPP_delta t        (rechte Kontakt-Tabelle gültig)
          0..7    0..3   -7..7     -7..7    0..2000000000s (Daten von Binärkanal)
           8    0..1199  -7..7     -7..7    0..2000000000s (Daten von SPS-Modul)
           2       3        4         5         7
```

```
SensorAt26=Alarm1 Einh  2 1  0  0  0  1023  1023   /an Adresse 2 Kanal 1 mit ATtiny Binäradapter aus [3] angeschlossen
z=770  27  F_Bad  2  1 -7 6 0                      /Bitanalyse Adresse 2, Kanal 1; BitZU = 7, negiert; BitKIPP = 6, normal
```

astabiler Multivibrator mit einstellbarem Zufall.



Zustand = periodisch EIN und AUS, Zeitdauer ist zufällig zwischen jeweiligen mininmal und maximal Zeiten

```
z=.. 42 .._EIN-Zeit-minimal_EIN-Zeit-maximal_AUS-Zeit-minimal_AUS-Zeit-maximal
              0..32767     0..2000000000s     0..32767      0..2000000000s
                 2               6               3               7
```

```
z=54  42  Zufall1   120  1800 7200 10800      /der Ausgangszustand von Modul 54  ist für 2min-30min (zufällig) EIN und für 2h-3h
  (zufällig) AUS. Danach wieder für 2-30min EIN usw.
```

Achtung! Dieser Modultyp kann in der SPS-Übersicht nicht auf Handbetrieb (Eingabetaste) geschaltet werden, d.h. einmal gestartet, läuft der Multivibrator ständig!

Raumsensor/Wärmemengensensor lesen. (ab Version 6.0.46)




```
f       = Raumsensor-/Wärmemengensensor-Messwert (je nach Art)
Zustand = EIN, wenn der Messwert vom Raumsensor gültig ist (gültige Daten
          innerhalb der letzten 10 min)
          AUS, wenn der Messwert vom Raumsensor ungültig ist oder der
          Raumsensor ausgefallen ist. Der Ausfall wird automatisch 10 min nach lesen
          der letzten gültigen Werte gesetzt.
Einheit = wird automatisch gesetzt
```

```
z=..  105  .._Subadresse_Art
                0..15    0..311
                  2        3
```

```
z=182  105  T_Flur    2  1    /Temperaturmessung [°C]
z=183  105  AF_Flur   2  5    /berechnete absolute Feuchte [g/m³]
```

Art	Parameter	Einheit
0	Raumsensortyp [1)]	-
1	Temperatur	°C
2	Temperatur	°F
3	Temperatur	K
4	Relative Feuchte	%
5	Absolute Feuchte	g/m³
6	CO2-Gehalt	%
7	CO2-Gehalt	ppm
8	±Sollwert (-6,1..+6,1)	grd
9	Helligkeit	Lux
10	Sättigungsdampfdruck	hPa

Art	Parameter	Einheit
11	Aktueller Taupunkt	°C
12	Frequenz (0,2..44)	Hz
13	Luftdruck	hPa
14..23	Bus-Data 1..10	KE
24	SOLL-Spannung (wenn VCC IC1=5V)	V
100	Ä.	/min
101	Änderung Temperatur	°C/min
102	Änderung Temperatur	°F/min
103	Änderung Temperatur	K/min
104	Ä. Relative Feuchte	%/min

Art	Parameter	Einheit
105	Ä. Absolute Feuchte	g/m³/min
106	Ä. CO2-Gehalt	%/min
107	Ä. CO2-Gehalt	ppm/min
108	Ä. ±Sollwert (-6,1..+6,1)	grd/min
109	Ä. Helligkeit	Lux/min
110	Ä. Sättigungsdampfdruck	hPa/min
111	Ä. Aktueller Taupunkt	°C/min
112	Ä. Frequenz (0,1..44)	Hz/min
113	Ä. Luftdruck	hPa/min
114..123	Ä. Bus-Data 1..10	KE/min
124	Ä. SOLL-Spannung	V/min

Ausschließlich für Subadresse 15 (Außensensor) möglich (ab Version 6.0.53):

Art	Parameter	Einheit
200	Regenimpulse gesamt	Imp
201	lfd. Tages-Regenimpulse	Imp
202	Regenmenge je m² gesamt	l
203	lfd. Tagesregenmenge je m²	l/d

Art	Parameter	Einheit
204	Regenmenge in den (vor)letzten 30 min	l/h
205	Tagesregenmenge gestern je m²	l/d
206	Windrichtung	° XX
207	Windrichtung, geglättet (5 min)	° XX

Art	Parameter	Einheit
208	aktuelle Windgeschwindigkeit	m/s
209	aktuelle Windgeschwindigkeit	km/h
210	maximale heutige Windgeschwindigkeit	km/h
211	maximale gestrige Windgeschwindigkeit	km/h

Ausschließlich für Wärmemengenmesser möglich (ab Version 6.0.60):

Art	Parameter	Einheit
300	Impulszähler	-
301	Impulse je Sekunde	Hz
302	Messzyklus	s
303	Temperatur2	°C

Art	Parameter	Einheit
304	Temperatur2	°F
305	Temperatur2	K
306	Durchfluss	l/s
307	Durchfluss	l/min

Art	Parameter	Einheit
308	Durchfluss	l/h
309	Wärmeleistung [2)]	W
310	Wärmeleistung [2)]	kW
311	Differenztemperatur dT	K

[1)] 0: AM23xx; 32: AM23xx und AS560x; 64: SHT3x; 96: SHT3x und AS560x; 192: Wärmemengenmesser 224: Raumsensor Fehler)

[2)] Wärmemenge[W] = dT[K] * Durchfluss[l/s] * cp[Wh/(kg*K)] * ρ[kg/l] *3600[s/h}
= dT[K] * Durchfluss[l/s] * 1,1625 * 0,99333 * 3600 [Ws/(l*K)]
= dT[K] * Durchfluss[l/s] * 4157[Ws/(l*K)]
Wärmemenge[kW] = Wärmemenge[W] / 1000

Angenommen die SPS-Module 736 und 734 sind vom Typ 105, dann erfolgen bei Ausfall eines ganzen Raumsensors folgende Fehlermeldungen:

05.05.2025 08:24:27 (Montag) Raumsensor LZ Ausfall
05.05.2025 08:28:46 (Montag) LZ_CO2 (736) Ausfall
05.05.2025 08:28:46 (Montag) LZ_Temp. (734) Ausfall

Raumsensor-Zustand vom Hausbus lesen. (ab Version 6.0.46)




```
Zustand      = EIN, wenn Raumsensor-Zustand ein ist und der Raumsensor in den letzten 5 min
               gültige Daten geliefert hat
f            = Steuerbyte des Raumsensors (1. gelesenes Datenbyte)

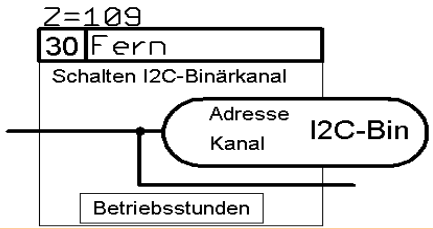
z=.. 106 .._Subadresse_Zustandsart
               0..15       0..7
                 2           3
z=184 106 Bew_Flur 2 1    /der Ausgangszustand von Modul 184 ist EIN, wenn der Raumsensor mit
                           Subadresse 2 eine Bewegung detektiert (Bewegungsmelder)
```

Der Zustand des Raumsensors wird durch eine UND-Verknüpfung der Zustandsart mit dem Status-Byte des Raumsensors ermittelt.

Zustandsart	***Parameter des Raumsensors***	***Bemerkungen***
1	Alarm gesetzt	
2	Infrarot-Bewegung erkannt (Bewegungsmelder aktiv)	
4	Externer Hinweis gesetzt	
8	Relais aktiviert	
16	Intern ist der Relaiszustand an Helligkeit und Bewegung gekoppelt	

4.7.4. Verknüpfung der SPS mit Hardware-Ausgängen

Binärkanal über Hausbus schalten. Raumsensor Statusbyte schalten.



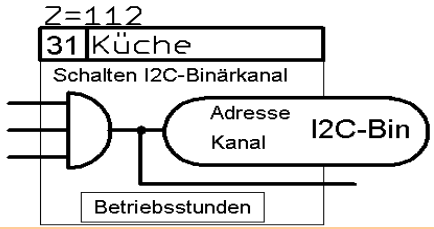

```
Zustand  = (Zustand x) mit gleichzeitigem Einschalten eines Binärausganges
f        = Betriebsstundenzähler Binärkanal [h]
Einheit  = [h] - wird automatisch gesetzt
t        = Betriebsstundenzähler (nur in Zustandsübersicht zu sehen)
```

```
z=..  30  .._[{-}Zustand x]_Adresse_Kanal
              -1199..1199       0..7    0..7        (normaler Binärkanal)
              -1199..1199      -1..-15  0..3        (Raumsensor)
                   4            2       3
```

```
Kanalzuordnung bei Raumsensoren:          0 - Relais schalten
                                          1 - Alarm schalten
                                          2 - externen Hinweis setzen
                                          3 - Zustand „LAMP" setzen
```

z=109 30 Fern 115 2 6 /Das Relais FERN wird eingeschaltet, wenn Modul 115 EIN ist, sofern an Binäradresse 2 Kanal 6 ein derartiges Relais hängt
z=110 30 Lüfter 115 -3 0 /Das Relais im Raumsensor 3 wird ein-/ausgeschaltet mit Modul 115

Binärkanal über Hausbus schalten, mit UND-Verknüpfung.



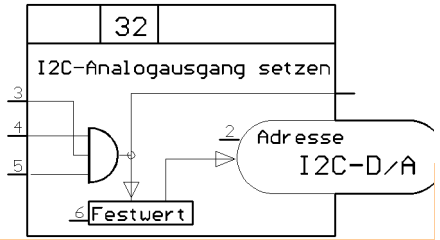

```
Zustand  = (Zustand x)UND(Zustand y)UND(Zustand z) mit gleichzeitigem
           Einschalten eines Binärausganges
t        = Betriebsstundenzähler (nur in Zustandsübersicht zu sehen)
```

```
z=..  31  .._[{-}Zustand x]_[{-}Zustand y]_[{-}Zustand z]_Adresse_Kanal
            -1199..1199      -1199..1199      -1199..1199      0..7    0..7
            -1199..1199      -1199..1199      -1199..1199     -1..-15  0..3      (Raumsensor)
                 4                5                6              2       3
```

z=112 31 Küche 106 108 -116 4 2 /Der Binärkanal 2 Binäradresse 4 wird eingeschaltet, wenn die Module 106 und 108 EIN sind und Modul 116 AUS ist.

```
Kanalzuordnung bei Raumsensoren: siehe vorherigen SPS-Modultyp
```

Analogkanal auf Festwert setzen, mit UND-Verknüpfung.

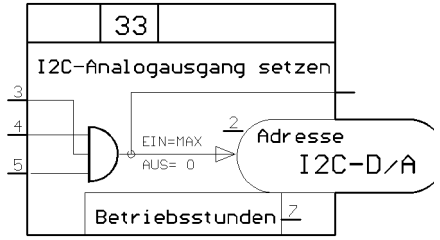



```
Zustand = (Zustand x)UND(Zustand y)UND(Zustand z) mit gleichzeitigem
          Setzen Analogwert auf Festwert
t       = Einschaltzeit (nur in Zustandsübersicht zu sehen)
```

```
z=.. 32 .._[{-}Zustand x]_[{-}Zustand y]_[{-}Zustand z]_Adresse_Festwert
            -1199..1199       -1199..1199       -1199..1199    0..7     real
                 4                 5                 3          2        6
```

Analog= A/D E/A 1 0 0 0 1 255 /Der Analogausgang an der Adresse 1 wird für den Bereich 0-1 festgelegt.
z=100 32 Test 18 18 18 1 0.5 /
z=101 32 Test -18 -18 -18 1 1.0 /Der Analogausgang (Adr.1) schaltet abwechselnd von 128 auf 255 Kodeeinheiten (in der Hausbusübersicht ersichtlich)

Analogkanal EIN/AUS schalten, mit UND-Verknüpfung.

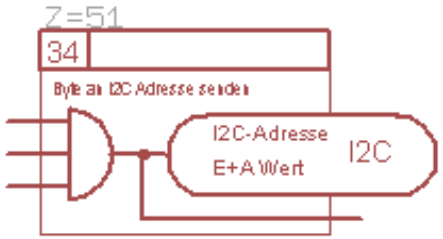



```
Zustand = (Zustand x)UND(Zustand y)UND(Zustand z) mit gleichzeitigem
          Setzen Analogwert auf:
          wenn EIN: D/A Ausgang = 255, bzw.5V; wenn AUS: D/A Ausgang = 0 V
          am analogen Universalinterface mit ATtiny26 entsprechend EIN bzw. AUS
t       = Betriebsstundenzähler  (nur in Zustandsübersicht sichtbar)
Einheit = [h] - wird automatisch gesetzt
f       = Betriebsstundenzähler [h] (kann für Anzeigen verwendet werden)
```

```
z=.. 33 .._[{-}Zustand x]_[{-}Zustand y]_[{-}Zustand z]_Adresse
            -1199..1199       -1199..1199    -1199..1199    0..7
                 4                 5              3          2
```

z=91 33 Vögel 90 90 90 5 /der analoge D/A-Ausgang an der Adresse 5 wird auf +5V gesetzt, wenn Modul 90 EIN ist. Schaltet Modul 90 auf AUS, so geht der D/A-Ausgang auf 0 V.

Byte auf I2C Bus ausgeben, mit UND-Verknüpfung.



Zustand = (Zustand x)UND(Zustand y)UND(Zustand z) mit gleichzeitiger Ausgabe von einem Byte an eine Adresse auf I2C Bus:
wenn Zustand von AUS auf EIN wechselt – Ausgabe von „EIN" Wert;
wenn Zustand von EIN auf AUS wechselt – Ausgabe von „AUS" Wert;

ca. 55 ms nach Übertragung des entsprechenden Bytes wird an die gleiche Adresse automatisiert eine „0" gesendet. Alle diese Werte werden entsprechend dem Standard der PCF8574(A) negiert gesendet. Praktisch erscheint dann an der entsprechenden I2C-Logik ein Impuls, der für weitere Steuerungen genutzt werden kann.

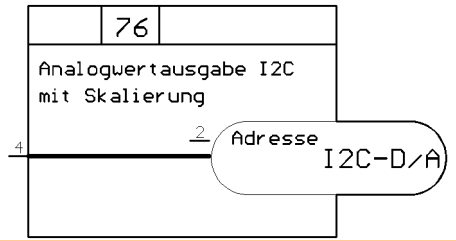
z=.. 34 .._[{-}Zustand x]_[{-}Zustand y]_[{-}Zustand z]_I2C Adresse_"EIN" Wert_"AUS" Wert
-1199..1199 -1199..1199 -1199..1199 0..255 0..255 0..255
2 3 4 5 6 7
32..39 – PCF8574P
54..61 – PCF8574AP

z=51 34 Effekt 12 1 1 54 8 16 /abwechselnd jede Sekunde (modul 12) wird an einen PCF8574A/ Subadresse 0 (I2C Adresse 0111000b = 56d) eine 8 bzw. 16 gesendet,
z=52 34 Effekt 12 1 1 34 8 16 /abwechselnd jede Sekunde (modul 12) wird an einen PCF8574/ Subadresse 2 (I2C Adresse 0100010b = 34d) eine 8 bzw. 16 gesendet,.

Dieser Modul dient der Ansteuerung fast beliebiger I2C Hardware, die normalerweise vom Programm nicht unterstützt wird. Mit entsprechender zusätzlicher Hardware können die übertragenen „Event-" Bytes in statische Signale umgewandelt oder weitergegeben werden (ein typischer Fall ist die Adressierung von Funksteckdosen).
Die Übertragung erfolgt im (Highspeed-)Zeitintervall unter Umgehung der normalen Bussequenz. Mit diesem Modul ist es theoretisch möglich, bis zu 256 Adressen * 256 verschiedene Werte/2 = 32768 verschiedene Ausgangskanäle anzusteuern.

Einschränkungen: Die Kanäle können nur über dieses SPS Modul geschaltet, nicht aber gelesen werden. Die Übertragung mehrerer „gleichzeitiger" Befehle aus der SPS heraus (beim Einsatz mehrerer dieser Module), die „zufällig" zur gleichen Zeit schalten, erfolgt sequenziell mit einer Rate von maximal 9 Befehlen je Sekunde. Die SPS Module werden dabei aufsteigend nach ihrer Nummerierung abgearbeitet. Die zu übertragenen Bytes werden nach dem Prinzip „fire and forget" auf den Bus gegeben, eine Fehlerüberwachung erfolgt nicht.

Ausgabe Analogwert an Analogausgang PCF8591 bzw. PWMWert an ATtiny26.



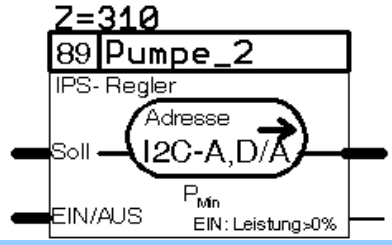

```
D/A Ausgang   =  Analogwert f von Modul x (mit automatischer Skalierung)
f             =  physikalischer Wert D/A Ausgang (bei linearer Kennlinie)
              =  f von Modul x (bei nicht linearer Kennlinie)
Zustand       =  EIN, wenn Analogwert > 0, sonst AUS
```

```
z=.. 76 .._Adresse_[Modul x]
            0..7    0..1199
              2        4
```

```
Sensor= Test     grd    1  0   90    -1.0    0    +1   255   /Der Analogeingang Adresse 1 Kanal 0 wird  auf -1..+1 [grd] skaliert.
Analog= A/D      qgrd   0    90    -1    0    1    255     /Der Analogausgang an der Adresse 1 wird für den Bereich -1 ..+1 [qgrd] festgelegt.
z=239  78  A/D0       1   0   0                           /Die SPS ermittelt den momentanen Wert des Analogeinganges Adresse 1 Kanal 0
z=240  76  analogtest  1  239                             /An den Analogausgang Adresse 1 wird der ermittelte Analogwert wieder ausgegeben
   - an dem entsprechenden Ausgang wird praktisch der gleiche Wert ausgegeben, wie er am Eingang 0 anliegt, allerdings mit der
   entsprechenden Bus- Verzögerung von einigen Sekunden.
Anzahl_PCF8591=2              /mit diesem Befehl in der hardware.ini (!) kann die Busreaktionszeit stark verkürzt werden...
```

Anmerkungen: Die Maßeinheit wird automatisch aus der Definition des Sensors (Analog=...) übernommen. Seit Version 6.0.37 wird auch eine nicht lineare Ausgangs-Kennlinie, die in der Definition (Analog=..) vorgegeben wurde, unterstützt. Bei Signalausfall am Modul x wird der analoge Ausgang am PCF8591 auf null und der PWM Ausgang des ATtiny26 auf AUS gesetzt. Für die Funktion des analogen Ausganges muss mindestens ein analoger Sensor der entsprechenden Adresse initialisiert sein (Analog=..)

IPS-Steuerung / Pumpenregelung einer nicht elektronischen Pumpe (Ä: ab Version 6.0.55) per Analogausgang PCF8591 mit PWM-Regler TL494 bzw. ATtiny26




```
D/A Ausgang   =  f{Sollwert, aktueller Leistung, Pmin}
Zustand       =  EIN, wenn Sollwert > 0, sonst AUS
f             =  Sollwert [%]
z=.. 89 .._[Modul Sollwert]_[Modul EIN/AUS]_ Pmin _ Adresse
              0..1199         -1199..1199    1..100   0..7
                2                  5            3       4
```

Analog=Pumpe_Hzg % 6 0 0 0 31 100 252 100 252 /Der Analogausgang an der Adresse 6 wird für den Bereich 0 ..+100 [%] festgelegt. Die entsprechenden Kodewerte für AUS sind 31 und für EIN 252

z=310 89 Pumpe_P2 309 308 25 2 /Der IPS Regler steuert über entsprechende Hardware die Leistung der Pumpe P2 an der Analog-Adresse 2 in Abhängigkeit vom Sollwert – übergeben durch Modul 309. Die minimale Leistung im Betrieb beträgt 25%. Modul 308 liefert das Freigabesignal zum Einschalten der Pumpe

Zur Funktion:

Die Impulspaketsteuerung (IPS) verwandelt eine herkömmliche an/aus Pumpe in eine elektronisch geregelte. In einem festen Zeitintervall (ca. 0,8s) wird das Verhältnis von Ein- und Aus-Zustand proportional zum Sollwert gesteuert.

Der IPS_Regler benötigt spezielle Hardware (beschrieben im Abschnitt 2.6 [3] bzw. beim 8- Kanal Sensor Interface [2]; alternativ auch mit Attiny26/ [1]). Der Sollwert wird über den entsprechenden Analogausgang direkt an die Pumpe weitergegeben. Über den Wert Pmin wird die minimale Leistung der Pumpe im Betrieb vorgegeben; das Modul EIN/AUS liefert das Freigabesignal zum Betrieb. Die Kennlinie des PWM Reglers wird mit der Kalibrierung „Analog=...“ eingestellt. Im Zweifelsfall sind die Kodewerte für „EIN“: 255 und für „AUS“: 0.

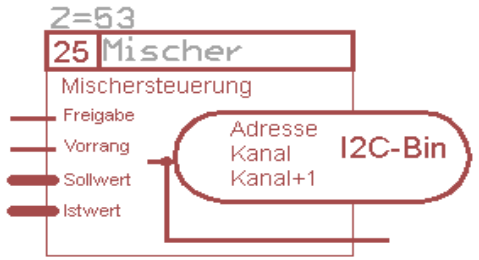
- Sollwert = 0 % ODER SPS ausgefallen: Pumpe sofort AUS
- Sollwert ≥ 95 % : Pumpe dauernd EIN (100 %)
- Pmin ≥ Sollwert > 0 % : Pumpenleistung = Pmin
- Pmin < Sollwert < 95 % : Pumpenleistung = Sollwert
- Kickstartfunktion 1: ist die Pumpe länger als 10 h AUS, wird nach Anlegen eines Sollwertes > 0 % die Pumpe für ca. 50 s auf 100 % gefahren.
- Kickstartfunktion 2: ist die Pumpe länger als 72 s AUS, wird nach Anlegen eines Sollwertes > 0 % und kleiner 70 % die Pumpe für ca. 13 s auf 100 % gefahren.
- Wird der SPS-Modul in den Handbetrieb versetzt (in der SPS-Übersicht per <Enter>, oder per Softkey vom Typ 1), so erfolgt eine korrekte Übertragung des „Hand-“ Sollwertes an die Pumpe: Sollwert=0% bei „Hand-Aus“ bzw. Sollwert=100% bei „Hand-Ein“ (Ä. Vers. 6.0.55)

2 Binärkanäle für einen (Heizungs-) Mischer über Hausbus schalten.

```
Kanal „HIN“   : Binäradresse/Kanal+1  (Temperatur wird vermindert)
Kanal „HER“   : Binäradresse/Kanal    (Temperatur wird erhöht)
Zyklus I            : 3s EIN, 21s AUS (Bezug: interner Modul 44)
Zyklus II           : 3s EIN,  9s AUS (Bezug: interner Modul 45)

wenn Freigabe = EIN und Vorrang = AUS dann:
  HIN ist zyklisch I  EIN, wenn Istwert > (Sollwert + 2,5)
  HER ist zyklisch I  EIN, wenn Istwert < (Sollwert - 2,5)
  HIN ist zyklisch II EIN, wenn Istwert > (Sollwert + 8)
  HER ist zyklisch II EIN, wenn Istwert < (Sollwert - 8)
  sonst AUS
wenn Freigabe = EIN und Vorrang = EIN dann:
  HIN = AUS, HER = EIN
wenn Freigabe = AUS dann:
  HIN = AUS, HER = AUS

Zustand   = EIN, wenn HIN = EIN oder HER = EIN  (Mischer fährt)
```



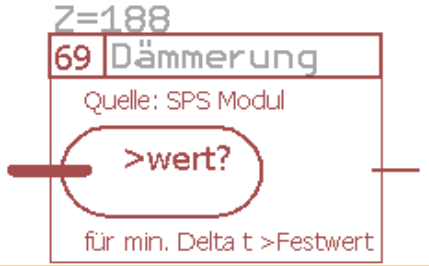
Es werden über zwei Binärkanäle (2 aufeinander folgende Kanäle der gleichen Adresse)die beiden Richtungen eines Heizungs-Mischers (in diesem Modul HIN und HER bezeichnet) angesprochen. Die Differenz zwischen Soll- und Istwert führt zum Nachfahren des Mischers. Um eine präzise Steuerung über die SPS zu gewährleisten, werden die beiden Kanäle getaktet – ist die Temperaturabweichung klein, so fährt der Mischer im Intervall von 24 s für 3 s, ist die Temperaturabweichung dagegen sehr groß, so fährt der Mischer zusätzlich schneller (Zyklus II). Die Unempfindlichkeit (Mischer steht) beträgt ± 2,5. Sollwert sowie Istwert werden als Temperaturen interpretiert. Hinweise zur praktischen Verwendung: der Temperaturfühler sollte so nah wie möglich am Mischerabgang platziert (<30cm) und nach außen isoliert sein. Im Test kam ein Mischer mit Aufsatzmotor der Fa. Meibes (66341) mit einer Stellgeschwindigkeit von 140 s (über den gesamten Stellbereich) zum Einsatz.

z=.. 25 .._[Freigabe]_[Vorrang]_[Sollwert]_Adresse_[Istwert]_KanalHER						
	-1199..1199	**-1199..1199**	**-1199..1199**	**0..7**	**-1199..1199**	**0..6**
	2	**3**	**4**	**5**	**6**	**7**
Binär=HIN 4 2 1						***/Binärkanal für Mischer „HIN“ definieren***
Binär=HER 4 3 1						***/Binärkanal für Mischer „HER“ definieren***
z=52 70 Sollwert °C +55.0						***/Vorgabe Sollwert +55 °C – in diesem Beispiel nur als Konstante vorgegeben***
z=53 25 Mischer 1 0 52 4 54 2						***/Mischer mit Binäradresse 4 Kanal 2 und 3 (2+1), Dauerfreigabe, kein Vorrang***
z=54 78 Istwert 5 1 0						***/Istwert als aktueller Wert von Analogadresse 5 Kanal 1 (Temperaturfühler!)***

4.7.5. Analoge Module

Diese Module verwenden in erster Linie analoge bzw. physikalische Werte inklusive der zugehörigen Maßeinheit um letztendlich realistische Steuer- und Regelungen zu realisieren.

SPS Modul größer Festwert für delta t.




```
Zustand = EIN, wenn Analogwert von Modul X für min. delta t [s] größer
          als Festwert, sonst AUS
f       = Festwert
e       = Maßeinheit
```

```
z=..  69  .._Einheit_[Modul x]_Festwert_t
          String[10] 0..1199     real    0.. 2000000000s
                8         2         6      7
```

z=188 69 Dämmerung ° 41 -4 30 /EIN, wenn Sonne länger als 30 s höher als -4° steht

SPS Modul kleiner Festwert für delta t.




```
Zustand = EIN, wenn Analogwert von Modul X für min. delta t [s] kleiner als
          Festwert, sonst AUS
f       = Festwert
e       = Maßeinheit
```

```
z=.. -69  .._Einheit_[Modul x]_Festwert_t
          String[10] 0..1199     real    0.. 2000000000s
                8         2         6      7
```

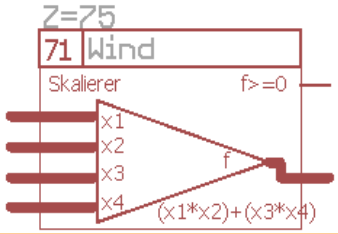
z=189 -69 duster % 40 35 30 /EIN, wenn Helligkeit 30s lang < 35% beträgt

Konstantenmodul, Vorgabe eines analogen Festwertes, einschließlich Maßeinheit.



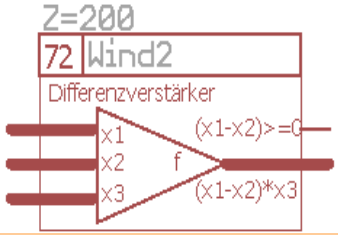

```
f       =  Analogwert
e       =  Maßeinheit
z=..  70  .._Einheit_Analogwert
          String[10]   real
                8        6
z=82  70  t_diff    °C     6     /Differenztemperatur von 6°C vorgegeben
```

Addierer/Multiplizierer/Skalierer. (wichtigste analoge Verknüpfung)




```
f       = x1*x2 + x3*x4
Zustand = AUS, wenn f<0;
          EIN, wenn f>=0.
e       = Maßeinheit
z=..  71  .._Einheit_[Modul x1]_[Modul x2]_[Modul x3]_[Modul x4]
           String[10]   2-5:   0..1199
                8          2           3          4          5
z= 75  71  Wind    km/h  74  13  0   0   /Umrechnung des Analogwertes von Modul 74 von m/s in km/h
```

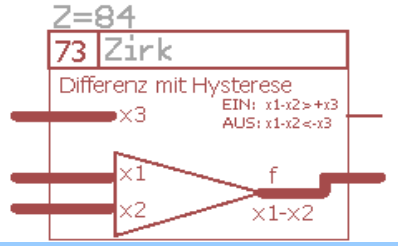
Differenzverstärker/ P-Regler. (entspricht einem analogen Operationsverstärker)




```
f       = (x1-x2)*x3
Zustand = AUS, wenn (x1-x2) <0;
          EIN, wenn (x1-x2)>=0.
e       = Maßeinheit
z=..  72  .._Einheit_[Modul x1]_[Modul x2]_[Modul x3]
           String[10]   2-4:    0..1199
                8          2          3          4
z= 200  72  Wind2  km/h     74     0     13     /Umrechnung des Analogwertes von Modul 74 von m/s in km/h
z= 231  72  Regler  %       230    231   8      /Verstärkung des Differenzwertes der Module 230 und 231 mit dem Faktor 100.
```

Differenz mit Hysteresis bzw. Zweipunktregler.

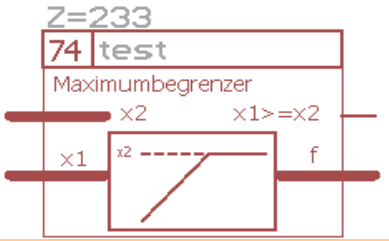
```
f       = (x1-x2)
Zustand = AUS, wenn (x1-x2) <-x3;
          EIN, wenn (x1-x2)≥ +x3.
e       = Maßeinheit
z=..  73  .._Einheit_[Modul x1]_[Modul x2]_[Modul x3]
           String[10]    2-4:   0..1199
                 8           2           3           4
```



z=84 73 Zirk_EIN °C 83 80 0 /2-Punkt Regler ohne Hysteresis. Angenommen, Modul 83 liefert den Sollwert und Modul 80 den Istwert. Dann ist der Ausgangszustand von Modul 84 auf EIN, wenn der Sollwert über dem Istwert liegt.

Maximumbegrenzer.

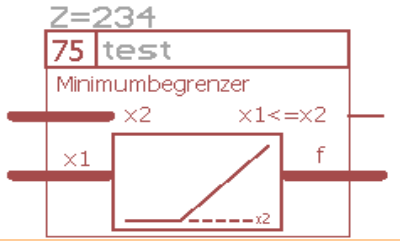
```
f       = x2, wenn x1≥x2;
        = x1, wenn x1< x2.
Zustand = AUS, wenn (x1< x2);
          EIN, wenn (x1≥ x2).
e       = Maßeinheit
z=..  74  .._Einheit_[Modul x1]_[Modul x2]
           String[10]    2-3:    0..1199
                 8           2           3
```



z=233 74 Test V 18 3 /Das Analogsignal von Modul 233 ändert sich wie das Dreiecksignal von Modul 18, allerdings nur bis maximal 0,5 V

Minimumbegrenzer.

```
f       = x2, wenn x1<=x2;
        = x1, wenn x1> x2.
Zustand = AUS, wenn (x1> x2);
          EIN, wenn (x1<=x2).
e       = Maßeinheit
z=..  75  .._Einheit_[Modul x1]_[Modul x2]
           String[10]    2-3:      0..1199
                 8           2           3
```



z=234 75 Test V 18 0 /Das Analogsignal von Modul 233 ändert sich wie das Dreiecksignal von Modul 18, allerdings nur bis minimal 0,0 V

Schwingungspaketsteuerung/ Pulsweitenmodulation (PWM)/ einstellbares Tastverhältnis, feste Zykluszeit. (sehr praktisch!)



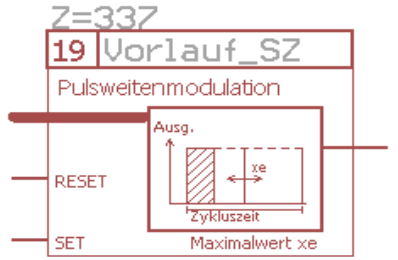
Diese Modultypen dienen der Ansteuerung eines schaltbaren Stellgliedes (z. B. Heizungsventile EIN/AUS schalten) durch einen linearen Regler.

Zustand = Das Tastverhältnis des Ausgangszustandes EIN/AUS ist proportional dem Analogwert von Modul X.
= AUS, bei f(Modul x)<= 0
= EIN, bei f(Modul x)>= MaximalwertXe
→ Bei f(Modul x)= MaximalwertXe/2 ist also der Ausgangszustand für die Hälfte der Zykluszeit EIN, für den Rest aus.

zusätzlich bei Typ 19:

Zustand = Bei Modul Reset= EIN → ist der Ausgangszustand ständig AUS
= Bei Modul Set = EIN → ist der Ausgangszustand ständig EIN (Reset hat aber Vorrang)

f = Zykluszeit (= const.) (nur Typ 77)
f = aktuelles PWM Tastverhältnis [%] (nur Typ 19)



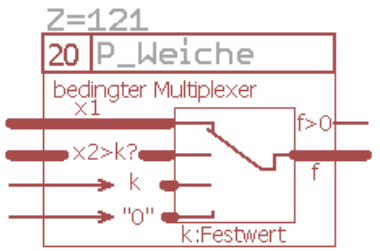

```
z=.. 77 .._[Modul x1]_MaximalwertXe_Zykluszeit
             0..1199      REAL        longint
                2          6           7
z=.. 19 .._[Modul x1]_[Modul RESET]_[Modul SET]_MaximalwertXe_Zykluszeit
             0..1199   -1199..1199    -1199..1199    Integer     longint
                2          3              4             5           7
```

```
Sensor= Test W  1  0  90   0   0   +1000   255  /Der Analogeingang Adresse 1 Kanal 0 wird  zu Testzwecken auf 0..1000 [W] skaliert.
z=240  78  Heizleist.    1      0      0       /Die SPS ermittelt den momentanen Wert des Analogeinganges Adresse 1 Kanal 0
z=241  77  Heizplatte   240    1000   18      /Die Leistung vom Imitationseingang (Adr.1 Kan.0) wird in ein Tastverhältnis umgewandelt
                                               und in der SPS im Zustand des Moduls 241  angezeigt
z=242  30  Kocher       241    3      7       /Der getastete Ausgang von Modul 242 wird auf den Binärkanal Adresse 3 Kanal 7 ausgegeben. Das dort angeschlossene Gerät lässt sich dann durch Variation am Analogeingang (1,0) im Tastverhältnis steuern.
```

Anmerkung: Die Zykluszeit wird nach unten begrenzt durch die Ausgabezeit an einen Binärkanal. Nach oben erhöht sich die Genauigkeit, es verschlechtert sich aber die Geschwindigkeit einer Regelung. Dadurch ergeben sich sinnvolle Zeiten zwischen 10s und 400s. Empfohlen wird 18s bei einer elektrischen Heizung wie z. B. einem Wasserkocher. Für Regelstrecken mit großen Zeitkonstanten (z. B. Heizung eines Hauses) kann die Zykluszeit auch entsprechend höher liegen.

Bedingter Multiplexer

```
f       =  x1, außer wenn:
        =  x2, wenn x2>x1 UND x2>k
           (d.h. der größere Wert wird aufgeschaltet)
        =  k,  wenn x1,x2 ungültig (>1e6, <0) ODER Modul_Festwert=EIN
        =  0,  wenn Modul_0 „EIN“ -Vorrang-
Zustand = EIN, wenn f > 0
Anmerkung: x1, x2, k sind gültige Werte ≥=0
```

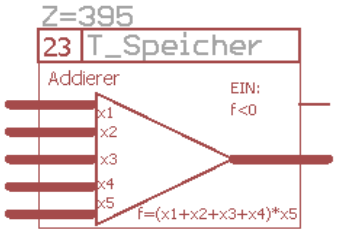


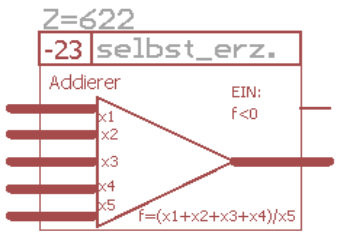

```
z=.. 20 .._Einheit_[x1]_[x2]_[Modul Festwert]_[Modul „0“]_Festwert
          String[10]  0..1199    -1199..1199  -1199..1199  0..2*10^9
               8      2     3          4                5         7
Z=121  20  P_Weiche  %     0     120   117   0     25    /Modul 120 errechnet die Soll-Pumpenleistung, Modul 117 liefert ein
                                                         Spülsignal, Die Leistung wird dabei auf mindestens 25% gesetzt; x1
                                                         wird nicht verwendet
z=122  89  Pumpe_HL       121   1     35    2           /Ansteuerung der Pumpe
```

Dieser Modul ist als praktische Ergänzung zum IPS Modul 89, also speziell für Pumpen- und Lüfterregelungen vorgesehen.

Addierer/Multiplizierer/Skalierer/Dividierer.

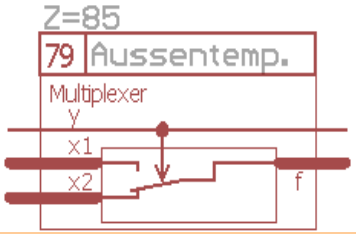
```
f       = (x1+x2+x3+x4)*x5      (Modultyp  23)
f       = (x1+x2+x3+x4)/x5      (Modultyp -23)
Zustand = AUS, wenn f<0;
          EIN, wenn f>=0.
e       = Maßeinheit
```






```
z=.. {-}23 .._Einheit_[Modul x1]_[Modul x2]_[Modul x3]_[Modul x4]_[Modul x5]
            String[10]   2-5,7:  -1199..1199
                 8         2          3          4          5          7
z= 394   70 Konstante  °  C  0.25                    /Vorgabe Konstante
z= 395   23 T_Speicher  °C  344 391 392 393 394     /Berechnung Durchschnittstemperatur Speicher, T1..T4 und Faktor 0.25
```

- Multiplexer, **Analogmultiplexer, Signalumschalter.**




```
f       = f(x1), wenn Zustand y = EIN;
        = f(x2), wenn Zustand y = AUS;
Zustand = Zustand von y
```

```
z=.. 79 .._Einheit_[Modul x1]_[Modulx2]_[Modul y]
          String[10] 0..1199   0..1199 -1199..1199
               8          2         3           4
```

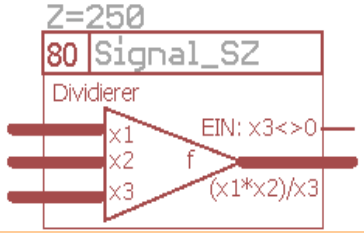
Beispiel 1:

```
z=242  79  testm       Hz     7     25     12          /am Ausgang von Modul 242 erscheint abwechselnd +10 und -10
```

Beispiel 2:

```
z=83 78 Aussen1        1      2     0                 /Außentemperatur 1 [°C] von Adresse 1 Kanal 2
z=84 78 Aussen2        1      3     0                 /Außentemperatur 2 [°C] von Adresse 1 Kanal 3
z=85 79 Aussentemp.    °C     83    84    86          /Bereitstellung der Außentemperatur 1 oder 2 in Abhängigkeit von einem Schalter an
                                                       Binärkanal 4 Adresse 3
z=86  2  Schalter      4      3     0                 /lesen Schalterzustand von Binäradresse 4 Binärkanal 3
```

- **Multiplizierer/ Dividierer.**



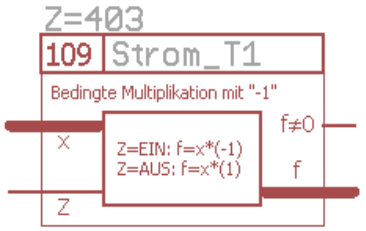

```
f       = (x1*x2)/x3      (Modultyp 80)
        = 100*(x1-x2)/x3  (Modultyp -80)
Zustand = EIN, wenn (x3<>0);
          AUS, wenn (x3= 0).
e       = Maßeinheit
```

```
z=.. {-}80 .._Einheit_[Modul x1]_[Modul x2]_[Modul x3]
             String[10] 0..1199   0..1199    0..1199
                  8         2          3          4
```

```
z=250  80  Signal_SZ   -     18    7     2  /Generierung Sägezahnkurve aus Dreiecksignal von Modul 18 für den Bereich -5 bis +5
```

Bedingte Multiplikation mit „-1“.

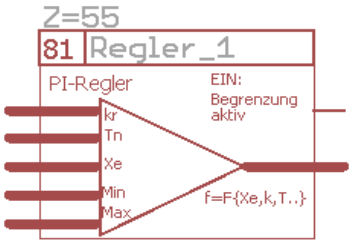
```
f       = x, wenn Z=aus;
        = -x, wenn Z=ein.
Zustand = EIN, wenn (x≠0, sonst AUS).
e       = Maßeinheit
```




```
z=.. 109 .._Einheit_[Modul Z]_[Modul x]
         String[10] -1199..1199 -1199..1199
                  8          2           3
```

```
SensorAt26=I10  A  1   0      2      0      0      2.49   1023   /Treiber für Stromsensor mit 5x1N5817 [A] Adr 1 Kan 0
SensorAt26=I11  A  1   1      2      0      0      2.49   1023   /Treiber für Stromsensor mit 5x1N5817 [A] Adr 1 Kan 1
SensorAt26=Iph  °  1   3      0      0      0      180    836    /Treiber für Phasensensor  [°] Adr 1 Kan 3 (Phase zwischen I10 und I11)
z=401  78  Strom_T1_roh      1      0      0                     /Strom Adr 1 Kan 0 lesen (Absolutbetrag)
z=402   3  Richtung     1      3      0      90     0             /Aus Phasendifferenz (Adr 1 Kan 3) Richtung ermitteln (<90°/>90°)
z=403 109 Strom_T1    A      -402   401                          /Aus Strombetrag und Richtung korrekten Strom ermitteln (+/-)
z=404 108 Leistung_T1 kW     403    244    100    2000          /Aus Strom und Spannung Leistung schätzen
```

PI- Regler. (Das Highlight der Regelungstechnik!)



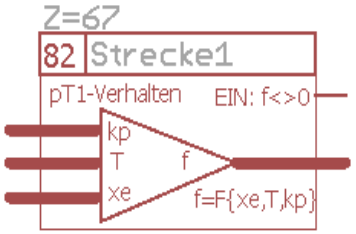
f = Funktion von(e,Tn,kr,MIN,MAX), $f = -k_r[Xe + \frac{1}{Tn}\int Xe\,dt]$

```
              Xe  - Eingangssignal/Regeldifferenz/Regelabweichung
              Tn - Nachlaufzeit
              kr - Verstärkungsfaktor
              MIN- minimales Ausgangssignal des Reglers
              MAX- maximales Ausgangssignal des Reglers
              Wenn MAX<MIN, dann wird MAX intern auf MIN gesetzt.
Zustand = AUS, wenn (MIN<f<MAX);
          EIN, wenn Begrenzung f auf MIN bzw. MAX aktiv.
e       = Maßeinheit des Stellwertes/Ausgangssignal (z. B. %)
```

z=.. 81 .._Einheit_[Modul kr]_[Modul Tn]_[Modul Xe]_[Modul MIN]_[Modul MAX] String[10] 2-5,7: 0..1199								
			8	2	3	4	5	7
z=49	*70*	*kr*	*-*	*25.36*				*/Verstärkung kr vorgeben*
z=50	*70*	*Tn*	*s*	*175*				*/Nachlaufzeit Tn vorgeben*
z=51	*70*	*min*	*W*	*0*				*/Minimum für Reglerausgang vorgeben*
z=52	*70*	*max*	*W*	*2500*				*/Maximum für Reglerausgang vorgeben*
z=53	*73*	*Rgl_diff*		*°C*	*60*	*61*	*0*	*/Regeldifferenz ermitteln: von Modul 61 kommt der Istwert, von Modul 60 der Sollwert (in diesem Beispiel nicht weiter dargestellt)*
z=55	*81*	*Regler_1*	*W*	*49*	*50*	*53*	*51* *52*	*/PI-Regler. Der Ausgang von Modul 55 kann mit Typ 77 in ein PWM- Signal umgewandelt und an einen Binärkanal zum Schalten eines Stellgliedes (z. B. Heizplatte) ausgegeben werden, oder direkt an ein analog gesteuerten Stellantrieb ausgegeben werden.*

Anmerkung: Da die Zykluszeit der SPS nicht konstant ist und die Genauigkeit der Zeitintervallmessung beim PC gering ist, wurde ein Kompromiss geschlossen. Der proportionale Anteil des Reglers wird bei jedem Durchlauf errechnet, während der integrale Bestandteil nur alle 3 Sekunden aktualisiert wird. Die Genauigkeit des I-Anteils beträgt damit maximal 1/(18,2*3) * 100 % = 1,8 %. Damit dürften alle praktischen Anwendungen realisierbar sein. Weiterhin wurde auf die Integration eines D-Anteils (PID-Regler) verzichtet, da alle externen Quellen nur mit 8-10 Bit gewandelt werden und somit die Berechnung in Bezug auf das Zeitverhalten äußerst ungenau werden würde. Wer den PI-Regler nur als P-Regler nutzen will, gibt für [Modul Tn] eine null ein. Für die Nutzung wichtig: Das Ausgangssignal des Reglers ist umgekehrt proportional dem Eingangssignal, also so, wie es sich für einen vernünftigen Regler gehört!

pT1- Verhalten, Glättung.



f = **Funktion von(e,T,kp)**
xe – Eingangssignal
T – Zeitkonstante [s]
kp – Übertragungsfaktor

$$C_0 = \frac{k_p}{\frac{T_1}{T_s}+1} \quad , \quad C_1 = \frac{1}{1+\frac{T_s}{T_1}} \qquad x_{a(k)} = C_0 \cdot x_e + C_1 \cdot x_{a(k-1)}$$

Ts: Zykluszeit ≈ 3s

Zustand = EIN, wenn Eingang xe ungleich 0 ist.
e = Maßeinheit des Ausgangssignal (z. B. °C)

z=.. 82 .._Einheit_[Modul kp]_[Modul T]_[Modul xe]
String[10] 2-4: 0..1199
8 2 3 4

z=65	***70***	***kp***	***-***	***0.05***			***/kp vorgeben***
z=66	***70***	***T***	***s***	***54***			***/Verzögerungszeit vorgeben***
z=67	***82***	***Strecke1***	***°C***	***65***	***66***	***55***	***/Der analoge Ausgang dieses Moduls verzögert das Signal vom PI-Regler (Modul 55) zusammen mit dem obigen Beispiel vom PI-Regler erhält man bei Vollaussteuerung des Reglers (=2500W) nach unendlich langer Zeit: 2500*0.05= 125°C. Weiterführende Literatur z.B unter www.softwaredidaktik.de/new2_spicelab/index.htm***

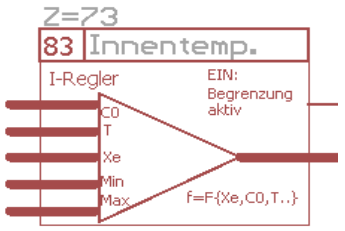
Anmerkung: Diese Funktion ist wichtig für die Simulation von praktischen Regelstrecken. So ähnelt der Temperaturverlauf in einem Haus der Kombination von pT-Verhalten und Totzeit. Damit kann man eine Heizungsregelung komplett mit diesem Programm simulieren und auf Herz und Nieren prüfen! Ähnlich wie beim PI-Regler wird die Berechnung des Analogwertes (Xa) nur alle 3s durchgeführt. Ich hoffe, dass dieser Wert in der Praxis ausreicht.

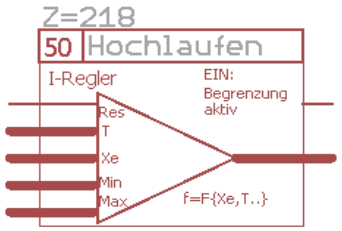
Integrator, Integralregler, I-Regler.

f = Funktion von(e,T,C₀,MIN,MAX), $f = \frac{1}{T}\int Xe\,dt + C_0$

Xe – Eingangssignal
MIN – minimales Ausgangssignal des Integrators
MAX – maximales Ausgangssignal des Integrators
f(83)=C0 bei Programmstart
f(85)=0, wenn Zustand vom angegebenen Resetmodul=Ein
f(50)=Xe, wenn Zustand vom angegebenen Resetmodul=Ein

T – Integrationszeitkonstante
C_0 – Startwert des Integrators





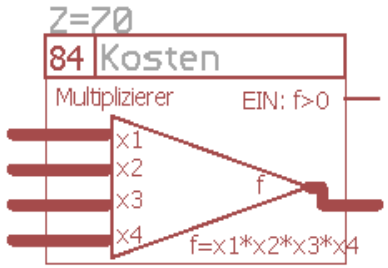
Zustand = AUS, wenn (MIN<f<MAX);
EIN, wenn Begrenzung f auf MIN bzw. MAX aktiv.
e = Maßeinheit des Ausgangssignal (z. B. %)

```
z=.. 83 .._Einheit_[Modul C0 ]_[Modul T]_[Modul Xe]_[Modul MIN]_[Modul MAX]
z=.. 85 .._Einheit_[Resetmod.]_[Modul T]_[Modul Xe]_[Modul MIN]_[Modul MAX]
z=.. 50 .._Einheit_[Resetmod.]_[Modul T]_[Modul Xe]_[Modul MIN]_[Modul MAX]
          String[10]   2-5,7: 0..1199
              8        2          3          4          5          7
```

Der Unterschied zwischen den Integratoren besteht in den Anfangsbedingungen. Während Typ 83 einmalig bei Programmstart auf einen Startwert C0 gesetzt wird und ab dann frei läuft, kann Typ 85 jederzeit auf „0" zurückgesetzt werden. Der Startwert beträgt ebenfalls „0". Typ 50 wird mit dem Resetmodul dagegen auf „Xe" gesetzt, der Startwert beträgt ebenfalls Xe. Die Integratoren werden aller 3 Sekunden aktualisiert.

```
/Simulation der Innentemperatur eines Hauses in Abhängigkeit von der Außentemperatur durch einen rückgekoppelten Integrator
z=69  70 Starttemp.     °C   20.0
z=71  70 Außentemp.     °C   -3.5
z=72  72 Energiefluß    kW   71    73   3                    /(Außen-Innentemperatur) * 0.5
z=73  83 Innentemp.     °C   69    8    72   25   8          /Simulation der Innentemperatur, Startwert:20°C (von Modul 69),
                                                             T=100s, Xe=Energiefluß (Modul 72), Min=-10°C (Modul 25),
                                                             Max=100°C (Modul 8)
z=74  70 T              §    3600                            /Umrechnungsfaktor von s in h
z=75  70 MAX            kWh  10000                           /Konstante für Maximalwert
z=76 85 Wärmemenge      kWh  42    74   72   0    75         /täglicher Wärmemengenzähler des Energieflusses
```

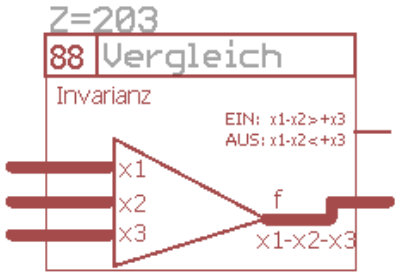
Multiplizierer 2.




```
f       = (x1*x2*x3*x4)   Zustand     = AUS, wenn (f<=0);  EIN, wenn f>0
e       = Maßeinheit
```

```
z=..  84  .._Einheit_[Modul x1]_[Modul x2]_[Modul x3]_[Modul x4]
          String[10]    2-3: 0..1199
                 8           2           3           4           5
 /Betriebskosten Tiefkühltruhe
z=65 70 Preis_kWh     EUR   0.20                                  /Energiepreis vorgeben
z=66 70 P_Truhe       kW    0.180                                 /Nennleistung der Tiefkühltruhe
z=67 -3 Truhe_EIN     2     2       3       0.0005      1       /EIN, wenn Temp. schneller als +0,0005°C je s fällt.
z=68 60 Tastung       67                                          /EIN/AUS Verhältnis Tiefkühltruhe ermitteln
z=69 70 Stunden       h     24                                    /Konstante 24 Stunden vorgeben
z=70 84 Kosten        EUR/d       65      66      68      69      /tägliche Betriebskosten Tiefkühltruhe Euro je Tag
```

Differenz mit Unempfindlichkeit. Keine Hysterese!



Praktische Verwendung: Ansteuerung von Heizungsmischern o. ä. Stellantrieben mit zwei Kanälen (je ein Kanal für jede Richtung). Alternative: SPS Modultyp 25.

```
f       = f=(x1-x2-x3)
Zustand = AUS, wenn (x1-x2) <+x3; → x1 ist (fast) gleich x2
          EIN, wenn (x1-x2)>=+x3.
e       = Maßeinheit
```

```
z=..  88  .._Einheit_[Modul x1]_[Modul x2]_[Modul x3]
          String[10]    2-4:  0..1199
                 8           2           3           4
z=58  78  VL_Haus     4     0       4                   /Temperatur am Mischer-Ausgang
z=202 70  Sollwert    °C    57.3                        /Solltemperatur am Mischer-Ausgang
z=203 88  Vergleich   °C    202     58      2           /Temperatur kleiner (Soll – 2°C)?  - kann einen Mischer (Heizung) auffahren
z=204 88  Vergleich   °C    58      202     2           /Temperatur größer (Soll + 2°C)?  - kann einen Mischer (Heizung) zufahren
```

4.7.6. Spezielle Module

Statistikmodul.

Statistische Analyse aller analogen Ausgänge (f) fortlaufender Module im Bereich (einschließlich) Modul x1 bis Modul xn. Es werden nur Werte im Bereich $-10^7..+10^7$ zur Berechnung verwendet (Gültigkeitskriterium).

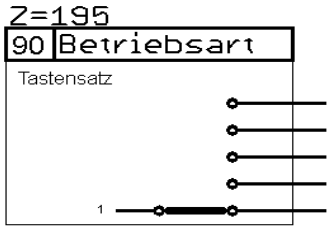
Typ	f	
1	Quersumme bzw. arithmetischer Mittelwert $\bar{x}_{arith}=\frac{1}{n}\sum_{i=1}^{n} x_i$	
2	maximaler Wert aus Folge f(x1) .. f(xn)	
3	minimaler Wert aus Folge f(x1) .. f(xn)	
4	(empirische) Varianz $\tilde{s}^2=\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2$	
5	Standardabweichung $\tilde{s}=+\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2}$	
6	Spannweite = x_{max}- x_{min}	

Zustand = ein, wenn f gültig, sonst aus.

```
z=.. 87 .._Einheit_[Modul x1]_[Modulxn]_Typ
        String[10] 0..1199   0..1199  1..6
             8          2         3       4
… {Kalibrierung der Sensoren Adresse 1 ist nicht dargestellt}
z=600 78 Aussen1      1      1      0      /Außentemperatur 1 [°C] von Adresse 1 Kanal 1
z=601 78 Aussen2      1      2      0      /Außentemperatur 2 [°C] von Adresse 1 Kanal 2
z=602 78 Aussen3      1      3      0      /Außentemperatur 2 [°C] von Adresse 1 Kanal 3
z=605 87 Aussen       °C     600    602    1       /Mittelwert der 3 Außentemperaturen
```

Neben der Erhöhung der Genauigkeit beim Einsatz mehrerer physikalischer Sensoren dient dieses Modul auch der Verbesserung der Zuverlässigkeit – ausgefallene Sensoren werden bei der Messung automatisch aussortiert.

Hilfsmodul für Tastensatz mit gegenseitiger Verriegelung für maximal 5 Tasten.

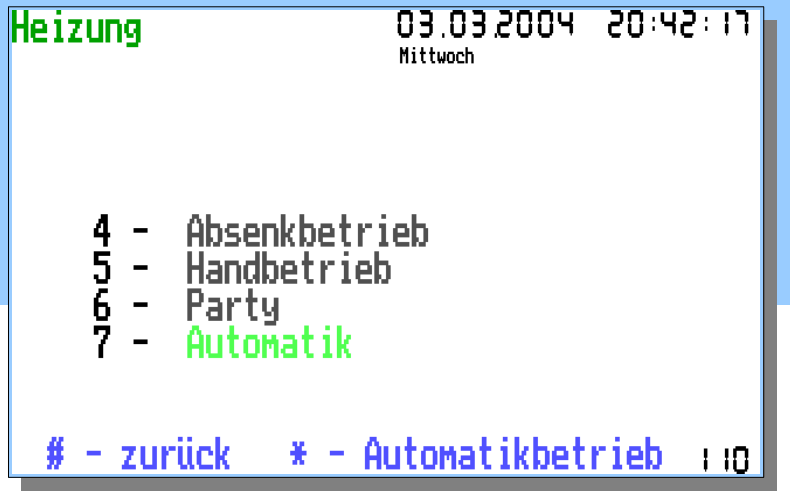


Die Module a-e müssen als Leermodul 91 vorgegeben sein.
Anfangszustand: Module b,c,d,e = AUS; Modul a = EIN;
Zustand: wenn ein Modul (a-e) von der Menüsteuerung oder vom SPS Modultyp 28 auf EIN gesetzt wird, schalten die anderen automatisch auf AUS. Die Ausgänge der Leermodule können für Schaltzwecke weiterverwendet werden.

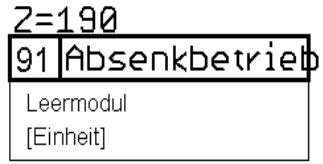
Der Schaltzustand wird in der Datei „gestern(2/3).daz" gespeichert, sodass er nach einem Programmneustart (z. B. nach Stromausfall) erhalten bleibt. Der Schaltzustand der (ebenfalls abgespeicherten) Module vom Typ 91 wird in diesem Fall überschrieben.

z=.. 90 .._[Modul a]_[Modul b]_[Modul c]_[Modul d]_[Modul e]
2-7: -1199..1199
2 3 4 5 7

/Beispiel: Betriebsartenumschaltung einer Heizungssteuerung
Menü11x= 110 Heizung /Diese Zeile programmiert im Anwendungsmenü eine Seite „Heizung"
Zeile=110 4 4 190 /Diese und die folgenden Zeilen erzeugen auf der Seite 110 („Heizung") 4 Tasten, die man betätigen kann
Zeile=110 5 4 191
Zeile=110 6 4 192
Zeile=110 7 4 193
z=190 91 Absenkbetrieb - /Die Module 190 bis 193 machen eigentlich gar nichts, außer den Text für die Tasten zu liefern
z=191 91 Handbetrieb -
z=192 91 Party -
z=193 91 Automatik -
z=195 90 Betriebsart 190 191 192 0 193 /Dieses Modul organisiert, dass immer nur (genau) eine Taste und damit der entsprechende Modul aktiv ist. Nicht benötigte Tasten werden auf 0 gesetzt.



Leermodul.



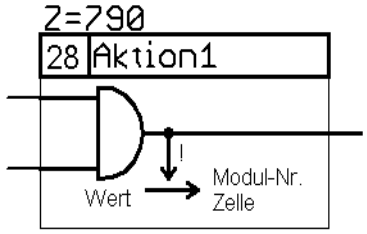
Zustand = wird von anderen Modulen oder Menütasten bestimmt, es werden nur die Bezeichnung und die Maßeinheit programmiert.
e = Maßeinheit
Hinweis: Bei Programmende wird der aktuelle Zustand (nicht ein eventuell vorhandener Wert) abgespeichert und bei Programmneustart wieder hergestellt.

```
z=.. 91 .._Einheit
         String[10]
              8
```

Beispiel: siehe Modul 90

Zuweisungs-Modul zum direkten Überschreiben von Speicherzellen in der SPS. Nur für Experten!

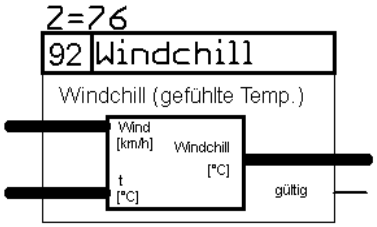
Dieses Modul schreibt beim Erreichen der Auslösebedingung einen programmierten Wert in eine beliebige Speicherzelle in der SPS. Damit können Konstanten, Zeiten, Bezugsmodule oder Modultypen gezielt verändert werden.
Zustand = EIN, wenn Z1 UND Z2 EIN sind, sonst AUS
Das Speichern des Wertes in das Zielmodul erfolgt nur beim Wechsel des Zustandes von AUS auf EIN, also impulsartig.
Speicherzelle: gemäß der formalen Parameterzuordnung unter Pkt. 4.7.1.




```
z=.. 28 .._[{-}Z1]_[{-}Z2]_Zielmodul_Speicherzelle_Wert
           -1199..1199    0..1199      0..7      real
              2      3        4           5        6
```

z=790 28 Aktion1 10 10 192 0 6 /immer freitags (10) wird der Tastensatz im Beispiel auf der vorangegangenen Seite auf Party geschaltet → in das SPS-Modul Nr. 192 wird in die Ausgabezelle (0) eine 6 geschrieben (ausgelöst, ein)

Windchill- Modul zur Ermittlung der gefühlten Temperatur.



f = spezielle Funktion zur Umrechnung von Windgeschwindigkeit[km/h] und Temperatur[°C] {und relativer Feuchte [%]}
Zustand = EIN, wenn f gültig (keine Über/Unterschreitung zulässiger Temperatur) sollte der Wind ungültig sein, so wird Wind=0 km/h angenommen
e = °C (wird automatisch gesetzt)

```
z=.. 92 .._[Modul Wind]_[Modul Temp.]_{[Modul Rel.Feuchte]}
           0..1199       0..1199          0..1199
              2             4                3
```

z=73 78 Temperatur 1 2 0 /Außentemperaturmessung [°C] von Adresse 1 Kanal 2
z=74 78 Wind 1 0 0 /Messung Windstärke [m/s] von Adresse 1 Kanal 0
z=75 71 Wind km/h 74 13 0 0 /Umrechnung m/s in km/h
z=76 92 Windchill 75 73 /Berechnung der gefühlten Temperatur [°C]
Anzeige=95 2 8 76 1 9 1 2 -20 +35 0 /Anzeige der gefühlten Temperatur bei „Hausparameter“

Beispiel: siehe Modul 90

Anmerkung: Unterhalb von +25°C wird die Windchill-Funktion auf der Basis von Sipple und Passel aktiv, oberhalb von +25°C wird ein vereinfachter Hitzeindex auf Basis der relativen Luftfeuchte errechnet:

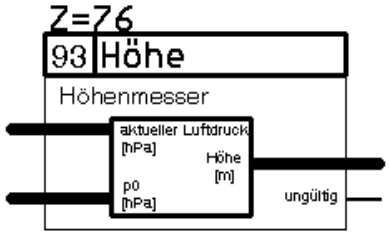
$T^* = -1{,}5 + 0{,}87*T + 0{,}085*F - 0{,}0065*T*F + 0{,}0003*T^2*F$

Dazu kann zusätzlich ein Bezugsmodul für die relative Luftfeuchte angegeben werden. Ist kein Modul vorhanden, so wird „40%“ als Festwert gesetzt.

Höhenmesser-Modul zur Ermittlung der geopotentiellen Höhe aus dem Luftdruck. (Flieger – Höhenmesser)

f = $H[m] = \frac{T_0}{\tau}\left(1 - \left(\frac{p_{st}}{p_0}\right)^{R}\right)$ (Hinweis: gültig nur bis 11000 m Höhe)

pst: aktueller Luftdruck [hPa], p0: Luftdruck H=0 [hPa], τ=0,0065°/m, R = 0,190295 (Standardatmosphäre 1976), T0=288,15 K
Ist p0=1013,25 hPa (QNH) wird die Standardhöhe ermittelt, ist p0=Platzdruck (QFE), so wird die relative Höhe über dem Platz ermittelt.

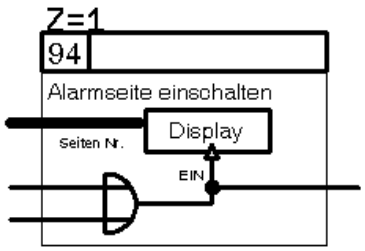


Zustand = EIN, wenn f ungültig
e = [m] (wird automatisch gesetzt)

z=.. 93 .._[Modul aktueller Luftdruck]_[Modul Luftdruck p0]
0..1199 0..1199
2 4

z=73 78 Luftdruck 1 2 0 /aktueller Luftdruck [hPa] von Adresse 1 Kanal 2
z=74 70 p0 hPa 1013.25 /Vorgabe Luftdruck p0
z=75 70 ft/m ft 3.28084 /Umrechnungsfaktor Meter in Fuß 1ft=0,3048m
z=76 93 Höhe_m 73 74 /Berechnung Höhe [m] aus Luftdruck [hPa]
z=77 72 Höhe_ft ft 76 0 75 /Berechnung Höhe [ft] aus Höhe [m]
Anzeige=95 2 8 77 1 9 1 2 0 +1000 /Anzeige der Höhe [ft]

Umschalten der Bildschirmseiten im automatischen Modus auf eine besondere Seite (Alarmseite), Umschalten auf eine besondere Bildschirmseite



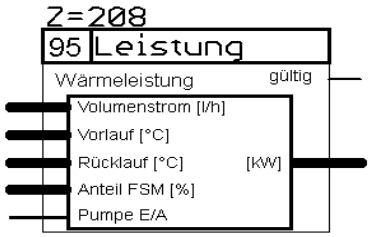

```
Zustand      = (Z1 ODER Z2);
```

Im automatischen Bildwechselmodus wird bis zur Betätigung durch die „ok"- Taste auf die vorgegebene Alarmseite umgeschaltet, wenn die Zustände von **Z1 ODER Z2** von AUS auf EIN schalten. Damit wird optisch das Erreichen einer bestimmten Situation signalisiert. Die Auslösung kann sowohl automatisch als auch von Hand erfolgen. Es können mit mehreren SPS-Modulen vom Typ 94 selbstverständlich auch verschiedene Alarmseiten programmiert werden, die bei unterschiedlichen Bedingungen ausgelöst werden. In den ersten 90s nach Programmstart, 1 min nach Tageswechsel sowie bei Ausfall der SPS inkl. 15s danach (z. B. bei negativer Zeit) ist die Funktion gesperrt, um Fehlauslösungen zu verhindern. Beim Typ (+)94 wird mit dem Umschalten auf die Alarmseite zugleich das Signal „Hinweis!" gesetzt, das Programm verbleibt im Automatikmodus, der Hinweis kann mit „OK" zurückgesetzt werden. Außerdem wird über die Datei alarm.txt eine e-mail ausgelöst (nur unter Linux). Der Scriptbefehl „-94" bewirkt dagegen ein sofortiges Umschalten in den Handmodus auf die entsprechende Seite, ohne das das Signal „Hinweis!" gesetzt wird oder eine e-mail ausgelöst wird. Ein Zurückschalten wie bei „94" mit „OK" ist nicht möglich, da man bereits im Menü ist. Die selbstständige Zurückschaltung in den Automatikmodus erfolgt nach 45 s.„+94" ist also für automatische Vorgänge und „-94" für Handbetätigungen, insbesondere für den Einsatz von Bustasten zum Blättern im Menü gedacht.

```
Z=.. {-}94 .._A-Seite_[{-}Z1]_[{-}Z2]
              0..255   -1199..1199
                 2       3       4
```

Beispiel 1
z=243 2 Taste_OK 0 0 3 /Taste_OK (Adresse 0, Kanal 0) >=3s gedrückt
z=244 -94 Astro 156 243 243 /Sternenkarte (S.156) aufrufen
Beispiel 2
...../Alarmanlage
z=228 12 Vorabalarm 227 202 222 1 /sofortiger Alarm
...../Alarmanlage
z=235 94 Seite111 111 228 228 /Bildschirmseite 111 wird eingeschaltet, wenn die Alarmanlage (Modul 228) auslöst

Wärmeleistungsmesser-Modul zur Ermittlung der Leistung für Klima/ Heizungstechnik.



f [kW] = Volumenstrom[l/h] * ($T_{vorlauf}$[°C]-$T_{Rücklauf}$[°C])*f{Frostschutzanteil[%]}

bzw. $P = V \cdot (T_v - T_r) \cdot \frac{4187}{3600000} \cdot (1 - 0{,}0009243 \cdot FSM)$

= **0**, wenn die Temperatur des Rücklaufs höher ist als die des Vorlaufs oder Pumpe E/A aus ist

Zustand = EIN, wenn f gültig (alle Sensoren intakt), sonst AUS

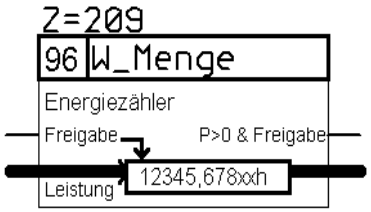
Der Volumenstrom wird in Liter je Stunde, der Frostschutzanteil in Prozent und die Temperaturen in °C vorgegeben.

```
z=..  95 .._Einheit_[Modul Volumenstrom]_[Modul TV]_[Modul TR]_[Modul FSM]_[Pumpe E/A]
         String[10]       0..1199          0..1199    0..1199    0..1199   -1199..1199
             8                 2                3          4          5          7
z=200 70  T1     °C   80                    /simulierte Vorlauftemperatur
z=201 70  T2     °C   30                    /simulierte Rücklauftemperatur
z=202 70  max    -    100000       /MAX
z=203 70  Vs     l/h  180                   /Volumenstrom
z=204 70  Fs     %    25                    /Frostschutzanteil
z=205 70  T      -    3600                  /Konstante
z=206 12  Pumpe  18 1 0 1          /simulierte Pumpe
z=207  6  15s    1  206 -206 15            /Verzögerung des Pumpensignals um 15s
z=208 95  Leistung kW  203 200 201 204 207      /Leistungsmesser
z=209 85  W_Menge  kWh  42 205 208  0 202      /Wärmemengenzähler (hier die Grundform als Integrator)
```

Bemerkung: Die Berechnungen beziehen sich auf eine Mischung von Wasser mit einem Frostschutzmittel. Als Wärmekapazität werden bei 0% : 4,187 KJ/KgK und bei 100 % : 3,800 KJ/KgK/ (ab Version 6.0.27) bzw. 4,100 (ältere Versionen vor 6.0.27) angenommen. Bei Verwendung von beispielsweise unverdünntem Tyfocor LS (3,85 KJ/KgK bei 85°C) als Solarflüssigkeit, ist somit als FSM 88 % vorzugeben.

Das Modul kann auch für andere Zwecke verwendet werden: z. B. zur Berechnung der Leistung eines Ölkessels bei bekanntem Öl-Volumenstrom [l/min]. Die Rücklauftemperatur und der Frostschutzanteil werden dabei auf „0" gesetzt und als Vorlauftemperatur der Faktor 479358 eingesetzt: ($479358 = \frac{3600000}{4187} \cdot 10{,}1 \cdot 60 \cdot 0{,}92$) Immer dann, wenn der Eingang „Pumpe E/A" EIN ist, erhält man als Ergebnis die Leistung in [kW] bei einem angenommenen Wirkungsgrad von ca. 92 % und einem Heizwert von 10,1 kWh/l. Der feuerungstechnische Wirkungsgrad ist dabei 100 % - Abgasverlust.

Energiezähler /Wärmemengenzähler /Tagesertragszähler /Betriebsstundenzähler mit Langzeitspeicherung. (geändert ab 6.0.35 und 6.0.52!)



f [Einheit+h] = Funktion von(Xe), $f = \frac{1}{3600}\int P\,dt$ **, wobei P ..**

{ 96, 97}: **P** = f(Xe), dh. die Leistung z. B. in [W],[kW] oder [J] wird direkt von Xe vorgegeben.

{-96,-97}: **P** = P_Nenn * Einschaltverhältnis f(Xe), wenn Modul Xe vom Typ 19,22,24,89 ist, andernfalls ist P = P_Nenn, wenn Modul Xe Ein ist, bzw. P = 0, bei Aus.

= const., wenn keine Freigabe

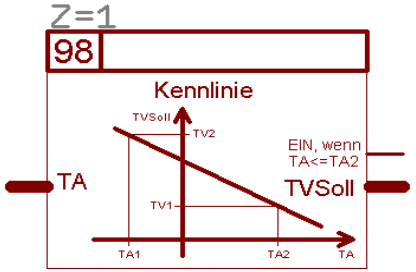
Zustand = AUS, wenn P<=0 oder keine Freigabe
= EIN, wenn P>0 und Freigabe erteilt

Einheit = die Einheit wird vom Modul Xe mit zusätzlich angehängtem „h" (96) bzw. „h/d" (97)übernommen, bzw. ist [kWh] (-96) oder [kWh/d] (-97). Ist in der Einheit des Moduls xe „/h" enthalten (z. B. Typ 86 - Impulsmessung), so wird „h" automatisch heraus gekürzt.

Bemerkungen: Die Energiezähler summieren die gemessenen Leistungen von Modul xe über die Zeit, solange das Freigabe-Modul „Ein" ist. Der ermittelter Wert wird bei Typ 96/-96 regelmäßig um 0:00 Uhr und bei Programmende gesichert und entsprechend bei Programmneustart wieder geladen. Der Wert geht also auch bei einem Systemabsturz nicht verloren. Typ 97/-97 ist als Tageszähler programmiert, er wird täglich um 0:00 Uhr auf null zurückgesetzt. Mittels -96,-97 kann man sich den Energieverbrauch von Pumpen (unter Angabe ihrer Nennleistung) direkt berechnen lassen. Die Energiezähler lassen sich in der SPS Übersicht mit der Taste „del" auf null stellen. Wenn das Eingangssignal negativ ist, so zählen die Zähler korrekt entsprechend rückwärts. Ist das Eingangsmodul vom Typ 86 (Impulsmessung), so erfolgt eine Aufsummierung des dort mit k angegebenen entsprechenden Volumens/Energiemenge. Die Zähler werden im Menüpunkt „lifing" automatisch gelistet.

```
z=.. 96   .._[Modul Xe]_[Modul Freigabe]              (Zähler für Gesamtertrag)
z=.. 97   .._[Modul Xe]_[Modul Freigabe]              (Zähler für Tagesertrag)
z=.. -96  .._[Modul Xe]_[Modul Freigabe]  _P-Nenn     (Zähler für Gesamtertrag)
z=.. -97  .._[Modul Xe]_[Modul Freigabe]  _P-Nenn     (Zähler für Tagesertrag)
               0..1199     -1199..1199     integer
                  2               3            4
z=208 95  Leistung kW   203 200 201 204 207 /dieses Modul berechnet laufend eine Leistung
z=209 96  W_Menge       208   1                 /Energiezähler - ermittelt die aufgelaufene Wärmemenge in kWh
z=210 97  Tagesertrag   208   1                 /Tagesertragszähler, kann für Anzeigen verwendet werden
z=378 89  Pumpe_Hzg 377 -332 15 6           /Ansteuerung einer Heizungspumpe mit Wellenpaket (getaktet)
z=470 -97 Pumpe_Hzg 378   1 90              /Messung Energieverbrauch Pumpe [KWh/d] mit Nennleistung P=90W
```

Vorgabe einer linearen (Heizungs-) Kennlinie in Abhängigkeit von der Außentemperatur.



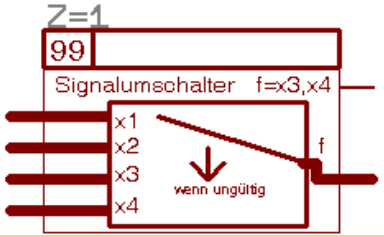
f = Funktion von(Modul TA) **TVSoll = -m * TA + n**
m =(TV2-TV1)/(TA2-TA1) n=TV1+m*TA2
Zustand = EIN, wenn TA<=TA2 ist.

Dieser Modultyp errechnet aus der Außentemperatur TA eine (Heizungs-) Vorlauftemperatur TvSoll in Abhängigkeit von Sollwerten. Die (Soll-) Wertepaare für die Heizkennlinie werden direkt im Scriptbefehl angegeben. Damit sind die Temperaturen nur als ganze Zahlen vorgebbar. Wird die Außentemperatur TA2 so gewählt, das sie der Heizungs-Abschalttemperatur (z. B. 20 °C) entspricht, so kann der schaltende Ausgangszustand des Moduls für die Freigabe bzw. Blockierung der Heizung genutzt werden. Das Kennlinienmodul ist natürlich auch für andere Verwendungen einsetzbar.

z=.. 98 .._Einheit_[Modul TA]____TA1____TV2____TA2____TV1
String[10] 0..1199 -255..+255
8 2 3 4 5 7

z=67 98 SollVorlauf °C 90 -15 +90 +20 +30 /Die Soll- Vorlauftemperatur wird errechnet aus der Außentemperatur im Modul 90

Signalumschalter/Notbetrieb.



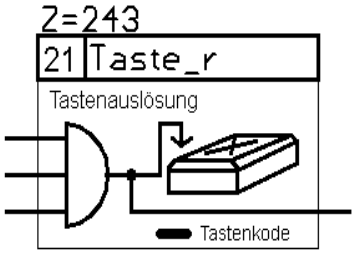
f = der erste gültige Wert der Reihe (x1, x2, x3, x4)
Zustand = AUS, wenn f=x1 oder f=x2;
= EIN, wenn f=x3 oder f=x4.
e = Maßeinheit
Das Modul übergibt den analogen Wert vom Modul x1 an den Ausgang, wenn der Wert gültig ist. Ist der Wert ungültig, so wird auf x2 (usw.) umgeschaltet.

z=.. 99 .._Einheit_[Modul x1]_[Modul x2]_[Modul x3]_[Modul x4]
String[10] 2-5: 0..1199
8 2 3 4 5

z=51 78 T_Auß_gest 1 1 20 /liefert die Tagesdurchschnitts- Außentemperatur (allerdings ungültig am Tag des Programmstartes)
z=52 78 T_Auß_aktu 1 1 4 /liefert die aktuelle Temperatur Außen (ungültig bei Sensorausfall)
z=53 99 Außentemp °C 51 51 52 7 /liefert immer eine Temperatur! - notfalls die Konstante +10 °C (vom Modul 7)
der Ausgangszustand wechselt von 0 auf 1, wenn T_Auß_gest nicht verfügbar: kann als Notbetriebssignal verwendet werden

Imitation eines Tastendrucks auf der Tastatur,

Zustand = (Z1 UND Z2 UND Z3);



Beim Wechsel des Zustandes von AUS auf EIN wird von der SPS der Tastenkode einmalig ausgeführt, so als wäre die entsprechende Taste auf der Tastatur betätigt worden. Damit kann die SPS das Programm *hauscomp.exe* selbst steuern. Das ist durchaus sinnvoll, wenn z. B. die PC-Tastatur überhaupt nicht vorhanden ist, oder aber der Rechner über Bustasten fern gesteuert werden soll. Denkbar ist auch das Abspeichern des aktuellen Zustandes der SPS (Taste „s")beim Auftreten bestimmter Zustände zur Fehlersuche. Das Update der *hauscomp.exe* über das Netzwerk erfordert anschließend ebenfalls einen Neustart des Programms. Hier ist die Taste „k" notwendig. Ist in der *autoexec.bat* der Aufruf der *hauscomp.exe* als Schleife geschrieben, so erfolgt ein automatischer Neustart, wenn über die Taste „k" das Programm beendet wurde.

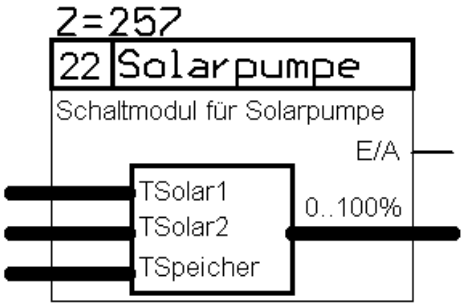
In den ersten 90 s nach Programmstart bzw. 3 min nach Tageswechsel ist die Funktion gesperrt, um Fehlauslösungen zu verhindern. Im Gegensatz zu der Verwendung von Bustasten, wo der Tastaturkode fest einem Binärkanal zugeordnet ist, ist hier die Verknüpfung innerhalb der SPS frei wählbar. Damit kann auch eine Mehrfachbelegung von Bustasten (z. B. nach Dauer des Tastendrucks) realisiert werden. *Hinweis:* Alle verwendbaren Tastaturkodes sind in 4.7.9 gelistet.

```
Z=.. 21 .._Tastenkode_[{-}Z1]_[{-}Z2]_[{-}Z3]
            0..255          -1199..1199
              2             3      4        5
```

Taste	*Tastenkode*	*Bedeutung*
„i“	105	Farbe (Bildschirm) invertieren
„k“	107	Programmende
„o“	111	Reset Hinweistext (Laufschrift)
„p“	112	Screenshot aktueller Seite in *.bmp
„r“	114	Erneutes Laden der *.ini Dateien (Restart)
„s“	115	Speichern des aktuellen Zustand der SPS in „sps.txt“

Beispiel :
z=242 2 Schalter1 3 2 15 /EIN, wenn Binärkanal Adr 3, Kan 2 (Schalter 1) für mindestens 15 s betätigt ist
z=243 21 Taste_r 114 242 1 1 /Auslösen Taste „r“ - Restart des Programms auf einem Steuerrechner ohne Tastatur mittels Betätigung der Taste „Schalter1“ (z. B. nach Korrekturen in der hauscomp.ini über das Netzwerk)

Schaltmodul für Solarpumpen.



Tsol: Temperatur am Solarkollektor, TSP: Temperatur Solarspeicher

f = (Tsol – Tsp) * 5 [%] d.h., bei 20 grd Differenztemperatur zwischen Kollektor und Speicher beträgt f = 100 %
Zustand = EIN, wenn (Tsol-Tsp)>= +Hysterese + Unempfindlichkeit
Zustand = AUS, wenn (Tsol-Tsp)< -Hysterese + Unempfindlichkeit,
wobei Tsol der größere Wert von Tsol1 bzw. Tsol2 ist.
Zustand = EIN und **f=30%**, wenn (Tsol1 oder Tsol2)< „Frost" (Frostschutz)
Zustand = EIN und **f=30%**, wenn Tsp > 87°C (Rückkühlfunktion)

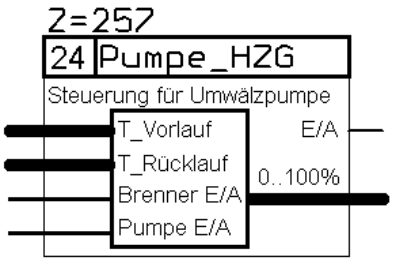
Tsol >= 85°C : Hysterese = 1,25 °C, Unempfindlichkeit = 4 °C.
Tsol < 85°C : Hysterese = 2,00 °C, Unempfindlichkeit = 8 °C.

Dieser Modultyp errechnet aus der Temperatur des Solarkollektors (Tsol) in Abhängigkeit von der Temperatur am Solarspeicher (Tsp) den Zeitpunkt zum Einschalten einer Solarpumpe (2-Punkt Regler) bzw. die Leistung für eine analoge oder PWM gesteuerten Solarpumpe. Der 2-Punktregler berücksichtigt dabei eine Hysterese und eine Unempfindlichkeit. Beide Werte sind von der Temperatur des Solarkollektors abhängig. Es können zwei Kollektorfühler angeschlossen werden, dabei wird der jeweils größere Wert verwendet. Der Speicherfühler muss die Temperatur im Pufferspeicher in der Wasserschicht unmittelbar über dem Solar-Wärmetauscher messen, andernfalls kommt es zu Fehlfunktionen der Pumpensteuerung. Der Wert „Frost" definiert die Temperatur, unterhalb derer die Frostschutzfunktion aktiviert wird. Bei gängiger Solarflüssigkeit sollte als Wert „-9" und bei reinem Wasser „+1" eingesetzt werden. Es sind nur ganze Zahlen möglich.

```
z=.. 22 .._[Modul Tsol1]_[Modul Tsol2]_[Modul Tsp]_Frost
             0..1199       0..1199      0..1199  -50..+10
                2             3            4        5
```

z=257 22 Solarpumpe 252 252 392 -8 /Modul 252 liefert die Temperatur des Kollektors, Modul 392 die der Einspeisestelle am Speicher, Frostschutz ab -8 °C

Schaltmodul für Umwälzpumpen.



f = (T_Vorlauf – T_Rücklauf) * 6 [%] (vor V.6.0.45: * 3,6)
aber mindestens 15 %, solange Pumpe E/A = EIN ist

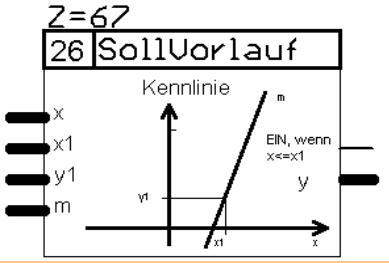
d.h., bei 10 grd Differenztemperatur zwischen Vor- und Rücklauf beträgt f = 60 %
Zustand = EIN, wenn f >= 15 % UND Pumpe E/A = EIN ist.

Dieser Modultyp errechnet aus der Vor- und Rücklauftemperatur in einem Heizkreis den Zeitpunkt zum Einschalten einer Umwälzpumpe (2-Punkt Regler) bzw. die Leistung für eine analoge oder PWM gesteuerten Pumpe. Ist das Signal „Brenner E/A" aktiv, so wird f fortlaufend (aller 3s um 5%) bis auf 100 % erhöht. Über das Signal „Pumpe E/A" kann der gesamte Modul ausgeschaltet werden. Ist die Rücklauftemperatur größer als die Vorlauftemperatur (z. B. bei einer Rücklaufanhebung durch Solarenergie) so beträgt f=15 %, zusätzlich wird die Pumpenleistung halbiert, indem während ungerader Minuten die Pumpe abgeschaltet wird.

```
z=.. 24 .._[Modul TVor]_[Modul TR]_[Brenner E/A]_[Pumpe_E/A]
                2             3            4             5
                    2..5: -1199..1199
```

z=257 24 Pumpe_HZG 255 256 365 333 /erzeugt das Schalt- bzw. Leistungssignal die für die Umwälzpumpe

Vorgabe einer linearen (Heizungs-) Kennlinie, parametrierbar (ab Version 6.0.30)

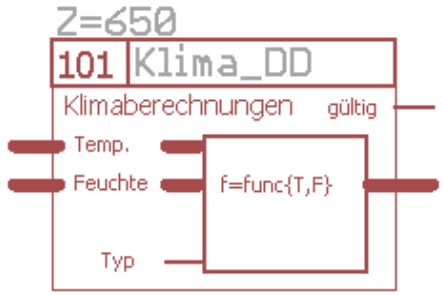


f = m(x-x1)+y1
Zustand = EIN, wenn x<=x1 ist.
Dieser Modultyp rechnet ähnlich SPS Modultyp 98, zusätzlich werden die notwendigen Konstanten nicht direkt angegeben, sondern aus anderen Modulen übernommen, sodass eine Kennlinienänderung zur Laufzeit möglich ist.

```
z=.. 26 .._Einheit_[Modul x]_[Modul x1]_[Modul y1]_[Modul m]
            8       2              3          4          5
         String[10]      2..5: -1199..1199
```

z=67 26 SollVorlauf °C 90 100 101 102 /Die Soll- Vorlauftemperatur wird errechnet aus der Außentemperatur im Modul 90 und den Konstanten in Modul 100..102

Klimaberechnungen / Taupunkt / Dampfdruck / Sättigungsdampfdruck / absolute Feuchte



Typ	1	2	3	4	5
f=	Dampfdruck	Sättigungs-dampfdruck	Temperatur Taupunkt	Absolute Feuchte	Maximale absolute Feuchte
Einheit=	„hPa“	„hPa“	„°C“	„g/m³“	„g/m³“

Zustand = EIN, wenn f gültig, sonst AUS

```
z=.. 101 .._[Temperatur]_[Luftfeuchte]_Typ
                0..1199        0..1199     1,2,3,4,5
                   2              3           4
```

```
z=650 101 Klima_DD      57 58 1   /Der Dampfdruck wird ermittelt (Temperatur aus SPS Modul 57, Luftfeuchte aus 58)
z=651 101 Klima_SDD     57 58 2   /Der Sättigungsdampfdruck wird ermittelt
z=652 101 Klima_TPkt    57 58 3   /Die Taupunkttemperatur wird ermittelt
z=653 101 Klima_AF      57 58 4   /Die aktuelle absolute Feuchte wird ermittelt
z=654 101 Klima_AFmax   57 58 5   /Die maximal mögliche absolute Feuchte (bei aktueller Temp.) wird ermittelt
```

Grundlage für die Berechnungen (Quelle: http://www.wetterochs.de/wetter/feuchte.html 22.02.2016, leicht modifiziert):

Bezeichnungen:

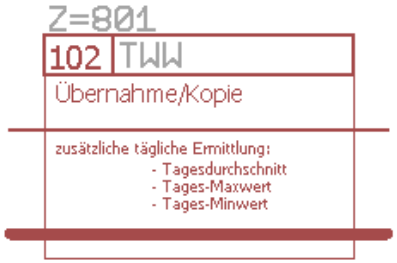
r = relative Luftfeuchte [%] T = Temperatur [°C] TK = Temperatur [K] (TK = T + 273.15)
DD = Dampfdruck [hPa] TD = Taupunkttemperatur [°C] SDD = Sättigungsdampfdruck [hPa] AF = absolute Feuchte [g/m³]
a = 7.5 b = 237.3 für T >= 0
a = 7.6 b = 240.7 für T < 0 über Wasser (Taupunkt)
R^* = 8314.3 J/(kmol*K) (universelle Gaskonstante) m_w = 18.016 kg/kmol (Molekulargewicht des Wasserdampfes)

Gültigkeit: -60°C < T < +110°C 0% < r < 100%

Formeln:

1. (Typ 2) SDD(T) = 6.1078 * 10^((a*T)/(b+T))
2. (Typ 1) DD(r,T) = r/100 * SDD(T)
3. r(T,TD) = 100 * SDD(TD) / SDD(T)
4. (Typ 3) TD(r,T) = b*v/(a-v) mit v(r,T) = log10(DD(r,T)/6.1078) DD ≠ 0
5. (Typ 4) AF(r,TK) = 10^5 * m_w/R^* * DD(r,T)/TK
6. (Typ 5) AF(TD,TK) = 10^5 * m_w/R^* * SDD(TD)/TK

Übernahme-Modul zum direkten Übernehmen von Werten von einem SPS-Modul in einen anderen (ergänzt um Min/Max/DS „Gestern“ ab 6.0.56)

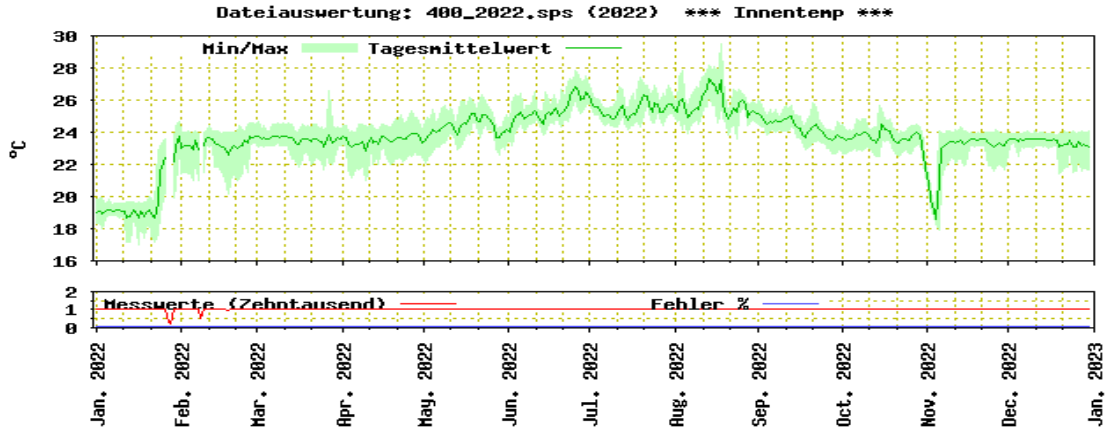


Dieses Modul dient der Verbesserung der Wartbarkeit von SPS-Strukturen. Damit ist es möglich, die Ausgänge von SPS-Modulen relativ einfach auf ein anderes Modul „umzulenken“ (bzw. von einem Sensor auf einen anderen) und dabei eine neue Bezeichnung zu vergeben. Zugleich werden im Hintergrund (in diesem Modul) aus f(ze) der Tages-Durchschnittswert, der Tages-Minimalwert und der Tages-Maximalwert (auch für „Gestern“) ermittelt. Alle diese Angaben werden täglich (bei Alog=on in der *hardware.ini*) vollautomatisch an eine entsprechende Datendatei angehängt. Unter Linux wird daraus dann ebenso täglich vollautomatisch eine Grafik erzeugt. Einschränkung für die Min/Max-Werte: Wird der Wertebereich von -3276,0 .. +3276,0 überschritten, so werden die Angaben aus technischen Gründen nicht berechnet. Für Temperaturen, Luftdruck u.ä. sollte das passen. Überschreitet der f(ze) Wert beim normalen Betrieb gelegentlich diesen Bereich, so wird der Tages-Durchschnittswert entsprechend fehlerhaft ermittelt. Der Modultyp kann auch durch Tasten beeinflusst werden (Hand/Automat).

```
f            =     f(ze)                  zi[z,3]  = MaxWertGestern*10  zr_alt[z]  = DurchschnittHeute
Zustand      =     Zustand(ze)            zi[z,4]  = MaxWertHeute*10    zl[z,1]    = DurchschnittGestern
Einheit      =     Einheit(ze)            zi[z,5]  = MinWertHeute*10    zh[z]      = MinWertGestern*10
                                          → z ist die Nummer des aktuellen Moduls (hier z.B. 400)
```

```
z=.. 102 .._[ZE]
          0..1199
            2
```

z=400 102 Innentemp 67 /der SPS-Modul 400 stellt die Werte vom Modul 67 mit dem Namen „Innentemp“ zur Verfügung



Die Max/Min/DS Werte können mit dem Anzeigetypen 3 und 4 direkt am PC angezeigt oder per SPS-Modultyp 107 extrahiert werden.
Linkes Bild: täglich erzeugte Grafik unter Linux. Sehr gut sichtbar sind die Max/Min und Durchschnittswerte. Grundlage der Grafik ist die Datendatei „400_2020.sps“, an der jeden Tag automatisch ein Datensatz angehängt wird. Bis Februar stand das Haus „leer“ ...

Variablen-Modul zum Lesen programminterner Variablen. Nur für Experten!

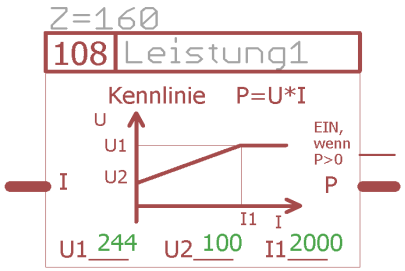
```
Dieses Modul stellt programminterne Variable aus Arrays zur Verfügung.
f            =     Wert interner Variable
```

```
z=..  107  .._was?_    Index1_    Index2_    Index3
             0..255     0..255     0..255     0..255
               2          3          4          5
```

z=791 107 Zykl_DS 0 2 /zeigt regelmäßig den Durschnittswert der Zykluszeit an

Was?	f	Zustand	Bemerkungen
0	TS[Index1]	Notbetrieb_EDB	TS[0] - Zykluszeit MIN (Messintervall 5 min) TS[1] - Zykluszeit MAX (Messintervall 5 min) TS[2] - Zykluszeit Durchschnitt(Messintervall 5 min) TS[7] - Zykluszeit aktuell
1	zi^[Index1,Index2]		SPS-Array zi[0..zmax,0..9]
2	zl^[Index1,Index2]		SPS-Array zl[0..zmax,0..2]
3	grb^[Index1]		Grafik-Array grb^[0..640]
4	grb2^[Index1]		Grafik-Array grb2^[0..640]
5	gr[Index1,Index2]		Grafik-Array gr[0..5,0..640]
6	wz^[Index1,Index2]		SPS-Array wz^[80..89,0..2]
7	c[Index1,Index2,Index3]		Übergabe-Array c[0..7,0..4,0..2]
8	ErrPCF91[Index1,Index2]		Fehler-Array ErrPCF91[0..7,0..6]
9	cue[Index1,Index2]		Überschreitungs-Array cue[0..7,0..4]
10	par[Index1,Index2,Index3]		Parameter-Array par[0..7,0..4,0..20]
11	a_ang[Index1]		Anschluss-Array a_ang[0..7]
12	b[Index1,Index2]		Binär-Array b[0..7,0..7]
13	ErrPCF74[Index1]		Fehler-Array ErrPCF74[0..7]
14	bue[Index1]		Überschreitungs-Array bue[0..7]
15	zl^[I1,1]+zi^[I1,2]-stzeit		Verbleibende Zeit bei Monoflop (SPS-Modultyp 15)
16	zl^[I1,1]+zh[I1]-stzeit		- „ - (SPS-Modultypen 44,61,62,64,65)
17	zr_alt[Index1]		SPS-Array zr_alt[0..zmax]

Faktor-Kennlinie für die Berechnung der elektrischen Leistung aus Stromsensoren



Dieses Modul „schätzt" die elektrische Leistung aus der einfachen Messung eines Stroms. Dabei wird bei Strömen < I1 eine geringere Leistung berechnet, um praktisch realistischere Ergebnisse zu erzielen (Die reine Strommessung in Wechselstromkreisen berücksichtigt leider keine Blindströme oder Ladeimpulse von Schaltnetzteilen). Wird in Verbindung mit Stromsensoren verwendet (Hardware-Band 2 Pkt. 1.8).

f = **U * I** [kW], wobei

U = U1 {I ≥ I1}, bzw.

U = $B+\frac{U1-U2}{I1}I$ {I ≤ I1} ist.

Für negative Ströme ist die Berechnung entsprechend symmetrisch.

U: - „Korrigierte" Nennspannung [V] des Wechselstromkreises, bei I1=0 erfolgt keine Korrektur
I: - Gemessener Strom [A] vom SPS-Modul I
U1: - Nennspannung [V] des Wechselstromkreises
U2: - Korrigierte (verminderte) Nennspannung [V] des Wechselstromkreises bei I=0
I1: - Strom [mA], Grenze zwischen automatischer Korrektur und korrekter Berechnung
Zustand = ein, wenn f>0, sonst aus

```
z=.. 108 .._Einheit        [Modul I]      U1         U2         I1
          String[10]       -1199..1199  Integer    Integer    Integer
               8                2          3          4          5
SensorAT26= U10  A  1  0  2  0  0  2.49  1023       /Treiber für Stromsensor mit SCT013 und 5x1N5817
z=102   78  Strom1   1      0      0               /Strom von analoger Adresse 1 Kanal 0 nehmen [A]
z=160  108  Leistung1 kW   102  244  100  2000    /Leistungsberechnung [kW]: 244V Nennspannung (U1), U2:100V, I1=2000 mA
```

Mit dem angegebenen Beispiel wird bei Strömen > 2 A (bzw. > 0,488 kW) die Leistung ganz normal per P=U1*I berechnet. Mit kleiner werdenden Strömen erfolgt eine Korrektur nach unten, sodass beispielsweise bei 0,1 A anstelle 0,024 kW nur noch 0,011 kW berechnet werden.

Signalanalyse / Mittelwert (IIR) / Anstiegsgeschwindigkeit / Maximum / Minimum ab V. 6.0.51



Dieses Modul ermittelt (je nach angegebenen Typ)verschiedene Parameter des (Ausgangs-) Datenstromes von Modul Xe. Die Berechnungen erfolgen alle 3s.

Modul Xe: - SPS-Modul-Nr für die Signalanalyse (Eingangssignal)
Einheit: - wird automatisch aus Xe und Typ ermittelt
Intervall: - Zeitintervall T [s] für die Berechnung des laufenden Mittelwertes, bzw. des Maximalwertes oder Minimalwertes. Je größer das Intervall ist, desto langsamer ändert sich der Mittelwert bzw. desto länger werden ermittelte Maximal- /Minimalwerte gehalten.
Typ: - Art der Signalanalyse
Trigger - Vorgabe eines Triggerwertes [Einh/min] für Signalanstieg & Prozent

Typ	1	2	3	4	5	6	7
f=	Mittelwert (Infinite Impulse Response IIR)	Anstieg	Anstieg	Anstieg	Maximalwert im Intervall [1]	Minimalwert im Intervall [1]	Mittelwert als Prozent [2]
Einheit=	„[Xe]“	„[Xe]/s“	„[Xe]/s“	„[Xe]/s“	„[Xe]“	„[Xe]“	„%“
Zustand=	EIN, wenn f gültig	EIN, wenn f > Trigger	EIN, wenn f < Trigger	EIN, wenn abs(f) > Trigger	EIN, wenn Max in letzten T/2	EIN, wenn Min in letzten T/2	EIN, wenn Prozentwert > Trigger

[1] Über- (unter-) schreitet der aktuelle Wert den letzten Maximal- (Minimal-) Wert, so wird dieser als neuer Wert **f** ausgegeben und der Zustand auf EIN gesetzt. Wird kein neuer Maximal- (Minimal-) Wert mehr detektiert, so geht nach Ablauf von T/4 der Zustand wieder auf AUS und der ausgegebene Wert **f** nähert sich wieder allmählich dem Mittelwert an.

[2] Der Mittelwert wird in Prozent ausgegeben, wobei der ausgegebene Mittelwert = tatsächliche Mittelwert*100% ist. Negative Ergebnisse werden auf „Null“ gesetzt. Zugleich erfolgt eine Auswertung bezüglich einer „Prozent-Schwelle“ (Triggerwert). Damit werden mehrere Rechenschritte in einem SPS-Modul kombiniert, was die statistische Auswertung von Zahlenfolgen im Bereich {0 … 1} wesentlich erleichtert.

```
z=.. 110 .. [Modul Xe]     Typ      Trigger       Intervall
           -1199..1199     1..7    Integer      5..2000000000
                  2          3        4              7
z=74   78 Wind              1      3      0          /Messung Windstärke [m/s], Analogkanal 1/3
z=360 110 WindBoen   74     5      420    200    /EIN, wenn Windböen>7 m/s (420/60 = 7), Messintervall = 200s
z=361 110 WindMittel 74     1      0      3600   /Ausgabe mittlere Windgeschwindigkeit [m/s] im Messintervall 1 Stunde
```

z=362	***110 Windanstieg (1h)***	***74***	***4***	***0***	***3600***	***/Ausgabe aktuelle Änderung der Windgeschwindigkeit im Vergleich zum Mittel***

Der Mittelwert (IIR) bzw. pT1- Verhalten, Glättung oder Tiefpass wird laufend intern bezogen auf das aktuelle (Betrachtungs-) Intervall berechnet. Die Ermittlung erfolgt nach einer der üblichen Mittelwert-Formeln als gleitender Durchschnitt: $new = (old * k - old + Xe)/k$, wobei k= Intervall/3,1 ist. Liefert das Modul xe dagegen eine Einheit in Grad [°], so wird in der Mittelwert-Ermittlung von einem Vollkreis (360°) ausgegangen. Dafür ist eine andere Berechnung notwendig, da z.B. bei old=359°, xe=1° und k=5: $(359*5-359+1)/5 = 287{,}4$ anstelle von korrekten 359,3 sich ergeben. In diesem Fall (ab Version 6.0.54) erfolgt die Mittelwertberechnung nach folgender Formel: $new = old + (Xe - old)/k$, mit automatischer Korrektur der Quadranten sowie an den Vollkreisgrenzen (0°, 360°). Hier ein Beispiel für die Glättung der Windrichtung über 100 min:

z=364	***105 Windrichtung***	***15***	***207***			***/ liefert eine Windrichtung von der Wetterstation***
z=365	***110 WR-Mittel***	***364***	***1***	***0***	***6000***	***/ glättet die Windrichtung über 100 min***

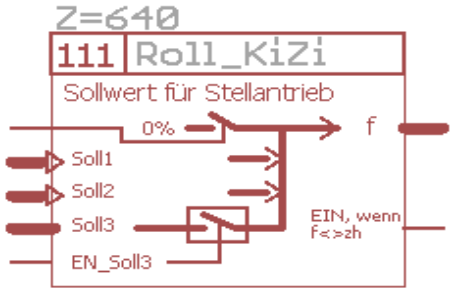
Bei der Ermittlung des Signalanstiegs (Analyse-Typen: 2, 3, 4) wird die Änderung des aktuellen Wertes von Xe bezogen auf den Mittelwert ermittelt: . Bitte beachten: Der Triggerwert wird ganzzahlig als Anstieg je Minute vorgegeben, der Ausgang f dagegen berechnet den Anstieg als [Einheiten] je Sekunde!

Fehlerfall:

Liefert der SPS-Modul xe ein Signal im Bereich -1e7 ≥ **f** xe ≥ +1e7 (d.h. Ausfall), so erzeugt der SPS-Modultyp 110 am Ausgang -1e7.

Info: Der gleitende Durchschnitt könnte auch mit $new = old + (Xe - old) * k2, wobei \quad k2 = \frac{3{,}1}{Intervall}$ berechnet werden (nur eine Multiplikation). Da das Intervall beim HC schwanken kann, ergibt sich aber kein zeitlicher Vorteil in der Programmierung.

Sollwertvorgabe für Stellantriebe oder Rollläden ab V. 6.0.52




```
Dieses Modul stellt, je nach Änderungen an den Eingängen, den entsprechenden Sollwert
(0 … 100 %) am Ausgang f zur Verfügung. 0%: Rollladen "Auf"; 100%: Rollladen "Zu"

f             = 0 %, wenn Null = EIN (Vorrang),
              = f{Modul_Soll1}, wenn f{Modul_Soll1} sich geändert hat,
              = f{Modul_Soll2}, wenn f{Modul_Soll2} sich geändert hat,
              = f{Modul_Soll3}, wenn EN_Soll3 = EIN.
              → Ist keine Bedingung erfüllt, so bleibt der letzte f-Wert dauerhaft
                erhalten.
Zustand       = EIN, wenn zh<>f (Bei Abweichung > 4 %)
Einheit       = wird automatisch auf [%] gesetzt
```

z=.. 111 ..	[Modul_Null]	[Modul_Soll1]	[Modul_Soll2]	[Modul_Soll3]	[Modul EN_Soll3]
	-1199..1199	0..1199	0..1199	0..1199	-1199..1199
	2	3	4	5	7

z=629	*12*	*RL_alle_auf*	*…*					*/Signal Rollladen auffahren*
z=630	*12*	*RL_alle_zu*	*…*					*/Signal Rollladen zufahren*
z=631	*70*	*Soll_RL_Zu*		*%*	*100*			*/Vorgabe, wieviel „%“ ist „Zu“*
z=632	*56*	*RL_Alle_man*	*%*	*0*				*/Menügesteuerte Variable für alle Rollläden*
…								
z=635	*56*	*RL_KiZi_man*	*%*	*0*				*/Menügesteuerte Variable für Rollladen Kinderzimmer*
…								
z=640	*111*	*RL_KiZi*	*629*	*632*	*635*	*631*	*630*	*/Ermittlung aktueller Sollwert Rollladen Kinderzimmer*

Anwendung: Für die Steuerung (z. B.) eines Rollladens wird mit diesem Modul ein Sollwert vorgegeben, der dann später (über andere Module) im Vergleich zum Istwert den entsprechenden Antrieb steuert. Der statische Eingang „Modul_Null“ setzt den Sollwert auf „0“ und fährt den Rollladen entsprechend auf. Zwei dynamische Eingänge „Soll1“ bzw. „Soll2“ werden nur aktiv auf den Ausgang geschaltet, wenn sich der entsprechende Eingangs-Wert ändert. Sinnvoll ist dabei die Soll-Vorgabe für alle Rollläden per Menüsteuerungen z. B. für „Soll1“ und ausschließlich für den gewünschten Rollladen per „Soll2“. „Soll3“ ist für das Zufahren vorgesehen: Wenn an „Soll3“ je nach Außentemperatur ein Wert von 100 % oder nicht ganz 100 % angelegt wird, so fährt der Rollladen nach kurzzeitiger Aktivierung des Freigabeeinganges (in Abhängigkeit von z. B. Zeit und Helligkeit in Verbindung mit einem Signalverkürzungsmodul) „EN_SOLL3“ entsprechend „zu“ oder „nicht-ganz-zu“. So kann sowohl das Schließen aller Rollläden als auch die entsprechende Lüftungsstellung bei warmen Wetter automatisch organisiert werden. Die interne Variable zh dient der Zwischenspeicherung des IST-Zustandes und kann nur von externen Modulen/Programmen beeinflusst werden. Wenn der (Soll-) Ausgang f nicht dem (Ist-) Wert zh entspricht (Abweichung > 4%), so signalisiert der Zustand die Abweichung. Ein spezialisiertes Anzeigemodul kann die beiden Werte (f , zh) auch farblich auswerten.

Wetterprognose (ab V. 6.0.53, korrigiert 6.0.54)

Dieses Modul liest nach Programmstart, sowie täglich um 00:30 MEZ bzw. 1:30 MESZ die Datei „prognos.dat“ und stellt die Daten der SPS zur Verfügung. Die Daten enthalten die Wetterprognose für „heute“ und für „morgen“. Die Datei „prognos.dat“ muss von anderen (externen) Programmen/Scripten erzeugt werden. *Hinweis: Im Zeitraum von Mitternacht bis zum Lesen von „prognos.dat“ werden korrekterweise als Daten von „heute“ die Daten von „morgen“ übernommen (die bis gestern gültig waren).*

Zustand = EIN, wenn Prognose vorhanden (Wert liegt im Bereich -999…+999), sonst AUS
Einheit = wird automatisch gesetzt

Art	Parameter „heute“	Einheit
0	Temperatur minimal	°C
1	Temperatur Durchschnitt	°C
2	Temperatur maximal	°C
3	Wolkenbedeckung	%
4	Sonnenstunden	h
5	Niederschlagsmenge	mm (l/m²)
6	maximale Windgeschwindigkeit	m/s
7	maximale Windgeschwindigkeit	km/h

Art	Parameter „morgen“	Einheit
8	Temperatur minimal	°C
9	Temperatur Durchschnitt	°C
10	Temperatur maximal	°C
11	Wolkenbedeckung	%
12	Sonnenstunden	h
13	Niederschlagsmenge	mm (l/m²)
14	maximale Windgeschwindigkeit	m/s
15	maximale Windgeschwindigkeit	km/h




```
z=.. 112 .. [Art]
             0..15
               2
```

z=542 112 TDS_heute 1 /voraussichtliche Tagesmitteltemperatur -heute-
z=543 112 Solar_heute 4 /Anzahl der voraussichtlichen Sonnenstunden -heute-
z=544 112 Solar_morg 12 /Anzahl der voraussichtlichen Sonnenstunden -morgen-

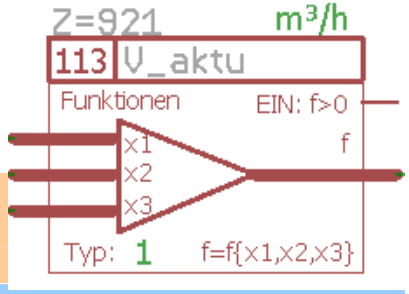
Bitte beachten: Gibt es keine Prognosedaten, so werden die Temperaturen auf „+10°C“, die Wolkenbedeckung auf „50%“ und die Sonnenstunden, die Niederschlagsmenge [mm], sowie die Windgeschwindigkeit [m/s] auf „0,5“ gesetzt. Die Windgeschwindigkeit [km/h] errechnet sich aus der Angabe der Windgeschwindigkeit [m/s] * 3,6.

Beispiel für den Inhalt einer Prognosedatei „prognos.dat“:

```
-0.95 2.24 4.63 97 .28 2.08 8.19 -2.63 -.23 3.75 63 5.73 0 3.72
  0    1    2   3   4   5    6    8    9    10 11  12  13  14          - Art
```

Funktionsblock, Luftdurchsatz, Volumenstrom für Lüftungsanlagen (ab V. 6.0.55)

```
f               = f {x1,x2,x3}, je nach Typ
                = 0, wenn x2=0, oder x1, x2, x3 ungültig
Zustand         = EIN, wenn (f>0), sonst AUS
e               = Maßeinheit
```



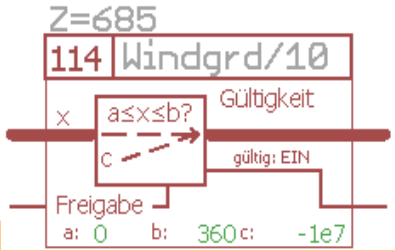

```
z=.. 113 .._Einheit_[Modul x1]_[Modul x2]_[Modul x3]_[Typ]
          String[10]   2-3: 0..1199
                 8          2           3           4        5
...
z=917   70 K_dPmax   Pa      145                              /Vorgabe max. gemessener Differenzdruck
z=919  110 DDruck3min        915    1      0      180    0    /Glättung gemessener Differenzdruck 3min (Quelle: 915)
z=920   70 Vmax      m³/h    270                              /Vorgabe maximaler Luftstrom
z=921  113 V_aktu    m³/h    919    917    920    1            /ermittelter aktueller Luftdurchsatz aus Luftdruckdifferenz
```

Typ	x1	x2	x3	Funktion f=	Bemerkungen
1	aktueller Differenz-druck (Δp)	maximaler Differenzdruck (Δp_{max})	maximaler Luftdurchsatz (V_{max})	Berechnung Luftdurchsatz aus Differenzdruck, ähnlich: $\dot{V}_{aktu}=\dot{V}_{max}\cdot\sqrt{\frac{\Delta p}{\Delta p_{max}}}$, $-1{,}2<\frac{\Delta p}{\Delta p_{max}}<1{,}2$	Genauigkeit der Berechnung: ±1,2%
2	aktuelle Lüfterleistung (P)	maximale Lüfterleistung (P_{max})	maximaler Luftdurchsatz (V_{max})	Berechnung Luftdurchsatz aus Differenzdruck, ähnlich: $\dot{V}_{aktu}=\dot{V}_{max}\cdot\sqrt[3]{\frac{P}{P_{max}}}$, $0\le\frac{P}{P_{max}}<1{,}2$	ideale Formel, Genauigkeit der Berechnung: ±1,2%
3	aktuelle Lüfterleistung (P)	maximale Lüfterleistung (P_{max})	maximaler Luftdurchsatz (V_{max})	Berechnung Luftdurchsatz aus Differenzdruck, ähnlich: $\dot{V}_{aktu}=\dot{V}_{max}\cdot f\{\frac{P}{P_{max}}\}$, $0\le\frac{P}{P_{max}}<1{,}2$	praktische Formel, für Ansteuerungen per Schwingungspaket

Dieses Modul dient der Berechnung des aktuell gemessenen (reduzierten) Luftdurchsatzes in Lüftungsanlagen in Abhängigkeit vom aktuell gemessenen Differenzdruck in Rohrleitungen oder der aktuellen (elektrischen) Leistung eines Lüfters in Bezug auf den vorgegeben maximalen Differenzdruck mit entsprechendem maximalen Luftdurchsatz. Die Berechnungen erfolgen über Polynomapproximationen, um Rechenzeit zu sparen. Die Angaben maximaler Differenzdruck/Luftdurchsatz werden vom Hersteller der Lüftungs-Baugruppe mitgeliefert.

Da in der Praxis der Wirkungsgrad elektrischer Ventilatoren je nach ihrer aktuellen elektrischen Leistung variiert, stimmt die theoretische Formel (Art=2) bei elektrischen Leistungen < 50% überhaupt nicht mehr mit der Praxis überein. Stattdessen ist die empirische Formel (Art=3) insbesondere für die verbreiteten Lüfter mit kostengünstigen Kondensatormotoren zu verwenden.

Gültigkeitsmodul (ab V. 6.0.56)




```
f               = Wert f von x, wenn a≤f(x)≤b und Freigabe=EIN (gültig)
                = c, wenn f(x)<a oder f(x)>b oder Freigabe=AUS (ungültig)
Zustand         = EIN, wenn f gültig, sonst AUS
Hinweise:       Die Maßeinheit wird von Modul x automatisch übernommen.
                Die Angabe „-1e7“ für c ist der HC-interne Standard für „ausgefallen“.
```

z=.. 114 ..._[Modul x]	_[Modul Freigabe]	_ a	_ b	_ c
0..1199	0..1199	a,b: integer		longint
2	3	4	5	7

```
z=641 105 Wind5min          15    207                       / Windrichtung geglättet von Wetterstation
z=643 110 Wind90min         641   1     0     5400          / Windrichtung Mittelwert 90min
z=645  72 Windgrd/10  °     643   0     6                   / Windrichtung [°]/10
z=684  69 Wind>10km/h       km/h  674   10    300           / schaltet, wenn Windgeschwindigkeit (kommt von 674) >10km/h
z=685 114 Windgrd/10        645   684   0     360   -1e7    / Windrichtung, wenn Windgeschwindigkeit >10km/h
in hardware.ini:
EDB=1 Aussenwerte   -15.0 +40.0       …  685  …            / Grafik: … Windgrd/10 …
```

Dieses Modul dient der Überwachung von Signalen. Sind sie nicht in einem sicheren Bereich, so wird ein festgelegter Wert zugewiesen. Verwendung: Überbrückung von Sensorausfällen, Verbesserung der Übersichtlichkeit in EDB-Grafiken. Linien in den Grafiken können nun einfacher nur bei Bedarf gezeichnet werden (Windrichtung nur bei stärkerem Wind; Wirkungsgrad Wärmetauscher nur bei eingeschaltetem Lüfter u. a. m.).

4.7.7. Festgelegte, interne Konstanten und Variable zur freien Verwendung in der SPS

Der Zustand der folgenden internen Module ist "EIN", wenn die entsprechende Bedingung erfüllt ist, ansonsten "AUS". Gleichzeitig werden über den Analogausgang (f) einige häufig benötigte Konstanten und Programmparameter bereitgestellt. Bitte nicht daran stören, dass die Bezeichnung sich mehr am Modulzustand als am Analogausgang orientiert.

Bezeichnung	**Modulzustand (EIN)**	**Modul- Analogausgang**
0 – AUS	immer AUS	f= 0
1 – EIN	immer EIN	f= 1
2 - Testmodus	ein: wenn Testmodus aktiv, sonst aus	f= 2
3 – AUTOMAT	,wenn AUTO (entspricht „A" in Statuszeile)	f= 0,5
4 – WERKTAG	Mo-Fr, kein Feiertag	f= 0,001
5 – Sonntag	sonntags	f= 0,01
6 - Montag	montags	f= 0,1
7 - Dienstag	dienstags	f= 10
8 - Mittwoch	mittwochs	f= 100
9 - Donnerstag	donnerstags	f= 1000
10 - Freitag	freitags	f= 3,14159 (Pi)
11 - Samstag	sonnabends	f= 6,28318 (2Pi)
12 – Blink1s	blinkt (nicht mit rotem Blinken verwechseln!)	f= Echtzeit (Tagessekunden*18,2)
13 - MESZ	wenn Sommerzeit	f= 3,6
14 – MESZ erlaubt	wenn MESZ-Korrektur erlaubt	f= 0,27778 (1/3,6)
15 – Ausfall SPS	wenn Fehlfunktion SPS, Fehlerflag	f= Stunde Programmstart
16 – DEMO	DEMO-Modus EIN	f= Tageseigenschaft (Bezeichnung Bereich)
17 – morgen:Wktg	morgen Werktag (Mo-Fr, kein Feiertag)	f= Tagessekunden
18 – 100 Zyklen	wechselt jeweils nach 100 Zyklen	f= Dreiecksignal -1..+1
19 – Taste OK	Taste „OK" am Hausbus ist gedrückt	f= delta t zwischen 2 Zyklen [s]
20 – Hinweis!	Signal „HINWEIS!" gesetzt	f= aktueller Zyklus Busroutine (0..255)
21 – min-Zyklus	Zu Beginn jeder Minute für genau einen Zyklus EIN, sonst AUS	f= Nr. aktuelle Bildschirmseite
22 – 3s-Zyklus	für einen Zyklus aller 3s EIN (für die Berechnung der I-Anteile; „I-Zyklus")	f= Differenzzeit zum letzten I-Zyklus (~3s)
23 – Farbinwert	Farbinvertierung eingeschaltet	f= -1
24 – Binär-Adr.(0) OK	Binär-Adr.(0) arbeitet korrekt	f= -2
25 - Binär-Adr.(1) OK	Binär-Adr.(1) arbeitet korrekt	f= -10
26 - Binär-Adr.(2) OK	Binär-Adr.(2) arbeitet korrekt	f= -100
27 - Binär-Adr.(3) OK	Binär-Adr.(3) arbeitet korrekt	f= -1000
28 - Binär-Adr.(4) OK	Binär-Adr.(4) arbeitet korrekt	f= Zeit in Sekunden seit 1.1.1997 0:00:00

Bezeichnung	Modulzustand (EIN)	Bezeichnung Modul- Analogausgang
29 - Binär-Adr.(5) OK	Binär-Adr.(5) arbeitet korrekt	f= Zeit automatischer Seitenwechsel
30 - Binär-Adr.(6) OK	Binär-Adr.(6) arbeitet korrekt	f= Sekunden
31 - Binär-Adr.(0) OK	Binär-Adr.(7) arbeitet korrekt	f= Minuten
32 - Analog-Adr.(0) OK	Analog-Adr.(0) arbeitet korrekt	f= Stunden
33 - Analog-Adr.(1) OK	Analog-Adr.(1) arbeitet korrekt	f= Wochentag (0..7), Sonntag=0
34 - Analog-Adr.(2) OK	Analog-Adr.(2) arbeitet korrekt	f= Tag des Monats
35 - Analog-Adr.(3) OK	Analog-Adr.(3) arbeitet korrekt	f= Monat
36 - Analog-Adr.(4) OK	Analog-Adr.(4) arbeitet korrekt	f= Jahr
37 - Analog-Adr.(5) OK	Analog-Adr.(5) arbeitet korrekt	f= aktuelle COM Adresse (1016:3F8)
38 - Analog-Adr.(6) OK	Analog-Adr.(6) arbeitet korrekt	f= Konstante Warteschleife I2C-Bus (zk)
39 - Analog-Adr.(7) OK	Analog-Adr.(7) arbeitet korrekt	f= Zeit [s] seit Programmstart
40 - Tag	errechnete Helligkeit>64,58%	f= aktuelle, errechnete Helligkeit [%]
41 - Dunkel	errechnete Helligkeit<30%	f= aktuelle Sonnenhöhe über Horizont [°]
42 - Azimut	Zu Beginn jeden Tages für genau einen Zyklus EIN, sonst AUS	f= aktuelles Sonnen- Azimut [°]
43 - UV-Index	Montag-Donnerstag: EIN	f= aktueller UV- Index
44 - Solarleistung	Takt 3s EIN, 21s AUS (T=24s)	f= geschätzte Sonnenenergie je m² (horiz.)
45 - Mondphase	Takt 3s Ein, 9s AUS (T=12s)	f= Alter des Mondes (Tage nach Neumond)
46 - SPS-Hand	„H" in Statuszeile gesetzt	f= Zählerstand S0-Zähler E0
47 - Notbetrieb	„Notbetrieb": Rechenzeit zu gering	f= Zählerstand S0-Zähler E1
48 - stat.Modus	statischer Anzeige-Modus aktiv	f= Zählerstand S0-Zähler E2
49 - BA"D"vorhanden	Busadapter „Deluxe" vorhanden	f= Zählerstand S0-Zähler E3
50 - P-S0-E0	S0-E0	f= Leistung/Verbrauch S0-Zähler E0
51 - P-S0-E1	S0-E1	f= Leistung/Verbrauch S0-Zähler E1
52 - P-S0-E2	S0-E2	f= Leistung/Verbrauch S0-Zähler E2
53 - P-S0-E3	S0-E3	f= Leistung/Verbrauch S0-Zähler E3
54 - BA"D"1Netzteil	Am Busadapter „Deluxe" nur ein Netzteil	f= Anzahl Schleifendurchläufe je Minute
55 - BA"D"ReservePC	Am Busadapter „Deluxe" Reserve PC aktiv	f= Tageseigenschaft Morgen (Bez.Bereich)
56 - Timerabweichung	Zeitinkonsistenz	f= Timerabweichung [%]
57 - Temp.-A	Sensorwert gültig	f= Ta [°C] Sensor AM2320 [1)]
58 - Feuchte-A	Sensorwert gültig	f= Fa [%] Sensor AM2320 [1)]
59 - Busproblem	Busproblem SDA und/oder SCL	f= Attiny-Fehlerquote [2)]

[1)] Diese Funktion ist gesperrt, wenn an Adresse 0B8h ein ATtiny261 (Binär26=XXXX 4 ..) angeschlossen ist. Ist kein ATtiny261 an Adresse 0B8h initialisiert, so ist der Treiber für den Sensor AM2320 automatisch aktiv.

2) Fehlerquote [%] der binären und analogen Attiny-Datentelegramme (keine Multisensor/Raumsensoren, nicht das Businterface „Deluxe"), sowie des Max1236 und DS1631, soweit sie initialisiert wurden. Wird aller 5 min berechnet, wenn mindestens ein Befehl „EDB=" in hardware.ini vorhanden ist. Die ermittelte Fehlerquote ist völlig unabhängig von der Ausfallanzeige einzelner Kanäle oder Interfaces und zählt jedes nicht beantwortete Datentelegramm im Vergleich zu den korrekt übermittelten: F= fehl*100/(fehl+korrekt+1) (Ab Version 6.0.59)

4.7.8. Übersicht über alle Betriebsstunden bzw. „lifing“- relevanten SPS Module und sonstige Zeit-/Speicherfunktionen

SPS-Modul Typ	*Kurzbezeichnung*	*Quelle:* [2]	*Speicherung* [4]	*Betriebsstunden relevante Funktion*	*Analogausgang (f)*	*Betriebs-stunden in SPS Übersicht* [3]	*„lifing“-Menüpunkt:*	*Besonderer Reset möglich?* [1]
4, 5, 37,38	Zeitmodul, Wecker	zi2, zi3, (zi4, zi5)	ja	-	Speicherflag		<Zeitmodule>	
6, 10, 91	Flipp-Flop, Leermodul	z	ja	-	-		-	Tastensteuerung
9	Betriebsstundenzähler	Binär-Kanal	ja	Fortlaufende Zählung	Betriebsstunden [h]	x	<Gesamtzähler>	
17	Betriebsstundenzähler	z	ja	Fortlaufende Zählung	Betriebsstunden [h]	x	<Gesamtzähler>	
30	Binärkanal schalten	z	ja	Fortlaufende Zählung	Betriebsstunden [h]	x	<Gesamtzähler>	
31	Binärkanal schalten	z	ja	Fortlaufende Zählung	-	x	<Gesamtzähler>	
33	Analogausgang EIN/AUS setzen	z	ja	Fortlaufende Zählung	Betriebsstunden [h]	x	<Gesamtzähler>	
56	Variablenspeicher	f	ja	-	Variable		<Q-Zähler/ Speicher>	
57	Quantumszähler	f	ja	-	Quatumsintegral		<Q-Zähler/ Speicher>	Reset-Eingang
58	Langzeitspeicher	f	ja	-	Gespeicherter Wert		<Q-Zähler/ Speicher>	Reset-Eingang
59	Betriebsstundenzähler	z	ja	Fortlaufende Zählung, Zeit letztes EIN	Zeitdauer letztes EIN [s]	x	<Gesamtzähler>	
60	Tastverhältnis Gesamt	z	ja	Ständige Berechnung	Tastverhältnis (0..1)		-	
66	Betriebsminuten Vortag	z	gT	Tägliche Zählung im Hintergrund, Anzeige Vortag	Betriebsminuten Vortag [min/d]		<Tageszähler>	Tagesbeginn (Zuweisung)
67	Tastverhältnis lfd. Tag	z	gT	Ständige Berechnung	Tastverhältnis (0..1)		-	Tagesbeginn
68	Betriebsminuten lfd. Tag	z	gT	Tägliche Zählung	Betriebsminuten lfd. Tag [min]		<Tageszähler>	Tagesbeginn
90	Tastensatz	f	ja	-	SPS Modulnummer, der gerade „ein“ ist		-	Menü-Tasten-steuerung
96, -96	Energiezähler (Gesamt)	f	ja	Fortlaufende Zählung möglich	Energiesumme		<Energiezähler>	
97, -97	Tagesenergiezähler	f	gT	Fortlaufende Zählung möglich	Energiesumme (tägl.)		<Energiezähler>	Tagesbeginn

1) Der Standard Reset für alle Zähler ist einheitlich:
   - Taste „del/Entf" im SPS Menü für ein Modul
   - Löschen der Datei „gestern.daz" für die Module 0..399
   - Löschen der Datei „gestern2.daz" für die Module 400..799
   - Löschen der Datei „gestern3.daz" für die Module 800..1199
   - über programmierten „Softkey" (Bildschirmtaste)

2) Die ermittelten Betriebsstunden beziehen sich entweder auf den Zustand (z) eines Quell-Moduls oder auf den Analogausgang (f) des Quellmoduls.

3) Neben der Anzeige der Betriebsstunden im unteren Teil der SPS Übersicht können auch Anzeigen vom Typ 29 und 30 an diese gelisteten SPS Module zur Darstellung der Betriebsstunden in anderen Menüs angeschlossen werden.

4) Das Laden und die Speicherung der relevanten Werte erfolgt immer automatisch in den „*.daz" -Dateien. Bei SPS Modulen mit der Angabe „gT" (gleicher Tag) werden die (alten) Werte beim Programmstart nur dann geladen, wenn das (vorangegangene) Programmende am selben Tag war UND das Programm über den letzten Tageswechsel lief. Damit wurde ein Kompromiss zwischen „aktiv" laufenden Anwendungen und Anwendungen „in der Entwicklung" gefunden. Eine Anzeige zu einem späteren Zeitpunkt ist dennoch möglich, wenn unmittelbar nach dem Programmstart unter „System", „Uhrzeit" „DEMO" eingeschaltet und dann „gestern.dat laden" aktiviert wird.

Für die Dokumentation wird ab Version 6.0.39e automatisch eine Textdatei „stunden.txt" mit einer Übersicht aller relevanten Daten erzeugt. Die Speicherung erfolgt zum Tagesabschluss um Mitternacht (wenn ALog=ein), zum Programmende und auf Anforderung über die Taste „s":

```
Übersicht über Betriebsstunden:
letzte Aktualisierung dieser Angaben: <2.12. 23:59>
SPSNr Typ        Wert Einheit       SPS-Bezeichnung
< 91>  31      213.14 h             Außenlicht
<189>  56     2850.00 l             Tank_Voll
<190>  17      219.10 h             Tankuhr
<213>  97       55.52 kWh/d         Heizung
<216>  97       13.28 kWh/d         Warmwasser
<226>  33     1455.54 h             Pumpe_Zirk
<227>  66       24.52 min(heute)    Pumpe_Zirk
<227>  66       18.12 min(gestern)  Pumpe_Zirk
                   . . .
<261>  96    13564.33 kWh           Gesamtertrag
<262>  97        0.00 kWh/d         Solarertrag
<333>  66       76.34 min(heute)    Öl_Brenner
<333>  66       54.47 min(gestern)  Öl_Brenner
<334>  59     4283.02 h             Brenner
<341>  31     3458.00 h             Pumpe_SP1
<350>  31      890.20 h             Pumpe_SP2
<382>  56    0.029297 l/min         Verbrauch
-ENDE-
```

4.7.9. Übersicht über alle verwendeten Tastaturkodes

Die Tastaturkodes können von der Tastatur, den Bustasten oder per Netzwerk übergeben werden. Erweiterte Kodes beginnen ab 1000, sie können nicht direkt per Netzwerk übergeben werden.

Kode	*Zeichen*	*Bedeutung*
0	-	-
9	<Tab>	Objektumschaltung im Gittermodus
10	<Ctr><Enter>	wird ignoriert, Taste unter Seamonkey
13	<Enter>	SPS-Menü: Modulstatusumschaltung
27	<Escape>	Entspricht „#“ - Rückschaltung im Menü
32	<Leertaste>	Weiterschaltung/Auslösung im Menü
35,39,92,124	#	Rückschaltung im Menü
42,125	*	Umschalten in Automatikbetrieb
43	+	Menüseite + 1
45	-	Menüseite - 1
48-57	0 .. 9	Ziffern, Menüsteuerung
72		Kursor hoch
73		Bild hoch / Page up
75		Kursor links
77		Kursor rechts
78	<Shift>n	„Speicherschutz“=ein, bzw. „No“ (kein Schreiben mehr auf Datenträger)
80		Kursor unten

Kode	*Zeichen*	*Bedeutung*
81		Bild runter / Page down
87	<Shift>w	„Speicherschutz“=aus, bzw. „weiter“ (Schreiben auf Datenträger erlaubt)
83	<Entf/Del>	SPS-Menü: Reset Betriebsstunden
97	a	Handmodus dauerhaft aktivieren/deaktivieren
103	g	Gittermodus einschalten/ausschalten
105	i	Farben invertieren
107	k	Programmende (dann automatischer Neustart, da Schleife in autoexe.bat)
108	l	Speicherung Laufschrift im Log
109	m	RAM Image speichern (nicht kompiliert)
111	o	Signal „ok“ - Hinweis löschen
112	p	Screenshot in Grafikdatei „<Seite>.bmp“
114	r	„Restart“ erneute Programminitialisierung
115	s	SPS Daten speichern unter „sps.txt“ und „stunden.txt“
116	t	Statischen Bildschirmmodus einschalten
255	-	BIOS Neustart
1033	<Strg><ALT>F	Als Taste ignorieren (Fensterumschaltung unter Linux)
1107	<ALT>< F4>	Programmende

5. Einschränkungen in der Programmbenutzung / Die Lizenz.RPT / rechtliche Hinweise /Updates

5.1. Rechtliche Hinweise

Das Programm *hauscomp.exe* ist bis einschließlich Version 6.0.32 ein Mix aus dem Shareware- und Freeware- Prinzip. Solange das Programm *hauscomp.exe* von privaten Anwendern, bzw. für nicht kommerzielle oder Ausbildungs - Zwecke genutzt wird, unterliegt es keinerlei Einschränkungen – es kann frei weitergegeben und beliebig benutzt werden. Die einzige technische Einschränkung besteht in der Laufzeitbegrenzung des Programms. Es läuft ununterbrochen maximal eine knappe Woche. Danach muss es erneut gestartet werden und das Spiel beginnt von vorn. Ein regelmäßiger Reset reicht damit völlig aus. **Ab Version 6.0.33 ist das Programm für nicht kommerzielle Zwecke komplett freigegeben (also Freeware), es gibt auch keine technischen Sperren mehr.**

Die Software steht nun unter der creative-common Lizenz:

Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 3.0 Deutschland (CC BY-NC-SA 3.0 DE) siehe auch:

https://creativecommons.org/licenses/by-nc-sa/3.0/de/legalcode

Kein Programm ist fehlerfrei. Damit ich aber eventuelle Programmfehler korrigieren kann, brauche ich auch entsprechende Hinweise! Selbstverständlich nehme ich gern Anregungen und Tipps entgegen. Sollte jemand eine besonders gute Anwendung programmiert haben, würde ich sie, das entsprechende Einverständnis vorausgesetzt, als Beispiel mit in die Dokumentation aufnehmen. Sollten sich im Programm Hinweise auf Schadcode finden, oder das Programm unvermutet Daten versenden, so sind diese Funktionen nicht von mir implementiert!

Noch ein letzter, aber für mich wichtiger Hinweis: Das Programm ist zwar sehr robust aber es entstand doch durch einen notorischen Laien und ich kann etwaige Fehler nicht ausschließen. Bitte, bitte dieses Programm nicht beim Militär oder in der Atomindustrie einsetzen!

5.2. Schreibgeschützte Datenträger

Das Programm *hauscomp.exe* startet auch von schreibgeschützten Datenträgern (CD/DVD). Dabei gibt es folgende Einschränkungen:

- Die automatische Erstellung von Logs (Laufband, Analogkanäle, SPS-Log) ist logischerweise nicht möglich.
- Die Daten des letzten Tages (gestern.*) können deshalb auch nicht gespeichert werden.
- Der Lizenz- key (lizenz.rpt) kann nicht geladen werden. das Programm läuft deshalb im zeitlich begrenzten Freeware- Modus (gültig für Versionen < 6.0.33).

In der Laufschrift erscheint bei Programmstart der Hinweis „*** Achtung, Datenträger ist schreibgeschützt! ***“. Ein eventueller Schreibschutz der Datei „font.abc“ (bis V. 6.0.33 „lizenz.rpt“) löst den gleichen Effekt aus.

5.3. Das Jahr 2038 bzw. 2065 Problem

Das Programm *hauscomp.exe* benutzt intern sehr intensiv alle möglichen Zeitvariablen um die geforderten Steuer- und Regelungen möglichst genau auszuführen. Unter anderem ist dabei eine sekundengenaue Systemzeit (B-Zeit), die in einem 32 Bit Format gespeichert ist. Sie zählt ab dem 1.1.1997 00:00 und kann damit den Zeitbereich bis zum 19.1.2065 überstreichen. Damit ist die Nutzung dieses Programms bis dorthin begrenzt. Durch interne Rechnungen kommt es aber bereits eine Woche vor dem theoretischen Ende zu Einschränkungen. Spätestens aber ab dem ***16.1.2065 19:50 MEZ*** hört der Doppelpunkt der Uhrzeit auf, zu blinken. Sollte es dann noch jemanden geben, der das Programm nutzt, so bleibt ihm nur die Uhrzeit des PCs entsprechend zurückzustellen.

Diese Zusammenhänge entsprechen denen auf UNIX Systemen (und damit auch LINUX) deren Unix-Zeit bereits mit dem 1.1.1970 begann. Diese Systeme, so sie die Zeit intern ebenfalls im 32 Bit Format speichern, arbeiten maximal bis zum 19.1.2038 3:14:08. Die Unix-Zeit wird im Programm *hauscomp.exe* ebenfalls verwendet – in der Ausgabe der *.csv Dateien. Die dortigen Angaben werden aber aus der B-Zeit über eine Gleitkommazahl errechnet, sodass es keine Ausfälle im Jahr 2038 geben wird! Vielleicht hebe ich irgendwann auch die interne Bezugszeit noch etwas an – sodass noch die Grenze nicht 2065, sondern etwa 20 – 25 Jahre später erreicht wird.

5.4. Updates

Die jeweils aktuellen Dateien und Handbücher stehen im Internet unter http://haus-computer.de/ zum Download zur Verfügung. Es reicht die aktuelle *hauscomp.exe* in das Anwenderverzeichnis zu kopieren und neu zu starten. Damit wäre das Thema Update geklärt.

6. Anwendungsbeispiele

6.1. Der individuelle Kalender (keine zusätzliche Hardware notwendig!)

Aufgabe: Anzeige eines tagesaktuellen Kalenders mit Hinweisen in der Laufschrift

hardware.ini:

```
Sommerzeit=auto
Seitenwechsel=10
Hintergrund=hell
COM=0
```

hauscomp.ini:

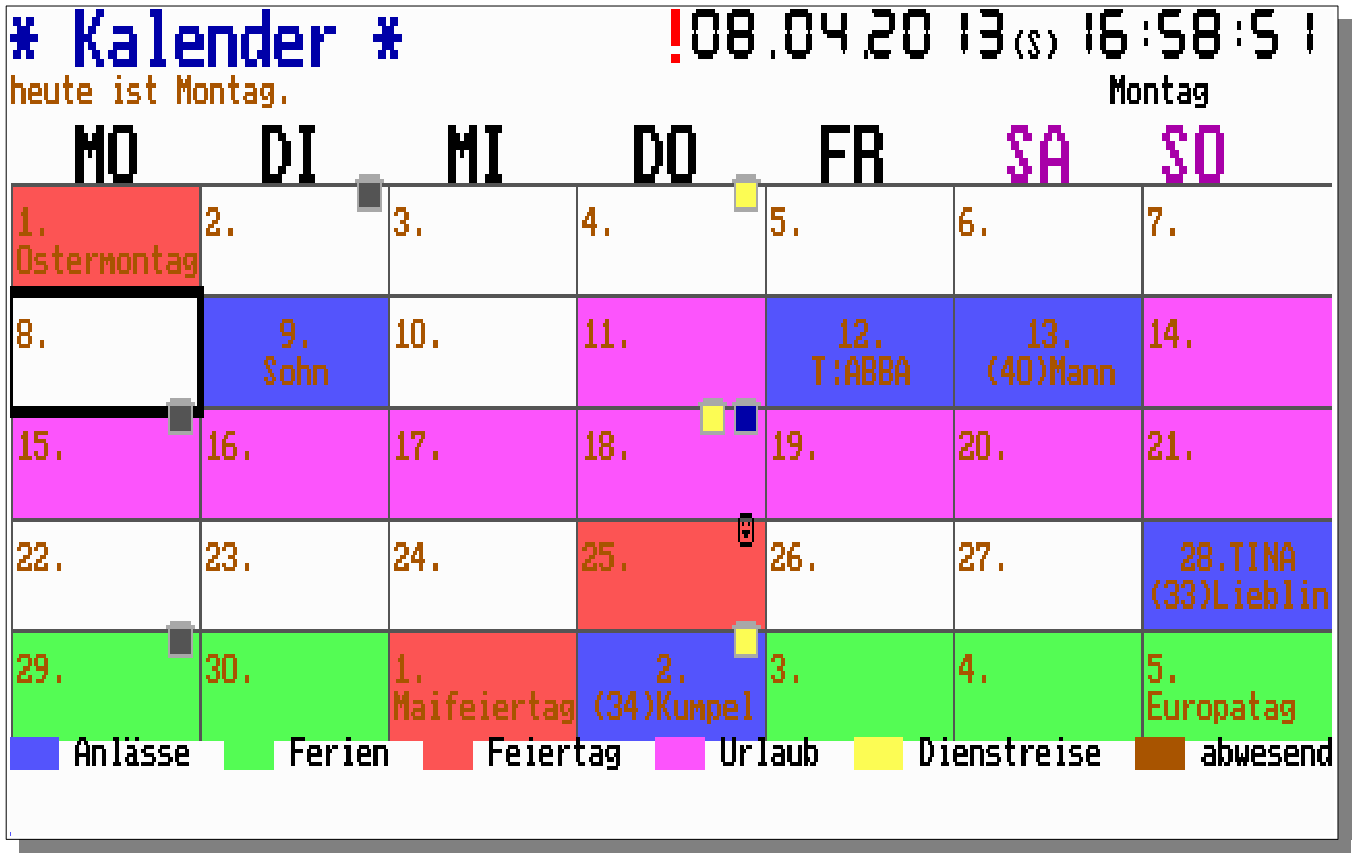
```
Norm=ANSI
Anzeigeseiten= 96 96 96                                  /nur den Kalender darstellen
Sommerzeit=off                                           /Zeitumstellung  über das Betriebssystem
Bundesland= 11                                           /Mecklenburg- Vorpommern

[Programmierung Kalender]
BezDatBereich=1 Anlässe 2 Ferien 4 Feiertag 6 Dienstreise 14 abwesend 5 Urlaub
DatBereich=11 04 2013   21 04 2013 5                     /geplanter Urlaub
DatBereich=29 04     13   07 05 2013 2                   /Ferien
DatBereich=25 04     13   25 04 2013 4                   /zusätzlicher Feiertag

*Datum= Tag Monat Wer?      Geburtsdatum
Datum=   09  04   Sohn
Datum=   12  04   T:ABBA
Datum=   13   4   Mann      1973
Datum=   28   4  TINA
Datum=   28   4   Liebling  1980
Datum=    2   5   Kumpel   1979

Müll=21 01 2013   2 08   /schwarze Tonne
Müll=02 04 2013   0 08   /schwarze Tonne Sonderleerung
Müll=24 01 2013   2 14   /gelbe    Tonne
Müll=24 01 2013   4 09   /blaue    Tonne
```

Mit den angegebenen *.ini Dateien ergibt sich am 8. April 2013 folgende Anzeige:



Die farbliche Tages-Darstellung ergibt sich zum einen aus der Farbvorgabe in der Legende (BezDatBereich=..), sowie den entsprechenden Zeiten (DatBereich=). Tage, die mit „Datum=“ benannt sind, werden automatisch blau dargestellt, Feiertage rot. Die symbolischen Einträge an den entsprechenden Tagen für die Müllabfuhr, sowie der Vollmond runden das Ergebnis ab.

Außer den angegebenen sind weiter keine Script-Befehle notwendig.

Diese kleine Variante eines Hauscomputers kann man wunderbar in den Autostart-Ordner z. B. unter WINDOWS XP einbinden.

Die Angabe „COM=0“ ist notwendig, um das Laufband ohne Hausbus zu aktivieren. Der Befehl „Seitenwechsel=“ kann entfallen, da ohnehin immer dieselbe Seite angezeigt wird.

Über die Tastatur sind selbstverständlich weiterhin alle Menüs erreichbar. Damit stehen natürlich auch die Astrofunktionen wie der Sternenhimmel und die Tagesinfo mit den Sonnenauf- und -Untergangszeiten zur Verfügung. Dabei sollte aber der genaue Standort mit dem „Geo=“ Scriptbefehl in der *hardware.ini* vorgegeben sein.

6.2. Das Geisterhaus …

Aufgabe: Während die Hauseigentümer abwesend sind, sollen einige Lampen möglichst realistisch die Abwesenheit verschleiern. Herkömmliche Zeitschaltuhren berücksichtigen nicht den Tagesgang. Zusätzlich soll gezählt werden, wie oft die Haustür offen war.

- *hauscomp.ini*:

  Zunächst wird der allgemeine Teil programmiert. Die angeschlossene Hardware besteht aus 2 PCF8574P an den Adressen 2 und 3, sowie einem PCF8591 an Adresse 1. Die 4 Analogeingänge sind mit drei Temperatursensoren und einem Helligkeitssensor versehen. Der analoge Ausgang wird nicht verwendet. Der Helligkeitssensor liefert knapp 5V (244Einheiten) bei maximaler Außenhelligkeit (=100%). Aus diesem Signal wird später von der SPS das Signal „Tag“ errechnet. Neuerdings kann man anstelle eines realen Helligkeitssensors auch die internen Module 41 (Tag) bzw. 42 (dunkel) verwenden.

  An die Binärkanäle sind an Dachboden, Wohnstube, Hof und Küche Schaltrelais angeschlossen, die die entsprechenden Lampen/ Geräte mit Spannung versorgen.

```
[GLOBAL]
*Norm=ANSI (äöüÄÖÜß - Umlaute lesbar)  ** Norm=ASCII   (Ž™š„”□á - Umlaute lesbar)*
Norm=ANSI
Hinweis!=2 3
TasteOK= 2 4
Sommerzeit=auto
PCF8591=PCF8591/0 Interface PCF8591/2 PCF8591/3 PCF8591/4 PCF8591/5 PCF8591/6 PCF8591/7
[Interface]
*Sensor= Bez.  Einheit Adresse Kanal Abfragefrequenz 2-3 Wertepaare*
Sensor=Helligkeit   %     1 0  90   0    12    100   244
Sensor=Temp.2      °C     1 1  90   0    63  +40.0   145     +80.0  217    /
Sensor=Temp.3      °C     1 2  90   0    63  +40.0   145     +80.0  217    /
Sensor=Temp.4      °C     1 3  90   0    63  +40.0   145     +80.0  217    /
Analog=A/D        E/A     1     0   0    0      1    255
PCF8574P=P_8574P/0 P_8574P/1 Intf_2 Intf_3 P_8574P/4 P_8574P/5 P_8574P/6 P_8574P/7
*Binär=Bezeichnung Adresse Kanal Aus+Eingang(1) nur Eingang(0)*
*zu Testzwecken sind alle Eingänge auch als Ausgänge ansteuerbar (auf 1 gesetzt)!*
Binär=Türkontakt      2 0  1
Binär=Hinweis!        2 3  1
Binär=Bestätig.       2 4  1
Binär=Dachboden       3 0  1
Binär=Wohnstube       3 1  1
Binär=Hof             3 2  1
Binär=Küche           3 3  1
```

Im folgenden Abschnitt werden die grafischen Ausgaben programmiert. Zunächst die 24h Datenlogger-Funktion für die verwendeten analogen

und binären Signale. Anschließend werden sinnvolle Anzeigen auf die Seite 95 (Hausparameter) gesetzt. Die 3 Temperatursensoren leuchten blau, wenn es kälter als +20°C bzw. -1°C ist und blinken rot, wenn es wärmer als +28°C wird. Der Wert des Türzählers sowie die Zustände Stube, Küche usw. kommen aus der SPS und werden ebenfalls auf dieser Seite (95) platziert:

hauscomp.ini (Fortsetzung)

```
[Programmierung Grafik]
*graph_a= Nr.(0-6)  Adresse Kanal Farbe MIN  Max
graph_a=     1          1     0    14   -10   120
graph_a=     2          1     1    15   -25  60
graph_a=     3          1     2    11   -25  60
graph_a=     4          1     3    10   -25  60
*graph_b= Bildzeile(1-16)  Adresse(0-7)   Kanal(0-7)   Farbe(0-15)*
graph_b=        4     2   3   4    ;
graph_b=        5     2   4   5    ;
graph_b=        9     3   0   9    ;
graph_b=        10    3   1   10   ;
graph_b=        11    3   2   11   ;
graph_b=        12    8  151  11   ;
graph_b=        13    8  160  13   ;Boden
graph_b=        14    8  164  14   ;Hof
graph_b=        15    8  156  15   ;Stube
graph_b=        16    8  157  10   ;Küche

[Programmierung Anzeige Hausparameter]
*Anzeige= Seite Anzeigetyp Adresse Kanal X  Y  P1  P2  MIN MAX  welcherpar
Anzeige=   95      2          1      0   1  1  1    2  -20 +120  0  ;Helligkeit
Anzeige=   95      2          1      1   1  2  1    2  +20 +28   0  ;Temperatur 2
Anzeige=   95      2          1      2   1  3  1    2  -1  +28   0  ;Temperatur 3
Anzeige=   95      2          1      3   1  4  1    2  -1  +28   0  ;Temperatur 4

Anzeige=   95      9          8     156  50  5  0    1   1  14    1  ;Stube
Anzeige=   95      9          8     157  50  6  0    1   1  14    1  ;Küche
Anzeige=   95      9          8     160  50  7  0    1   1  14    1  ;Boden
Anzeige=   95      9          8     164  50  8  0    1   1  14    1  ;Hof
Anzeige=   95      9          8     151 250  1  0    1   1  14    1  ;Tag/Nacht
Anzeige=   95      9          8     170 250  5  0    1   1  14    1  ;Haustür
Anzeige=   95      0          8     172   1  9  0    1   0  32000 1  ;Türzähler
```

Jetzt wird es interessant. Für die Realisation der Aufgabe wurden die Module 151 bis 173 benutzt. Modul 151 erzeugt das Signal „Tag“. Die

Module 152 und 153 stellen jeweils ein Zeitfenster zur Verfügung. Kernstück der Zufälligkeit sind die Module 154, 155 und 158. Sie geben dem Haus die Individualität. Zufall2 steht z. B. für die Aufenthaltsdauer in der Küche: man ist mindestens 2 min aber höchstens 20 min in der Küche und kommt frühstens nach 50 min wieder zurück. Die Küchenlampe wird von Modul 157 angesteuert: Es darf draußen nicht hell sein, nicht vor 16:00 Uhr und nicht nach 21:35 Uhr. Dazwischen bestimmt der Zufall, wann die Lampe angeht. Nach diesem Muster werden auch die anderen Ausgänge angesteuert. Die Hoflampe hat eine Besonderheit: wenn von den restlichen 3 Lampen auch die letzte ausgeht, so schaltet nachts die Hofbeleuchtung für 15 min ein, auch wenn zwischenzeitlich wieder eine der anderen Lampen angeht. Der Phantasie sind praktisch kaum Grenzen gesetzt.

Der Türkontakt wird von Modul 170 erkannt und als Signal „Haustür“ zur Verfügung gestellt. Der Zählermodul 172 zählt bei jeder Betätigung +1. Der aktuelle Stand wird von den Anzeigen dargestellt. Modul 171 verhindert das unbeabsichtigte Zählen, wenn die Tür mehrfach in kurzer Zeit geöffnet und geschlossen wird.

hauscomp.ini (Fortsetzung)

```
[Programmierung SPS]
z=151  3 Tag          1   0   4  20  1000            /Tageslicht heller 20% für min. 16,7min
z=152  5 t1600_2135  16  0  21  35                   /Ein: im Zeitraum von 16:00 bis 21:35
z=153  5 t2120_2135  21 20  21  35                   /Ein: im Zeitraum von 21:20 bis 21:35
z=154 42 Zufall1     120  1800 7200 10800           /EIN: 2min-30min  AUS: 2h-3h
z=155 42 Zufall2     120  1200 3000  5400           /EIN: 2min-20min   AUS: 50min-1,5h
z=156 31 Stube       -151    165   1     3  1       /Stube=Ein, wenn kein Tag UND verkn4 UND 1
z=157 31 Küche       -151    152  155     3  3      /Küche=Ein, wenn kein Tag und (52) UND Zufall2
z=158 42 Zufall3     3000 7200  180  1800           /EIN: 30min-2h     AUS: 3min-30min
z=159 12 Verkn1      -156   -157  158    158        /EIN: wenn Stube und Küche AUS sind UND Zufall3

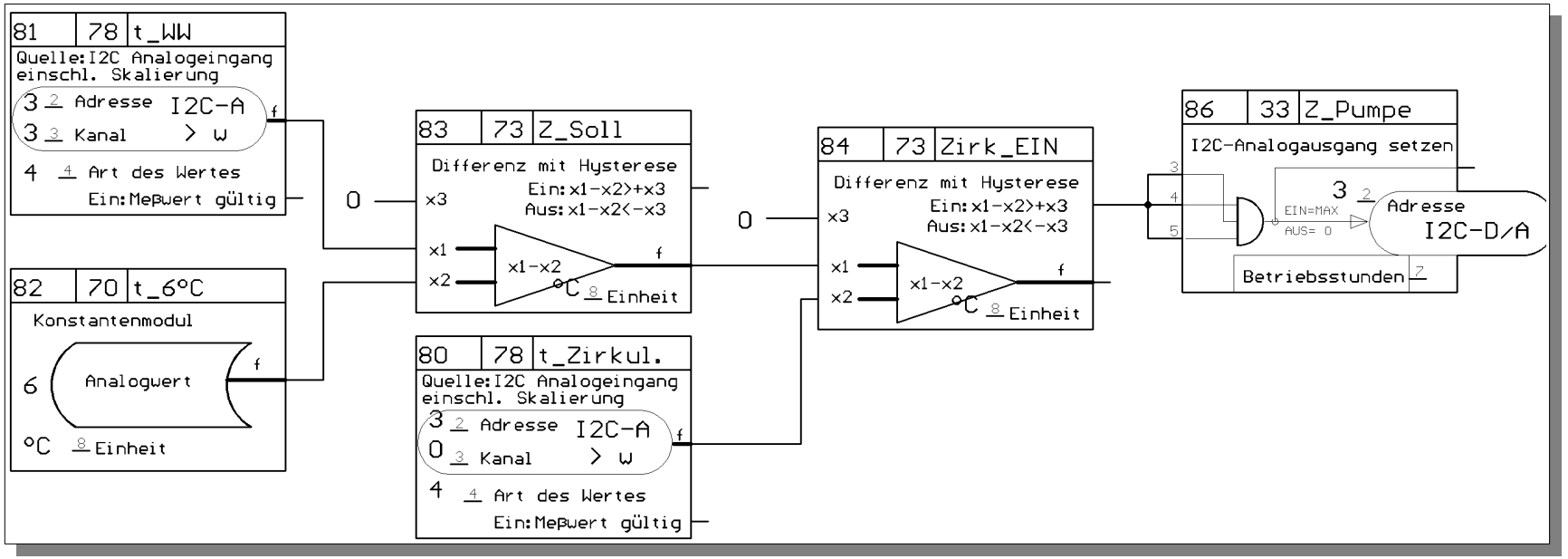
z=160 31 Boden       -151    159  152     3  0      /Boden=Ein, wenn kein Tag UND verkn1 UND (52)
z=161 12 Verkn2      -160   -156   0   -157         /EIN: wenn Stube UND Küche UND Boden AUS sind
z=162 61 Verz1        161   900                     /EIN: für 15min, wenn (61) ausgeht
z=163 12 Verkn3       161    161  162     1         /EIN: wenn verz1 ODER verkn2
z=164 31 Hof         -151    163  152     3  2      /Hof=Ein, wenn kein Tag UND Verkn3 UND (52)
z=165 12 verkn4       154    152  153     1         /EIN: wenn (Zufall1 UND t1) ODER t2
z=170  2 Haustür       2     0   2                  /EIN: wenn Tür für min 2s schaltet
z=171 61 VerzTür      170    160                    /Verlängerung Türsignal auf 60s
z=172 40 Türzähler    171     0   0    1000         /Türzähler
z=173 42 Zufallx     1  1 2 2
```

6.3. Die Steuerung einer Warmwasser-Zirkulationspumpe

Aufgabe: In einem Haus soll die Warmwasser-Zirkulationspumpe so gesteuert werden, dass sie einerseits möglichst wenig läuft und anderseits warmes Wasser immer sofort verfügbar ist.

- Notwendige Hardware: 1 x PCF8591 mit zwei Temperatursensoren an den Analogeingängen 0 und 3. Über den D/A-Ausgang und einem Halbleiter-Relais wird die Zirkulationspumpe geschaltet. Die IC-Adresse wird mit „3“ festgelegt.
- Grafischer Entwurf:

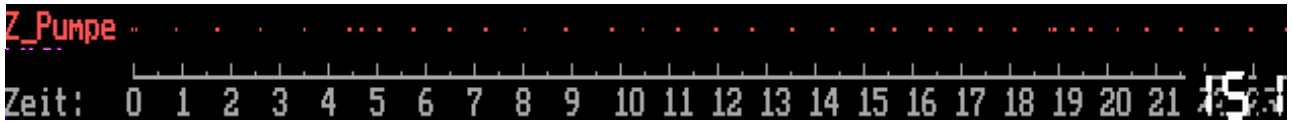
  Die Module 80 und 81 liefern die Temperatur des Warmwasser-Speichers und die Temperatur der Zirkulationsleitung. Nach Abzug von 6°C von der Warmwasser-Temperatur erfolgt der Vergleich mit der Temperatur der Zirkulationsleitung. Ist diese kleiner, so wird die Pumpe eingeschaltet. Die Pumpe pumpt dann warmes Wasser zyklisch durch das Rohrsystem und erwärmt dabei wieder die Zirkulationsleitung auf die Temperatur des warmen Wassers. Der Kompromiss „möglichst wenig“ wird mit den erwähnten 6°C erreicht.



- *hauscomp.ini* (Fragment):

```
...
Sensor=Zirkulation   øC  3 0  30   -24.0  33      4.2   84    44.4   155
Sensor=Warmwasser    øC  3 3  30   -24.0  33      4.2   84    44.4   156
Analog=Zirkulation  E/A  3     0       0   0        1  255
...
graph_b=    8     8 86  12   ;Zirkulationspumpe in Binärgrafik anzeigen
...
/SPS-Zirkulationspumpe
z=80 78 t_Zirkul    3 0 4              /Temperatur Zirkulationsleitung messen
z=81 78 t_WW        3 3 4              /Temperatur Warmwasser messen
z=82 70 t_6øC       øC 6               /Differenztemperatur vorgeben
z=83 73 Z_Soll      øC 81 82 0         /Sollwert Zirkulationstemperatur errechnen
z=84 73 Zirk_EIN    øC 83 80 0         /2-Punkt Regler ohne Hysteresis (Zirkulation)
z=86 33 Z_Pumpe     84 84 84 3         /Zirkulationspumpe ansteuern
...
```

In der Praxis läuft diese Steuerung seit Jahren störungsfrei. Die Pumpe ist etwa 1-2 mal in der Stunde für maximal 40 s aktiv. Durch den damit verringerten Strombedarf der Pumpe hat sich der Hauscomputer bei mir bereits amortisiert. Auf dem Screenshot der 24h Grafik sind deutlich die Einschaltzeiten zu sehen, sogar der erhöhte Warmwasserbedarf gegen 4:30 Uhr und 19:00 Uhr ist erkennbar. Dieses einfache Beispiel lässt sich durch zusätzliche Freigabesignale (die am SPS Modul 86 angeschlossen werden), weiter verbessern. So könnte die Pumpe z. B. in der Nacht oder bei Abwesenheit blockiert werden.



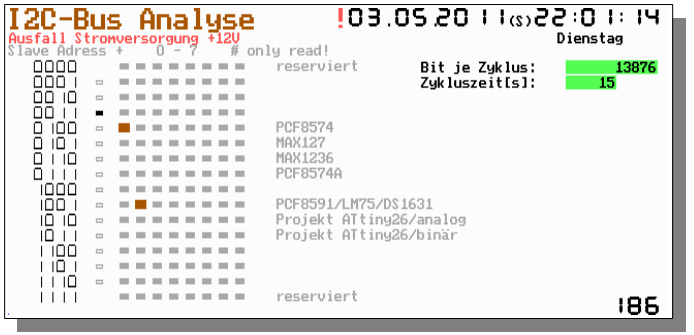
6.4. Test/ Monitoring I²C-Hardware



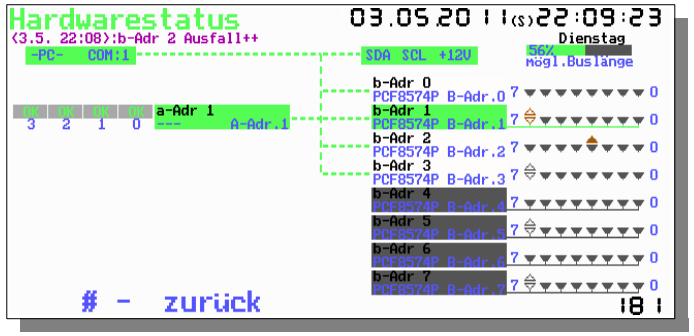
Aufgabe: Beim Aufbau von I2C-Hardware soll mit dem PC die Funktion des I²C-Busses überwacht werden; welche Komponenten melden sich mit welcher Adresse an, welche Analogkanäle liefern welche Werte usw.

- Überwachbare ICs: PCF8574, PCF8591, LM75, MAX6625, MAX7500, DS1631, DS1631A, DS1731, MAX1236, ATtiny26 (mit spezieller Firmware)
- Notwendige Hardware: PC mit Anschluss- oder Busadapter. Für die nachfolgende Beschreibung waren zusätzlich ein PCF8574 und ein PCF8591 jeweils mit der Adresse 1 am Bus angeschlossen.
- Start des Programms:

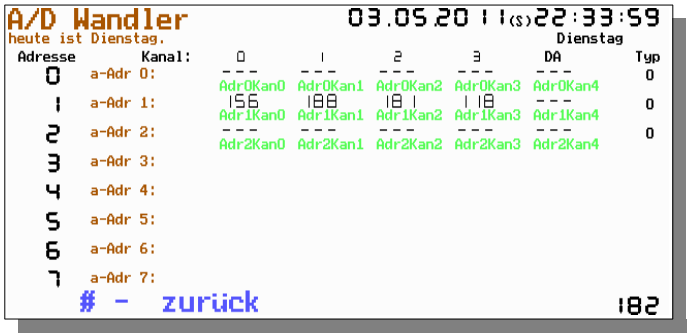
  Einfach die ini Dateien im Verzeichnis Beispiel_6.4 in das Verzeichnis „Software“ kopieren und die *hauscomp.exe* starten. Bitte die Besonderheiten der Busfunktion unter den verschiedenen Betriebssystemen beachten! Letztlich startet das Programm automatisch mit der Testseite 18. Man hat nun die Möglichkeit verschiedene Aspekte des I2C-Busses anzuzeigen:



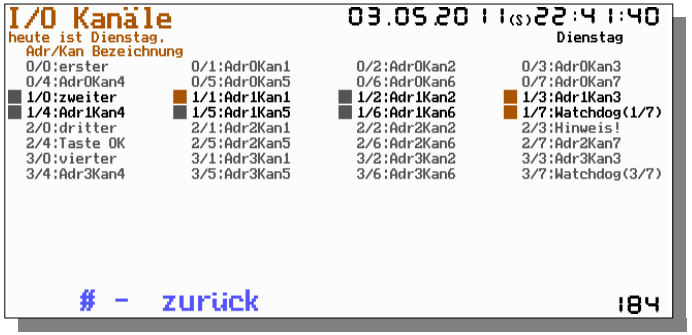
- Über das Menü I2C-Busanalyse erhält man zunächst eine Übersicht über alle am Bus angeschlossenen ICs, ob sie vom Programm unterstützt werden, oder nicht. Einträge in den ini- Dateien sind dabei ohne Belang. Die COM Adresse sollte schon stimmen.
- Das Menü Hardwarestatus dagegen zeigt die Adressen und Kanäle, die am Bus erwartet werden und sich tatsächlich melden. Die Einträge in der Hardware- und *hauscomp.ini* werden dabei weitgehend ausgewertet. Die Dateien im Beispiel 4 sind bewusst unvollständig gehalten, um die verschiedenen Möglichkeiten zu zeigen. So werden z. B. alle vorhandenen PCF8591 ohne weiteren Scriptbefehl bis zur Adresse 3 angezeigt, während bei den PCF8574 zumindest ein Kanal je Adresse mit dem Scriptbefehl „Binär=...“ initialisiert werden muss. Im Hintergrund werden alle PCF8574 mit der Adresse 0 bis „Anzahl_PCF8574-1“ bzw. alle PCF8591 mit der Adresse 0 bis „Anzahl_PCF8591-1“ lt. *hardware.ini* ständig abgefragt und angezeigt. Die Anzahl der analogen PCF8591 auf die tatsächlich vorhandenen zu begrenzen, macht dabei Sinn, da ein Abfragezyklus aller möglichen 8 „analogen“ Schaltkreise ca. 10 s dauert. Die nicht abgefragten, aber lt. *hauscomp.ini* angegebenen ICs sind grau unterlegt, alle ICs, die sich dagegen korrekt melden, grün.



- Im Menü A/D Wandler sind die digitalen Kodewerte, also nicht die physikalischen (!) nach analogen Adressen und Kanälen gelistet. Liefert ein Analog-Kanal keinen oder einen Kodewert von 0 oder 255 zurück, so wird er mit 3 Bindestrichen als fehlerhaft markiert. Andere als der genannten PCF8591 müssen in der *hauscomp.ini* mit dem entsprechenden „Sensor..“ Befehl initialisiert werden um ebenfalls in der Anzeige zu erscheinen.



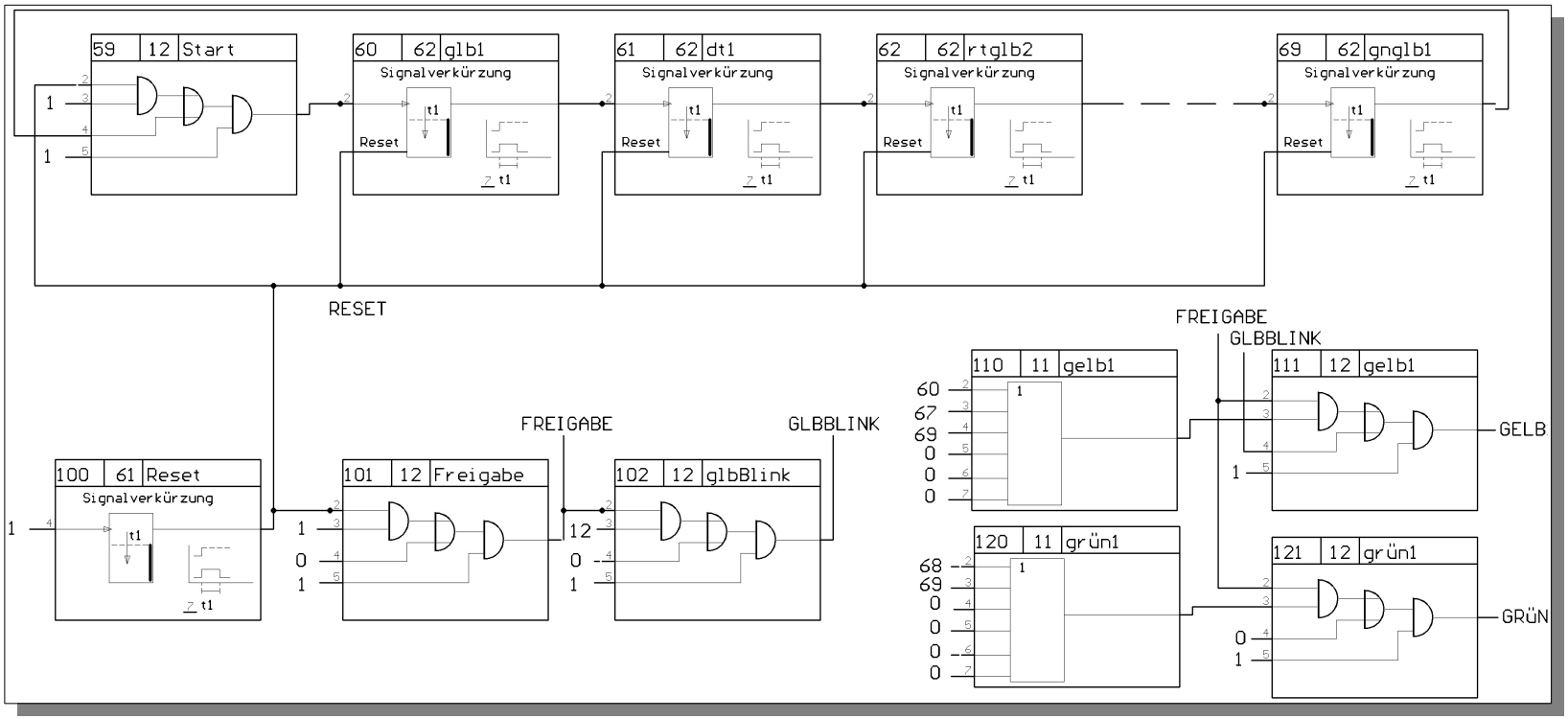
- Das Menü I/O Kanäle listet alle binären Adressen und Kanäle mit ihren programmierten Namen. Kanäle, die sich melden, besitzen zusätzlich ein Statussymbol, ob sie „ein“ oder „aus“ sind.



6.5. Eine einfache Ampelsteuerung (keine Hardware notwendig)

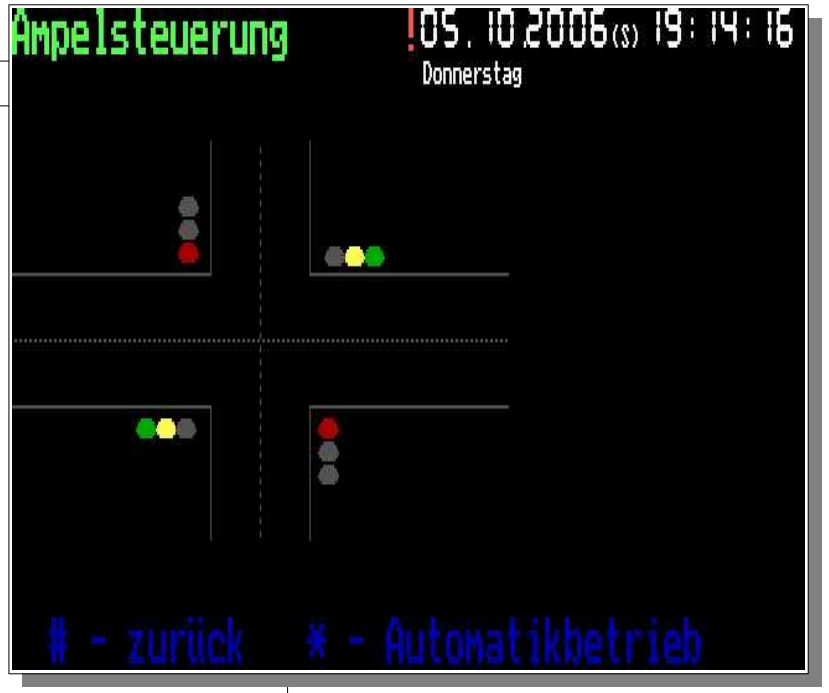
Aufgabe: Eine einfache Verkehrsampel soll durch eine Ablaufsteuerung realisiert werden.

Lösung: Durch aneinanderreihen von Monoflops (Modultyp 62) wird eine Ablaufsequenz generiert. Eine Auswertung dieser Zustände erfolgt durch die SPS Module Nr. 110 ff. Die Module 100-102 stellen einen ordentlichen Programmstart sicher. Das Freigabesignal dient zum Umtasten der Ampel zwischen normalen Betrieb und Blinken aller gelben Ampeln. Beispielhaft sind nur die Ansteuerungen der gelben und grünen Leuchten von Ampel 1 gezeichnet.



Die grafische Darstellung erfolgt durch entsprechende Kreise und Linien auf der Anwendungs-Seite 110. Die Steuerung kann selbstverständlich auch an entsprechende Hardware angeschlossen oder anderweitig modifiziert werden.

Hinweis: Die Module 59..69 wurden in 259..269 geändert, damit die aktuelle *hauscomp.exe* läuft (Februar 2018).



hardware.ini	***hauscomp.ini***

```
COM=0
Laufschrift=off
Sommerzeit=off
Seitenwechsel=59
```

```
/Softwarescript für Ampelsteuerung
** Norm=ANSI   (äöüÄÖÜß - Umlaute lesbar)**
** Norm=ASCII  (Ž™š„”□á - Umlaute lesbar)**
Norm=ANSI
Anzeigeseiten=110 110 110
/Anwendungsseite 110
Menü1x= 110 Ampelsteuerung
Anzeige=110 15    0   0   0   80   8 0 160  80 0 /Kreuzung
Anzeige=110 15    0   0 240   80   8 0 400  80 0 /Kreuzung
Anzeige=110 15    0   0   0  120   8 0 160 120 0 /Kreuzung
Anzeige=110 15    0   0 240  120   8 0 400 120 0 /Kreuzung
Anzeige=110 15    0   0   0  100   8 1 400 100 0 /Kreuzung
Anzeige=110 15    0   0 160   40   8 0 160  80 0 /Kreuzung
Anzeige=110 15    0   0 160  120   8 0 160 160 0 /Kreuzung
Anzeige=110 15    0   0 240   40   8 0 240  80 0 /Kreuzung
Anzeige=110 15    0   0 240  120   8 0 240 160 0 /Kreuzung
Anzeige=110 15    0   0 200   40   8 1 200 160 0 /Kreuzung
Anzeige=110 17    8 121  142  60  1 2 8 02 8 /gn  Ampel oben
Anzeige=110 17    8 111  142  67  1 2 8 14 8 /glb
Anzeige=110 17    8 133  142  74  1 2 8 04 8 /rt
Anzeige=110 17    8 121  255 141  1 2 8 02 8 /gn  Ampel unten
Anzeige=110 17    8 111  255 134  1 2 8 14 8 /glb
Anzeige=110 17    8 133  255 127  1 2 8 04 8 /rt
Anzeige=110 17    8 151  292  75  1 2 8 02 8 /gn  Ampel rechts
Anzeige=110 17    8 141  276  75  1 2 8 14 8 /glb
Anzeige=110 17    8 162  260  75  1 2 8 04 8 /rt
Anzeige=110 17    8 151  108 127  1 2 8 02 8 /gn  Ampel links
Anzeige=110 17    8 141  124 127  1 2 8 14 8 /glb
Anzeige=110 17    8 162  140 127  1 2 8 04 8 /rt
/SPS
z=259 12 Start    100   1 269 1   /Ablaufsequenz startet hier
z=260 62 glb1     -259 100 3
z=261 62 dt1      -260 100 4
z=262 62 rtglb2   -261 100 2
z=263 62 gn2      -262 100 5
z=264 62 gnglb2   -263 100 3
z=265 62 glb2     -264 100 3
z=266 62 dt2      -265 100 4
z=267 62 rtglb1   -266 100 2
z=268 62 gn1      -267 100 4
z=269 62 gnglb1   -268 100 3
z=100 61 Reset     1 10              /für 10s nach Programmstart Reset
z=101 12 Freigabe  -100 1  0 1
z=102 12 glbBlink  -101 12 0 1
z=110 11 gelb1  260  267  269  0  0  0  /ab hier entsprechende Signalverknüpfungen
z=111 12 gelb1 101 110 102  1
z=120 11 grün1  268  269   0  0  0  0
z=121 12 grün1 101 120   0  1
z=131 11 rota   262  263  264 265 266  0
z=132 11 rot1  131  261  267  0  0  0
z=133 12 rot1  101 132   0  1
z=140 11 gelb2  262  264  265  0  0  0
z=141 12 gelb2 101 140 102  1
z=150 11 grün2  263  264   0  0  0  0
z=151 12 grün2 101 150   0  1
z=160 11 rotb   267  268  269 260 261  0
z=161 11 rot2  160  262  266  0  0  0
z=162 12 rot2  101 161   0  1
```

6.6. Das Webinterface

Wird der PC mit dem Programm *hauscomp.exe* in einem Netzwerk betrieben, gibt es folgende Möglichkeiten der Kommunikation mit diesem Programm:

1. Über das Netzwerk die Dateien *hardware.ini* oder *hauscomp.ini* im aktuellen Verzeichnis der laufenden *hauscomp.exe* editieren und speichern. Um Mitternacht werden die Änderungen im Rahmen des automatischen Tagesabschlusses in das laufende Programm übernommen. Diese Möglichkeit ist die wichtigste Form der Wartung und weiteren Ausbaus des Hauscomputers.
2. Über das Netzwerk alle vom Programm erzeugten Dateien (logs usw.) herunterladen und mit externer Software (EXCEL und Co) auswerten.
3. Mit dem Befehl „Netzwerk=“ (in der *hardware.ini*) und einer Index.htm im aktuellen Verzeichnis der *hauscomp.exe* ist es möglich, über ein Netzwerk aktuelle Informationen auf entfernte Rechner zu übertragen. Das Programm *hauscomp.exe* aktualisiert dabei regelmäßig die Dateien „Auto1-3.bmp“, „aktuell.txt“, „sps.txt“ und „Tagesinf.txt“ im gleichen Verzeichnis im angegebenen Intervall. In der Praxis benötigt man dann nur noch einen Webbrowser (der Zugriff auf dieses Verzeichnis hat) um live am Geschehen des Hauses dabei zu sein. Der umgekehrte Weg – die Fernsteuerung des Hauses ist damit nicht möglich. Mit geeigneten Programmen (Homepage Gestaltung) oder einem einfachen Texteditor kann die „index.htm“ eigenen Bedürfnissen angepasst werden.
4. Mit dem Befehl „live= on“ (in der *hardware.ini*) wird die Datei „taste.txt“ vom Programm *hauscomp.exe* regelmäßig auf Inhalt abgefragt. Wird über das Netzwerk ein Tastenkode (siehe Abschnitt 2.2) in diese Datei geschrieben, so wird der entsprechend zugehörige Befehl ausgeführt und anschließend der Dateiinhalt der Datei „taste.txt“ wieder gelöscht. Gleichzeitig ist 2 s später ein aktuelles Screenshot über die Datei „live.bmp“ verfügbar. Es sind alle Tastenbefehle, einschließlich der Menüführung ausführbar.
5. Fernsteuerung des abgesetzten Rechners über einen VNC Viewer. Damit lässt sich das Programm über das Netzwerk bedienen. Siehe auch unter Pkt. 1.10.4.

Voraussetzung für die Netzwerkfunktionen ist ein Betriebssystem, das netzwerkfähig ist. Hinweise dazu siehe auch Pkt. 1.9. - 1.11.

Die Dateien „Auto1-3.bmp“ stellen ein Screenshot der drei möglichen frei programmierbaren Autoseiten dar. Erfolgreich erprobt wurde das Interface mit DOS98, win98SE, winXP-professional sowie in der DosBox und Dosemu unter LINUX. Praktisch installiert man ein bootfähiges Betriebssystem mit den entsprechenden Netzwerktreibern auf dem Steuerrechner, gibt das Verzeichnis mit der Hauscomputer-Software für den Netzwerkzugriff frei und fügt die *hauscomp.exe* der Autostartgruppe hinzu. Bewährt hat sich ein Rechner IGEL 3/2 – damit erhält man einen guten Kompromiss zwischen Rechenleistung, Stromaufnahme und Preis. Zu beachten ist, dass die Lebensdauer von motorlosen Festplatten (z. B. Flash) von der Zahl der schreibenden Zugriffe abhängt. Die Aktualisierung mittels „Netzwerk=“ sollte also mehr im Minutentakt als im Sekundenbereich liegen.

In [6] werden detailliert alle Möglichkeiten des erweiterten Webinterfaces bis hin zur CGI-Steuerung und Verschlüsselung mit SSL beschrieben.

Darstellung der Datei „Index.htm“ im Webbrowser:

Hauscomputer - Webinterface

Tagesinformationen:

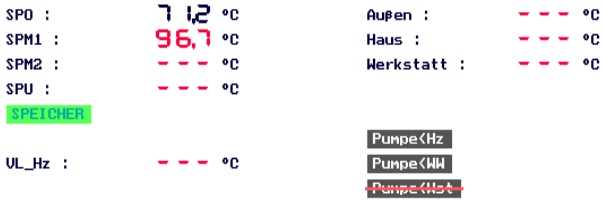
```
heute ist Donnerstag, Sommeranfang.
Tageseigenschaft:        2 (Ferien)
Wir haben Mitteleuropäische Sommerzeit, d.h. MEZ=MESZ-1 !
Standortkoordinaten:     12° 13' (öL)  53° 57' (nB)
Sonnenaufgang heute:     04:38 Uhr MESZ
Sonnenuntergang heute:   21:46 Uhr MESZ
Sonne im Süden heute:    59.44° um 13:01 Uhr MESZ
Sonnenhöchststand heute: 59.51° um 13:12 Uhr MESZ
UV-Index bei wolkenlosem Himmel (Mitteleuropa) heute Mittag maximal: 7.56
        das bedeutet Sonnenbrand in der Mittagszeit nach ca.
                     Hauttyp I  :  14 min
                     Hauttyp II :  18 min
                     Hauttyp III:  25 min
                     Hauttyp IV :  34 min
Das Programm läuft seit     0 Tagen.
```

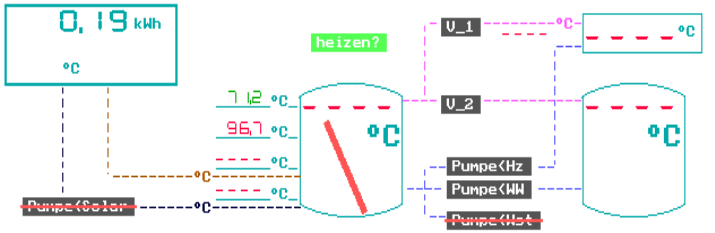
letzte Meldungen:

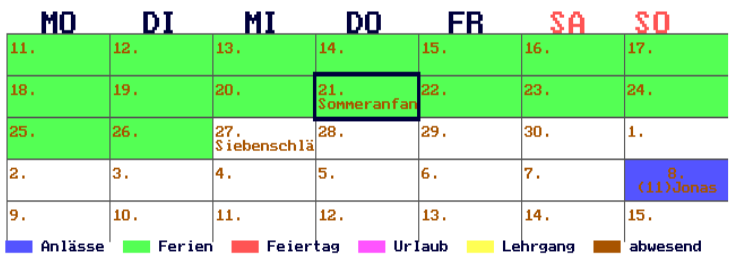
```
. Achtung, Problem! Totalausfall Hausbus: -Ausfall Stromversorgung +12V- Achtung, Hinweis:
<21.6. 15:24>:SPM2 (53) Wert größer MAX++
<21.6. 15:24>:SPU (54) Wert größer MAX++
<21.6. 15:24>:WW_Speicher (59) Wert größer MAX++
<21.6. 15:24>:VL_Haus (58) Wert größer MAX++
<21.6. 15:24>:Raumtemp (57) Wert größer MAX++

letzte Aktualisierung dieser Angaben: <21.6. 15:26>
```

Übersichten:







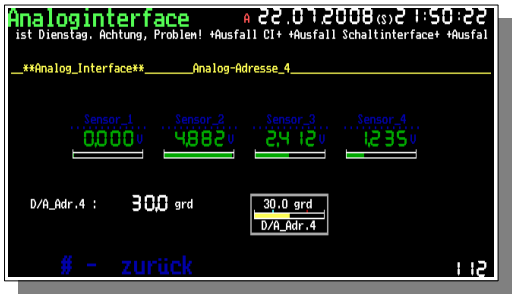

```
18 100 Zyklen      [X]  -0.880 c%
19 Taste OK        [X]  0.055
20 Hinweis!        [ ]  55.000
21 min-Zyklus      [ ]  0.000
22 3s-Zyklus       [ ]  3.021
23 Farbinvert      [ ]  -1.000
24 PCF8574P (0) OK [ ]  -2.000
```

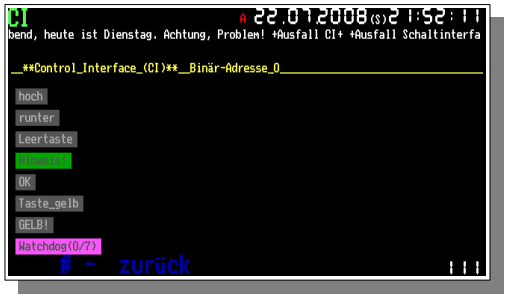
6.7. Ein Testprogramm für die Funktionsprüfung aller Interfaces (außer ATtiny26)

Im Verzeichnis /Beispiel_6.7/ befinden sich die ini-Dateien für den Test aller Interface aus dem Hardware-Band 1. Sie sind nur in das Verzeichnis ../Software/ zu kopieren und das Programm zu starten. Das Testprogramm erwartet folgende Hardware-Konfiguration (gegebenenfalls über Jumper entsprechend einstellen) :

Interface	*Analog-Adresse(n)*	*Binär-Adresse(n)*	*Bemerkungen*
Control Interface (CI)	**-**	**0**	
Analogerweiterung (Interface)	**4**	**-**	ohne Sensoren überprüfbar
A+D Interface	**1**	**2+3**	Mit 4xKTY81-210, Rn=24k, Rb=2,1k
Schaltinterface	**-**	**0+1**	Enthält auch ein CI
8-Kanal-PWM-Interface	**2+3**	**-**	PWM Ausgänge variieren ununterbrochen; Sensoren wie in *hauscomp.ini* anschließen!
PT1000 Präzisionsinterface	**0**	**-**	
Bussensor LM75	**5**	**-**	
Bussensor DS1636	**6**	**-**	







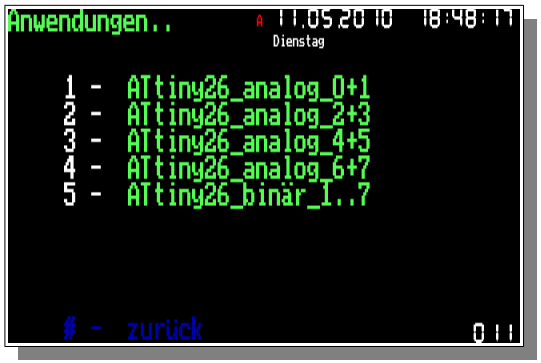
Nach Programmstart erreicht man über ein Auswahlmenü die einzelnen Teste. Um Schalt-Aktivitäten auszulösen ist z.T. eine Funktionstaste zu aktivieren.

6.8. Ein Testprogramm für die Funktionsprüfung aller Interfaces mit ATtiny26

Im Verzeichnis /Beispiel_6.8/ befinden sich die ini-Dateien für den Test aller Interface aus dem Hardware-Band 0. Sie sind nur in das Verzeichnis ../Software/ zu kopieren und das Programm entsprechend zu starten. Ggf. muss in der *hardware.ini* der COM Anschluss korrigiert werden. Der Test erwartet folgende Hardware-Konfiguration:

Interface	*Analog-Adresse(n)*	*Binär-Adresse(n)*	*Bemerkungen*
Analoges Interface	**0+1**	-	Mit 5-8 x KTY81-122*
Analoges Interface	**2+3**	-	Mit 5-8 x KTY81-122*
Analoges Interface	**4+5**	-	Mit 5-8 x KTY81-122*
Analoges Interface	**6+7**	-	Mit 5-8 x KTY81-122*
Binäres Interface	-	**1..7**	Test Adresse 0 ist in *hauscomp.ini* vorbereitet aber inaktiv (eingerückt)

(*) je nach Konfiguration







Hinweis: Der Zustand der Eingänge E1 bzw. E2 wird erst nach dem Auslesen von min. 10 Messwerten abgefragt!
Der Test der analogen Interface enthält ein Sub- Menü für die Ansteuerung der PWM Kanäle. Dabei sind verschiedene Modi möglich, u.a. die Vorgabe eines Wertes von Hand. Der Kanal K3U ist jeweils für die Frequenzmessung vorkonfiguriert.

Für den Test der binären Interface ist Punkt 5 der Anwendungsseite zuständig. Hier sind alle 56 Kanäle für 7 binäre Interface gelistet. Nach zweimaligen Betätigen der Taste „0“ wird ein automatischer Test gestartet, bei dem alle Kanäle sequentiell durchgeschaltet und der Status angezeigt wird. Wird an einen Eingang (Kanal 6 bzw. 7) eine Spannung gelegt, so wird ebenfalls eine Anzeige generiert. Werden während der Testphase Änderungen an der Hardware vorgenommen, kann es sein, dass die Anzeigen scheinbar nicht aktualisiert werden. In diesem Fall muss man kurz auf eine andere Anzeigeseite und wieder zurückwechseln. Dabei erfolgt eine Löschung des Bildschirmes sowie die nachfolgende Aktualisierung.
Für die Überprüfung der Funktion des Signals „Hinweis!“ (Status LED am Interface blinkt grün) ist die Binär-Adresse 0 mit einem PCF8574 initialisiert, der aber nicht angeschlossen sein muss.



Ab Version 6.0.37 wurde im Testprogramm noch eine Seite „Textanzeigen“ eingefügt. Sie enthält praktisch alle verfügbaren Tastaturkodierungen für spezielle Fälle. Falls also mal ein Zeichen benötigt wird, kann man es ohne umständliche Suche auf der Tastatur einfach über „copy and paste“ in die eigene *hauscomp.ini* übernehmen. Einzelne Unterschiede in der grafischen Darstellung zum Dateilayout sind in den unterschiedlichen ANSI/ASCII Standards begründet, nicht alles ist kompatibel...



6.9. Auswertung von Heizungs- oder Solardaten

Beim Einsatz der Software *hauscomp.exe* für die Steuerung oder das Monitoring im Bereich von Heizungs- bzw. Solaranlagen fallen gewöhnlich eine große Anzahl spezifischer Daten wie Temperaturen, Durchflussmengen oder Schaltzustände an. Für die Auswertung dieser Daten gibt es bereits fertige Software am Markt. Es lag also nahe, lediglich eine kompatible Log-Datei zu erzeugen, die dann entsprechend ausgewertet wird. Ausgewählt habe ich die frei verfügbare Software „Solvis-Viewer“ (http://www.dialog-edv.de/public/sc2/) von Oliver Sörensen, die zum jetzigen Zeitpunkt (Juli 2013) in der Version 0.67 vorlag. Als zweite Variante kann ich auf das offene gnuplot Projekt verweisen (http://www.gnuplot.info/) aktuell in der Version 4.6, dass eine bessere grafische Darstellbarkeit gegenüber „calc“ oder EXCEL ermöglicht. Einzige derzeitige Einschränkung zur Log- Datei: unter DOS wird der Dateiname mit Großbuchstaben erzeugt, der „Solvis-Viewer“ erwartet aber „mi...“, also kleingeschrieben. Somit müssen die Log- Dateien von Hand umbenannt werden. Unter Linux tritt das Problem natürlich nicht auf... Eine Datei ist knapp 2 MB groß. Bei 52 Wochen sind so mit etwa 100 MB Datenfülle je Jahr zu rechnen. Im Verzeichnis „.. \Beispiel 6.9\“ befinden sich die Beispiel-Dateien „mi130429.txt“ - von einem originalen Solvis-Regler erzeugt, der Kopf wurde manuell mit „#“ eingerückt, sowie die Datei „mi130722.txt“, die mit *hauscomp.exe* erzeugt wurde und für Testzwecke nur die Datensätze von 1 ½ Tagen enthält.
Jetzt zu einem praxisnahen Beispiel...
Grundlage für die Datensammlung ist das Projekt „kombinierte Heizungssteuerung“. Damit ist bereits eine funktionierende Steuerung mit den entsprechenden SPS-Modulen gegeben. Zusätzlich wurden in der *hauscomp.ini* und *hardware.ini* die entsprechenden Zeilen für die Erzeugung der speziellen Log- Datei eingetragen:

hardware.ini	***hauscomp.ini***
HzgLog=on	HzgLogS01= 344 343 393 404 405 253 407 252 374 271 223 315 HzgLogS13= 402 353 374 353 403 0 400 388 401 305 388 358 HzgLogA01= 265 341 378 390 226 350 359 318 319 274 300 HzgLogA12= 329 328 364 192 212 215 95 299 301 259 269

Somit wird wöchentlich eine Text-Datei (z. B. „mi130722.txt“) mit 46 (minütig) aufgezeichneten Parametern in folgender Form erzeugt:

```
#Logformat:        4110000
#Softwareversion:          6.0.38B8
#System:           Hauscomputer
#Datum Zeit       S1      S2      S3      S4      S5      S6      S7      S8      S9      S10             …
#Datum Zeit       °C#0.1  °C#0.1  °C#0.1  °C#0.1  °C#0.1  °C#0.1  hPa#0.1 °C#0.1  °C#0.1  °C#0.1          …
…
22.07.13  12:22:09        667     568     560     309     681     602     3000    820     273             …
22.07.13  12:23:09        669     568     560     333     752     585     3000    835     272             …
…
```

Welche Parameter sich tatsächlich hinter den programmierten SPS Modulen verbergen, kann man sich zur Kontrolle unter „Funktionen..“, „HzgLog-Parameter“ anzeigen lassen:

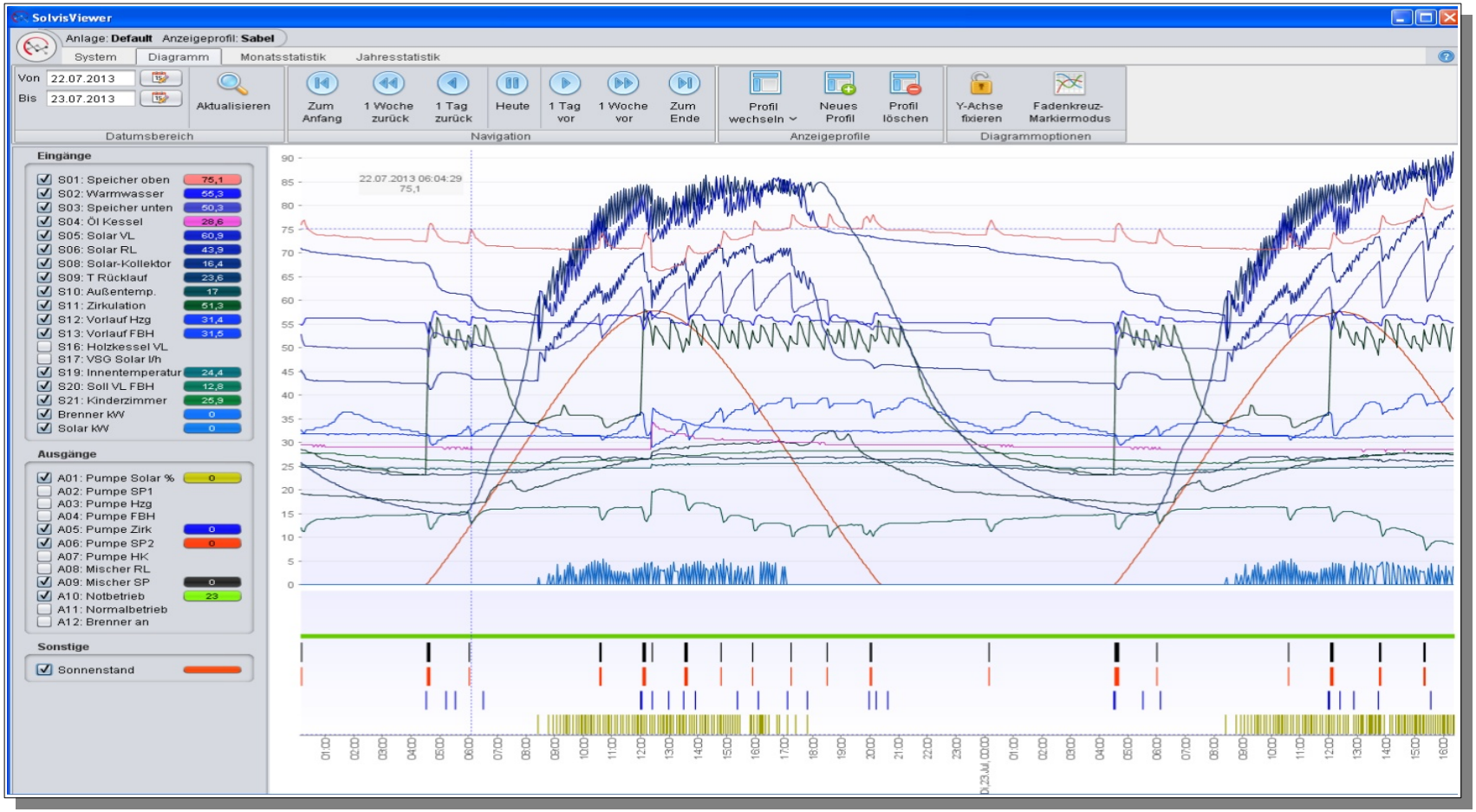
Die weiß dargestellte Axx Parameter bedeuten dabei die Speicherung des E/A Zustandes des entsprechenden SPS-Moduls, die blau dargestellten die Speicherung des analogen Wertes. Die Zuordnung der Parameter sollte sich aus Gründen der Kompatibilität in etwa an den Vereinbarungen in der folgenden Tabelle halten. Auf jeden Fall ist der zulässige Wertebereich zu beachten:



Par	Bedeutung	Bereich	Einheit	Par	Bedeutung	Bereich	Einheit	Par	Bedeutung	Bereich	Einheit
S01	Speicher oben	-99..+999	°C	S16	Temp. Holzkessel	-99..+999	°C	A07	Ladepumpe Holzofen	0..+100	%
S02	Warmwasser	-99..+999	°C	S17	Durchfluss Solar	-999..+9999	l/h	A08	HK 1 Mischer auf	0..+100	%
S03	Speicher unten	-99..+999	°C	S18	Durchfluss Wasser	-99..+999	l/min	A09	HK 1 Mischer zu	0..+100	%
S04	Hzg Puffer oben	-99..+999	°C	S19	Raumfühler 1	-99..+999	°C	A10			
S05	Solar VL	-99..+999	°C	S20	Raumfühler 2	-99..+999	°C	A11			
S06	Solar RL	-99..+999	°C	S21	Raumfühler 3	-99..+999	°C	A12	Brenner an	0..+100	%
S07	Solar Druck	-99..+999	°C	S22		-99..+999		A13	Brenner Stufe 2 an	0..+100	%
S08	Kollektor	-99..+999	°C	S23		-99..+999		A14			
S09	Hzg Puffer unten	-99..+999	°C	S24		-99..+999		A15		-99..+999	
S10	Außentemperatur	-99..+999	°C					A16	Brenner Leistung	-99..+999	
S11	Zirkulation	-99..+999	°C	A01	Pumpe Solar	0..+100	%	A17		-99..+999	
S12	Vorlauf HK 1	-99..+999	°C	A02	Pumpe WW	0..+400	%	A18			
S13	Vorlauf HK 2	-99..+999	°C	A03	Pumpe HK 1	0..+100	%	A19			
S14	Vorlauf Holzkessel	-99..+999	°C	A04	Pumpe HK 2	0..+100	%	A20			
S15	Rücklauf Holzkessel	-99..+999	°C	A05	Pumpe Zirkulation	0..+100	%	A21			
				A06	Pumpe HK 3	0..+100	%	A22			

Nun zur Auswertung mit dem Programm Solvis-Viewer. Nach Programmstart wird als Erstes unter dem Menüpunkt „Diagramm“ ein eigenes Profil angelegt. Das hat den Vorteil, dass später die Zuordnung der Parameter, d. h. die Namen und eventuelle Formeln nicht neu eingegeben

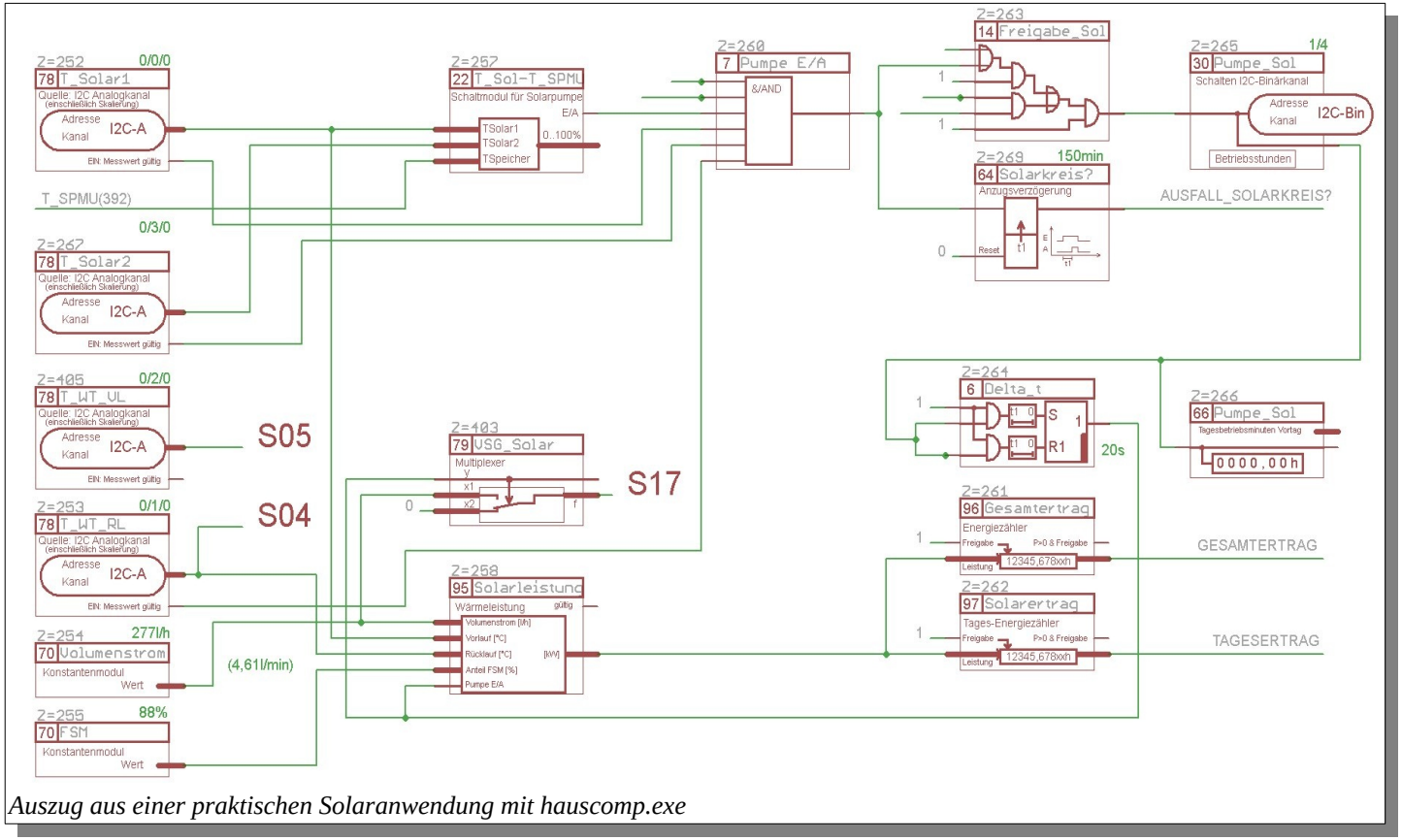
werden müssen. Unter dem Menüpunkt „System“ werden nun die eigenen Ein- und Ausgänge definiert. Im Anschluss kann man die Log- Datei unter dem Punkt „Minuten Log Datei lesen“ laden. Mindestens 6 Datensätze müssen in der Datei enthalten sein. Für die Auswertung stehen die Funktionen „Diagramm“, „Monatsstatistik“ und „Jahresstatistik“ zur Verfügung. Nachfolgend beispielhaft das „Diagramm“ mit der Datei „mi130722.txt“. Es lassen sich die einzelnen Signale ein- und ausblenden und man kann auch sehr gut hineinzoomen – kurz gesagt, eine super Auswertung!



Für den Vergleich der Effizienz verschiedener Energieträger kann man die Monats- und Jahresauswertung heranziehen. In diesem Beispiel wurde

die Solarleistung wie folgt ermittelt: S05 und S06 sind die präzisen Temperaturen unmittelbar am Ein- und Ausgang des Wärmetauschers (Pufferspeicher). Zusätzlich wird der, von der SPS ermittelte Solardurchfluss, per S17 an den Solvis-Viewer übergeben. Sollte man keinen Durchflussmesser haben und nur eine Solarpumpe im Start-Stopp Betrieb betreiben, so kann man sich wie folgt behelfen:

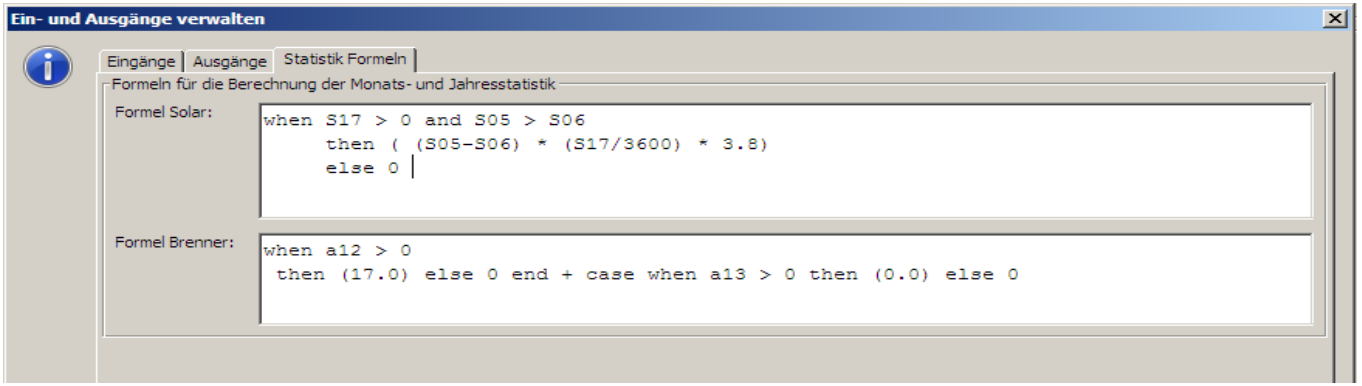
Die ohnehin vorhandene Konstante des Volumenstroms (254) wird in der SPS an einen Multiplexer gegeben, der diesen Wert nur dann weitergibt, wenn die Solarpumpe „Ein“ ist. Eine Zeitverzögerung (261) berücksichtigt, dass die Solarflüssigkeit nicht sofort anläuft bzw. stehen bleibt.



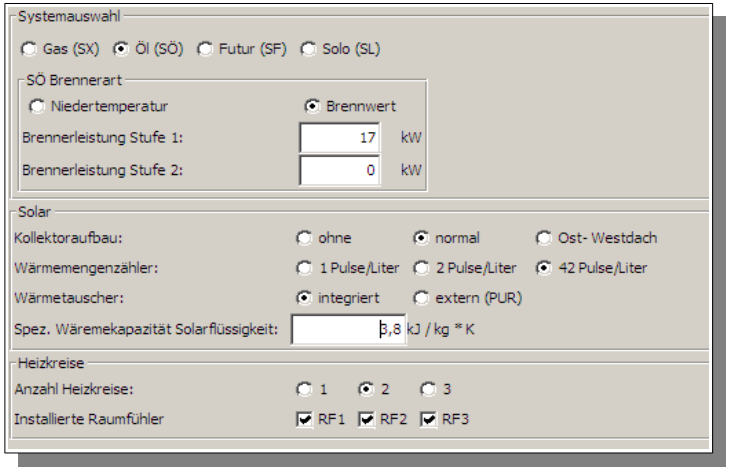
Auszug aus einer praktischen Solaranwendung mit hauscomp.exe

Modul 258 ermittelt laufend die aktuelle Solarleistung (kW) und die Module 261 und 262 wandeln diese Leistung in die entsprechende Energiemenge (kWh) für den laufenden Tag bzw. insgesamt um.

Im Solvis-Viewer muss man jetzt einmalig noch unter „System“, „Ein-Ausgänge“ die Formel für die Solarberechnung einstellen, und zwar einmal bei „Eingänge“ und einmal bei „Statistik Formeln“:



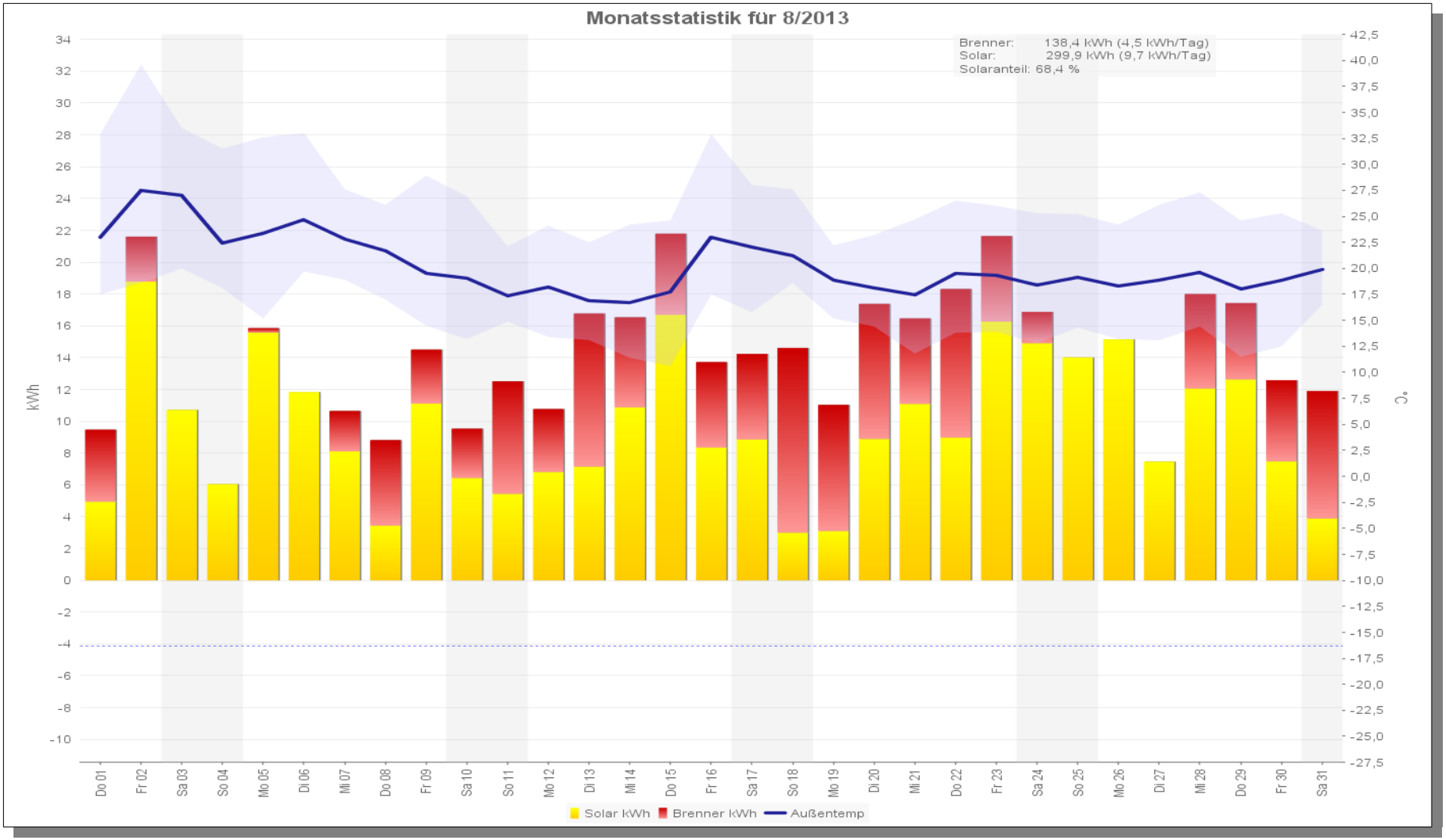
Der im obigen Bild zu sehende Faktor „3,8“ unterscheidet sich von dem Faktor, der im Leistungsmodul (258) verwendet wird (dort beträgt er 4,187). Hintergrund ist das Problem, dass die Logaufzeichnungen des Solvis-Viewers lediglich einmal je Minute gespeichert werden und es so zu großen Ungenauigkeiten bei der Feststellung der „Auszeit“ der Solarpumpe bei bewölktem Himmel also im Start/Stopp Betrieb kommt. In der Praxis muss also ein Kompromiss in der Genauigkeit gefunden werden. Da die Leistungsermittlung durch die SPS wesentlich genauer erfolgt, kann man mit ihrer Hilfe einen für die eigene Anwendung günstigen Faktor ermitteln. Die Abweichungen zwischen beiden Systemen sind zum Teil beträchtlich und können methodisch leider nicht korrigiert werden. Wer es genauer haben will, kommt um eine geregelte Pumpe und/oder einen Volumenstrommesser nicht herum. Den jeweils letzten Tag einer Aufzeichnung berechnet der Solvisviewer übrigens (warum konnte ich nicht nicht klären) um fast das Doppelte verkehrt. Vielleicht hängt es damit zusammen, dass an diesem Tag weniger als 24h Aufzeichnung vorhanden sind. Nicht betroffen sind vollständige Wochenlogs.



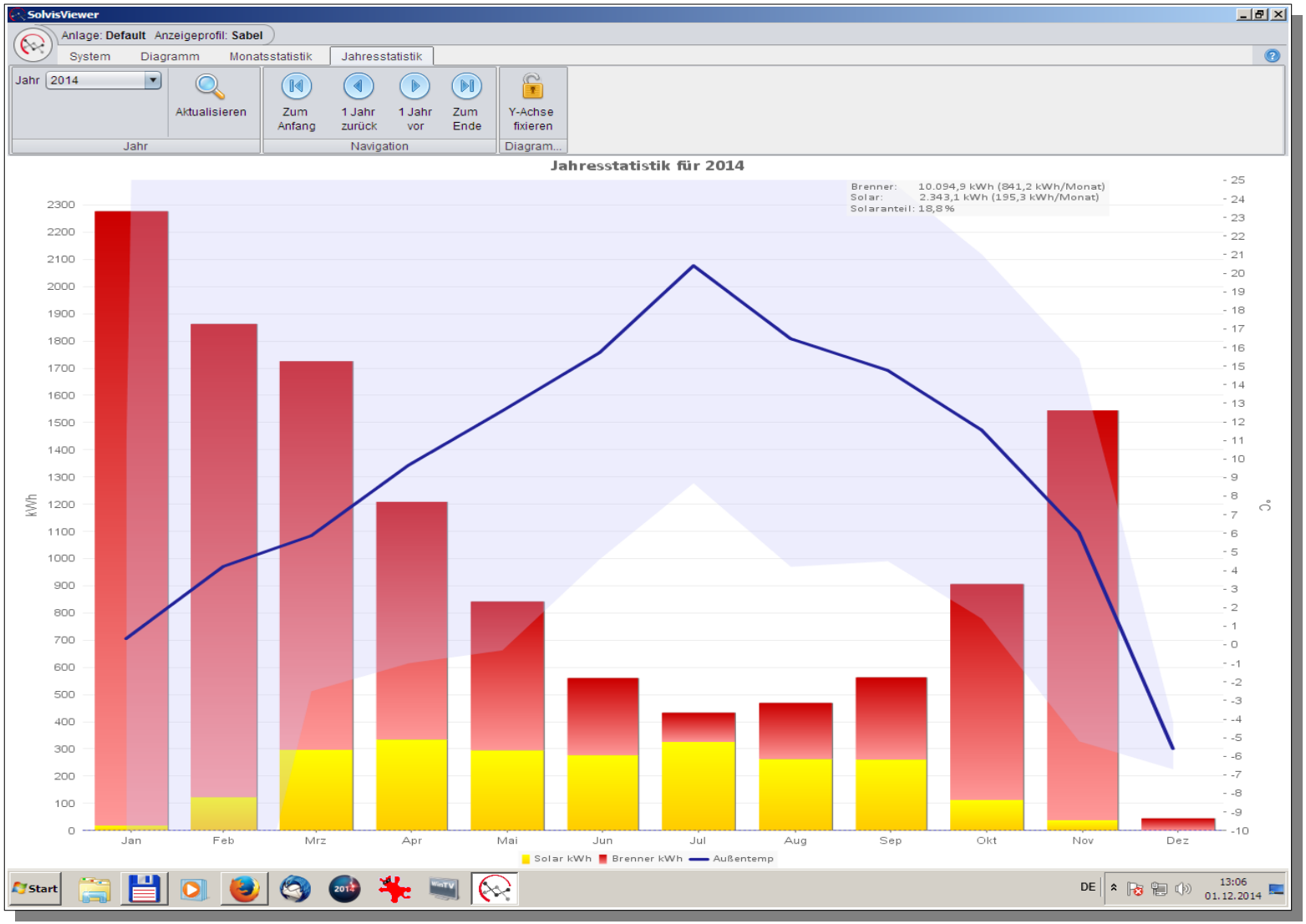
Letztlich dient der Solvisviewer aber auch der optisch eleganten Darstellung und da kommt es auf 10 % oder 15 % Abweichung nicht so sehr an.

Natürlich müssen für die korrekte Anzeige der Kilowattstunden unter „Solvis-System“ auch die Brennerleistung und die spezifische Wärmekapazität der Solarflüssigkeit korrekt eingetragen sein:

Und hier die Monatsauswertung:



Das „Highlight“ des Programms ist die Jahresübersicht, die aus den vielen Tausenden von Messwerten einfache klare Aussagen erbringt:

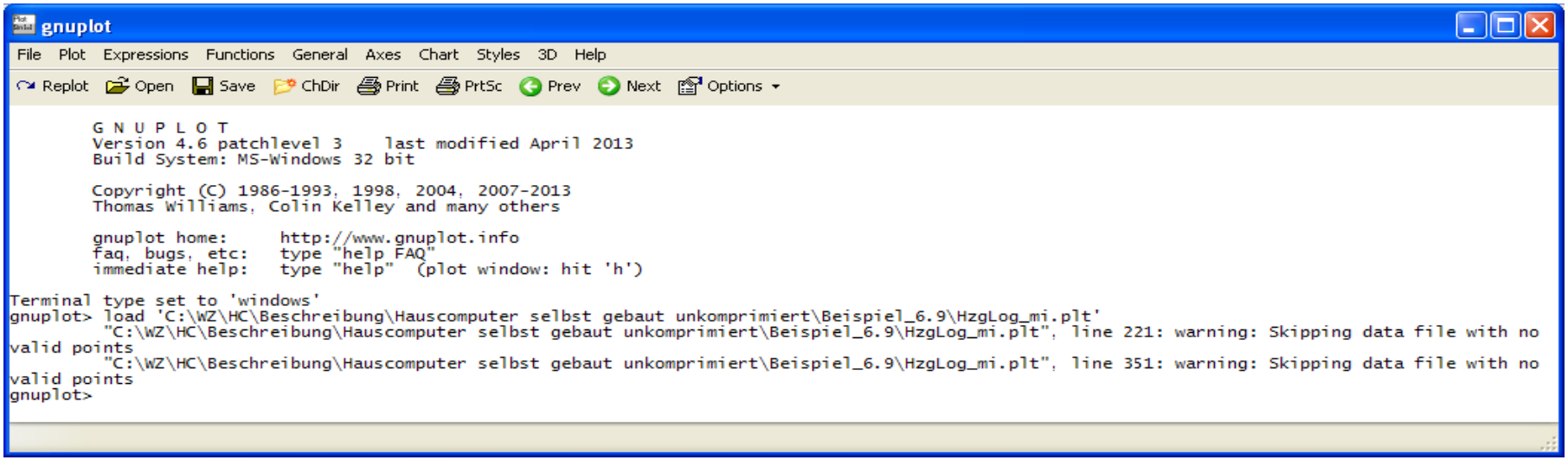

SolvisViewer
Anlage: Default Anzeigeprofil: Sabel
System
Diagramm
Monatsstatistik
Jahresstatistik
Jahr 2014
Aktualisieren
Zum Anfang
1 Jahr zurück
1 Jahr vor
Zum Ende
Y-Achse fixieren
Jahr
Navigation
Diagram...
Jahresstatistik für 2014
Brenner: 10.094,9 kWh (841,2 kWh/Monat)
Solar: 2.343,1 kWh (195,3 kWh/Monat)
Solaranteil: 18,8 %
kWh
°C
Jan
Feb
Mrz
Apr
Mai
Jun
Jul
Aug
Sep
Okt
Nov
Dez
Solar kWh
Brenner kWh
Außentemp
Start
DE
13:06
01.12.2014

Die nächste Auswertung erfolgt mit „gnuplot“. Eigentlich ist alles ganz einfach – Das Auswertescript „HzgLog_mi.plt“ (zu finden unter „.../Beispiel_6.9“) kopiert man zusammen mit dem zu analysierenden Log- File in ein beliebiges Verzeichnis. Dann editiert man in dem Auswertescript in der folgenden Zeile den Dateinamen:

filename='mi130722.txt'

Nur so „weiß“ später das Script, in welcher Datei sich die Daten befinden. Zusätzlich gibt es noch die Option „dayno=..“. Wird sie verwendet, so sind später in der Grafik lediglich die Daten des entsprechenden Wochentages (0..6) zu sehen. Lässt man diesen Befehl weg, oder rückt ihn mit „#“ ein, so wird die ganze Woche angezeigt.

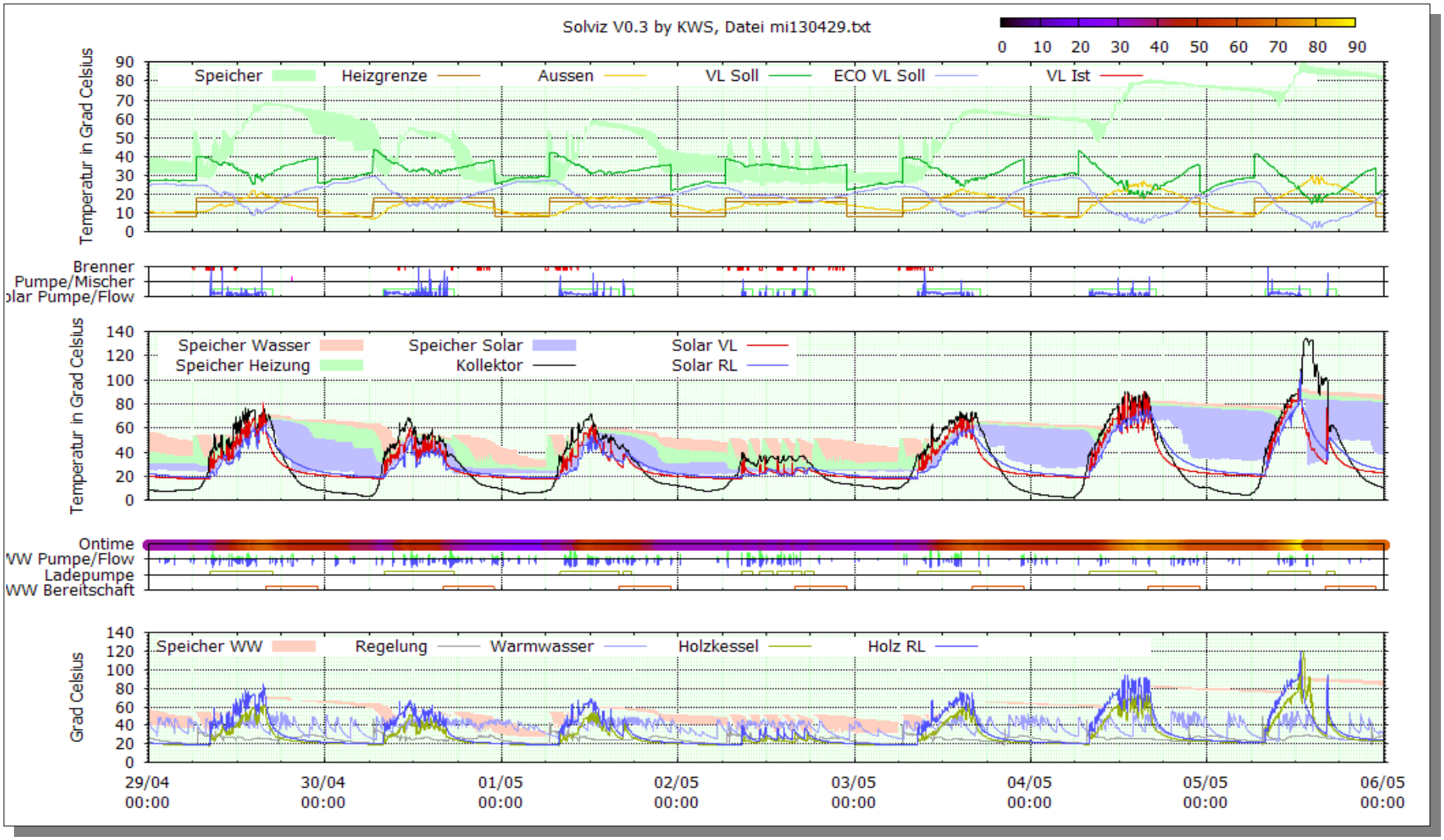
Nach der Installation der gnuplot- Software startet man gnuplot und lädt das Auswertescript unter „File/Open“:



Das Script verarbeitet vollautomatisch die Log- Datei und zeigt nach einiger Zeit in einem neuen Fenster das Ergebnis an. Eventuelle Warnungen werden im Befehlsfenster mit angegeben. Sollte das Script allerdings mit einem Fehler in der Abarbeitung abbrechen (anstelle von „gnuplot>“ steht „multiplot>“ als Eingabeaufforderung), so muss man erst den Befehl „unset multiplot“ eingeben (und mit Enter bestätigen), bevor man erneut ein Scriptfile laden kann.

Übrigens – darf die Datei, die die Daten enthält (in unserem Beispiel also die Log- Datei), ausschließlich aus den entsprechen Zahlenkolonnen bestehen. Eventuell vorhandene zusätzliche Angaben, wie z. B. ein Dateikopf mit Textangaben muss zwingend mit einer Raute („#“) am Zeilenanfang beginnen. Das Programm *hauscomp.exe* setzt die Raute bereits automatisch. Hoffentlich ist das auch zu späteren Versionen des Solvis-Viewers noch kompatibel...

So sieht die grafische Analyse der Log- Datei „mi130429.txt“ mit dem Auswertescript „HzgLog_mi.plt“ unter „gnuplot“ aus:



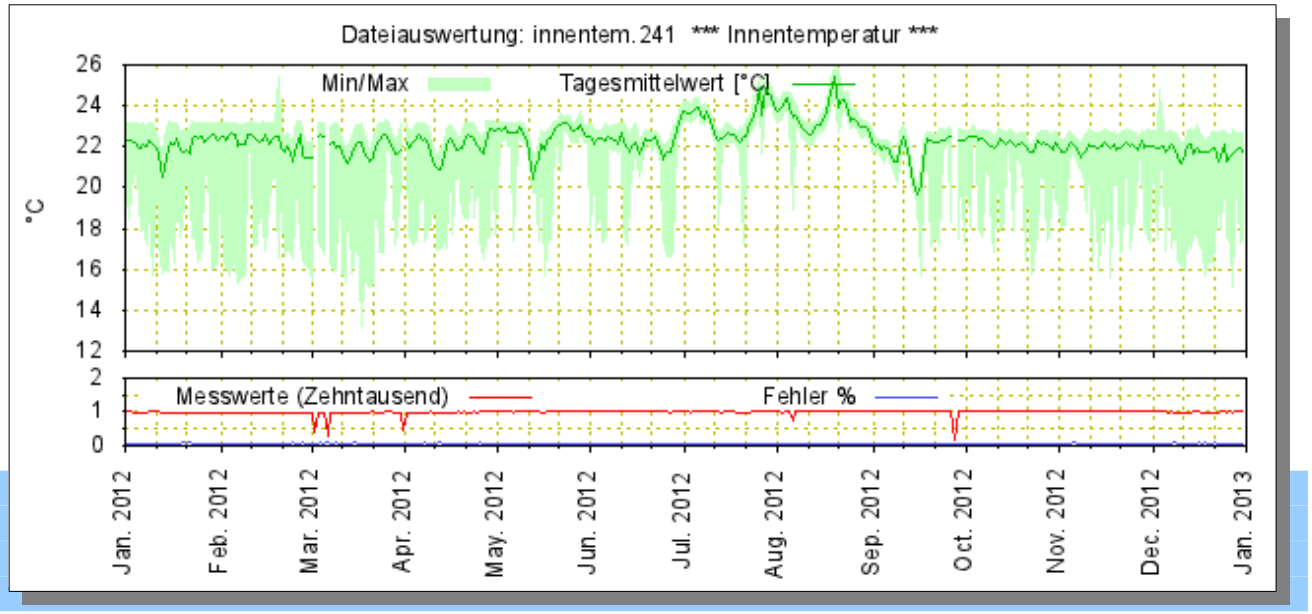
Für eigene Zwecke kann man die Scriptdatei natürlich nach Belieben anpassen...

6.10. Statistische Auswertung von Sensordaten

Das Programm *hauscomp.exe* sammelt täglich für jeden Sensor die minimalen, maximalen und Durchschnittswerte, vorausgesetzt, in der *hardware.ini* ist der Scriptbefehl „ALog= on“ enthalten. Hier ein Auszug aus der Datei „INNENTEM.241“:

```
Datum;Innentemp.[øC]-MIN;Innentemp.[øC]-MAX;Innentemp.[øC]-Tagesmittelwert;Innentemp.-Anzahl Messwerte;Fehlerquote [%]
01.01.2012;20,806;23,198;22,235;10008; 0,03
...
```

Eine einfache Umwandlung dieser Daten in eine Grafikdatei, vom Typ „*.png“ ist mit dem Programm „gnuplot“ ebenfalls möglich. Installation und Start der Software entspricht dem unter Pkt. 6.9. bereits beschriebenen. Als Scriptfile ist die Datei „Sensor_als_png.plt“ aus dem Verzeichnis „..\Beispiel_6.10\“ zu starten. Sie bezieht sich wiederum auf die Sensor-Datei „INNENTEM.241“ mit den gesammelten Daten im gleichen Verzeichnis. Als Ergebnis erhält man die nebenstehende Grafik:



Interessant ist dabei, dass Tage mit zu wenigen Daten (z. B. Programmstart) automatisch ausgeblendet werden und auch die Fehlerquote angezeigt wird. Mit EXCEL oder Calc ist diese Aufgabe nicht so einfach zu lösen...

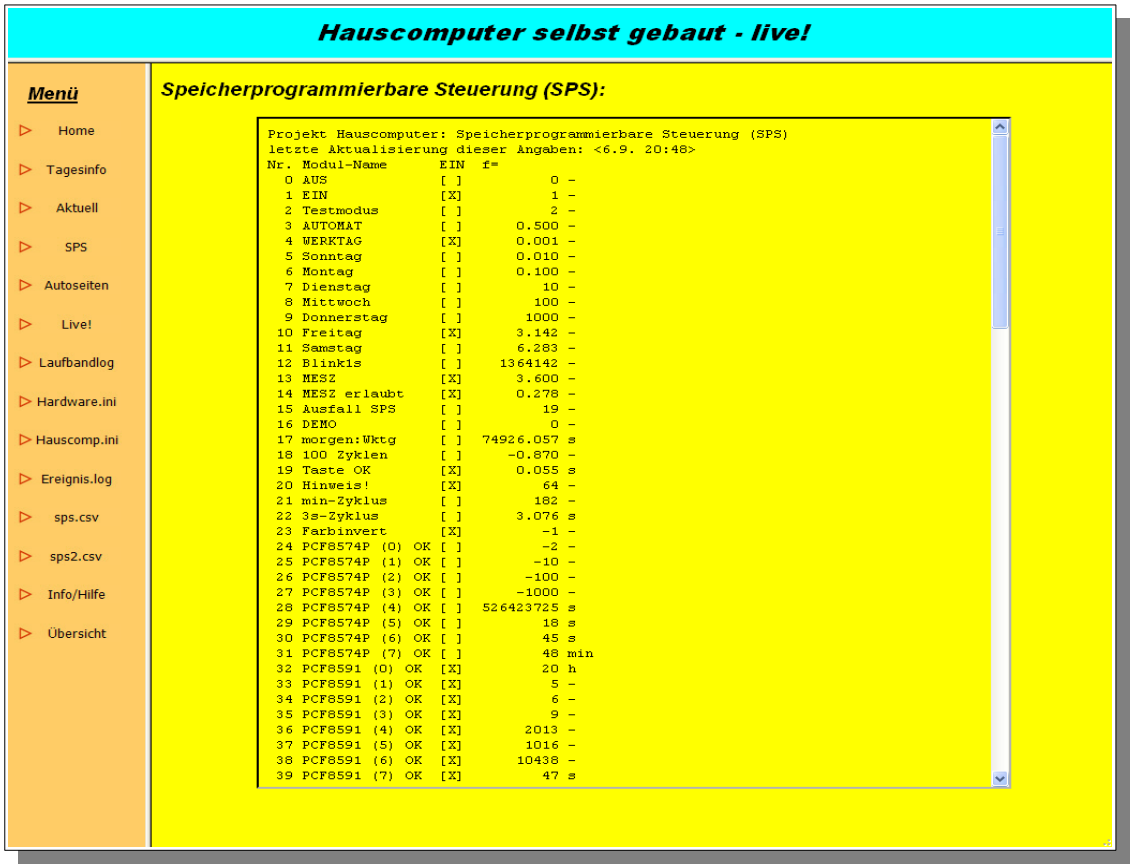
Für andere Daten- Dateien sind die nachstehenden Scriptzeilen entsprechend zu ändern:

```
INfile='innentem'
INendung='241'
Bezeichnung='Innentemperatur'
Einheit='°C'
```

Das Linux-Image für den Hauscomputer organisiert diese Grafiken bereits täglich vollautomatisch kurz nach Mitternacht (→ Statistik → Daten). Neben analogen Sensoren werden dabei auch alle SPS-Module vom Typ 66 (Betriebsminuten des Vortages), 97 (Tagesertragszähler), 1002 (Übernahmemodul) erfasst, sowie SPS-Module in deren Einheit sich „/d“ findet.

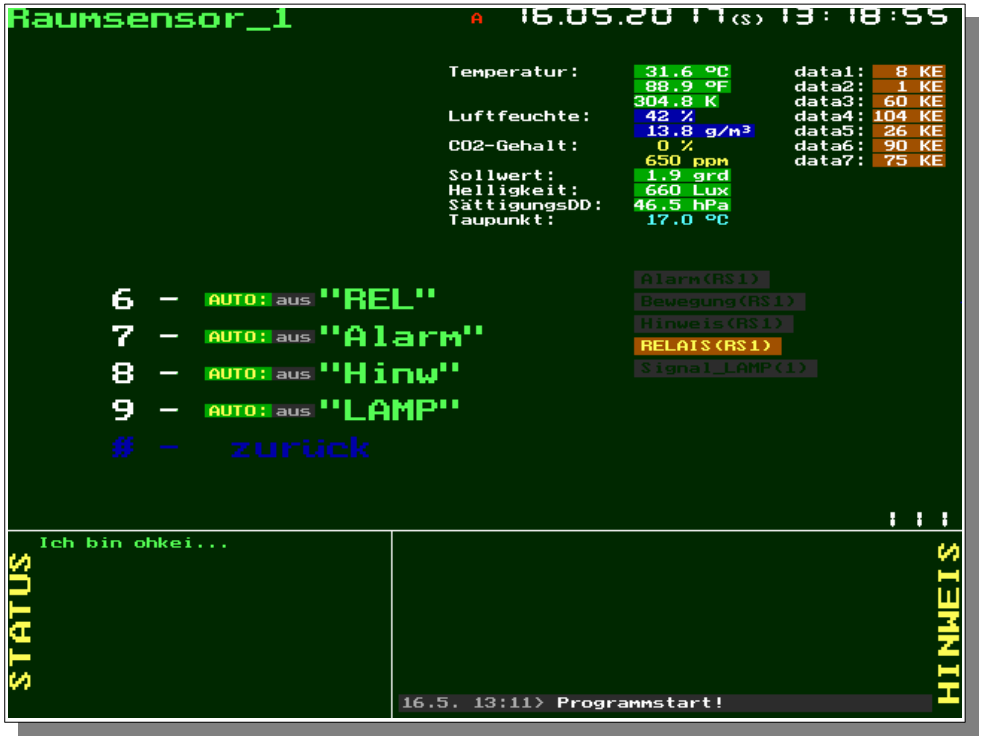
6.11. Das erweiterte Webinterface

Unter 6.6. wurde bereits eine einfache Web-Darstellung beschrieben, die ausschließlich aus einer Datei „index.htm“ bestand. Hier nun eine erweiterte Form, mit der der Zugriff auf die zahlreichen Log-Dateien möglich ist. Dazu müssen lediglich die Datei „index.htm“ und der Ordner „/htm/“ von der Projekt DVD („..\DVD\Beispiel_6.11\..“)in das Anwenderverzeichnis mit der *hauscomp.exe* kopiert werden. Da der Dateitransfer bei laufenden Betrieb umfangreicher ist, kann es unter reinem DOS im Netzwerkbetrieb zu längeren Ladezeiten kommen. Unter Linux/Dosemu und entsprechend schnellem Rechner sollte es keine Probleme geben. In [6] wird ein erweitertes Webinterfaces bis hin zur CGI-Steuerung und Verschlüsselung mit SSL beschrieben.

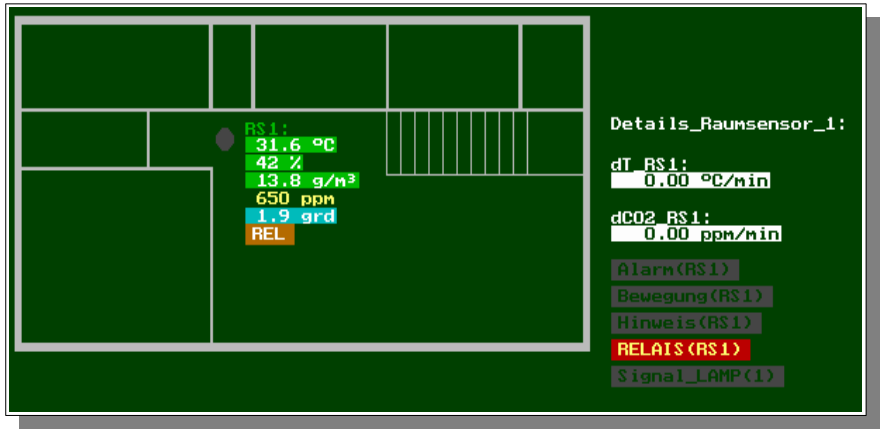


6.12. Ein Testprogramm für Raumsensoren / Außensensor / Multisensoren

Im Verzeichnis /Beispiel_6.12/ befinden sich die ini-Dateien für den Test von Multisensoren. Dabei werden die Subadressen 0, 1, 2, 12 und 15 unterstützt. Zusätzlich zur ohnehin bereits vorhandenen Übersichtsseite 186 (Funktionen../Raumsensoren) kann man nun per Anwendermenü (Anwendungen../Raumsensor_x) auch die zusätzlichen Funktionen für die programmierten Multisensoren auslösen (Relais schalten, Alarm setzen usw.). Auch werden hier fast alle übermittelten und berechneten Parameter angezeigt (Bild rechts).



Auf der Seite /Grafik/Hausparameter gibt es ein Beispiel für praktische Anwendungen:



Voraussetzung für die Anzeigen im Obergeschoss eines Einfamilienhauses ist ein angeschlossener Raumsensor (Subadresse 1). Die „650 ppm“ (im Bild gelbe Anzeige) beziehen sich auf den CO2-Wert – den gibt es natürlich nur, wenn ein entsprechender Sensor verbaut ist. Der Grundriss ist als Vektorgrafik mit Linien umgesetzt. Die Subadresse 12 ist für einen Wärmemengenmesser vorbereitet. Da die Subadresse 15 der Raumsensoren als Außensensor dient, gibt es als „Schmankel“ auch eine Seite „Wetterstation“. Siehe auch die entsprechende Bauanleitung im Hardware-Band 5.

Die Gestaltung greift genauso auf die bewährten Scriptbefehle „Anzeige=…“, „Text=…“ für die Grafik zurück. Quelle der Daten sind sowohl direkte Messwerte, als auch per SPS gesammelte und -bearbeitete. So führt z.B. die Kombination …

text=	***119***	***0***	***47***	***15***	***1***	***Temperatur:***						
Anzeige=	***119***	***25***	***9***	***15***	***135***	***40***	***2***	***1***	***0***	***40***	***1***	***/°C***

… zur Anzeige des direkten Messwertes „Temperatur: 2.0 °C“ im rechten Bild.

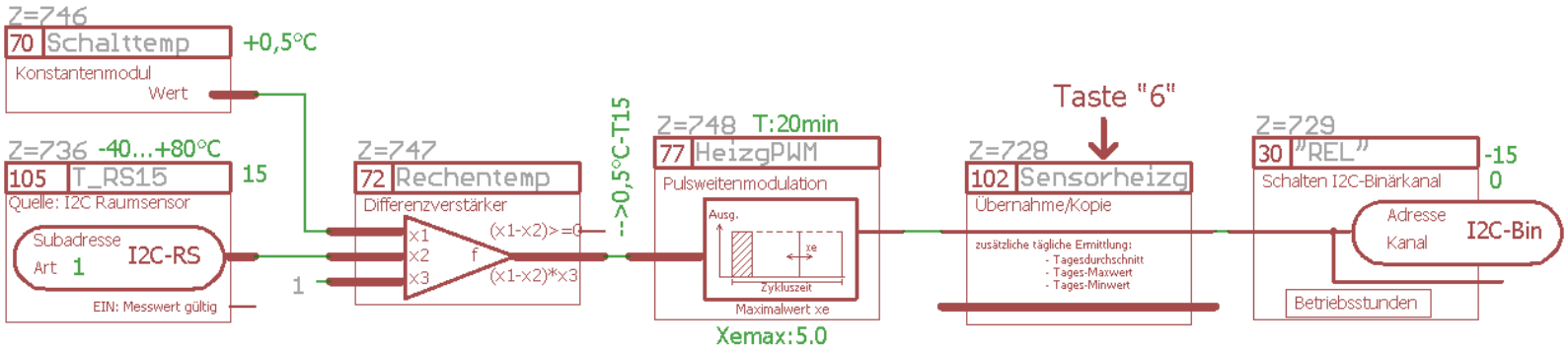
So werden die verschiedenen Wetterdaten, wie Temperatur, Feuchte, Wind und Niederschlag erfasst.

Neben der Anzeige von Messwerten kann man aber auch per Taste „6 … 9“ interne Zustände schalten. So ist es z.B. möglich, das Alarm- oder Hinweissignal an der Wetterstation zu setzen und damit einen zusätzlichen akustisch/optischen Aktor in Betrieb zu setzen.

Das Relais in der Wetterstation wird in diesem Beispiel genutzt, um eine Heizung im Regensensor automatisch bei Frostgefahr zu aktivieren. Dann kann Schnee (so er denn auf dem Sensor tatsächlich auftaut) als Niederschlagsmenge gemessen werden.

Ein paar Module in der SPS erledigen die obige Aufgabe problemlos: Oberhalb der Einschalttemperatur (+0,5°C) ist die Heizung immer aus, unterhalb -4,5°C ist sie immer an und dazwischen wird sie per PWM in Abhängigkeit von der Temperatur im Intervall 20 min regelmäßig ein/aus geschaltet (SPS-Auszug unten). Per Taste 6 kann man zusätzlich die Betriebsart beeinflussen.





7. Fehlersuche / Korrekturen am System / bekannte Probleme / Tipps

- Allgemeines

  Das Programm, wenn es denn einmal läuft, macht wenig Probleme. In der Laufschrift werden eindeutige Hardware – Probleme im Klartext angezeigt. Notfalls kann man die Testseit über „Funktionen/Hardware-Status...“ aufrufen. Hier müssen alle an den Hausbus angeschlossenen Schaltkreise gelistet sein. Jeder Kanal eines PCF8574P kann dann über „schalten..“ und dem entsprechenden Unterpunkt einzeln ein- bzw. ausgeschaltet werden. Die analogen Kanäle überprüft man über „Analogsensoren..“ und dem entsprechenden Unterpunkt. Für weitergehende Tests kann man verschiedene Computer nutzen. In diesem Fall sollte für jeden Computer die korrekte *hardware.ini* existieren.

- Fehler in der *hauscomp.ini* oder *hardware.ini*

  Die wichtigsten Ursachen die zu Fehlern führen, sind Schreib- oder Denkfehler in den beiden *.ini - Dateien. Zur Fehlersuche ist es in der Regel ausreichend, die zuletzt geänderten Scriptzeilen durch ein Leerzeichen am Anfang einzurücken und damit inaktiv zu schalten. Nach erneutem Programmstart kann man überprüfen, ob der Fehler noch da ist, oder nicht. Ist die fehlerhafte Zeile lokalisiert, so überprüft man nochmal ihre Syntax mit einem Blick in das Handbuch. Die vielen Zahlen verleiten dazu, mal ein Parameter zu vergessen. Sind auch wirklich Zahlen und keine Buchstaben an den entsprechenden Stellen? Hilft auch das nicht, so den zweiten Blick in die vielfältigen Beispiele werfen. Sollte auch das nichts nutzen – morgen ist auch noch ein Tag! Die Regelungmaterie ist ja nicht so einfach und vielleicht benötigt man einfach einen anderen Blickwinkel. Bei hartnäckigen Fehlern eine E-Mail an mich und eine Lösung wird gefunden. Es gibt schließlich keine fehlerfreien Programme. Die Parameter der geladenen Scriptbefehle „Anzeige=“ lassen sich übrigens direkt auf Bildschirmseite 9 begutachten.

- Fehlersuche bei einem laufenden Hauscomputer

  Wenn der Hauscomputer seiner eigentlichen Bestimmung zugeführt ist, sollte man ihn nicht unnötig neu starten. Man sollte zunächst mit den bereits genannten Möglichkeiten den Fehler einkreisen und wenn möglich parallel auf einem anderen Computer testen. Dazu kann man die Diskette aus dem Hauscomputer verwenden. Das der Hausbus auf dem anderen Computer ja fehlt, stört bei Fehlersuche in der SPS überhaupt nicht, da man hier jeden Modulausgang von Hand ein- bzw. ausschalten kann (Enter – Taste!). Während dieser Test- Phase am anderen Rechner kann der normale Hauscomputer weiterlaufen, aber spätestens um Mitternacht erwartet er die geänderte Diskette zurück!

- Analyse der analogen Graphen / Neuskalierung von analogen Eingängen bei laufendem Hauscomputer

  Die Diskette wird dem Hauscomputer bei laufendem Programm entnommen, auf einem anderen Rechner die *hauscomp.exe* gestartet. Über dem Menüpunkt „System..“ wird AUTO abgeschaltet, DEMO zugeschaltet, und die „gestern.dat“ geladen. Danach kann man über „Grafik..“ die kompletten graphischen Daten des Vortages analysieren/ ausdrucken. Sollen die Skalierungsdaten der analogen Eingänge verändert werden, so überschreibt man sie einfach in der *hauscomp.ini* mit einem Editor. Der Hauscomputer muss dann nicht neu gestartet werden. Ein Neustart ist allerdings erforderlich, wenn die Adress- und Kanalzuordnung insbesondere von graphischen Parametern sich ändern! Hierbei kann es erforderlich werden, vor dem Programmstart das Datum der 3 Dateien gestern.* zu ändern, um zu verhindern, dass sie beim Programmstart geladen werden. Alternativ überschreibt man sie mit den mitgelieferten gestern.* Dateien oder löscht sie einfach. Zur Neuskalierung von

Sensoren sollte man die notwendigen Wertepaare längerfristig ermitteln. Also z. B. die Außentemperatur im Winter bei -20 bis -10°C, im Sommer bei ca. 20-25°C und irgendwie um +5°C. Man benötigt noch ein gutes Vergleichsthermometer und den zugehörigen Kodewert aus der „Hausbus“ Menüseite.

- Zufällige (oder regelmäßige um den Tageswechsel) Programmabstürze, Vergrößerung der Systemlast, extrem falsche Uhrzeit, Zeit bleibt länger als 1 Stunde „hängen“ o. ä. Bitte zunächst die interne Uhr des PCs überprüfen. Irgendwann ist die Batterie, die die Uhr und das BIOS stützen, leer und ein Austausch wird fällig. Zweite Möglichkeit: Speicherfehler oder ein beginnender Ausfall des Netzteils (gar nicht soo selten!). In jedem Fall sollte das Programm zum Vergleich auf einem anderen Rechner getestet werden. Spannungseinbrüche und Störsignale auf dem Hausbus sieht man sehr gut in der 24h Grafik oder edb0 Grafik (unter Linux)!
- Fehler bei Umstellung auf / von Sommerzeit oder bei der Korrektur der Uhrzeit

  Wird die Uhrzeit des PC, aus welchen Gründen auch immer, zurückgestellt, kommt es programmintern zunächst zu einer negativen Zeitrichtung mit der entsprechenden Fehlermeldung in der Laufschrift. Zusätzlich wird der Zustand von SPS-Modul Nr. 15 aktiv. Um Fehlfunktionen zu vermeiden, werden für den Zeitraum bis zum Wiedererreichen der ursprünglichen Zeit, Teile der SPS „eingefroren“. Alle Binärausgänge sind in dieser Zeit abgeschaltet. Die Watchdogs bleiben aktiv. Beim Vorstellen der Uhr tritt dieser Effekt nicht auf. Dafür produziert der SPS-Modultyp 105 eine Fehlermeldung (ab 6.0.52 behoben).
- „Falsche“ Zeichen auf dem Bildschirm unter Windows XP:

  Die neueren PCs und Windows-Versionen unterstützen nur noch eingeschränkt DOS-Programme. Unter anderem ist der Zeichensatz nicht vollständig emuliert. Deshalb testet das Programm beim Programmstart, ob eine „1“ richtig auf dem Bildschirm dargestellt wird. Ist das nicht der Fall, werden alle Zahlen und Sonderzeichen mithilfe der Datei „Font.abc“ um kodiert. Das kostet etwas Rechenzeit. Sollten trotzdem fehlerhafte Zeichen auftauchen, bitte ich um eine Info.
- Programmfehlerkodes „RUNTIME ERROR“:

  002 - Dateien GESTERN.* oder LIZENZ.RPT oder FONT.ABC fehlen 203 – Hauptspeicher zu klein, Programm wird nicht gestartet

  005 - Dateizugriff durch Betriebssystem verweigert
- „RUNTIME ERROR 207“ (Fehler bei Umwandlung Gleitkomma in Integer/Longinteger):
  Trat im Mai 2025 bei einem PC mit intensiver Nutzung des Datentransfers per *datain.txt* auf. Ab Version 6.60B2 sind nun zahlreiche Tests auf Bereichsüberschreitungen eingefügt, Fehler behoben.
- Absturz DosEmu unter Linux mit Fehlermeldung „leavedos (54|10x36) called - shutting down“ in */root/.dosemu/boot.log*.
- „Dateien *hardware.ini* oder *hauscomp.ini* fehlen“: Die Dateien fehlen oder der aktuelle Pfad wurde in der autoexec.bat nicht korrekt auf das Verzeichnis der Hauscomputer Dateien gesetzt (also in der vorletzten Zeile z. B. A:\HC eintragen, wenn alle entsprechenden Dateien im Verzeichnis „A:\HC“ liegen).

Bekannte Probleme:

Am Tag der automatischen Zeitumstellung von Winter- auf Sommerzeit wird bei der Ermittlung der analogen Parameter, wie z. B. die Maximal- und Minimalwerte die Bezugszeit ganztägig auf die Winterzeit berechnet.

Weiterhin wurde bei einem älteren PC ein Umstellungsproblem am Sonntag, dem 29.02.2004 um 0:30 beobachtet. Dieser Rechner blieb für ca. einen Tag im Zustand negativer Zeit hängen und lief anschließend selbständig weiter. Der Fehler konnte nicht reproduziert werden. Die automatische Wochenkorrektur wurde deshalb auf Sonnabend 2:30 Uhr vorverlegt. Ab Version 6.0.7. ist das Problem durch Einführung einer Plausibilitätsüberprüfung in der Zeitsteuerung mit an Sicherheit grenzender Wahrscheinlichkeit behoben.

- Während der Laufzeit des Programms erscheinen vereinzelte „weiße Punkte“ auf dem Bildschirm, die erst bei einem Seitenwechsel gelöscht werden. Ursache ist der „rabiate“ Zugriff beim Grafikaufbau auf den Bildschirmspeicher. Diese Zugriffsart beschleunigt den Programmablauf auf älterer Hardware beträchtlich und hat außer den weißen Punkten keine weiteren Auswirkungen. Ab Version 6.0.37 durch veränderten Zugriff vermindert, ab 6.0.40 müsste es ganz weg sein...
- Programmabsturz mit „RUNTIME ERROR“ oder Lesefehler vom Laufwerk gegen Mitternacht (Geisterhaus?)

  Um Mitternacht (BIOS-Zeit), und nur dann*, versucht das Programm auf den Datenträger (Diskette) zuzugreifen um Updates zu laden und die Tagesdaten zu speichern. Kann das Betriebssystem den Zugriff nicht realisieren, kommt es zu Fehlermeldungen. Ursache sind in der Regel Staubablagerungen im Diskettenlaufwerk, die zum fehlerhaften Lesen des Datenträgers führen. Staub ist praktisch überall vorhanden und verursacht mit Abstand die meisten Ausfälle beim langfristigen Betrieb eines Hauscomputers. Abhilfe: Zukleben des Disketteneinschubes mit Klebeband oder Verwendung einer Flashdisk oder eines bootfähigen USB-Sticks. (*Hinweis: Netzwerk=0, live=off und SPSLog=0)
- „Hängenbleiben“ des Programms um Mitternacht ohne Fehlermeldung auf einem Compaq PC mit Flashdisk. Ursache: Timingproblem beim Schreiben auf den Datenträger. Das Problem ist ab der Version 6.0.1 gelöst: Während des (schreibenden) Speicherzugriffs wird der Haus-Bus abgeschaltet.
- Sporadische Programmabstürze, die nur im Netzwerkbetrieb (DOS oder Windows), nicht aber unter reinem DOS ohne Netzwerk auftreten: Ursache ist der gleichzeitige Zugriff auf Programmdateien, der zum Auslösen eines DOS internen Flags führt, das von der *hauscomp.exe* nicht beeinflusst werden kann. Mit Überarbeitung des Dateihandlings ab Version 6.0.28 wurden diese Fehler minimiert. In Version 6.0.30 führen sie nochmals zu I/O Fehlern, ab Version 6.0.31 werden diese Fehler im laufenden Betrieb (Netzwerkbetrieb bzw. Livemodus) erkannt und umgangen. Da im Netzwerkbetrieb Dateien regelmäßig aktualisiert werden müssen, fallen die kurzzeitigen Aussetzer nicht auf. Bei der Speicherung der *.csv Dateien können deshalb auch in sehr seltenen Fällen Datensätze fehlen. Sporadische Dateizugriffsfehler können noch beim Programmstart bzw. -Ende auftreten. In diesen Fällen bleibt nur ein „händischer“ Programmneustart. Ab Version 6.0.37 wurden nochmal Maßnahmen zur Verbesserung des Dateihandlings umgesetzt. Die generelle Unterdrückung dieser Dateizugriffsfehler wäre möglich, es blieben dann aber eventuelle Fehler in der Speicherung der Betriebsstunden (gestern.daz) unentdeckt. Unter Linux wurden derartige Fehler bisher nicht beobachtet.
- Nach dem Tageswechsel von Sonnabend auf Sonntag bleibt die Uhr auf Sonnabend stehen. Festgestellt an einem 486er COMPAQ DX33. Ursache: Das Bios des PCs unterstützte nicht den Interrupt für die automatische Zeitkorrektur (sonnabends um 2:30 Uhr). Lösung: auf die automatische Wochenkorrektur verzichten, bzw. „Wochenkorrektur=0“ setzen.

- Kurz vor Mitternacht beim Tageswechsel von Sonnabend auf Sonntag und „Wochenkorrektur=-106“ bleiben das ganze Programm oder nur Teiledavon „hängen“. Ursache ist eine Bereichsüberschreitung eines Arrays bei negativer Wochenkorrektur am Sonnabend. Ab Version 6.0.18 behoben.
- Beim Programmstart bleibt der Bildschirm schwarz, wenn versucht wird, eine *hauscomp.exe* Version 16 oder niedriger mit einer „Gestern.daz“ (mit einer Größe von 1024 Byte) zu starten. Lösung: Version 6.017 ff. verwenden oder die Datei „Gestern.daz“ löschen. Ursache ist die vergrößerte SPS ab Version 6.0.17. Der Übergang von Version 6.0.36 zu 6.0.37 (SPS von 399 auf 799 Module vergrößert) ist dagegen in beide Richtungen durch die zusätzliche Datei „gestern2.daz“ kompatibel. Das betrifft auch die Erweiterung auf 1199 mit der Datei „gestern3.daz“.
- Der I2C Bus funktioniert nicht mehr: Das Signal SDA ist dauerhaft auf „Low“

  Ursachen: 1. HAL verwehrt den Zugriff unter Windows; oder 2. ein I2C Baustein hat sich „aufgehängt“ - bisher nur bei LM 75 beobachtet (ca. 1 bis 2 mal je Tag). Lösung für 1.: automatische Hardwareerkennung für Modem aktivieren, möglicherweise funktioniert die Schnittstelle dann wieder. Lösung für 2.: kurzzeitig (1-2 s) Stromversorgung der I2C Bausteine abschalten. Auf den Einsatz des LM 75 verzichten.
- Fehlerhafte Werte der Analog-Sensoren bei Einsatz des ATtiny26 mit der Firmware I2C_A1.0. Lösung: in *hardware.ini* Scriptbefehl „Xfaktor=7“ hinzufügen.
- Es gibt in Version 6.0.30 und 6.0.31 einen Bug in der Ansteuerung der Adresse 0 mit PCF8574P - deshalb kommt es insbesondere im Schaltinterface zur Anzeige „rot“ und „flackern“ der Relais, da in diesem Interface die Watchdog am empfindlichsten auf diesen Fehler reagiert. Ab 6.0.32 behoben.
- Im DOS-Netzwerk ist der Zugriff auf den Steuerrechner stark verlangsamt. Abhilfe: in der *hardware.ini* den Scriptbefehl INT21=on nutzen.

Besonderheiten unter Windows:

- Unter Windows 3.11: Fehlermeldung bei Programmstart „Programm gefährdet Systemintegrität“. Lösung: pif-Datei erstellen und in das gleiche Verzeichnis wie die *hauscomp.exe* kopieren. Eine Musterdatei (Hauscomp.pif) befindet sich im Verzeichnis „Windows_3.11“. Sie erwartet die *hauscomp.exe* im Verzeichnis „c:\hauscomp“. Der Programmstart erfolgt dann über „Hauscomp.pif“. Die Hauscomp.pif Datei lässt sich mit dem Programm „pifedit.exe“ (im Windows Verzeichnis) editieren.
- Unter Windows XP bzw. Windows 2000: Bedingt durch die andere Timerprogrammierung unter Windows XP und des damit, je nach eingesetzter Hardware, verringertem Zeitfenster ist der Bustakt des Hausbusses im Programm verdoppelt worden. Die maximal zulässige Länge des Hausbusses verringert sich damit auf die Hälfte. Erkennbar ist dieser Zustand an „XP=true“ im Zeitkern.
- Unter Windows 98SE: der Zugriff auf die seriellen Schnittstellen erfolgt manchmal scheinbar korrekt, aber am Bus erscheint kein Taktsignal. Ursache ist Windows, der sogenannte HAL verweigert direkte Portzugriffe, je nachdem welche internen Sperr-Flags irgendwann gesetzt wurden. Darüber hinaus versucht Windows beim Programmstart COM1 für eine Maus zu initialisieren, obwohl z. B. eine PS2- Maus angeschlossen ist und setzt damit immer Sperr-Flags für COM1. Abhilfe: Über Systemsteuerung/Modem die Hardwareerkennung starten – die natürlich kein Modem findet, aber die Sperr-Flags zurücksetzt. Danach geht der Zugriff wieder- bis zum nächsten Programmstart. Für den Autostart benötigt man eine zweite serielle Schnittstelle. Bei einer Cross TV X-Box (GX1 Prozessor 300Mhz) konnte im BIOS eine physikalisch nicht vorhanden serielle Schnittstelle mit 2F8 initialisiert und als COM2 die tatsächliche mit den üblichen 3F8 eingestellt werden.

Der Gerätemanager unter Windows zeigt COM1 mit 3F8 und einen COM mit Resourcenkonflikt (ebenfalls 3F8). Die *hardware.ini* wurde auf COM1 (=3F8) gestellt. Damit verläuft der Autostart (*hauscomp.exehauscomp.exe* in Autostartgruppe) erfolgreich. Der dabei angezeigte Fehler im Gerätemanager hat keine weiteren Auswirkungen.

- Unter Windows 98SE: bei Autostart erfolgt die grafische Ausgabe des Programms unter Umständen schwarz/weiß, bei Start aus dem Dateimanager dagegen korrekt in Farbe. Ursache unbekannt. Abhilfe: Ab Version 6.0.22 wird unter win98SE die Grafik kurz nach dem Programmstart ein zweites Mal initialisiert. Wahrscheinlich ein Timingproblem unter Windows und/oder Hardware abhängig.
- Unter Windows, im Fenstermodus und auf langsamen Rechnern bleibt das Programm förmlich stehen. Lösung: Programm nur im Vollbildmodus laufen lassen.
- Unter Windows im Netzwerkbetrieb: Während der Aktualisierung der Dateien bleibt das Programm längere Zeit stehen (bis zu 2 min je Seite). Ursache: Bei „shared" Grafik ist der Prozessor mit dem Auslesen der Grafik überlastet. Abhilfe: Einbau einer separaten Grafikkarte. Die Zeit für die Aktualisierung der entsprechenden Dateien liegt dann bei max. 1 s. Ab der Version 6.0.25 ist die Aktualisierung der Screenshots durch Einbau von Assembler Code wesentlich verbessert, die Zeit liegt nun auch bei „shared" Grafik unter 1 s.
- Unter Windows XP und dem Programm „DosBox" auf einem IBM Pentium 233 Mhz Type 8364 erfolgte die zyklische Abfrage des Timers 5 mal langsamer als normal. Erkennbar ist dieser Zustand im Zeitkern („Dalles/Timer" >> 0 bzw. Tagessekunden zählen nicht im Sekundenrythmus). Das Programm läuft damit fehlerhaft. Ursache ist die nicht „IBM"-kompatible Hardware des PCs.
- Unter Windows 98SE wird an einem IGLE PC (Geode- Prozessor) ein SDA Korrekturadapter erkannt, der aber gar nicht vorhanden ist. Ursache ist die RS232 Treiberroutine von Windows in Verbindung mit dem Chipsatz. Abhilfe: in *hardware.ini* die Scriptzeile „SDAKorrektur=off" aufnehmen.
- Unter Windows XP und Linux DosEmu: Beim Dateiabgleich (meist um Mitternacht) kommt es selten zu Datei-Fehlermeldungen. Ursache: XP/Linux gaukelt kurzzeitig ein falsches Laufwerk und damit einen falschen Dateipfad vor. Die Wahrscheinlichkeit ist höher, wenn die Netzwerkfunktion und der Live-Modus aktiv sind. Eine Fehlermeldung ist die Folge und ein Neustart von Hand wird erforderlich. Nach intensiver Suche nach einer Problemlösung konnte eine Querverbindung zur Laufschrift gefunden werden. Deshalb ist unter diesen Betriebssystemen die Scriptzeile „Laufschrift=off" in die *hardware.ini* einzufügen. Ab Version 6.0.39e ist das Problem generell behoben.
- Beim Öffnen von *.BMP Screenshots kommt es in verschiedenen Programmen/Browsern zu fehlerhaften Anzeigen. Ab Version 6.0.37 behoben.
- Anzeige „Entschuldigung dieser Fehler sollte nicht passieren" beim Programmstart mit Verweis auf eine Datei „heute.dai" oder „heute.dab". Ursache ist ein vorzeitiger Betriebssystemabsturz/Bootvorgang beim letzten Abspeichern der 3 „heute.xxx" Dateien, wobei die erste Datei (heute.dar) noch korrekt gespeichert wird. Hier hilft nur alle 3 Dateien von Hand löschen und dann Neustarten.
- Im Netzwerkbetrieb und bei Verwendung des Scriptbefehls „Hintergrund=hell" kommt es zu unterschiedlichen Farb-Darstellungen (z. B. sind braun und gelb vertauscht). Für diesen Fall ist „Hintergrund=hellgrau" die bessere Wahl.
- Wird im Scriptbefehl „Netzwerk= .." die Option „RAM" angegeben (Linuxsystem) und der Programmstart erfolgt unter DOS/Windows, so erfolgt die Fehlermeldung: „IOFehler beim Zugriff auf D:\aktuell.txt # 381/152". Die Option „RAM" ist dann vor dem Neustart zu entfernen.

8. Sprechphrasen für Sprachausgabe mit RN – SPEAK Board (nur in Version 6.0.7/8 implementiert!)

1. Zahlen, Datum

Nr.	***Text***	***Nr.***	***Text***	***Nr.***	***Text***
0	(300 ms Pause)	22	hundert	44	September
1	null	23	tausend	45	Oktober
2	ein	24	te	46	November
3	zwei	25	erste	47	Dezember
4	drei	26	dritte	48	Sonntag
5	vier	27	ste	49	Montag
6	fünf	28	eins	50	Dienstag
7	sechs	29	und	51	Mittwoch
8	sieben	30	plus	52	Donnerstag
9	acht	31	minus	53	Freitag
10	neun	32	Komma	54	Sonnabend
11	zehn	33	Uhr	55	Feiertag
12	elf	34	vor	56	vorgestern
13	zwölf	35	gegen	57	gestern
14	zwanzig	36	Januar	58	heute
15	dreißig	37	Februar	59	morgen
16	vierzig	38	März	60	Minuten
17	fünfzig	39	April	61	Stunden
18	sechzig	40	Mai	62	am
19	siebzig	41	Juni	63	es
20	achtzig	42	Juli	64	ist
21	neunzig	43	August	65	der

9. Stichwortverzeichnis

2

24h-Grafik....127f.

A

Adressen....16
alarm.txt....63, 74
Alarmseite....44, 177
AM2320....17
Ampelsteuerung....211
Analog/Digital Wandler....90
Analogsensoren....47
Anemometer....96
Anzeige....108
astabiler Multivibrator....132
Astrofunktionen....20, 80
Außensensor....154, 229
Automatikbetrieb....57

B

Balkenanzeige....112
Beaufort....111
Bedingter Multiplexer....166
Betriebsminuten....143f.
Betriebsstunden....129
Betriebsstundenzähler....140, 150f., 179
Bundesland....124
Bus....9
Busproblem....13
Bussensoren....17
Bussequenz....10f.
Bustakt....70
Bustasten....88f., 177, 181
Byte....14

C

COM-Port....40, 70

D

Datei „gestern.daz“....129
Datei „gestern2/3.daz“....129
DEMO....46
Digital/Analog Wandler....90
Diskettenlaufwerk....233
DS1631....17, 96
Durchflussmessung....149

E

Echtzeitmonitor....60
Eichgrafik....47
Eigendiagnose....62
Einschaltzeit....129
Energiemenge....129
Energiezähler....74, 96
Ereignis.log....62, 73

F

Flashdisk....233
Fontanzeige....56
Frequenzmessung....97, 216

G

Geisterhaus....204
Generator....141
Gitter....41
Glättung....109
Grafik....127

H

Hauptmenü....45
hcdata.txt....85
Heizkennlinie....180
Hintergrund....44, 72
HINWEIS!....43, 62f.

I

I-Regler....171
Integer....14
interne Konstanten und Variable....194
IPS-Steuerung....160

K

Kalender....202
Kanäle....16
Konstanten....194
Kursortasten....48

L

Laufband.log....41, 62, 73
Laufschrift....72
Laufzeitbegrenzung....200
Legende....125
liesmich.txt....40
lifing....54
Linux....29
Lizenz.RPT....200
LM75....17, 96
Local Apparent Sidereal Time....20
lokalen Ortssternzeit....20

M

MAX1236....16, 92
Menüauswahlmodus....41, 45
Menüdarstellung....83
Menüseite....41
Mischer....161
Mittelwert....109
Mitternacht....83, 233
Modulmatrix....129

Modus.....45, 72, 108, 177
Monitoring.....209
Mülltonnenfunktion.....126
Multisensor.....229
Multitasking.....7
N
Netzwerkfähigkeit.....27, 35
Netzwerkfunktionen.....213
Notbetrieb.....22, 32
O
Oszilloskop.....61
P
Parameteranzeige.....113
PC-Watchdog.....81
PCF8574.....**10,** 16, 88
PCF8591.....**10,** 16, 90
Planeten.....20
Probleme.....232
Programmfehlerkodes.....232
Pumpenregelung.....160
R
Raumsensor.....229
Raumsensor Statusbyte schalten.....156
Raumsensoren.....17, 53
Real.....14
Reset Hinweistext.....41
RUNTIME ERROR.....232
S
S0-Zähler.....74, 96, 195
Schlüsselwörter.....14, 70
Hardware.ini.....
Alarm=.....74
ALog=.....76
Anzahl_PCF8574P=.....71
Anzahl_PCF8591=.....71
Autokill=.....74
BS=.....71
Bussequenz=.....71
Bustakt=.....70
COM=.....70
EDB=.....82
ELog=.....73
Geo=.....80
Hintergrund=.....72
HzgLog=.....79
INT21=.....71
ISOVGA=.....75
Laufschrift=.....44, 73
live=.....80, 213
LLog=.....73
Netzwerk=.....80, 213
NN=.....80
PCWatchdog=.....81
PfadTaste=.....80
S0Intervall=.....74
SDAKorrektur=.....71
Seitenwechsel=.....72
SensLog=.....79
Sommerzeit=.....72
SPSLog=.....77
SPSLog2=.....77
VGA=.....75
Vorrang01=.....92
Watchdog=.....72
XFaktor=.....81
zk=.....74
zkonst=.....73
Hauscomp.ini.....
Analog=.....93
Analogwert=.....104
Anemometer=.....96
Anzeige=.....108
Anzeigeseiten=.....84
BezDatBereich=.....125
Binär=.....88
Binär26=.....88
Bundesland=.....124
Bussensor=.....96
DataIN=.....69, 87
DatBereich=.....125
Datum=.....124
Echtzeitmonitor=.....128
Frequenz26=.....97
graph_a=.....127
graph_a2=.....127
graph_b=.....128
graph_b2=.....128
Handstartseite=.....84
HCdata=.....69, 86
HCdata2=.....86
HCdata3=.....86
Hinweis!=.....85
HzgLogA01=.....97
HzgLogA12=.....97
HzgLogS01=.....97
HzgLogS13=.....97
Menü11x=.....99
Menüform=.....44, 83
Müll=.....126
Norm=.....83
PCF8574P=.....88
PCF8591=.....90

Raumsensor=....91
Raumsensor2000=....91
RaumsensorWMM=....91
Regenloeffel=....91
S0x=....96
Sensor=....90
Sensor2=....91
SensorAt26=....94
SensorMAX1236=....92
SensorMax1236K=....92
SensorMax1236L=....92
SensorMAX1236M=....92
Taste=....100
Tastengruppe=....104
TasteOK=....85
Test=....84
text=....99
textgross=....99
Windgrd=....95
Windkode=....95
WindNord=....95
Wochenkorrektur=....84
z=....130
Zeile=....100
Schreibschutz....200
Sensoren....90
seriellen PC-Schnittstelle....70
Softkey....100, 104
Softwaremodul....129
Solarpumpe....182
Sommerzeit....43, 60, 72, 232
Speicher....140
Sprachausgabe....60, 236
SPS- Module....
0....132
1....132
10....133
101....184
102....185
105....153
106....155
107....186
108....187
109....168
11....134
110....188
111....190
112....191
113....192
114....193
12....134
13....152
14....134
15....141
16....135
17....151
18....146
19....165
2....146
20....166
21....181
22....182
23....166
24....183
25....161
26....183
27....152
28....175
29....135
3....147
30....156
31....156
32....157
33....157
34....158
37....149
38....150
39....136
4....149
40....136
41....136
42....152
43....138
44....145
45....138
46....137
5....150
50....171
56....139
57....139
58....140
59....140
6....132
60....141
61....141
62....142
63....142
64....143
65....143
66....144
67....144
68....144

69.....162
7.....133
70.....163
71.....163
72.....163
73.....164
74.....164
75.....164
76.....159
77.....165
78.....148
79.....167
8.....133
80.....167
81.....169
82.....170
83.....171
84.....172
85.....171
86.....149
87.....173
88.....172
89.....160
9.....150
90.....174
91.....175
92.....175
93.....176
94.....177
95.....178
96.....179
97.....179
98.....180
99.....180
Abfallverzögerung.....143
absolute Feuchte.....184
Addierer.....163, 166
Alarmseite.....177
Analogkanal auf Festwert setzen.....157
Analogkanal EIN/AUS schalten.....157
Analogkanal größer Festwert.....147
Analogkanal kleiner Festwert.....148
Analogkanal lesen.....148
Analogmultiplexer.....167
Anprechverzögerung.....143
Anstiegsgeschwindigkeit.....188
Anzugsverzögerung.....143
astabiler Multivibrator.....141, 152
Ausgabe Analogwert.....159
Bedingte Multiplikation mit „-1“.....168
Bedingter Multiplexer.....166
Betriebsstundenzähler.....140
Betriebsstundenzähler für Binärkanal.....150
Betriebsstundenzähler für SPS Modul.....151
Binärkanal über Hausbus schalten.....156
Binärkanal vom Hausbus lesen.....146
Bistabiles Relais.....142
Bitanalyse Analogkanal.....151
Byte auf I2C Bus ausgeben.....158
Dampfdruck.....184
Differenz.....164, 172
Differenzierglied.....141f., 145
Differenzverstärker.....163
Dividierer.....166f.
Einsmodul.....132
Energiezähler.....179
Faktor-Kennlinie.....187
Fensterdiskriminator.....137
Fensterkomparator.....136f.
Flip- Flop, bistabil.....133
Funktionsblock.....192
gefühlten Temperatur.....175
Glättung.....170, 189
Gültigkeitsmodul.....193
Hilfsmodul für Tastensatz.....174
Höhenmesser-Modul.....176
Hysteresis.....164
I-Regler.....171
Imitation eines Tastendrucks.....181
Impulsmessung.....149
Integrator.....171
IPS-Steuerung.....160
Kennlinie.....180, 183
Klimaberechnungen.....184
Konstantenmodul.....163
Leermodul.....132, 175
Luftdurchsatz.....192
Maximum.....188
Maximumbegrenzer.....164
Minimum.....188
Minimumbegrenzer.....164
Mischer über Hausbus schalten.....161
Mittelwert (IIR).....188
Monoflop.....141, 145
Multiplexer.....167
Multiplizierer.....163, 166f.
Multiplizierer 2.....172
Multivibrator.....132, 141, 152
Notbetrieb.....180
ODER.....133f.
ODER/UND/ODER.....135
P-Regler.....163

PI- Regler.....169
Prozentwert.....188
pT1- Verhalten.....170, 189
Pulsweitenmodulation (PWM).....165
Quantum- Zähler.....139
Raumsensor.....153
Raumsensor-Zustand vom Hausbus lesen..155
Rollläden.....190
Sättigungsdampfdruck.....184
Schaltmodul für Solarpumpen.....182
Schaltmodul für Umwälzpumpen.....183
Schwingungspaketsteuerung.....165
Signalanalyse.....188
Signalumschalter.....167, 180
Signalverkürzung.....141
Skalierer.....163, 166
Skalierung.....148
Sollwertvorgabe.....190
Speicher.....133, 140
Speicher/ Flip-Flop, bistabil.....132
SPS Modul größer Festwert.....162
SPS Modul kleiner Festwert.....162
Statistikmodul.....173
Stellantriebe.....190
Stromstoßschalter.....142
Tagesbetriebsminuten.....144
Taktgeber.....141
Tastverhältnis.....141, 165
Taupunkt.....184
Teiler 2:1.....142
Terminzeit.....149
Tiefpass.....189
Übernahme-Modul.....185
Überwachung.....138
Umschalten der Bildschirmseiten.....177
UND.....133
UND/ODER.....134f.
Unempfindlichkeit.....172
Variablen- Speicher.....139
Variablen-Modul.....186
Vergleicher.....136f.
Volumenstrom.....192
Vorgabe einer linearen (Heizungs-) Kennlinie.
180, 183
Wärmeleistungsmesser-Modul.....178
Wärmemengensensor.....153
Wecker.....149f.
Wetterprognose.....191
Windchill- Modul.....175
Zähler.....136
Zeitmodul.....149f.
Zufall.....152
Zuweisungs-Modul.....175
Zweipunktregler.....164
.....187
-18.....146
-2.....146
-23.....166
-27.....152
-3.....148
-46.....137
-69.....162
-80.....167
-96.....179
-97.....179
Startwert.....88, 93
statische Anzeigemodus.....22, 44, 73
Stoppuhr.....111

T

Tageseigenschaft.....125
Tagesertragszähler.....179
Tagesmittelwert.....109
Tastaturkode.....88
taste.txt.....44, 63, 80
Tasten.....100
Tastengruppe.....100, 104
Tastensatz.....174
Tastverhältnis.....144
Temperatursensoren.....17
Testmodus.....84
Thermometer.....111
Trennzeichen.....15
Treppenlichtautomat.....143

U

Unempfindlichkeit.....172
Unix-Zeit.....201

V

Variable.....194
virtuelle Helligkeitssensor.....21
Vollmond.....21
Volumenstromberechnung.....149

W

Wärmemengenzähler.....**91,** 179
Watchdog.....72
Webinterface.....213, 228
Windows XP.....232
Windrichtungssensors.....95

Z

Zeigerinstrument.....111
Zeitinkonsistenz.....61
Zeitinkonsistenz.....43
Zeitkern.....61

Zirkulationspumpe.....207
„
„live“ Modus.....63, 80