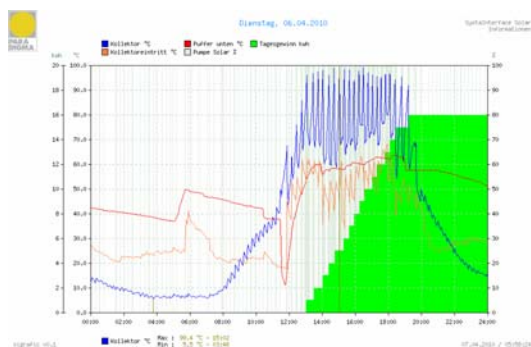


Systainterface Data Loader & Systainterface Grafik Maker für Linux

Entwickelt für das Systainterface der Firma



Inhaltsverzeichnis

1. Vorwort	3
2. Historie	4
3. Funktionsumfang der Programme	5
3.1. Datenlogger 'siloader'	5
3.2. Grafik Programm 'sigrafic'	5
4. Voraussetzung Hardware	6
5. Voraussetzung Software	7
6. Informationen zum Programm Packet	8
7. Voraussetzung zur Installation der Software	9
8. Installation des Software Paketes	11
9. Programm 'siloader'	12
9.1. USB seriell Treiber	12
9.2. MySQL Datenbank	12
9.3. Konfigurationsdatei siloader.conf	13
9.4. Programm Parameter 'siloader'	16
9.5. Statusangaben im Daemon Modus	18
10. Programm 'sigrafic'	20
10.1. Konfigurationsdatei sigrafic.conf	20
10.2. Programm Parameter 'sigrafic'	24
10.3. Erzeugen von dynamischen Webseiten	26
11. FAQ (häufige Fragen)	28

1. Vorwort

Für die Auswertung und grafische Aufbereitung der Daten des *Systainterface* der Firma *emsystem engineering* entstand dieses Projekt.

Für genauere Informationen zum Systainterface und der dort vorhandenen Software kann auf der Webseite <http://www.emsystem.de> nachgesehen werden.

Ziel war es eine Software zu entwickeln, das die Daten des Systainterfaces permanent abrufen und diese für eine spätere Aufbereitung in einer Datenbank ablegt.

Da die Firma *emsystem engineering* bereits entsprechende Software für MS Windows bereitstellt, wurde hier eine Lösung unter einer Linux Umgebung angestrebt.

Zur Realisierung wurden zwei Programme entwickelt, die für die Speicherung der Daten sowie der grafischen Aufbereitung der Daten verantwortlich sind.

Dieses Dokument beschreibt daher die Installation und Funktionsweise der beiden Linux Programme *siloader* und *sigrafic*

Die gesamte Software aus diesem Archiv steht zudem unter der *GNU General Public License* was bedeutet, dass der komplette Sourcecode verfügbar ist, dieser frei weitergegeben werden muss und selbstverständlich beliebig erweitert oder geändert werden darf.

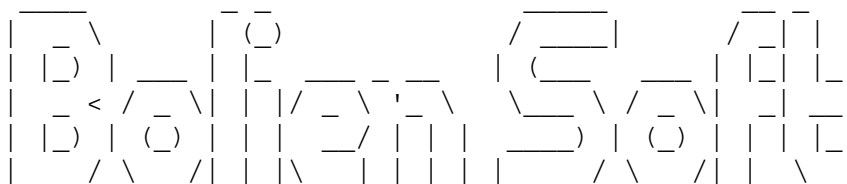
Zudem übernimmt der Autor keine Gewährleistung zur richtigen und fehlerfreien funktionsweise der Programme.

Einen besonderen Dank will ich an dieser Stelle noch Herrn Martin Steppuhn von der Firma *emsystem engineering* aussprechen, für die freundliche Unterstützung dieses Projektes sowie der Bereitstellung dieser Software auf seiner Homepage.

Für Feedback & Vorschläge zu diesem Projekt bin ich gerne per Mail unter wbolien@bolien.de zu erreichen.

Unter www.bolien.de/solar kann man sich aktuelle Grafiken meiner Anlage (besitze selber nur eine Systs Solar Steuerung) anschauen.

Wolfgang Bolien April 2010



2. Historie

Änderungen sowie Bug fixes werden hier aufgelistet

04.2010 Version 0.1 von *siloder* & *sigrafic*

3. Funktionsumfang der Programme

Die beiden Programme *siloader* und *sigrafic* stellen verschiedene Funktionen bereit, die hier kurz aufgezeigt werden.

3.1. Datenlogger 'siloader'

Das Programm *siloader* dient dazu alle Daten die das Systainterface von den Heizungsreglern SystaComfort oder SystaCompact und Solarreglern SystaSolar abrufen zu sammeln. Es werden folgende Funktionen unterstützt:

- Start im Singlemodus (einzelner Abruf von Daten) oder im Daemon Modus (permanenter Abruf der Daten)
- Export der Daten in eine MySQL Datenbank
- Export der Daten in eine CSV Datei
- Start von externen Programmen bei Datenupdates oder Fehlern
- Frei wählbare Zeiten zum Abrufintervall vom Systainterface sowie zum Update der MySQL Datenbank
- Frei wählbare Zeiten für den Start des externen Update Programms
- Lokaler Datencache zur Reduktion des Datenverkehrs im lokalen Netz
- Konfiguration der Funktionsweise über eine Konfigurationsdatei
- IPC Dienste für das Programm *sigrafic*
- Anlegen einer Logdatei falls gewünscht
- Mitschnitt der Daten vom Systainterface falls gewünscht
- Anzeige aktueller Parameter im Daemon Modus

3.2. Grafik Programm 'sigrafic'

Mit dem Programm *sigrafic* können die Daten die vom *siloader* Programm in der MySQL Datenbank abgelegt wurden grafisch aufbereitet werden. Es kommen folgende Funktion zum Einsatz:

- Erzeugen verschiedener grafischer Diagramme aus den vorhandenen Daten der MySQL Datenbank
- Erstellen auf Basis von Tages, Wochen, Monats und Jahreszyklen
- Erzeugen einer dynamischen HTML Seite zur Anzeige mittels Webserver im lokalen Netz
- Anzeigen der aktuellsten Werte der Systa Regler in einer Statusübersichtsgrafik
- Frei definierbare Farben und Werte die gezeigt werden sollen
- Kommunikation zwischen den beiden Programmen mittels IPC (Sockets)
- Konfiguration der Funktionsweise und der anzuzeigenden Werte über eine Konfigurationsdatei
- Anzeige des aktuellen Status des *siloader* Servers

4. Voraussetzung Hardware

Geeignete Hardware gibt es natürlich unzählige. Was aber sicher für jeden wichtig ist, ist ein niedriger Stromverbrauch, da der Rechner ja permanent in Betrieb sein muss um kontinuierlich Daten vom Systainterface auszulesen.

Aus den vielen Möglichkeiten seien hier zwei genannt:

- **Fujitsu Siemens Futro S200** (oder S220 - S400)

Dieser Rechner wird zwar schon eine weile nicht mehr hergestellt, man kann ihn aber sehr oft noch gebraucht erwerben.

Er bietet den Vorteil keine Festplatte und keine Lüfter zu haben, sondern das Betriebssystem von einer CF Karte zu booten.

Er hat ein eingebautes IDE Interface (falls eine Festplatte nachgerüstet werden soll) und eine RJ45 Netzwerkschnittstelle.

Natürlich sind auch mehrere USB Schnittstellen zum Anschluss des Systainterfaces vorhanden. Zudem kann eine PCI Karte z.B. für eine WLAN Anbindung eingebaut werden. Er hat einen Stromverbrauch von ca. 10 W



- **Fox Board G20**

Hier handelt es sich um ein 'embedded' Linux System, das ebenfalls ohne Festplatte und Lüfter auskommt. Das Linux Betriebssystem (Debian Linux) wird auf einer SD Karte bereitgestellt.

Es werden Netzwerkschnittstellen und USB Anschlüsse bereitgestellt, was diese Hardware ebenfalls für den Einsatz geeignet macht.

Zudem wird nur ein Stromverbrauch von ca. 1 W gebraucht.



Es sei an dieser Stelle erwähnt, das für die Installation der Linux Programme eventuell ein weiterer Linux Rechner (oder eine VM unter Windows z.B. Virtualbox) verfügbar sein muss, da alle Programme ausschliesslich im Quellcode vorhanden sind und zuvor für das Zielsystem übersetzt werden müssen.

5. Voraussetzung Software

Für die Installation der hier beschriebenen Software wird ein Linux Betriebssystem vorausgesetzt, das aber keine grafische Oberfläche haben muss.

Für dieses Projekt kam die Linux Distribution Voyage Linux (<http://linux.voyage.hk>) zum Einsatz, das auf Debian Version 5.0 (Etch) basiert. Der Vorteil dieser Distribution ist, das sie speziell zur Installation auf so genannten 'readonly' Datenträgern wie CF oder SD Karten ausgelegt ist.

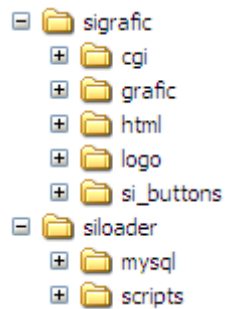
Es können aber auch andere Distributionen z.B. SuSE Linux verwendet werden. Die Anforderungen an die Software sind nicht besonders hoch. Es müssen nur Libraries zur Anbindung an USB (Systainterface) sowie zur Kommunikation mit einer SQL Datenbank vorhanden sein.

Programmiert (in der Programmiersprache C) und übersetzt wurden alle Programme auf einer separaten Linux Umgebung, da hier bereits alle nötigen Programme (Compiler, Linker, Headerdateien) vorhanden sind.

Was genau benötigt wird und wie dieses Installiert wird, ist im Kapitel 7 'Voraussetzung zur Installation der Software' beschrieben.

6. Informationen zum Programm Packet

Dieses Softwarepaket liegt als Linux GZ Archiv vor. Nach dem entpacken (Was auch unter MS Windows gemacht werden kann) finden sich folgende Verzeichnisse vor:



Es handelt sich hierbei um:

- sigrafic Sourcecode zum grafischen Auswertungsprogramm
- sigrafic/cgi CGI Script für einen lokalen Webserver
- sigrafic/grafic Beispielgrafiken
- sigrafic/html Beispiele für statische Webseiten
- sigrafic/logo Beispiele von Logos die in den Grafiken angezeigt werden können
- sigrafic/si_buttons Buttons die zur Erzeugung von dynamischen Webseiten benutzt werden können

- siloader Sourcecode zum Datenlogger
- siloader/mysql Scripte zum anlegen der MySQL Datenbank
- siloader/scripts Beispiele für Shellscripte (Daemonstart, Errorscript)

7. Voraussetzung zur Installation der Software

Wie das Softwarepaket installiert wird, hängt maßgeblich davon ab ob der Rechner auf der später die Programme laufen auch derjenige Rechner ist auf dem der Sourcecode übersetzt (kompiliert) werden soll.

In allen Fällen ist aber ein vorhandenes Linux Betriebssystem Voraussetzung zum Betrieb und zur Übersetzung der Programme.

Wie bereits erwähnt kann dies eines der im Umlauf befindlichen Distributionen wie z.B. Debian, SuSE Linux oder auch Voyage Linux sein. Wobei eine grafische Oberfläche (Gnome, KDE) nicht benötigt wird, aber auch nicht stört.

Natürlich ist es auch möglich für die Übersetzung der Programme eine VM Umgebung wie z.B. Virtualbox oder VM Ware zu benutzen.

Klar ist aber, dass die später zu startenden Programme diesem Programm Packet nur in Quellform vorliegen und zuvor für das Zielsystem übersetzt werden müssen. Der Rechner auf dem übersetzt wird, muss aber nicht der Rechner sein auf dem die Programme später laufen werden.

Es ist zudem möglich das grafische Auswertungsprogramm *sigrafic* auf einen anderen Rechner als wo der Datenlogger *siloader* läuft zu installieren, da *sigrafic* die Daten ja vom SQL Server abholt oder per IPC mit dem Datenlogger kommuniziert.

Wie man sieht, sind viele Varianten möglich, das muss jeder selber entscheiden.

Voraussetzung für den korrekten Start der Programme ist das Vorhandensein folgender Libraries:

siloader MySQL Client Librarie

sigrafic MySQL Client Librarie
Grafische Librarie GD
PNG & JPG Librarie

Zum übersetzen der Programme sollte eine entsprechende Entwicklungsumgebung auf dem Linux Rechner vorhanden sein, dazu zählen unter anderem der *GNU C Compiler* und entsprechende Tools wie *make* usw.

Außer den Libraries die für den Betrieb der Programme nötig sind, werden auch die Development Sourcen der Libraries benötigt.

Unter Debian Linux 5.0 (Etch) kann dies mit Hilfe des Packet Programms *apt-get* realisiert werden.

Folgende Kommandos sollten als User 'root' ausgeführt werden um die entsprechenden Pakete zu installieren:

```
apt-get install libmysqlclient15off
apt-get install libgd2-xpm
apt-get install libpng12-0
apt-get install libjpeg62
```

zusätzlich nötige Pakete in der Entwicklungsumgebung:

```
apt-get install libmysqlclient15-dev
apt-get install libgd2-xpm-dev
apt-get install libpng12-dev
apt-get install libjpeg62-dev
```

In anderen Linux Systemen erfolgt die Installation von Software eventuell anders z.B. mittels YAST unter SuSE Linux. Das muss individuell geklärt werden.

Damit eine Kommunikation des **siloader** Rechners mit dem Sustaininterface erfolgen kann, muss auch der **usbseriell** Treiber geladen sein. Dazu mehr im Kapitel 9 Programm 'siloader'.

Es soll auch nicht unerwähnt bleiben, dass zum Speichern der Daten des Datenloggers ein MySQL Server im Netzwerk vorhanden sein muss. Dieser muss natürlich nicht zwingend auf dem Rechner auf dem **siloader** oder **sigrafic** läuft vorhanden sein.

Will man zusätzlich auch eine lokale dynamische Webseite unterhalten, sollte auch ein entsprechender Webserver wie z.B. Apache mit CGI Unterstützung vorhanden sein.

Wie dies zu bewerkstelligen ist, würde den Rahmen dieser Dokumentation sprengen, ich empfehle daher die vielfältigen Möglichkeiten des Internets zu nutzen.

8. Installation des Software Paketes

Nach der Vorbereitung des Linux Systems wie im vorherigen Kapitel beschrieben, kann das Softwarepaket z.B. unter dem Verzeichnis `/usr/src` entpackt werden.

Es findet sich dann die Verzeichnis Struktur wie im Kapitel 6 beschrieben vor.

Da sich im Wurzel Verzeichnis der Installation bereits ein **makefile** befindet sollte mittels des Linux Kommandos **make** die Übersetzung der beiden Programme **siloader** und **sigrafic** beginnen.

Das Resultat sollten die beiden Programme **siloader** und **sigrafic** sein, die in den gleichnamigen Verzeichnissen zu finden sind, falls der Compiler seine Arbeit ohne Fehler verrichtet hat.

Es wird empfohlen diese beiden Programme nun an eine entsprechende Stelle wie z.B. `/usr/local/bin` des Zielsystemes zu kopieren.

Damit die Programme auch ihre entsprechenden Konfigurationen finden, müssen die beiden Dateien **siloader.conf** und **sigrafic.conf** aus den entsprechenden Verzeichnissen nach `/etc` des Linux Zielrechners kopiert und angepasst werden.

In beiden Konfigurationsdateien können Kommentare eingefügt werden, diese Zeilen sollten immer mit dem Zeichen `#` beginnen.

Falls **siloader** und **sigrafic** auf verschiedenen Rechner laufen muss natürlich nur die entsprechende Konfigurationsdatei nach `/etc` kopiert werden.

Was die Parameter in den Konfigurationsdateien zu bedeuten haben wird in den Kapiteln 9 Programm 'siloader' und 10 Programm 'sigrafic' beschrieben.

Falls der Datenlogger **siloader** im so genannten Daemon Modus gestartet werden soll, kann die Datei **siloader.sh** aus dem Unterverzeichnis **siloader/scripts** verwendet werden.

Diese Datei wird in einem Debian System nach `/etc/init.d` kopiert und in den Startverzeichnissen z.B. `/etc/rc2.d` ein symbolischer Link angelegt (siehe Dokumentationen zu Debian) damit **siloader** beim Booten des Rechners automatisch startet.

In anderen Linux Umgebungen kann der Start eventuell auch anders erfolgen, das muss gegebenenfalls geprüft werden.

9. Programm 'siloaders'

Kurz gesagt dient das Programm **siloaders** dazu alle Daten die das Systainterface liefert in einer MySQL Datenbank abzulegen. Dazu muss das Systainterface per USB Anschluss an den Rechner (hier Host genannt) angeschlossen werden und eine MySQL Datenbank zu Verfügung stehen (Die aber auf einem anderen Rechner laufen kann).

9.1. USB seriell Treiber

Damit der Hostrechner auch eine Verbindung zum Systainterface (per USB) aufbauen kann muss eine USB seriell Emulation auf dem Host gestartet werden, da die Kommunikation zum Systainterface seriell erfolgt.

Zu diesem Zweck gibt es bei Debian 5.0 (Etch) ein Kernelmodul das eine serielle Schnittstelle am USB Anschluss nachbildet. Der Treiber kann mittels **modprobe usbserial** geladen werden. Mehr Informationen dazu unter <http://wiki.debian.org/usbserial>. Dieser Treiber sollte auch auf anderen Linux Distributionen vorhanden sein.

9.2. MySQL Datenbank

Ist diese Hürde erst einmal genommen, sollte als nächstes eine Datenbank auf dem vorhandenen MySQL Server angelegt werden. Hierzu gibt es allerdings zuerst einiges zu beachten.

Je nachdem welche Systs Regler man hat, sollte man die Struktur der Datenbank Tabellen wählen.

Das Systainterface liefert zwar immer genau die gleiche Anzahl an Daten ab, falls man aber z.B. wie bei mir nur einen Solarregler hat, macht es keinen Sinn Daten, die nur leer sind, in der Datenbank zu speichern.

Zum anlegen der Datenbank Tabellen findet man im Unterverzeichnis **siloaders/mysql** drei verschiedene SQL Scripts die je nach Anlagen Konstellation folgendermassen zu nutzen sind:

- Hat man den Regler SystsComfort oder SystsCompact benutzt man die Datei **siloaders_1.sql** zum Anlegen der Datenbank Tabellen.
- Betreibt man nur den Regler SystsSolar sollte die Datei **siloaders_2.sql** zum Anlegen der Datenbank Tabellen benutzt werden.
- Falls man die Regler SystsComfort oder SystsCompact und SystsSolar betreibt ist **siloaders_3.sql** zum Anlegen der Datenbank Tabellen geeignet.

In allen drei Fällen kann das Anlegen der Tabellen erst erfolgen wenn zuvor eine Datenbank mit dem Namen **systs** angelegt wurde. Zur Datenbank sollte auch ein entsprechender Benutzer mit Passwort eröffnet werden, der vollen Zugriff auf diese DB hat. Ist dies erfolgt kann mittels:

```
mysql --database=systs --user=systs --password=systs < siloaders_1.sql    oder  
mysql --database=systs --user=systs --password=systs < siloaders_2.sql    oder  
mysql --database=systs --user=systs --password=systs < siloaders_3.sql
```

die passende Tabelle auf der Datenbank **systs** angelegt werden. Die Parameter **--user** und **--password** sollten den zuvor angelegten Benutzer und das entsprechende Passwort enthalten.

Als nächstes sollte nun die Konfigurationsdatei angepasst werden, die als */etc/siloader.conf* zu finden sein sollte.

9.3. Konfigurationsdatei siloader.conf

Folgende wichtigen Parameter (solche die nur den Daemon Modus betreffen sind hier blau dargestellt) sollten in der Datei angepasst werden:

systaconf	Dieser Parameter legt fest welche Regler die eigene Anlage besitzt und was auf der MySQL Datenbank abgelegt wird. Bitte einen Wert von 1 bis 3 eintragen. Der hier eingetragene Wert sollte mit der zuvor angelegten MySQL Datenbank übereinstimmen (siehe auch 9.2 MySQL Datenbank)
port	Hier wird der USB Port zur seriellen Kommunikation mit dem Systainterface eingetragen. Der Wert ist abhängig vom verwendeten Anschluss und dem Treiber wird aber in der Regel ttyUSB0 sein.
interval	Wie oft die Daten vom Systainterface abgeholt werden, wird hier in Sekunden angegeben. Falls eine hohe Aktualisierungsrate gewünscht ist, sollte man auch öfters in der Minute die Daten abholen. Es gilt aber zu bedenken dass eine hohe Rate auch ein schnelleres Anwachsen der Datenbank nach sich zieht.
comparemodus	Da es nicht unbedingt nötig ist, Datensätze die sich nicht vom vorigen unterscheiden, auf der Datenbank abzulegen (falls man z.B. eine Solaranlage betreibt die nachts keine grossen Schwankungen macht) kann hier eine Toleranz in °C angegeben werden, die sich ein Temperaturwert ändern muss, um einen neuen Datensatz auszulösen. Mittels dieser Technik ist es möglich das Datenaufkommen in der Datenbank zu reduzieren. Es sei aber angemerkt, das jede andere Änderung im Systainterface (z.B. Pumpe geht An oder Aus) auf alle Fälle einen neuen Datensatz generiert.
datacache	Da es nicht zweckmässig ist, nach jedem Datensatz vom Systainterface diesen auf der Datenbank abzulegen, (Es wird eventuell viel Netzwerklast und Datenbanklast erzeugt) bietet der Datenlogger die Möglichkeit alle gesammelten Daten vom Systainterface in einem lokalen Datacache zu sammeln und diesen zyklisch in die Datenbank zu schreiben (und danach zu löschen). Ob der Cache als Memorycache oder als Dateicache arbeitet wird hier eingestellt. Für welchen Cache man sich entscheidet muss jeder selber entscheiden. Der Dateicache hat aber den Vorteil, das beim beenden des Datenloggers dieser erhalten bleibt und beim nächsten Start wiederverwendet wird. Bei Verwendung des Dateicache muss zudem der komplette Name der Cachefile angegeben werden.

updatemodus

Wie oft die gesammelten Daten vom Systainterface (eigentlich vom Datencache) in der Datenbank abgelegt oder in die CSV Exportdatei geschrieben werden, wird hier eingestellt.

Es stehen insgesamt 4 verschiedene Modi zur Verfügung die sich jeweils unterscheiden.

Modus 1 wird benutzt um bis zu 30 Zeitangaben zu definieren (hh:mm) in denen jeweils die Daten in die Datenbank geschrieben werden.

Bei Modus 2 können auch bis zu 30 Zeiten angegeben werden, hier allerdings nur die Minutenangaben (mm), wann in die Datenbank abgelegt wird. Im Gegensatz zu Modus 1 spielt hier die Stunde keine Rolle. Nützlich falls man zu jeder Stunde z.B. alle 15 Minuten die Daten ablegen will.

Modus 3 ist dafür gedacht, nach einer eingestellten Zeit, (z.B. alle 30 Minuten) die Daten auf die Datenbank zu schreiben.

Modus 4 wird nicht von der Zeit bestimmt, sondern der Anzahl der Records die vom Systainterface abgeholt wurden. Bei einer Angabe von z.B. 100 wird also nach 100 Records vom Systainterface in die Datenbank geschrieben. Nach dem erfolgreichen Schreiben der Daten in die Datenbank wird danach der Datencache gelöscht.

Kann, aus welchen Gründen auch immer, keine Verbindung zur Datenbank aufgebaut werden, so wird das Errorprogramm einmal gestartet (falls angegeben) und dieser Updatezyklus ausgesetzt. Der Datencache wird in diesem fall natürlich nicht gelöscht. Beim nächsten Updatezyklus ist die Datenbank eventuell wieder vorhanden und die Daten werden in die Datenbank übertragen und der Datencache gelöscht.

dbserver
dbname
dbuser
dbpass

Hier sollten die verwendeten Datenbank Parameter wie Server und Benutzer angegeben werden. Wie bereits erwähnt kann der DB Server auch auf einer anderen Maschine laufen. (Dies wird sicher die Regel sein, da bei Einsatz eines kleinen Hostrechners keine lokale Festplatte vorhanden ist)

exportfile

Das Programm bietet zusätzlich die Möglichkeit alle Daten in eine CSV Datei (Comma separated Value) abzulegen. Diese Möglichkeit kann auch genutzt werden, falls keine MySQL Datenbank vorhanden ist.

updateprog

Da der Datenlogger je nach Angaben in der Konfiguration die gesammelten Daten zyklisch in der Datenbank ablegt, ist es hier möglich ein Programm anzugeben, das nach jeder Aktualisierung der Datenbank gestartet wird. (z.B. das grafische Auswertungsprogramm *sigrafic* oder der Upload der Grafiken mittels FTP auf eine Webseite)

updateprogmodus

Wann das Update Programm gestartet wird, kann hier beeinflusst werden. Es gibt insgesamt 3 Modi.

Modus 1 wird benutzt um bis zu 30 Zeitangaben zu definieren (hh:mm) an denen das Update Programm gestartet wird.

Bei Modus 2 können ebenfalls 30 Zeiten definiert werden, hier aber nur die Minutenangaben (mm), an denen das Update Programm in jeder Stunde gestartet werden soll. Bei Modus 3 wird das Update Programm gestartet nachdem die Daten erfolgreich auf die Datenbank geschrieben wurden. Es werden also keine Zeiten definiert.

errorprog

Falls Fehler im Datenlogger auftreten (Verbindungsfehler zum Systainterface oder zur Datenbank) so kann ein Programm gestartet werden das darauf reagiert (z.B. eine Mail versendet)

socketsservice

Soll dem grafischen Auswertungsprogramm *sigrafic* die IPC Kommunikation zu Verfügung gestellt werden, so kann dies hier mit Angabe einer 1 eingeschaltet werden.

Weiter sollten auch die restlichen Parameter in dieser Datei geprüft und angepasst werden. Entsprechende Angaben zur Bedeutung sind als entsprechender Kommentar in der Datei zu finden.

9.4. Programm Parameter 'siloader'

Sind alle vorherigen Dinge gemacht worden kann der Datenlogger gestartet werden. Wird das Programm ohne Parameter gestartet wird eine entsprechende Hilfe angezeigt, die die zu Verfügung stehenden Parameter anzeigt.

```
-----
SystaInterface (www.emsystech.de) Data Loader
siloader v0.1, Copyright (C) 2010 Wolfgang Bolien (wbolien@bolien.de)
-----

Usage :
siloader -s    Show SystaInterface Overview & Info (also in daemonize mode)
siloader -x    Fetch SystaInterface Data into DB and/or Exportfile
siloader -d    Start the program in daemonize mode
siloader -k    Stop the program if in daemonize mode
siloader -r    Store all Data immediately (only in daemonize mode)
siloader -u    Execute update program immediately (only in daemonize mode)
siloader -c    Reload the config file (only in daemonize mode)
siloader -f    additional parameter to set config file
                 (e.g. -f /home/siloader/siloader.conf)
siloader -?    Display this help
```

Die Programm Parameter bedeuten im Einzelnen:

- s Zeigt die aktuellen vom Systainterface empfangenen Daten an. Falls das Programm zuvor im Daemon Modus gestartet wurde, werden zusätzlich Angaben zum Status des Daemons gezeigt. Was diese Angaben bedeuten wird im Kapitel 9.5 Statusangaben im Daemon Modus erklärt.
- x Mit Hilfe dieses Parameters wird der Datenlogger veranlasst den aktuellen Datensatz vom Systainterface abzuholen und in der Datenbank sowie in die CSV Exportdatei zu speichern (falls in der Konfigurationsdatei angegeben). Falls das Programm bereits im Daemon Modus gestartet wurde, macht dies natürlich keinen Sinn und es wird nichts gemacht.
Dieser Modus ist sicher für jene interessant die aus welchen Gründen auch immer die Daten per einzeltem Aufruf (z.B. von einem anderen Programm) vom Systainterface abholen wollen.
- d Hier wird der Datenlogger nun als Daemon gestartet und lädt die Daten nun automatisch nach den Angaben in der Konfigurationsdatei vom Systainterface. Wie es sich für ein Daemon Programm gehört werden entsprechende 'run' und 'lock' Dateien angelegt. Ein zweites starten des Programms wird so verhindert.
- k Mit diesem Parameter wird ein bereits gestarteter Daemon wieder beendet und z.B. die USB Schnittstellen Parameter wieder in den Ursprungszustand versetzt.
- r Falls der Datenlogger bereits als Daemon gestartet wurde, wird dieser veranlasst alle bisher gesammelten Daten vom Systainterface (Datencache) unverzüglich auf die Datenbank zu schreiben.
- u Das externe Update Programm wird gestartet, falls es konfiguriert ist und der Datenlogger als Daemon gestartet wurde.
- c Dieser Parameter dient dazu den ebenfalls zuvor gestarteten Daemon dazu veranlassen die Konfigurationsdatei neu einzulesen. Auf diese weise können Änderungen an der laufenden Konfiguration vorgenommen werden ohne den Daemon zu beenden. Es ist aber zu beachten, dass einige Parameter auf alle Fälle einen Neustart des Daemons voraussetzen.

- f Standardmässig wird die Konfiguration des Datenloggers aus der Datei */etc/siloader.conf* gelesen. Falls aber eine andere Datei dazu verwendet werden soll, kann dies hiermit getan werden (z.B. -f /home/siloader/siloader.conf)
- ? Zeigt diese Hilfe an.

Wird das Programm gestartet (Single Modus oder Daemon Modus) und ein Parameter in der Konfigurationsdatei ist falsch gesetzt z.B. falscher Pfad oder keine Verbindung zur Datenbank, dann wird eine entsprechende Fehlermeldung gezeigt und auch kein Daemon gestartet (falls mit -d angegeben)

Bei korrekter Funktionsweise des Programms, sollten sich irgendwann Daten in der MySQL Datenbank vorfinden, oder eine entsprechende CSV Exportdatei vorhanden sein (falls so konfiguriert).

Um diesen Vorgang im Vorfeld zu testen (kein Start des Programms als Daemon), kann auch mittels **siloader -x** das Lesen der Daten vom Sustaininterface, sowie das Speichern in der Datenbank gemacht werden.

9.5. Statusangaben im Daemon Modus

Wird im Daemon Modus der Status mittels **siloader -s** abgefragt werden unter anderem folgende Informationen gezeigt:

Daemon started at	Gibt die Zeit an, wann das Programm im Daemon Modus gestartet wurde.
Last requests time	Zeigt die Zeit, wann das letzte mal Daten vom Systainterface abgeholt wurden.
Requests made	Wie viel Requests bisher vom Systainterface gemacht wurden.
Records loaded	Anzahl der bisher vom Systainterface geladenen Records. Da eventuell nicht alle Requests vom Systainterface geladen werden (siehe Konfigurations-Parameter comparemodus), kann sich dieser Zähler vom vorangegangenen unterscheiden.
Records in Cache	Wie viel Records sich zurzeit im Datencache befinden (Dateicache oder Memorycache), wird hier angezeigt.
Last DB update time	Wann wurde das letzte mal auf die MySQL Datenbank geschrieben. Es wird (manual) gezeigt, falls der letzte DB update mittels siloader -r gestartet wurde.
Next DB update in	Falls der Daemon im Updatemodus '3' gestartet wurde (siehe Konfigurations-Parameter updatemodus) wird hier gezeigt in wie viel Minuten der nächste DB Update stattfindet.
DB Records stored	Anzahl aller bisher gespeicherten Records in der Datenbank.
CSV Records exported	Anzahl aller bisher gespeicherten Records in der CSV Export Datei.
Last Update Program execution time	Zeigt an, wann das externe Update Programm zuletzt ausgeführt wurde (Nur bei updateprogmodus '1' oder '2')
Checksum errors	Zahl der bisher empfangenen ungültigen Systainterface Records (Checksum Fehler).
Timeout errors	Anzahl der bisher aufgetretenen Systainterface Timeouts.
DB store errors	Falls Daten nicht auf die Datenbank geschrieben werden konnten (eventuell war die DB offline), wird hier die Zahl solcher Fehler gezeigt.

```
-----
SystaInterface Daemon Status          12.04.2010 - 11:34:25
-----

Daemon started at      : 12.04.2010 - 10:05:42

Last requests time     : 11:34:15
Requests made          : 217
Records loaded         : 190
Records in Cache       : 17
Last DB update time    : 11:30:00
DB Records stored      : 23478
CSV Records exported   : 359
Last Update Program
execution time         : 11:00:00

Checksum errors        : 0
Timeout errors         : 0
DB store errors        : 0
```

Die Daten die hier angezeigt werden, können auch als Grafik mit dem Programm [sigrafic](#) (siehe Kapitel 10 Programm 'sigrafic') erzeugt und als Webseite angezeigt werden.

Im Anschluss an diese Daten werden dann die aktuellen Sensorwerte (abhängig von der Anlagen Konfiguration) angezeigt.

```
-----  
SystaInterface Status & Info          (last received record !)
```

```
SystaSolar  
-----
```

```
Date - Time : 12.04.2010 - 20:00
```

Solar	Störung
-----	-----
TSA: 41.6 °C	Fühler : --
TSE: 24.5 °C	Solar : --
TWU: 41.0 °C	
Status: Frostschutz (5)	
Leistung: 0.0 kw	
Tagesgewinn: 10 kwh	
Ertrag: 371 kwh	

10. Programm 'sigrafic'

Wurde wie im vorherigen Kapitel beschrieben der Datenlogger erfolgreich in Betrieb genommen, befinden sich nun entsprechende Datensätze in der MySQL Datenbank. Selbstverständlich können diese Daten nun beliebig weiter verarbeitet oder genutzt werden. Zu diesem Zweck wurde dem Datenlogger ein weiteres Programm zur Seite gestellt, das aus diesen Daten entsprechende Grafiken aufbereitet.

Es werden im Moment insgesamt 6 verschiedene Grafiken zur Verfügung gestellt, die alle separat angepasst werden können z.B. welche Werte angezeigt werden sollen. Diese Grafiken zeigen jeweils verschiedene Zeiträume die ebenfalls bei der Erzeugung der Grafiken angegeben werden können. Diese sind Tages, Wochen, Monats, Jahresübersicht sowie eine aktuelle Infografik und eine Server Status Grafik.

Auch hier sollte vor der Nutzung des Programms als erstes die entsprechende Konfigurationsdatei die als */etc/sigrafic.conf* zu finden sein sollte angepasst werden.

10.1. Konfigurationsdatei sigrafic.conf

Folgende Parameter sollten in der Datei angepasst werden:

systaconf	Wie bereits in der Konfiguration des Datenloggers, sollte hier die Konfiguration der Anlage (welche Regler sind vorhanden) festgelegt werden. Bitte einen Wert von 1 bis 3 eintragen. Der hier eingetragene Wert sollte mit der angelegten MySQL Datenbank übereinstimmen (siehe auch 9.2 MySQL Datenbank)
headline_1 headline_2	Oben rechts in den erzeugten Grafiken kann ein entsprechender Text angezeigt werden (2 Zeilen) der hier festgelegt wird.
logo_name_day logo_name_week logo_name_month logo_name_year logo_name_over	Es besteht die Möglichkeit sich in den verschiedenen Grafiken jeweils ein entsprechendes Logo anzuzeigen. Hier wird für die verschiedenen Grafiken die Position (X,Y Koordinaten) und der Name festgelegt.
jpgpath pngpath	Es können entweder Grafiken im JPG Format oder im PNG Format erzeugt werden. Hier wird jeweils festgelegt wohin diese gespeichert werden. Es ist zu beachten das man sich auf ein Format festlegen muss und deshalb ein Eintrag auszukommentieren (mit # als erstes Zeichen) ist. Da PNG Grafiken auch Transparenz unterstützen, wird dieses Format bevorzugt.
name_day name_week name_month name_year name_over	Hier wird für jede Grafikdatei die erzeugt wird der Dateiname festgelegt. Zudem wird jeweils angegeben welche Daten von den Systa Reglern zur Anzeige verwendet werden sollen. Somit hat man für jede Grafik die volle Flexibilität was den Inhalt betrifft.

add_date	Wenn nötig kann an jeden Dateinamen der Grafiken der jeweilige Anzeigezeitraum angehängt werden. Dies kann helfen die Grafiken für z.B. eine Webseite zu unterscheiden. Gültige Werte sind 0 (für Aus) oder 1 (für Ein).
ta .. stoerung_solar	Es folgen nun alle von den Systa Reglern bereitgestellten und vom Datenlogger abgelegten Werte. Diese Werte können mit ihrem Namen jeweils in den entsprechenden Grafiken verwendet werden (siehe Parameter name_day bis name_over). Es wird hier festgelegt wie die Werte in den Grafiken erscheinen (z.B. Stärke der Linien) sowie die verwendete Farbe und der entsprechende Name. Bei der Definition der Farben ist zu beachten, das hier immer drei Werte angegeben werden (RGB Farbschema).
solar_sum_week solar_sum_month solar_sum_year	Falls die Anlage einen Systa Solar Regler verwendet, können in den jeweiligen Grafiken die Gesamtwerte der bisher erzeugten Leistung (in Khw) angezeigt werden. Diese Daten werden zwar nicht vom Systainterface geliefert, aber können aus den Daten in der Datenbank ermittelt werden. Verwendet werden können die Werte jeweils in den Parametern name_week , name_month und name_year .
solar_date_max	In der Infografik kann zusätzlich der bisher höchste Solar Tagesgewinn angezeigt werden. Falls gewollt, muss wie gehabt dieser Name beim Parameter name_over angegeben werden.
db_info	Auch kann in der Infografik die Anzahl der bisher abgelegten Datenbank Records angezeigt werden.
days months	Tages und Monatsnamen wie sie in den Grafiken angezeigt werden. So kann auch auf eine andere Sprache ausgewichen werden.
day_x mon_x c_back t_back	Es folgen die Namen für die X Achse der Grafiken. Welche Farbe der Hintergrund der Grafiken hat (RGB Angaben) und ob dieser transparent ist (geht nur im PNG Ausgabeformat), wird hier festgelegt.
c_matrix c_dash c_head c_date c_foot c_sign	Die Farben für die Tages, Wochen, Monats und Jahresübersicht wird hier im RGB Format angegeben.
o_header o_type_back o_type_text o_sensor_text	Für die Infografik werden hier separate Farben angegeben.

y_scale_temp	
y_scale_perc	
y_scale_power	
y_scale_power_y	
y_scale_power_sum_week	
y_scale_power_sum_month	
y_scale_power_sum_year	Die Y Achse der Grafiken zeigt jeweils die Werte für Temperatur, Prozent und Leistung an. Wo die entsprechende Skala beginnt, welchen Schritt sie hat und wie viel dieser Schritte benutzt werden wird hier eingestellt. Die Angaben sollten natürlich auf die zu erwartenden minimalen und maximalen Werte abgestimmt werden, da sonst die Kurven in den Grafiken abgeschnitten werden.
dash_hor	
dash_ver	In den Grafiken für die Tages, Wochen, Monats und Jahresübersicht kann eine horizontale und vertikale 'Dash' Linie angezeigt werden. Hier wird mittels Werten von 1,2 oder 0 angegeben wie diese aussieht (durchgezogen oder gestrichelt).
maxmin_1(2)	
c_min_1(2)	
c_max_1(2)	
mark_min_1(2)	
mark_max_1(2)	Es ist möglich sich für Temperaturwerte in den Grafiken einen Min / Max Wert anzeigen zu lassen. Hier werden die Angaben dazu gemacht. Es ist zu beachten das der zu verwendete Wert beim Parameter maxmin_1(2) angegeben wird und das bis zu zwei Min / Max Werte in den Grafiken gezeigt werden können.
step_day	
step_week	
step_month	
step_year	Ein wichtiger Parameter was das Aussehen der Temperaturwerte in den Grafiken anbelangt wird hier angegeben. Da zur Erzeugung einer Grafik eventuell viele Daten aus der Datenbank verwendet werden, ergibt sich bei der Verwendung aller dieser Werte unter Umständen eine sehr schwankende (unruhige) Darstellung der Kurven. Um dies zu steuern, kann für jede Grafik separat angegeben werden, in welchen Abständen der Werte ein Mittelwert gebildet wird. Dies führt zu einer weniger schwankenden Anzeige. Was hier eingestellt wird hängt auch sehr davon ab, wie oft die Daten vom Systerinterface abgeholt werden. Eine Angabe von 1 bedeutet, dass jeder Datensatz verwendet wird. Eine Angabe von z.B. 10 heisst, dass jeweils 10 Datensätze gesammelt werden und dann der Mittelwert in die Grafik einfließt. Für die Darstellung der Grafiken ist dies also ein bedeutender Parameter der individuell ermittelt werden sollte.
dbserver	
dbname	
dbuser	
dbpass	Wie auch beim Datenlogger, werden hier die Angaben zur Verbindung mit der MySQL Datenbank gemacht.

serveraddress	<p>Für die IPC Kommunikation mit dem Datenlogger <i>siload</i>er kann hier der Hostname oder die IP Adresse des Rechners auf dem dieser läuft angegeben werden.</p> <p>Der Vorteil einer IPC Kommunikation liegt darin, dass für die Infografik die letzten vom Datenlogger abgeholten Werte vom Systainterface in die Grafik einfließen.</p> <p>Wird dies nicht benötigt (auskommentieren mittels #) so werden für die Infografik die letzten zur Verfügung gestellten Daten aus der Datenbank verwendet. Diese Daten können ja unter Umständen bereits einige Minuten alt sein, da der Datenlogger einen Datencache besitzt und nur zyklisch diesen Cache auf die Datenbank schreibt.</p>
name_status	Der Name der Grafikdatei, falls eine Grafik mit dem Server Status erzeugt wird.
logo_name_status	Dateiname und Position (X,Y Koordinaten) für das Logo auf der Server Status Grafik.
c_status_matrix	
c_status_text	Hintergrundfarbe sowie Textfarbe (RGB Angaben) für die Server Status Grafik (Body Teil)

Die Parameter zur Erstellung von dynamischen Webseiten werden im Kapitel 10.3 Erzeugen von dynamischen Webseiten besprochen.

10.2. Programm Parameter 'sigrafic'

Sind alle entsprechenden Parameter angepasst worden, kann das Grafikprogramm benutzt werden. Wird das Programm ohne Parameter gestartet, werden sofort Grafiken der Art Tages, Wochen, Monats und Jahresübersicht sowie die Infografik erstellt.

Zum anzeigen einer entsprechende Hilfe, wird *sigrafic* mit dem Parameter *-?* gestartet

```
-----
SystaInterface (www.emsystech.de) Grafic Maker
sigrafic v0.1, Copyright (C) 2010 Wolfgang Bolien (wbolien@bolien.de)
-----

Usage :
sigrafic -d [dd] [mm] [yyyy] Create a day graphic for a given date
sigrafic -w [ww] [yyyy]      Create a week graphic for a given week number
sigrafic -m [mm] [yyyy]      Create a month graphic for a given month
sigrafic -y [yyyy]           Create a year graphic for a given year
sigrafic -o                  Create a overview graphic
sigrafic -s                  Create a daemon status graphic
additionally command line switches :
-c config                    Name of the config file
-h                            Create HTML Code
-v                            Verbose mode
-?                            Display this help
Without parameters (-d -w -m -y -o -s) a day, week, month, year
and overview graphic for the actual date was made.
```

Die Programm Parameter bedeuten im Einzelnen:

- d [dd][mm][yyyy]** Mit diesem Parameter wird das Programm veranlasst eine Tagesgrafik zu erstellen. Die Zusatzparameter [dd], [mm] und [yyyy] können dabei benutzt werden um den entsprechenden Tag der zu erstellen ist zu definieren. Ohne Angabe der Zusatzparameter wird eine Grafik für den aktuellen Tag erstellt. Mit der Angabe von [dd] wird eine Grafik für den angegebenen Tag sowie dem aktuellen Monat und Jahr erstellt. Es ist so also flexibel möglich bis zu 3 Angaben zum Datum zu machen, wobei die weggelassenen Zusatzparameter mit dem aktuellen Datum ergänzt werden.
- w [ww][yyyy]** Eine Wochengrafik wird erstellt. Hier ist die Angabe der Woche und des Jahres möglich. Wird beides weggelassen wird die Grafik für die aktuelle Kalenderwoche erstellt.
- m [mm][yyyy]** Hier wird eine Monatsgrafik erstellt. Auch hier ist die Angabe des Monats und des Jahres möglich. Wird beides weggelassen wird die Grafik für den aktuellen Monat erstellt.
- y [yyyy]** Eine Jahresgrafik wird erstellt. Entweder für das aktuelle Jahr, oder bei der Angabe von [jjjj] für diese Jahr.
- o** Es wird eine Infografik erstellt. Bei aktiver IPC Kommunikation werden die neusten Daten beim Datenlogger *siloader* abgeholt, anstatt aus der Datenbank. Falls keine Verbindung zum Datenlogger zustande kommt, werden die zuletzt abgelegten Daten aus der Datenbank benutzt.
- s** Eine Grafik die den Status des Server Daemons anzeigt wird erzeugt. Nur bei aktivierter IPC Kommunikation kann dieser Modus benutzt werden.

Zu diesen Parametern können noch die folgenden angegeben werden:

- c Falls die Konfiguration aus einer anderen Datei gelesen werden soll, kann hier der Name der zu verwendeten Konfigurationsdatei angegeben werden (z.B. -c /home/sigrafi/sigrafi.conf). Ohne Angabe wird versucht aus */etc/sigrafi.conf* zu lesen.
- h Eine dynamische HTML Seite wird erzeugt. Genauer dazu im Kapitel 10.3 Erzeugen von dynamischen Webseiten
- v Keine Anzeige von Statusmeldungen in der Kommando (Shell) Umgebung.

Alle erzeugten Grafiken werden im entsprechenden Ordner abgelegt der in der Konfigurationsdatei angegeben wurde. Falls kein Datum an den Dateinamen angehängt wird (ebenfalls in der Konfiguration anzugeben) werden die Grafiken jeweils überschrieben.

Auch hier gilt das bereits gesagte wie beim Datenlogger *siload*. Falls Fehler in der Konfigurationsdatei gefunden werden, wird eine entsprechende Fehlermeldung gezeigt und das Programm abgebrochen.

10.3. Erzeugen von dynamischen Webseiten

Als besondere Variante bietet das Grafikprogramm die Möglichkeit eine Dynamische Webseite zu erstellen.

Es handelt sich hierbei um eine flexible Navigation durch die verschiedenen Grafiken (Tages, Wochen, Monats und Jahresübersicht sowie die Infografik) die als HTML Seite angezeigt werden. Diese Seite kann direkt im Webbrowser angezeigt werden und hat wiederum Links die eine Navigation zu anderen Daten und Grafiken erlauben.

Um diesen Modus zu nutzen sollte ein eigener Webserver vorhanden sein (z.B. Apache mit CGI Support) und das Script **sigrafic.cgi** das sich im Unterverzeichnis **sigrafic/cgi** befindet genutzt werden.

Unter Debian Linux 5.0 (Etch) muss **sigrafic.cgi** nach **/usr/lib/cgi-bin** kopiert und angepasst werden. Insbesondere der Eintrag **\$sigrafic_prog** muss auf das **sigrafic** Programm verweisen.

Zusätzlich sollte im Verzeichnis **/var/www** ein Verzeichnis zum Speichern der dynamischen Webseiten erstellt werden (z.B. **sigrafic**)

Soll auch Javascript zur Anzeige von Rollover Buttons genutzt werden, so sollte das Unterverzeichnis **sigrafic/si_buttons** samt Inhalt ebenfalls nach **/var/www** kopiert werden.

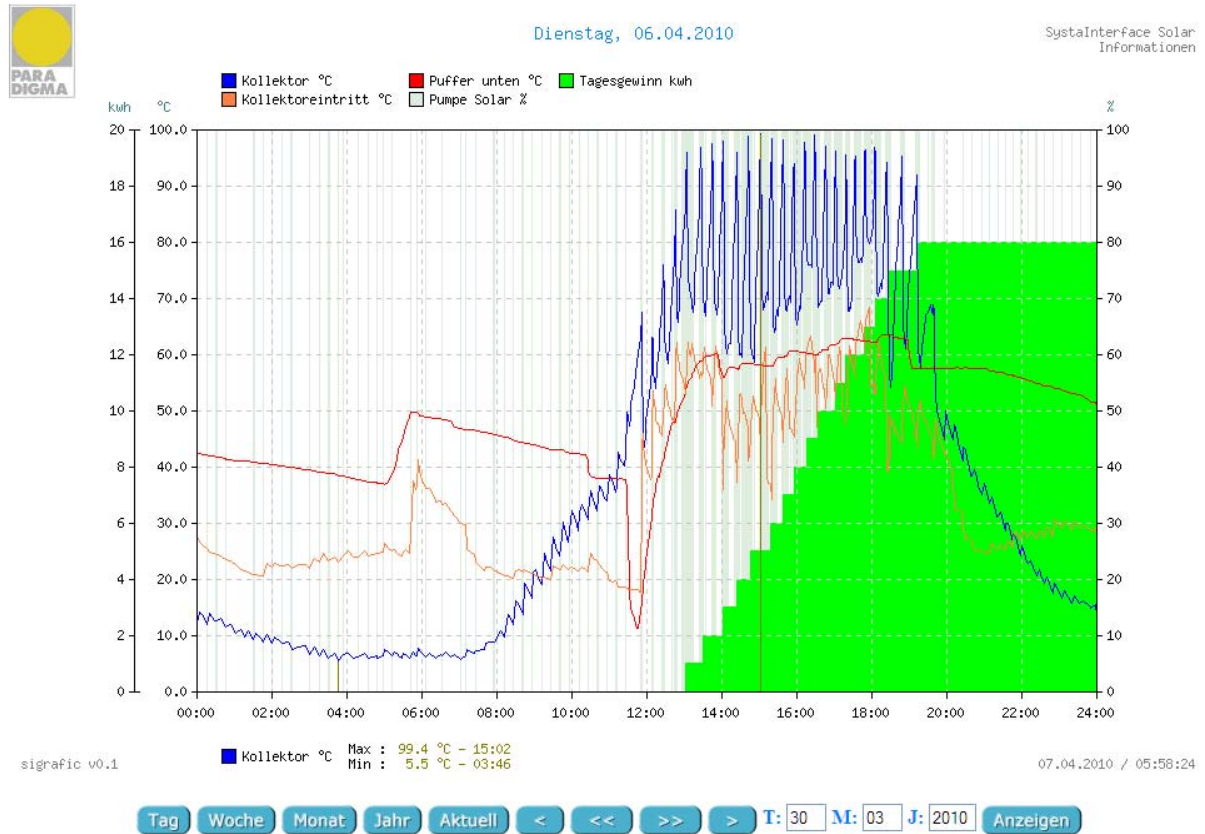
Es sind nun also die beiden neuen Verzeichnisse **/var/www/sigrafic/** und **/var/www/si_buttons/** vorhanden. Die Linux Berechtigungen dieser Verzeichnisse muss den eigenen Gegebenheiten angepasst werden.

Danach sollte die Konfigurationsdatei **/etc/sigrafic.conf** angepasst werden und diese neuen Verzeichnisse eingetragen werden. Die Parameter im Einzelnen sind:

html_path_abs	
html_path_rel	Hier wird der Pfad angegeben wo <i>sigrafic</i> die erzeugten Grafiken abspeichert. Es muss der absolute Pfad (für <i>sigrafic</i>) sowie der relative Pfad (für den Webserver) angegeben werden.
html_graf	Sollen PNG oder JPG Grafiken erzeugt werden, muss hier angegeben werden.
cgi_prog	Der relative Pfad auf das CGI Programm
html_title	Welcher Titel im Webbrowser angezeigt wird, kann hier eingestellt werden.
html_c_back	
html_c_text	Die Farben für den Hintergrund und den Text der Webseite werden hier festgelegt.
html_fsize	
html_bold	Falls keine Javascript Navigation verwendet wird, wird hier die Fontgrösse und Verwendung der Schriftart Fett für die Navigation angegeben.
html_back	Falls gewünscht kann hier auf eine Hintergrundgrafik verwiesen werden, die hinter die Grafiken unseres Grafikprogramms gelegt werden.
html_path_buttons	Wird Javascript zur Navigation verwendet, muss hier der relative Pfad zu den verwendeten Rollover Buttons angegeben werden.
html_buttons	Bei der Navigation ohne Javascript wird hier festgelegt wie die einzelnen Navigationstexte heissen.
html_status_button	Falls gewünscht kann ein Button zum erzeugen der Server Status Grafik angezeigt werden. In diesem Falle ist hier eine 1 zu setzen.
html_align	Die Grafiken können entsprechend ausgerichtet werden.

Bei erfolgreicher Installation kann die dynamische Webseite dann im Webbrowser mittels der URL http://www.eigener_server/cgi-bin/sigrafic.cgi angezeigt werden.

Hier ein Beispiel:



Die angezeigten Navigationsbuttons werden mit den entsprechenden Parameter für das CGI Script belegt, so steht eine flexible Navigation zur Verfügung.

Es kann zudem der Tag der Monat oder das Jahr eingegeben werden und mittels

[Anzeigen](#)

diese Grafik erstellt und angezeigt werden.

Zum blättern der Tage, Monate oder Wochen steht die Navigation mit den Buttons

[<](#) [<<](#) [>>](#) [>](#)

zur Verfügung

11. FAQ (häufige Fragen)

Dieses Kapitel wird erweitert, sobald entsprechende Fragen der Benutzer auftreten.