

Alle Rechte bei Thomas Babel, Hallenberg

UVRstat/UVRStatmini

Grundsätzliches/Voraussetzungen

Die beiden Programme UVRStat und UVRStatmini dienen dazu, die Daten der Heizungsregelung *uvr1611* der Firma Technische Alternative (www.ta.co.at) zu erfassen und zu protokollieren.

Die Kommunikation der beiden Programme mit der Regelung setzen einen an die *UVR1611* angeschlossenen und korrekt konfigurierten *bootloader* voraus. Dies bedeutet, der *bootloader* muss per tcp/ip erreichbar sein. (Dazu verweise ich hier auf die Dokumentation der Firma Technische Alternative.)

Ziel der Programme ist es, eine langfristige Datenerfassung aller Parameter der Heizungsregelung zu realisieren. Die Daten aller Sensoren (Ein- und Ausgänge) der Regelung können entweder in einer Firebird SQL-Datenbank gespeichert werden oder als JSON-Format in eine/mehrere Textdateien gespeichert werden. Die weitere Verarbeitung und Analyse der Daten kann dann mit geeigneten Programmen (Tabellenkalkulation oä.) erfolgen.

Sollen die Daten in der Firebird SQL-Datenbank erfasst werden, ist selbstverständlich ein korrekt installierter SQL-Server innerhalb des Netzwerkes erforderlich.

Zusätzlich können die ausgelesenen Daten auch aktuell auf einen Webservice hochgeladen werden, wo sie dann über einen üblichen Webbrowser darstellbar sind.

Die beiden Programme wurden mit *FPC/Lazarus* entwickelt. Es sind Versionen für Windows oder Linux (Debian) erhältlich. *UVRstatmini* läuft auch auf einem *Raspberry PI*.

UVRStatmini

Das Programm *UVRStatmini* läuft unter Windows, Linux (Debian) und auch auf dem *Raspberry Pi* (ebenfalls Linux).

Im Unterschied zu *UVRStat* bietet *UVRStatmini* keine Anbindung an die Firebird SQL-Server.

Bei einem fertig konfigurierten *UVRStatmini* sehen die aktuellen Daten wie folgt aus:

The screenshot shows the 'UVRStat-mini / Version: 2014.12.3.4' application window. It features a 'Livedaten' tab and a 'Aktuelle Daten holen' button. Below the button are four checkboxes: 'Timer aktivieren', 'Daten per JSON übertragen', 'Mit Fortschrittsbalken', and 'Lokales Speichern'. The main area displays 'Live-Daten der uvr um: 2014-12-03 10:36:02' and two tables of data.

Eingang	Wert	Ausgang	Wert
Kollektorfühler	1,4 C	Solarpumpe	Aus
Fühler Combisol	13,9 C	Ladepumpe Combisol	Aus
Puffer 1 Oben	55,9 C	Ventil Combisol 1	Aus
Puffer 1 Mitte 1	46,8 C	Ventil Combisol 2	Aus
Puffer 1 Mitte 2	36,6 C	Brenneranforderung	An
Puffer 1 Unten	36,4 C	Ladepumpe Frischwasserstat...	Aus
Vorlauf HK	41,1 C	Heizkreispumpe	An
GB5	12 W/qm	Mischer auf	Aus
Außenfühler	0,4 C	Mischer zu	Aus
Kesselfühler Öl	67,0 C	Ladepumpe Ölkessel	Aus
Kesselfühler Holz	43,8 C	Ventil Holz/Öl	Aus
Abgasrohr Holz	23,3 C	Ölkessel frei (und2)	An
Puffer 2 unten	35,0 C	Warmwasser fehlt	An
Warmwasser	47,2 C	Solar-Drehzahl	0
Strömungsschal...	Aus	Combisol-Drehzahl	0
Puffer 2 mitte	36,8 C	Frischwasser-Drehzahl	0
		Heizkreis-Drehzahl	30

At the bottom, there is a status bar showing 'Löschtimer aktiv alle 14400 Sekunden' and 'Anzahl in Json-Datenliste: 1'. A JSON snippet is visible: `{ "babel-hallenberg" : [{ "Datum" : "2014-12-03", "Zeit" : "10:36:02", "Sensor 1" : "1,4 C", "Sensor 2" : "13,9 C", "Sensor 3" : "55,9 C", "Sensor 4" : "46,8 C", "Sen:`

Es werden in zwei Grids die aktuellen Daten der Ein- und Ausgänge dargestellt. Zusätzlich können einige Optionen gesetzt werden:

1. Zyklisches Auslesen der UVR1611
2. Versenden der Daten im JSON-Format
3. Anzeige eines Fortschrittsbalken bis zum nächsten automatischen Auslesen der Regelung
4. Lokales Speichern der ermittelten Daten im JSON-Format

Damit die Daten empfangen werden können, sind in der Sektion „Einstellungen“ die korrekten Daten zur Kommunikation mit der UVR1661 einzutragen:

Die „Adresse der UVR“ kann entweder eine IP-Adresse oder auch eine URL sein, unter der die UVR1661 erreichbar ist (über einen dynamic DNS-Dienst des Routers). Damit ist es möglich, auch entfernte Heizungsregelungen zu überwachen und die Daten zu speichern; die Regelung muss lediglich per TCP/IP erreichbar sein.

Sind die entsprechenden Zugangsdaten für den ftp-Zugang zu einem Webspaces eingegeben können die Daten auch dorthin hochgeladen werden. Die aktuellen Daten werden hierbei in eine lokal vorhandene html-Datei eingespielt und diese html-Datei dann per ftp hochgeladen.

Dabei sind dann (in dem gewählten Zeitraum) immer die aktuellen Daten mit dem Browser einzusehen.

Für eine Langzeitüberwachung können mit UVRStatmini die Daten auch noch im JSON-Format an ein extern lauschendes Programm verschickt werden, welches die empfangenen Daten dann weiterverarbeitet.

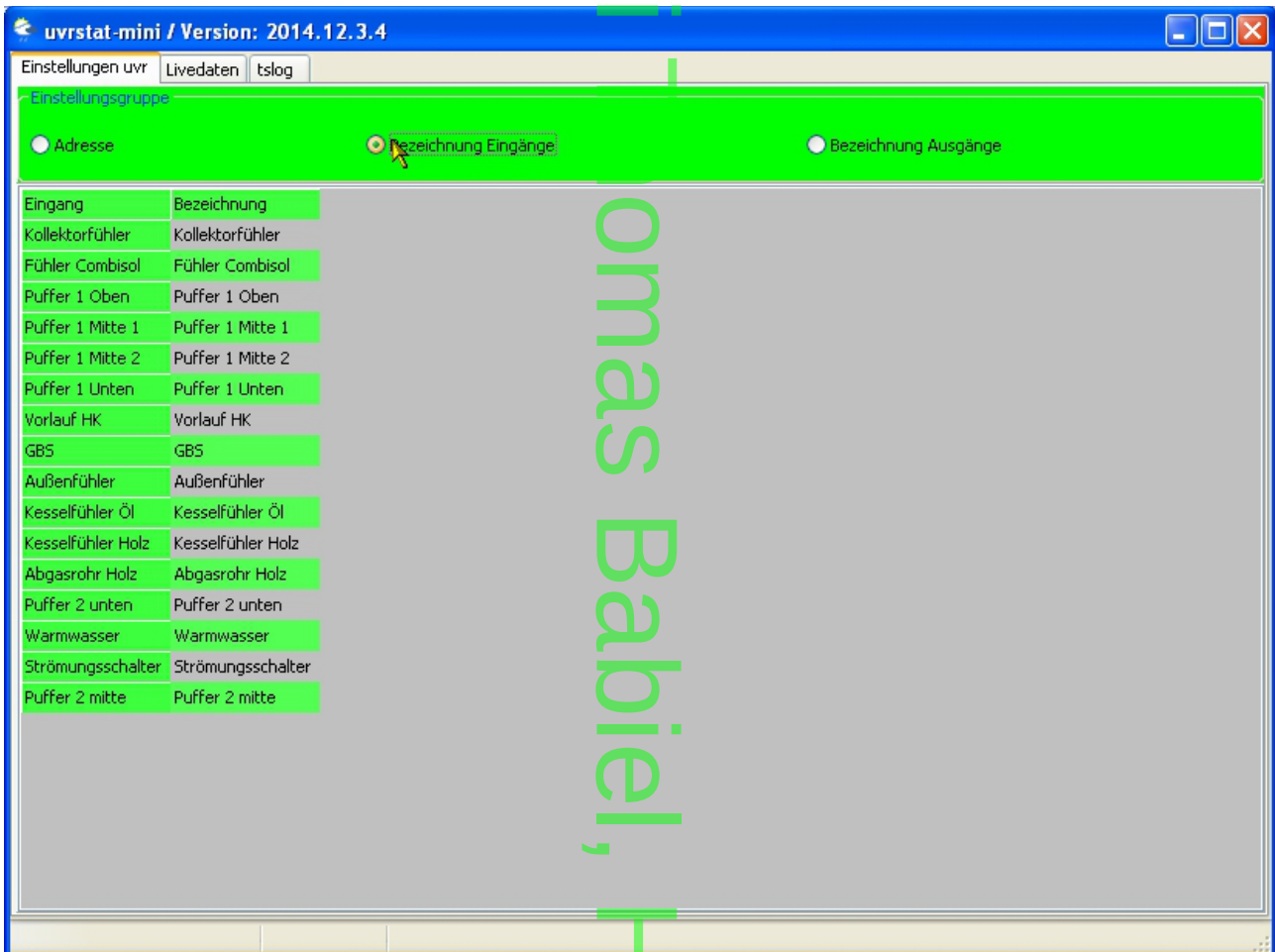
Ein Programm, welches das leistet ist hier auch erhältlich (noch ohne Doku). Auch dieses Programm läuft unter Windows und (Debian)-Linux.

Ein JSON-Datensatz sieht wie folgt aus:

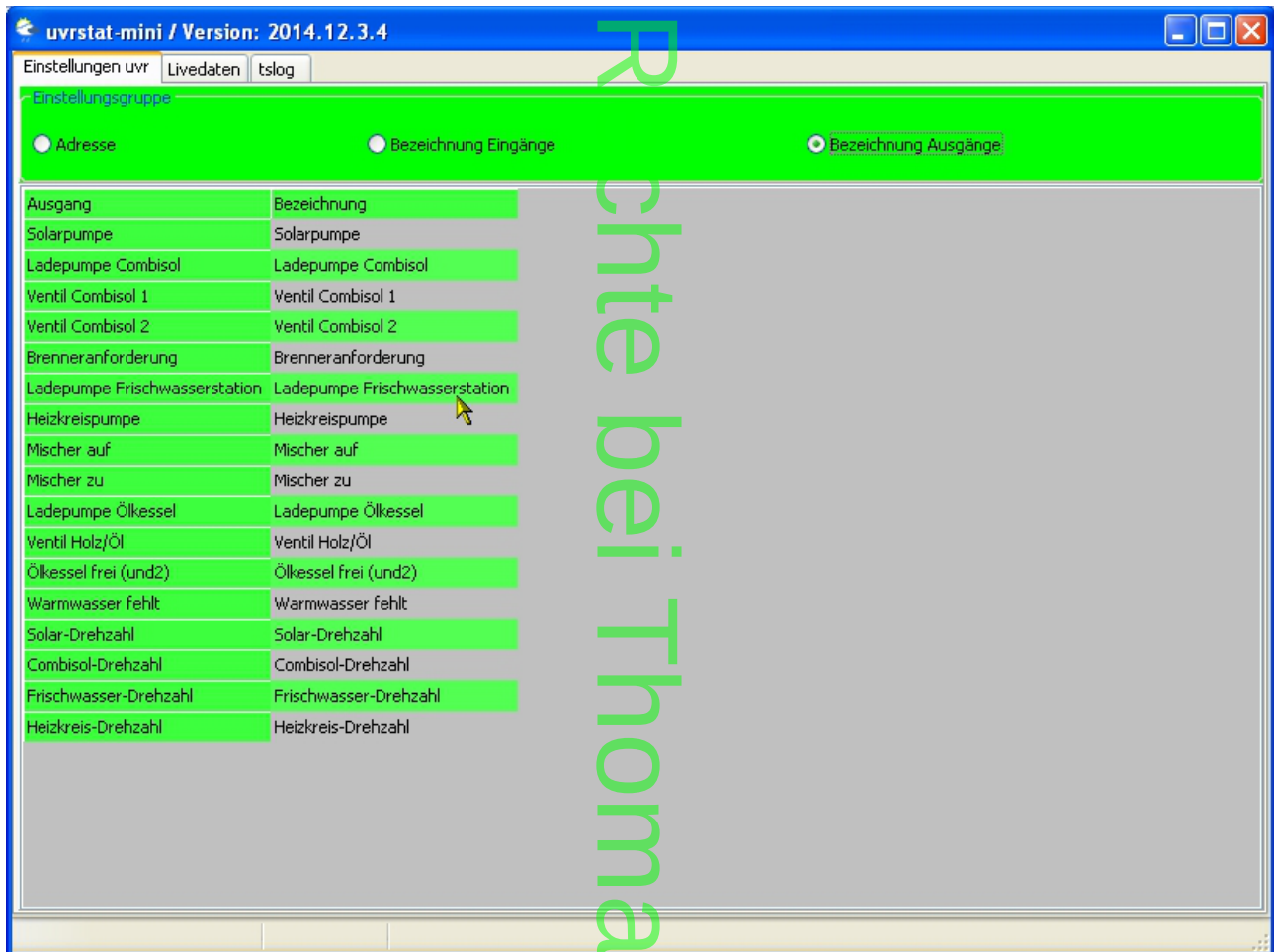
```
{ "Bezeichnung der uvr" : [{ "Datum" : "2014-12-03", "Zeit" : "11:36:36", "Sensor 1" :
"2,8 C", "Sensor 2" : "13,6 C", "Sensor 3" : "54,4 C", "Sensor 4" : "54,3 C", "Sensor 5"
: "47,2 C", "Sensor 6" : "36,6 C", "Sensor 7" : "46,2 C", "Sensor 8" : "28 W\qm",
"Sensor 9" : "0,4 C", "Sensor 10" : "62,4 C", "Sensor 11" : "42,8 C", "Sensor 12" :
"23,7 C", "Sensor 13" : "34,6 C", "Sensor 14" : "45,4 C", "Sensor 15" : "Aus", "Sensor
16" : "36,7 C" }, { "Ausgang 1" : "Aus", "Ausgang 2" : "Aus", "Ausgang 3" : "Aus",
"Ausgang 4" : "Aus", "Ausgang 5" : "Aus", "Ausgang 6" : "Aus", "Ausgang 7" : "An",
"Ausgang 8" : "Aus", "Ausgang 9" : "Aus", "Ausgang 10" : "Aus", "Ausgang 11" : "Aus",
"Ausgang 12" : "An", "Ausgang 13" : "Aus", "Ausgang 14" : "0", "Ausgang 15" : "0",
"Ausgang 16" : "0", "Ausgang 17" : "30" } ] }
```

Ebenso ist es möglich, dass *UVRStatmini* die empfangenen Daten lokal abspeichert. Dies geschieht ebenfalls im JSON-Format.

Damit die Darstellung etwas suggestiver ist, können für die einzelnen Ein- und Ausgänge auch „sprechende Bezeichnungen“ vergeben werden:



Dies geschieht für die Eingänge (s.o.) und auch für die Ausgänge (s.u.).



Die hier festgelegten Bezeichnungen für die Ein- und Ausgänge werden in die Grids der Registerkarte der Live-Daten benutzt.

Die Vorlage der html-Datei zum Hochladen auf dem Webservice kann ebenfalls selbst erstellt und an die eigenen Bedürfnisse angepasst werden.

Die html-Datei kann mit einem beliebigen Editor erzeugt werden. Für die einzelnen Sensoren (Eingänge) werden Platzhalter in der Form `%Sn%` und für die Ausgänge Platzhalter in der Form `%An%` in den html-Quelltext eingeführt.

Vor dem Hochladen auf dem Webservice werden die aktuell ermittelten Werte der Ein- und Ausgänge dann in die html-Datei an den entsprechenden Platzhaltern eingefügt.

Nachfolgend ein Beispiel, das das Prinzip verdeutlichen mag:

C:\lazarus\uvr\p\heizungswerte1.html (Heizungswerte Hallenberg) - Internet Explorer Viewer

Herzlich willkommen auf der Heizungsseite Hallenberg

Kollektortemperatur: %S1%	Solarpumpe: %A1%
Combisol: %S2%	Ladepumpe Combisol: %A2%
Puffer 1 Oben: %S3%	Vent. 1 Combisol: %A3%
Puffer 1 Mitte 1: %S4%	Vent. 2 Combisol: %A4%
Puffer 1 Mitte 2: %S5%	Brenneranforderung: %A5%
Puffer 1 Unten: %S6%	Ldp. Felix: %A6%
Puffer 2 Mitte: %S16%	HK-Dz: %D7%
Puffer 2 Unten: %S13%	HK Pumpe 1: %A7%
Vorlauf HK: %S7%	Mischer auf: %A8%
GBS: %S8%	Mischer zu: %A9%
Aussenfühler: %S9%	Ldp. Ölkessel: %A10%
Kesselfühler Öl: %S10%	Ventil Holz/Öl: %A11%
Kesselfühler Holz: %S11%	Warmwasser fehlt: %A13%
Ölkessel frei: %A12%	Solar-Dz: %D1%
Abgasrohr Holz: %S12%	Combisol-Dz: %D2%
Warmwasser: %S14%	Frischwasserstation-Dz: %D6%
Strömungsschalter: %S15%	Erstellt mit <i>uvrstatmini</i> am: %Datum% um %Uhr%

Die Software *uvrstatmini* ist ein von Thomas Babiak, edv-Beratung entwickeltes Programm (lauffähig für Windows und Linux), welches die Heizungsregelung *uvr1611* der Firma [Technische Alternative](#) ausliest, die ermittelten Daten archiviert, graphisch aufbereitet und ggf. auch online stellt. *Uvrstatmini* läuft jetzt auf einem Raspberry-Pi. Es gibt auch Versionen für Linux/Debian oder Windows. Die Software wurde entwickelt mit Lazarus/FPC