

1wire-USB-Master und FHEM

Mit dem beliebten und kostenlosen Hausautomatisierungssystem von FHEM.de lassen sich Sensorwerte aus dem 1wire-USB-Master mit wenigen Klicks visualisieren.

Im folgenden Beispiel werden am 1wire-USB-Master zwei 1wire-Sensorbus-Module angeschlossen. Das erste ist vom Typ SB-M, dieses enthält einen Temperatur-Sensor, einen Lichtsensor, einen opt. Sensor für relative Luftfeuchtigkeit und einen Relaisausgang für 230V~. Das zweite Modul ist ein SB-SS-S0 mit einem klappbaren Stromsensor für 230/380V~ und gibt den Stromverbrauch als Zählstand [Wh] zurück, sowie den aktuellen Stromverbrauch [W]. Die Module benötigen +12VDC Versorgungsspannung und es sind lediglich die Klemmen „GND“ und „SB“ mit dem 1wire-USB-Master „DQ“ und „GND“ zu verbinden.



Der 1wire-USB-Master kann an der USB-Buchse eines PCs betrieben werden unter Windows oder Linux, in unserem Beispiel ein Raspberry-Pi B, auf dem wir die Software 1wire-USB-Master und FHEM nach Anleitung installiert haben.

Mit „dmesg | grep usb“ finden wir das USB-Device und wir benötigen die 1wire-IDs der beiden Module. Dazu tippen wir in ein Terminal:

```
sudo killall 1wire-USB
sudo /home/pi/1wire-USB /dev/ttyUSB0 /tmp -l
```

und bekommen unter anderem die beiden IDs zurück:

```
o$?
i$0;o;10FC117001080030;
i$1;o;287FC15A0300006A;
i$S0;0;0;
```

Wir tragen diese in die 1wire-USB-config.txt ein:

```
#Syntax
287FC15A0300006A;sb-m-0;SB-M-Modul
S01;L1;800
S02;L2;1000
10FC117001080030;sb-ss-1;SB-SS-S0-Modul
```

bereits jetzt stehen die Sensorwerte als File zur Verfügung:

```
cat /tmp/sb-m-0v.txt
22.625;45;100;0;0
T[°C];rLf[%];rLi[%];rLq[%];limit/LED
cat /tmp/sb-ss-1v.txt
974;39;1
counter [Wh]; consumption[Wh]; limit/LED
```

aus beiden Files lesen wir die erste Zeile mit der 1wire-USB-rrd-sh ein und hängen diese mit einem Zeitstempel versehen an das Logfile /home/pi/fhem-log.txt. Dazu ist die 1wire-USB-rrd-sh zu ändern:

```
#!/bin/sh
##### nur hier sind die eigenen Daten einzutragen
PROG_PATH="/home/pi" # Pfad in dem die 1wire-USB läuft
DATA_PATH="/tmp" # Pad zu den Daten in einer RAM-Disk tmpfs (/etc/fstab)
SERIAL="/dev/ttyUSB0" # serielles Interface des 1wire-USB-Master-Adapters
# und in der 1wire-USB-config.txt die ID der angeschlossenen 1wire-Sensoren eintragen
# und die 1wire-USB-rrd-sh über den cron als root alle 5 min starten
```





```
#####
##### ist alle 5Minuten zu starten (/etc/crontab)
```

```
cd $PROG_PATH

killall rrdtool

if [ -e 1wire-USB.rrd ]; then
# starte 1wire-USB
aktiv=`ps --no-headers -C 1wire-USB`
if [ "$aktiv" = "" ];then
  ${PROG_PATH}/1wire-USB $SERIAL ${DATA_PATH}/-I &
fi
# read data
sb_m_0=$(cat ${DATA_PATH}/sb-m-0v.txt)
sb_m_0=${sb_m_0%T*}
sb_m_0=$(echo "$sb_m_0" | tr '[:]' ' ')
sb_ss_1=$(cat ${DATA_PATH}/sb-ss-1v.txt)
sb_ss_1=${sb_ss_1%counter*}
sb_ss_1=$(echo "$sb_ss_1" | tr '[:]' ' ')
#echo "update sbm=$sb_m_0, sbss=$sb_ss_1"

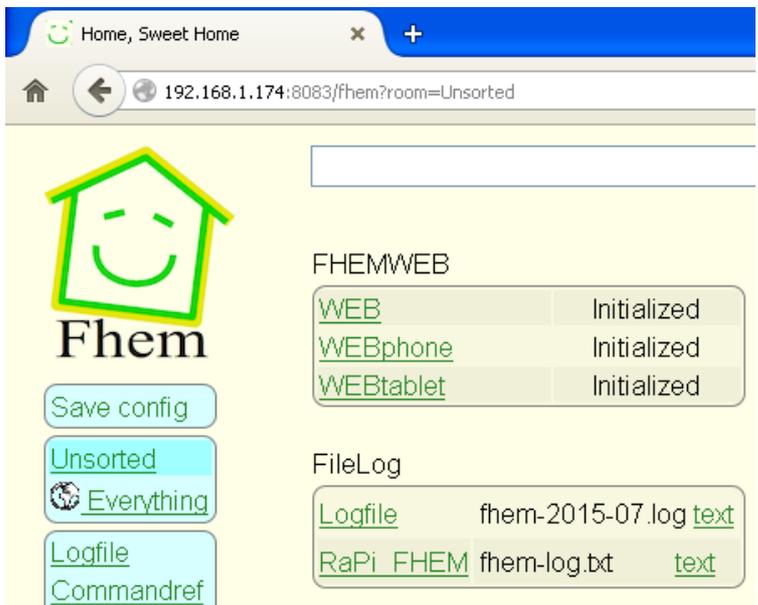
# write Logfile fuer FHEM
dat=$(date +"%Y-%m-%d_%H:%M:%S")
echo "$dat RaPi_FHEM 1wireUSB: $sb_m_0 $sb_ss_1" >> fhem-log.txt

# write data to database
```

Die Logdatei wird nun alle 5Minuten um eine Zeile erweitert:
 2015-07-12_13:45:01 RaPi_FHEM 1wireUSB: 22.625 45 100 0 0 973 39 1
 2015-07-12_13:50:02 RaPi_FHEM 1wireUSB: 22.625 45 100 0 0 976 39 1
 2015-07-12_13:55:01 RaPi_FHEM 1wireUSB: 22.750 45 100 0 0 979 39 1

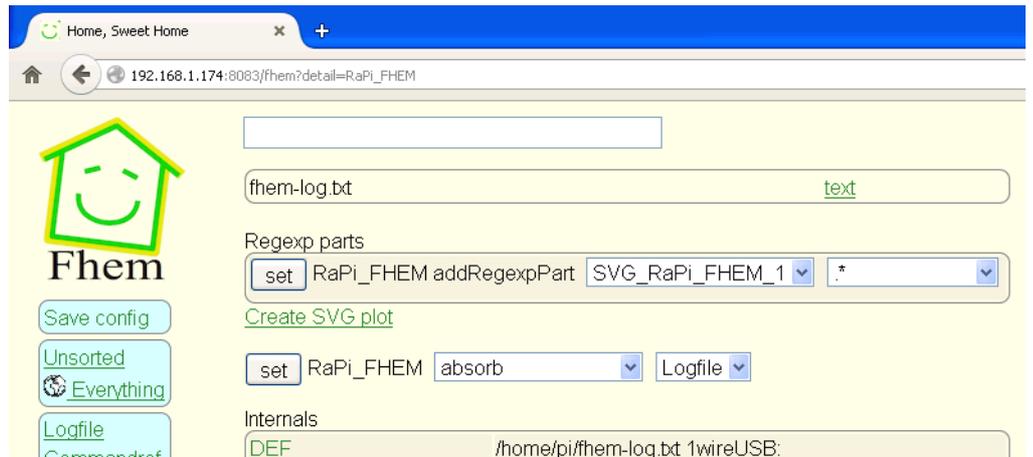
Jetzt öffnen wir mit einem Browser FHEM und mit „Edit files“ editieren wir das „config file fhem.cfg“ und erweitern es um die Zeile:
 define RaPi_FHEM FileLog /home/pi/fhem-log.txt 1wireUSB:

abspeichern mit „Save fhem.cfg“ und mit „unsorted“ wird der 1wire-USB-Master gelistet unter „RaPi_FHEM“. Und „text“ in der Zeile zeigt bereits das Logfile fhem-log.txt.

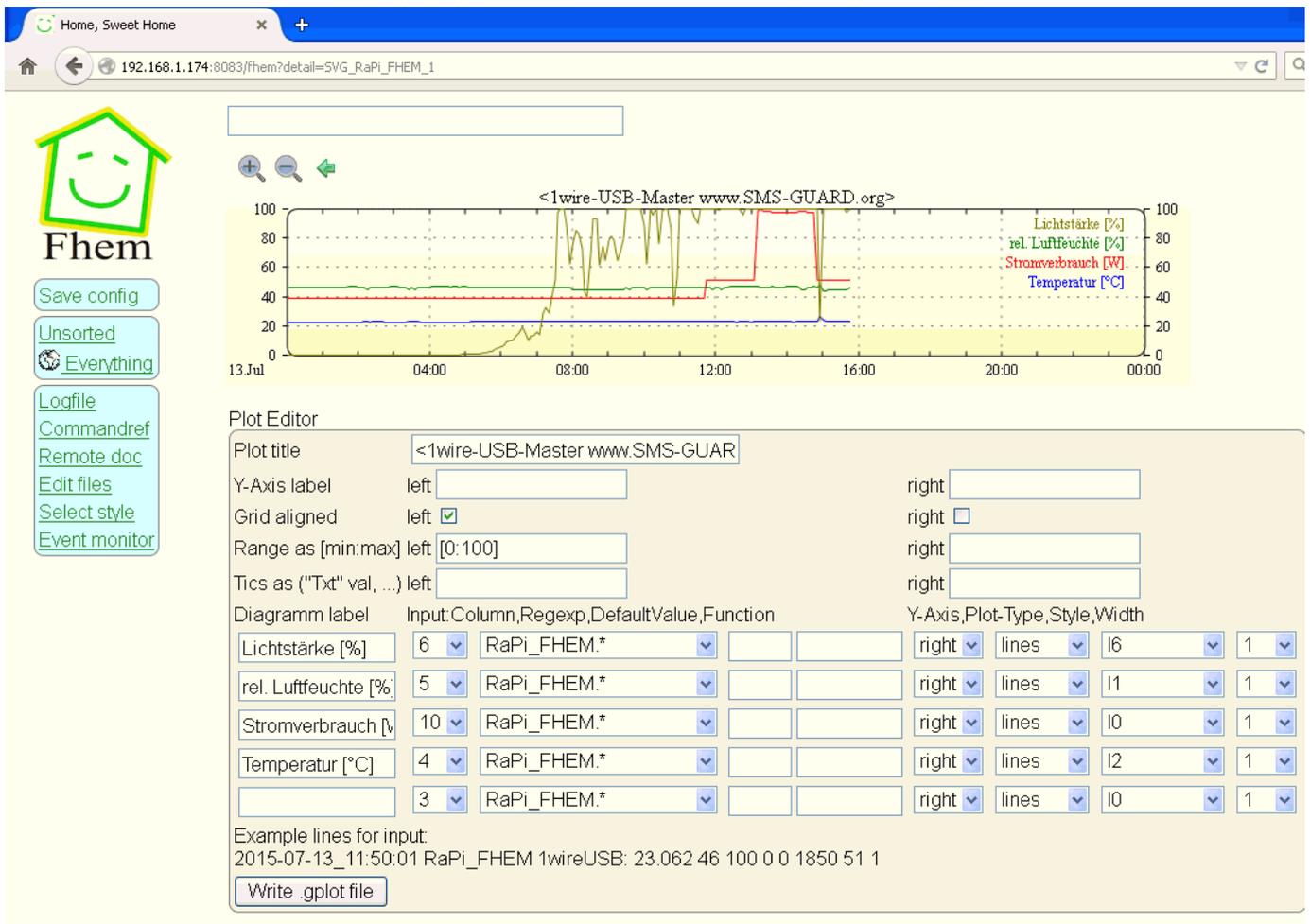




Wir klicken auf „RaPi_FHEM“ und bekommen:



Wir klicken auf „Create SCG plot“ und gelangen zu den Diagramm-Einstellungen:



Mit klicken auf „Write gplot file“ ist auch schon die Arbeit getan. Nun sind die 1wire-USB-Sensoren in FHEM als Diagramm verfügbar.

Eine weitere Möglichkeit wäre das Relais auf dem SB-M-Modul über FHEM anzusteuern. Dazu wird in die fhem.cfg angefügt:

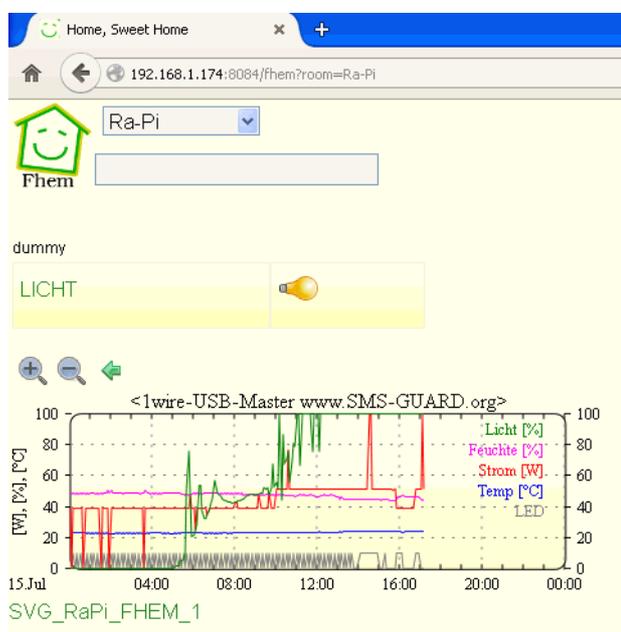


```
define LICHT dummy
attr LICHT room Ra-Pi
attr LICHT setList on off
attr LICHT webCmd on:off
define FileLicht notify LICHT {\
open (DATEI, ">/tmp/sb-m-0o.txt") or die $!;\
print DATEI $EVENT eq "on"?1":0";\
close (DATEI);;\
}
```

will man das Schaltsymbol für den Schalter auf einer Seite mit dem Diagramm sehen wären in der fhem.cfg die Zeilen dafür folgend anzupassen:

```
define RaPi_FHEM FileLog /home/pi/fhem-log.txt 1wireUSB:
define SVG_RaPi_FHEM_1 SVG RaPi_FHEM:SVG_RaPi_FHEM_1:CURRENT
attr SVG_RaPi_FHEM_1 room Ra-Pi
```

und mit „Save fhem.cfg“ sichern und aus dem Menü „Ra-Pi“ wählen, das schaut dann auf der Smartphone-Seite 192.168.1.174:8084 so aus:



Die Licht-details sind rechts dargestellt und in die FileLicht details ist folgendes script einzutragen:



```

LICHT {
open (DATEI, ">/tmp/sb-m-0o.txt") or die $!;
print DATEI $EVENT eq "on"? "1": "0";
close (DATEI);
}
    
```

Home, Sweet Home

192.168.1.174:8084/fhem?detail=FileLicht

FileLicht details

Internals

```

DEF LICHT {
  open (DATEI,
">/tmp/sb-m-0o.txt") or
  die $!;
  print DATEI
  $EVENT eq
  "on"? "1": "0";
  close (DATEI);
}
    
```

NAME	FileLicht
NOTIFYDEV	LICHT
NR	22
NTFY_ORDER	50-FileLicht
REGEXP	LICHT
STATE	2015-07-15 17:01:02
TYPE	notify

attr FileLicht room

Probably associated with LICHT dummy

Select icon
Extend devStateIcon
Device specific help

Mit klicken auf die Glühbirne wird jetzt das Relais im SB-Modul geschaltet.

Klappt das nicht, kann man noch die Rechte des Files anpassen mit:

```
sudo chmod 666 /tmp/sb-m-0o.txt
```

Home, Sweet Home

192.168.1.174:8084/fhem?room=Ra-Pi

Ra-Pi

Fhem

dummy

LICHT

<1wire-USB-Master www.SMS-GUARD.org>

SVG_RaPi_FHEM_1

Wir bedanken uns für die freundliche Unterstützung im FHEM-Forum!