

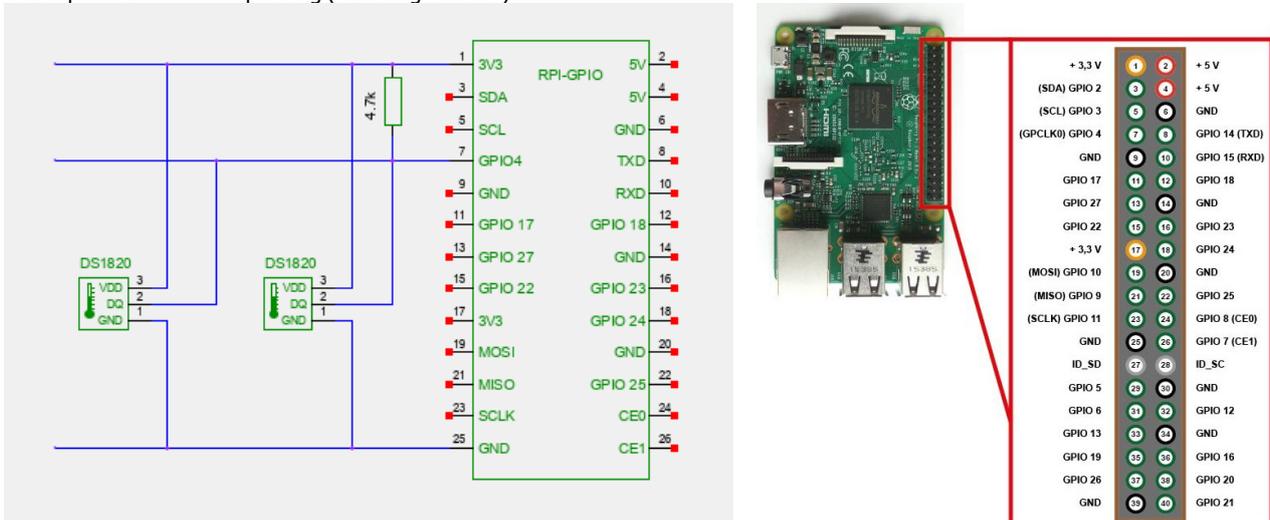
1-wire Temperatursensoren DS18B20 mit Raspberry, Influx und Grafana

Zur Vorbereitung habe ich mich bei untenstehenden Link (Quelle) schlau gemacht.
Da ich mit der Pin-Belegung keine Fehler machen wollte habe ich auch noch dieses Bild gefunden.

Bildquelle: <https://www.elektronik-kompendium.de/sites/raspberry-pi/fotos/raspberry-pi-15b.jpg>

Quelle: <https://www.kompdf.de/weather/pionewiremini.html>

Ich empfehle die aktive Speisung (dreiadriges Kabel)



Anschluss des DS18B20 an den GPIO Port des Raspberry Pi mit aktiver Speisung (dreiadriges Kabel)

Den Datenanschluss DQ des DS18B20 verbindet man auch hier direkt mit dem Anschluss GPIO4 der GPIO-Schnittstelle des Raspberry Pi. GND liegt am Masseanschluss GND, wogegen V_{DD} mit dem 3,3 Volt Anschluss 3V3 verbunden ist. Der Widerstand zwischen 3V3 und GPIO4 dient jetzt lediglich als Pullup-Widerstand zur Einstellung eines definierten Potentials auf der Datenleitung.

Meine DS18B20 Sensoren haben bereits 3 adriges Kabel und die Sensoren sind in einer Edelstahlhülse verpresst. Dadurch sind diese auch für einen produktiven Einsatz vorbereitet.

Ich habe die Kabelummantelung entfernt und 5 Sensoren parallel zusammengefügt. D.h. jeweils die 5 gelben, die 5 roten und 5 schwarzen zusammen verpresst.

Die roten sind normalerweise + , die schwarzen – und die gelben sind die Datenkabel.

Wir nutzen die linke Pin-Leiste (ungerade)

Dann wird an den Pi ein rotes(+) Jumperkabel auf Pin 1 gesteckt, ein schwarzes(-) an Pin 39

Ein gelbes Kabel kommt an Pin 7 (gpio=4)

Zwischen Pin 1 und Pin 7(gpio_4) sollte zusätzlich ein 4,7Ohm-Widerstand eingebaut werden.

Damit ist die Verkabelung bereits perfekt, und man kann alles anschließen.

Jetzt habe ich die 3 Kabel aus dem Pi herausgeführt und mit den 3 Kabel des Sensorenpackchens verbunden.

Möchte man die Anzahl erweitern, so schliesst man einfach nur weitere DS18B20-Sensoren parallel dazu.

Dies muss man dann im shell-script bekannt machen.

Hinweis für später:

Festlegen der Sensoradressen, beliebige Anzahl möglich, da array

```
declare -a SENSORIDENTIFICATOR
```

```
SENSORIDENTIFICATOR[0]="28-0215034ea8ff"
```

```
SENSORIDENTIFICATOR[1]="28-02150350f1ff"
```

```
SENSORIDENTIFICATOR[2]="28-02150351b0ff"
```

```
SENSORIDENTIFICATOR[3]="28-0215037508ff"
```

```
SENSORIDENTIFICATOR[4]="28-031504bc05ff"
```

```
#SENSORIDENTIFICATOR[5]="id des 6. Sensor" Hier braucht man nur erweitern
```

Das weitere script bleibt unverändert.

Nun geht es weiter mit der Vorbereitung des Pi und Bereitstellung der Software.

1-wire Softwaretreiber

Die für die Ansteuerung des 1-wire Temperatursensors erforderlichen Kernelmodule sind Bestandteil der Linux-Distribution [Raspberry Pi OS](#).

Zunächst ist die Aktivierung des Device Tree Overlay für 1-wire notwendig. Dazu editiert man per

```
sudo nano /boot/config.txt
```

und fügt folgende Zeile hinzu:

```
dtoverlay=w1-gpio,gpiopin=4
```

mit STRG+X speichern, mit „J“ bestätigen und schließen

Nach dem erforderlichen Reboot vergewissert man sich mittels des Kommandos `lsmod`, ob die erforderlichen Kernelmodule `w1_gpio` und `w1_therm` geladen sind:

```
lsmod | grep w1
```

die Ausgabe sollte in etwa so aussehen

```
w1_therm      20480  0
w1_gpio       16384  0
wire          36864  2 w1_gpio,w1_therm
mit der Abfrage
```

```
ls /sys/bus/w1/devices
```

erhält man beispielsweise die Ausgabe der Sensoradressen. Bei mir sieht diese so aus.

```
28-0215034ea8ff 28-02150350f1ff 28-02150351b0ff 28-0215037508ff 28-031504bc05ff w1_bus_master1
```

Ich benötige diese Adressen, um sie in ein bash-script einzupflegen, welches die Sensorwerte in eine Influxtabelle schreibt.

Diese Datenbank muss selbstverständlich erst in Influx angelegt, und in Grafana bekannt gemacht werden.

Ich habe eine Datenbank „temperaturen“ gewählt

```
pi@solaranzeige:~$ influx
Connected to http://localhost:8086 version 1.8.3
InfluxDB shell version: 1.8.3
> create database temperatures
> show databases
name: databases
name
----
_internal
steuerung
solarreader
sensoren
solaranzeige
temperaturen
> quit
```

In Grafana mit ADD DATASOURCE bekannt machen.

Das shell script nenne ich `sensoren.sh` und erstelle es mit

`sudo nano /solaranzeige/sensoren.sh` in diesem Ordner

```
#!/bin/bash
#Konfiguration
#####
# Geschrieben wurde es durch Stefan Töngi, getestet durch Enrico Engemann
#
# Dieses Teilprogramm ist freie Software. Sie können es unter den Bedingungen
# der GNU General Public License, wie von der Free Software Foundation
# veröffentlicht, weitergeben und/oder modifizieren, entweder gemäß
# Version 3 der Lizenz oder (nach Ihrer Option) jeder späteren Version.
#
# Die Veröffentlichung dieses Teilprogramms erfolgt in der Hoffnung, daß es
# Ihnen von Nutzen sein wird, aber OHNE IRGEND EINE GARANTIE, sogar ohne
# die implizite Garantie der MARKTREIFE oder der VERWENDBARKEIT FÜR EINEN
# BESTIMMTEN ZWECK. Details finden Sie in der GNU General Public License.
#
# Ein original Exemplar der GNU General Public License finden Sie hier:
# http://www.gnu.org/licenses/

INFLUXHOST="localhost"
INFLUXPORT=8086
INFLUXUSER="root"
INFLUXPASS="rootpw"
INFLUXDATABASE="temperaturen"
INFLUXTABLE="sensoren"

# Festlegen der Sensoradressen, beliebige anzahl moeglich, da array
declare -a SENSORIDENTIFICATOR
SENSORIDENTIFICATOR[0]="28-0215034ea8ff"
SENSORIDENTIFICATOR[1]="28-02150350f1ff"
SENSORIDENTIFICATOR[2]="28-02150351b0ff"
SENSORIDENTIFICATOR[3]="28-0215037508ff"
SENSORIDENTIFICATOR[4]="28-031504bc05ff"

# Ab hier keine Aenderung mehr

# Url bilden, die aendert sich in der Schleife ja nicht
url="http://$INFLUXHOST:$INFLUXPORT/write?u=$INFLUXUSER&p=$INFLUXPASS&precision=s&db=$INFLUXDATABASE"

# Aktuelle Zeit auslesen
timestamp=$(date +%s)

# Laufe ueber alle Sensoren
for item in "${SENSORIDENTIFICATOR[@]}";
do :
    echo "lese Wert von Sensor $address";
    value=$( cat /sys/bus/w1/devices/$item/temperature)
# Teile durch 1000
    korrigiert=`expr $value / 1000`
    echo "Wert von Sensor $item lautet $value"
# Erstelle den insert String
    insertdata="$INFLUXTABLE,sensoradress='$item' sensorwert=$korrigiert $timestamp"
    echo "Sende Daten: $insertdata"
# Sende alles per curl an den Influx Server
    curl -i -XPOST "$url" --data-binary "$insertdata"
done
```

mit STRG+X speichern, mit „J“ bestätigen und schließen
danach Rechte vergeben

`sudo chmod 755 /solaranzeige/sensoren.sh`

Mit `sudo crontab -e` öffnet am den cronjob und fügt folgende Zeilen ein

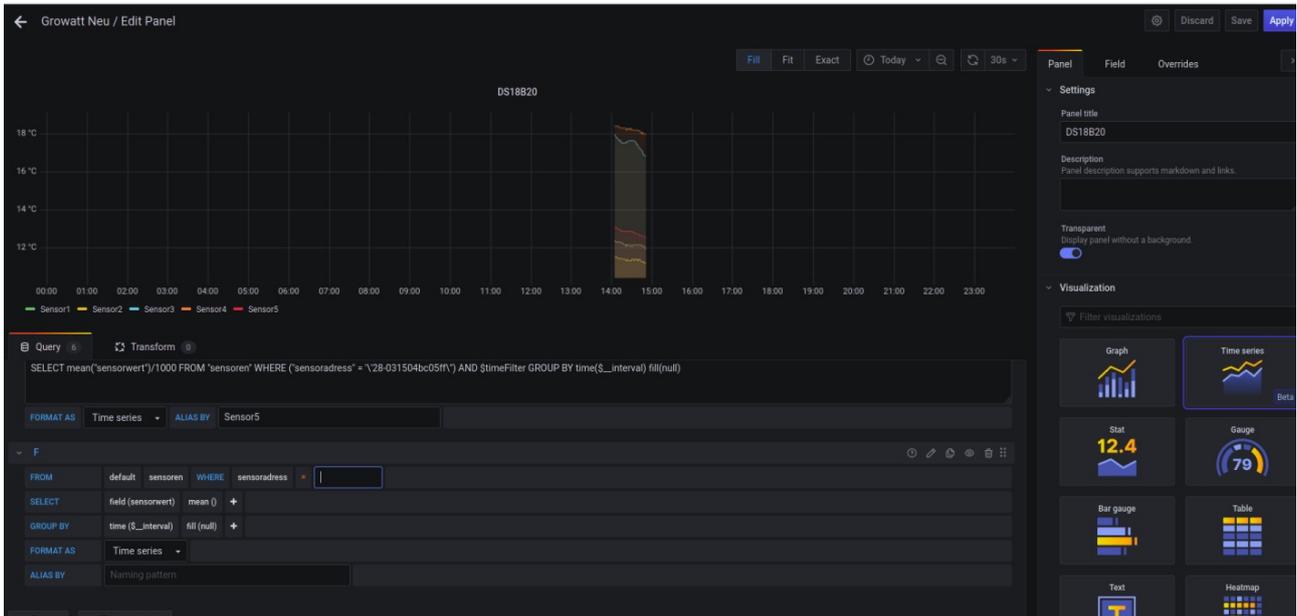
```
*/5 * * * * /solaranzeige/sensoren.sh
```

Dadurch wird im 5 Minutenintervall für den Übertrag in die Datenbank vorbereitet.

Zum Schluss Leerzeile nicht vergessen, da sonst die Änderung nicht wirksam wird.

Danach Speichern und Beenden.

Nun sind die Temperaturwerte im Grafana abrufbereit.



Man wählt seine Datasource, das Measurement sensoren und kann bei WHERE die sensoradress wählen.

Im field gibt der sensorwert sein Geheimnis frei.

Da die DS18B20-Sensoren eine Toleranz von +/- 0,5K besitzen ist es unerheblich, dass ohne Nachkommastelle gespeichert wird. Das Diagramm zeigt dennoch den richtigen Trend.

Herzlichen Dank an Stefan Töngi und viel Spass und Freude beim Nachbau.

Enrico