

# 1. - db\_summen.py -

Was macht das Script?

Mit Hilfe des Scriptes kann man Werte aus verschiedenen Influxdb Datenbanken auslesen, diese "mathematisch" behandeln und das Ergebnis in eine Datenbank (zB. "Summen") zurückschreiben. Daten können dabei von beliebig vielen DB's geholt und verarbeitet werden. Haupteinsatz im Zusammenspiel mit "Solaranzeige" ist das Bilden von Summen der Werte bei Einsatz mehrerer WR zur entsprechenden Weiterverarbeitung/Visualisierung mit "Ulanzi" etc..

Die **Installation** des Tool's ist einfach:

1. falls nicht vorhanden Ordner für Scripte im Homeverzeichnis anlegen:

```
mkdir /home/pi/scripts
```

2. Ordner Tools anlegen

```
mkdir /home/pi/scripts/Tools
```

3. Script herunterladen

```
cd /home/pi/scripts/Tools/  
wget https://raw.githubusercontent.com/Deepintheeast/Ulanzi-Solaranzeige-Connector/main/Tools/db_summen.py
```

4. Abhängigkeiten installieren

⇒ Wie finde ich heraus, welche Debian und SA Version ich habe?

- Solaranzeige: `-uname -a` : 6.1.21
- Debian: `cat /etc/debian_version` : 11.8
- 

⇒ Welche nehme ich, da ich Solaranzeige V6 und Debian 11 habe ???

für Debian 11 Solaranzeige Ver. 5.x

```
sudo pip3 install influxdb
```

für Debian 12 Solaranzeige Ver. 6.x

```
sudo apt install python3-influxdb
```

⇒ Hat hiermit funktioniert

## Konfiguration und Test.

### Datenbank „Summen“ erzeugen:

```
pi@solaranzeige:~ $ influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> create database Summen
> quit
```

Es müssen folgende Einstellungen im Script vorgenommen werden:

Da wir das Script ja über Cron (zeitgleich zum Auslesen der Regler/WR) jede Minute laufen lassen wollen müssen wir eine Verzögerung einbauen die die Ausführung unseres Scriptes solange pausiert bis die Werte der Regler(WR) ausgelesen und gespeichert sind. Das wird hier

```
# Verzögerung in Sekunden bis alle "Regler" ausgelesen sind
time.sleep(25)
```

vorgenommen. (hier 25 Sekunden)

Den genauen Wert bekommt man durch einen Blick ins Log mit

```
tail -f /var/www/log/solaranzeige.log
```

Hier kann man sehr gut erkennen nach wieviel Sekunden das Auslesen der Regler beendet ist, diesen Wert noch 3-5 Sekunden dazugeben und den Wert an Stelle der "25" eintragen.

Jetzt müssen wir nur noch festlegen welche Daten aus welcher Datenbank geholt, wie diese "mathematisch" verknüpft und wohin das "Ergebnis" geschrieben werden soll! Das passiert im Script ab Zeile 71.

```
# Wertepaar 1 holen, berechnen und nach Summen schreiben
result = handler.read_data("solaranzeige", "PV", "PV1_Leistung")
result_1 = round(float(result), 2)
print(result_1)
result = handler.read_data("solaranzeige2", "PV", "PV1_Leistung")
result_2 = round(float(result), 2)
print(result_2)
wert = result_1 + result_2
handler.write_data("Summen", "PV", "PV1_Leistung", wert, time_stamp_db)
print(wert)
```

⇒ "PV", "PV1\_Leistung etc. muss durch die Werte ersetzt werden, die in der Regler Datenbank stehen, bzw. die man gerne haben möchte, um damit zu rechnen. Die Werte sind möglicherweise für jeden Regler unterschiedlich

Hier wird nun in der 2. Zeile der Wert aus Datenbank "solaranzeige", Measurement "PV", Wert "PV1\_Leistung" geholt als "result\_1" noch in eine Gleitkommazahl mit 2 Stellen gewandelt, gespeichert und per "print" auf der Konsole zur Kontrolle angezeigt! In den nächsten 3 Zeilen passiert das selbe mit dem Wert aus Datenbank

"solaranzeige2). Sollen mehr als 2 Werte behandelt werden kann das ganze hier für weitere Werte adäquat erweitert werden! Das auskommentierte Beispiel "Werte 3 holen" am Ende des Scriptes zeigt das für 3. Datenbanken. In den letzten 3 Zeilen werden hier im Beispiel die Werte "addiert", in die Datenbank "Summen", Measurement "PV", Wert "PV1\_Leistung" geschrieben und zur Kontrolle auf der Konsole ausgegeben.

Es können beliebig viele solche Blöcke erstellt und verarbeitet werden!

Bevor man das ganze als Cron Eintrag automatisiert sollte es erst einmal ausgiebig auf der Konsole getestet werden! Dazu wechselt man in das "Tools" Verzeichnis

```
cd /home/pi/scripts/Tools und kann dann das Script wie folgt starten:  
python3 ./db_summen.py
```

(kleiner Tipp, zum testen eventuell die "Verzögerung" durch voransetzen einer #time.sleep(25) auskommentieren ;-))

Wenn man mit der Funktion des Scriptes dann zufrieden ist, das ganze dann durch Erstellen eines "Cron Eintrages" automatisieren. Dazu

```
crontab -e
```

aufrufen und um diese Zeile ergänzen

```
* * * * * python3 /home/pi/scripts/Tools/db_summen.py >/dev/null 2>&1
```

Ab jetzt wird das Script zu jeder vollen Minute gestartet und macht hoffentlich den gewünschten Job!

## 2. - showdb.py -

Da sich immer wieder die Frage stellt welche Daten alle in der InfluxDB stecken und man zB. beim Einrichten des Ulanzi-Connectors auch die Angaben des "Measurements" und der entsprechenden "Werte" benötigt habe ich dazu mal ein kleines Tool in Python erstellt!

Das Script wird einfach mit Angabe des Datenbanknamens aufgerufen!

Die Installation des Tool's ist einfach:

```
# falls nicht vorhanden Ordner für Scripte im Homeverzeichnis anlegen:
mkdir /home/pi/scripts

# Ordner Tools anlegen falls nicht vorhanden
mkdir /home/pi/scripts/Tools

# in Ordner wechseln und Script herunterladen
cd /home/pi/scripts/Tools/
wget https://raw.githubusercontent.com/Deepintheeast/Ulanzi-Solaranzeige-Connector/main/Tools/showdb.py

# benötigte Abhängigkeiten installieren
# für Debian 11 Solaranzeige Ver. 5.x
sudo pip3 install influxdb
# für Debian 12 Solaranzeige Ver. 6.x
sudo apt install python3-influxdb

# Aufruf des Scriptes mit Angabe der auszulesenden Datenbank hier "solaranzeige"
python3 /home/pi/scripts/Tools/showdb.py solaranzeige
```