Raspberry Pi als USV-Server für Home Assistant

Diese Dokumentation beschreibt die Einrichtung eines Raspberry Pi 1 Model B+ (mit Raspberry Pi OS Legacy Lite 32-Bit) als ressourcenschonenden USV-Server für eine **APC Smart-UPS 1500**.

In dieser Konfiguration wurde bewusst **auf den Standard-Dienst NUT verzichtet**, da NUT bei modernen APC Smart-UPS Modellen über USB ohne massiven Aufwand nur rudimentäre Basisdaten ausgibt. Stattdessen kommt der spezialisierte **APC UPS Daemon (apcupsd)** im **Modbus-Modus** zum Einsatz. Dieser schaltet die vollständige Datenvielfalt der professionellen USV (wie Netzspannung, prozentuale Last und Temperaturen) fehlerfrei frei.

1. USV-Hardware vorbereiten

1. Die APC Smart-UPS 1500 per USB-Kabel mit dem Pi verbinden. 2. **Wichtig:** Am Front-Display der USV in das Menü **Configuration** navigieren, den **Menu Type** auf **Advanced** umstellen und anschließend die Option **Modbus** auf **Enable** setzen. Erst dadurch gibt die USV alle Messdaten über USB frei.

2. apcupsd auf dem Raspberry Pi einrichten

Dienst installieren:

```
sudo apt update && sudo apt install apcupsd -y
```

Konfiguration für den USB-Modbus-Betrieb und den Netzwerkzugriff anpassen:

```
sudo nano /etc/apcupsd/apcupsd.conf
```

Folgende Parameter exakt so setzen (den Rest der Datei unberührt lassen):

```
UPSCABLE usb
UPSTYPE modbus
DEVICE

NETSERVER on
NISIP 0.0.0.0
NISPORT 3551
```

(Hinweis: Hinter DEVICE bleibt der Eintrag bei USB-Modbus bewusst leer. `NISIP 0.0.0.0` öffnet den Port für Anfragen aus dem lokalen Netzwerk von Home Assistant).

Dienst aktivieren und starten:

```
sudo systemctl enable apcupsd
sudo systemctl restart apcupsd
```

Erfolgstest auf dem Pi ausführen:

```
apcaccess status
```

Die Ausgabe liefert nun alle Live-Werte der USV (z. B. LINEV, LOADPCT, ITEMP).

Wichtig: Echtzeit-Erkennung bei Stromausfall (POLLTIME)

Standardmäßig ist das Abfrage-Intervall (Polling) in der Konfiguration deaktiviert oder sehr träge eingestellt (60 Sekunden). Um eine sekundenaktuelle Reaktion bei einem Stromausfall in Home Assistant zu garantieren, muss die `POLLTIME` manuell aktiviert und verkürzt werden.

Datei öffnen:

```
sudo nano /etc/apcupsd/apcupsd.conf
```

Nach der Zeile **POLLTIME** suchen (ggf. die Raute # am Zeilenanfang entfernen) und den Wert auf **10** (Sekunden) setzen:

```
#  POLLTIME <rate in seconds>
POLLTIME 20
```

Dienst neu starten, um die Änderung zu übernehmen:

```
sudo systemctl restart apcupsd
```

* **Ergebnis:** Der Pi scannt die USV nun alle 10 Sekunden. Statusänderungen (z. B. der Wechsel von `ONLINE` zu `ON BATT`) werden dadurch innerhalb weniger Sekunden an Home Assistant übertragen und kritische Automationen sofort ausgelöst.

3. Integration in Home Assistant

1. In Home Assistant zu **Einstellungen → Geräte & Dienste** navigieren. 2. Auf **Integration hinzufügen** klicken und nach **APC UPS Daemon (apcupsd)** suchen. 3. Die IP-Adresse des Raspberry Pi (z. B. `192.168.10.248`) und den Port `3551` eintragen und absenden. 4. Das Gerät listet nun über 20 Sensoren direkt auf.

4. Leistungstrick: Echten Watt-Verbrauch berechnen

Da die USV die Last nur in Prozent (`LOADPCT`) ausgibt, die maximale Nennleistung des Modells aber bekannt ist (1000 Watt bei der 1500VA-Version), wird in Home Assistant ein virtueller Sensor für den aktuellen Verbrauch in Watt angelegt.

1. In HA zu **Einstellungen** → **Geräte & Dienste** → Reiter **Helfer** wechseln. 2. Auf **Helfer erstellen** → **Template** → **Template für einen Sensor** klicken. 3. Folgende Daten eintragen:

- **Name:** USV Aktueller Verbrauch Watts
- **Zustandstemplate:**

```
{{ (states('sensor.apcups_last') | float(0) * 1000 / 100) | round(1) }}
```

- (Hinweis: Den Sensornamen `sensor.apcups_last` bei Bedarf an den exakten Entitätsnamen der eigenen Installation anpassen).*
- **Maßeinheit:** W
- **Geräteklasse:** Power
- **Zustandsklasse:** measurement

Der Sensor steht nun für Dashboards und das Home Assistant Energie-Dashboard zur Verfügung.

5. Automatische Selbstheilung (Dienst-Überwachung)

Falls sich der USB-Treiber oder der Dienst aufgrund von hoher Last verschluckt oder einfriert, wurde eine automatische systemd-Überwachung eingerichtet. Diese startet den Dienst im Fehlerfall ohne manuelles Eingreifen nach 10 Sekunden automatisch neu.

Erweiterungskonfiguration öffnen:

```
sudo systemctl edit apcupsd
```

Folgenden Block einfügen und speichern:

```
[Service]
Restart=on-failure
RestartSec=10s
```

Konfigurations-Manager neu laden:

```
sudo systemctl daemon-reload
```

6. Modbus-Freeze-Überwachung (Watchdog-Skript)

Bei APC Smart-UPS Modellen kann es im USB-Modbus-Betrieb nach einigen Stunden dazu kommen, dass die Messwerte einfrieren (obwohl der Zeitstempel aktuell bleibt). Da die Netzspannung (`LINEV`) im Normalbetrieb permanent minimal schwankt, erkennt das folgende Skript einen absoluten Stillstand der Werte als Fehler und startet den Dienst vollautomatisch im Hintergrund neu.

Schritt 1: Überwachungs-Skript erstellen

Datei anlegen:

```
sudo nano /usr/local/bin/check_apcupsd_freeze.sh
```

Folgenden Inhalt einfügen und speichern:

```
#!/bin/bash
CACHE_FILE="/tmp/apcupsd_last_linev"
CURRENT_LINEV=$(/sbin/apcaccess status | grep "LINEV" | awk '{print $3}')

if [ -z "$CURRENT_LINEV" ]; then
    exit 0
fi

if [ -f "$CACHE_FILE" ]; then
    OLD_LINEV=$(cat "$CACHE_FILE")
    if [ "$CURRENT_LINEV" == "$OLD_LINEV" ]; then
        sudo systemctl restart apcupsd
        echo "$(date): apcupsd Modbus-Freeze erkannt. Neustart
durchgeführt." >> /var/log/apcupsd_freeze.log
    fi
fi

echo "$CURRENT_LINEV" > "$CACHE_FILE"
```

Skript ausführbar machen:

```
sudo chmod +x /usr/local/bin/check_apcupsd_freeze.sh
```

Schritt 2: Automatischen Zeitplan einrichten (Cronjob)

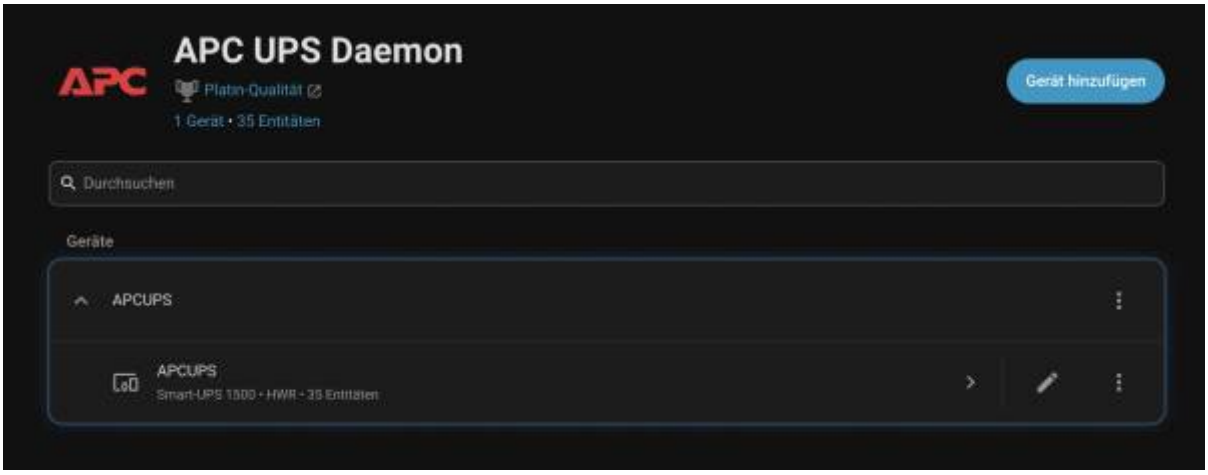
Die System-Zeitschaltuhr als Root öffnen:

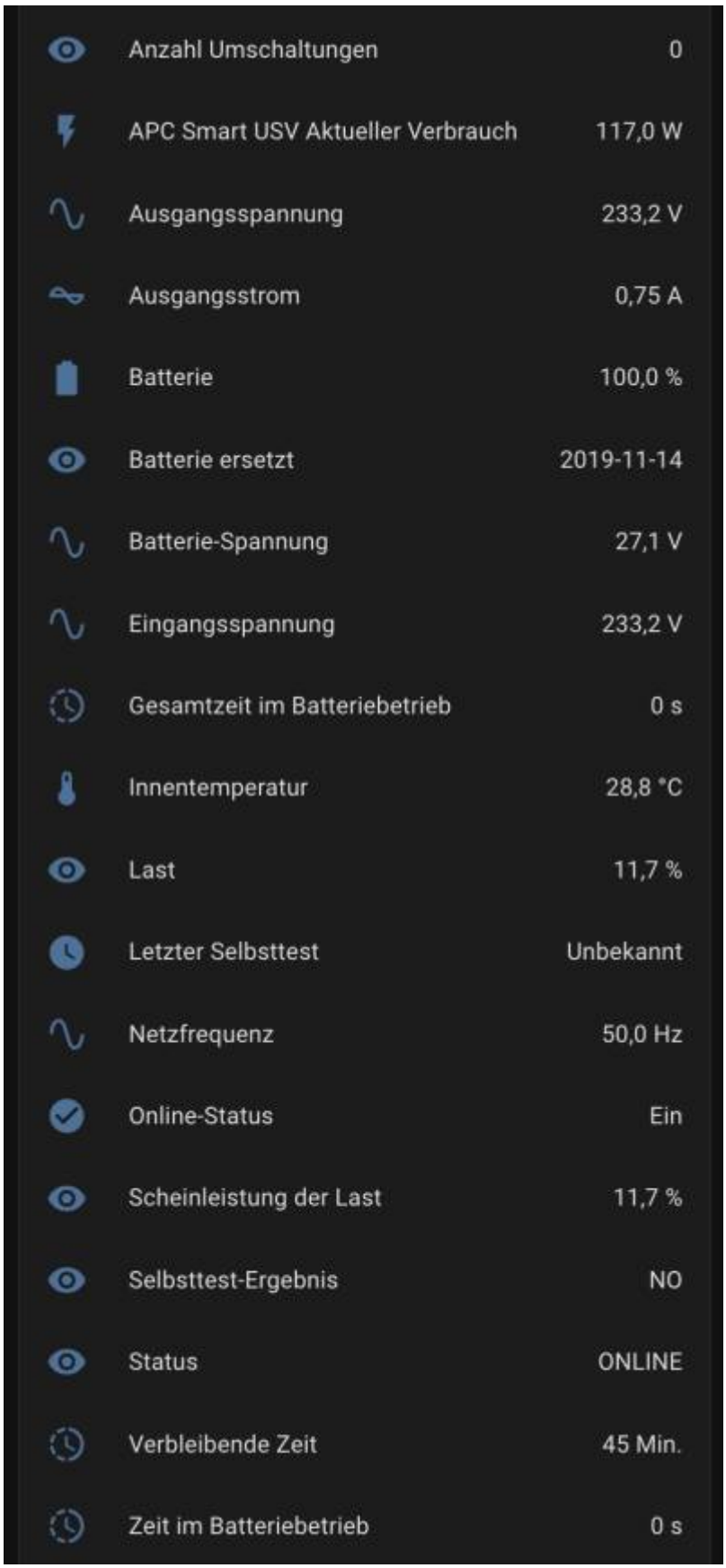
```
sudo crontab -e
```

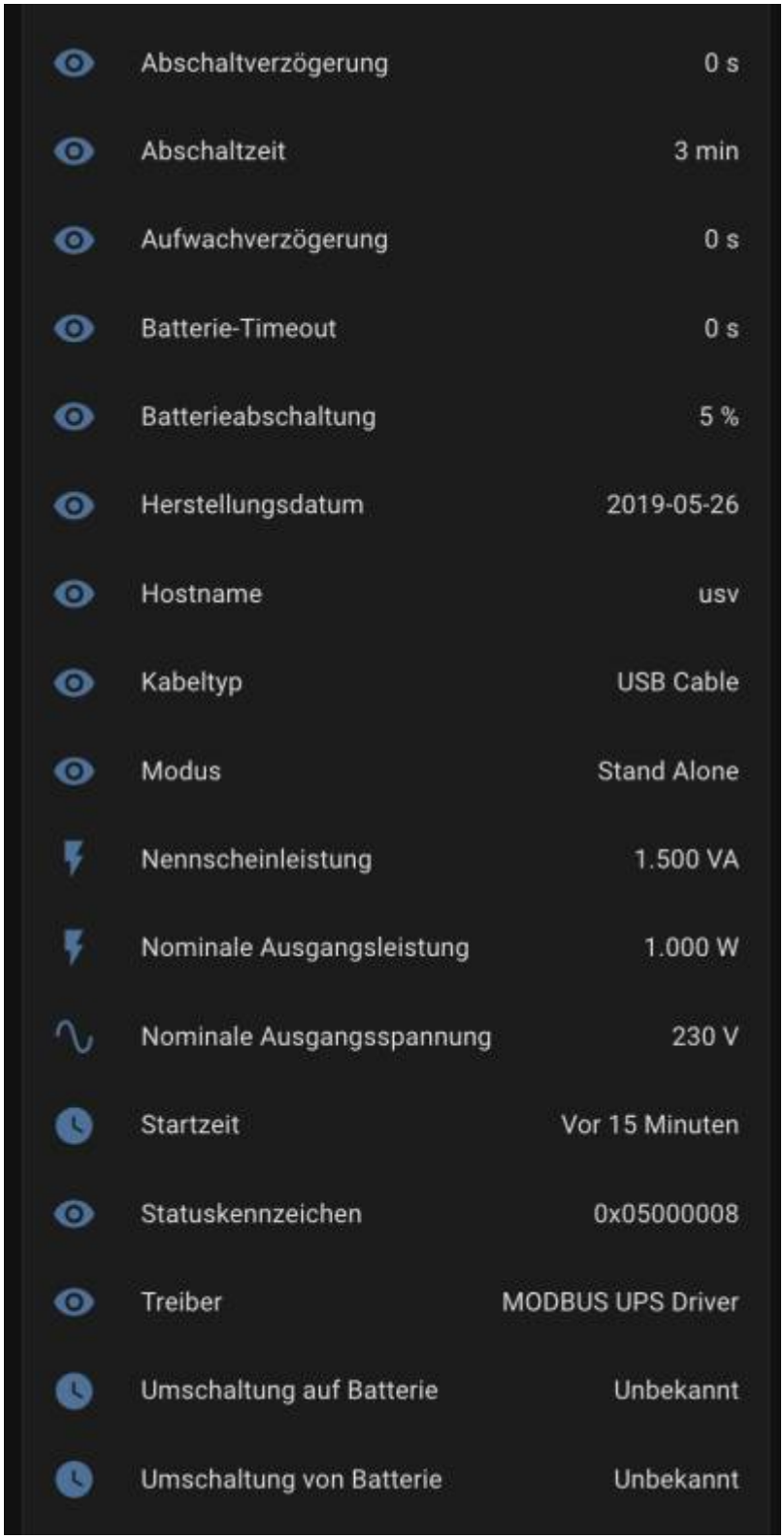
Ganz unten am Ende der Datei folgende Zeile einfügen, damit der Check alle 15 Minuten läuft:

```
*/15 * * * * /usr/local/bin/check_apcupsd_freeze.sh
```

Bilder


APC UPS Daemon
Platin-Qualität
1 Gerät • 35 Entitäten
Gerät hinzufügen
Durchsuchen
Geräte
APCUPS
APCUPS
Smart-UPS 1500 • HWR • 35 Entitäten


Anzahl Umschaltungen 0
APC Smart USV Aktueller Verbrauch 117,0 W
Ausgangsspannung 233,2 V
Ausgangsstrom 0,75 A
Batterie 100,0 %
Batterie ersetzt 2019-11-14
Batterie-Spannung 27,1 V
Eingangsspannung 233,2 V
Gesamtzeit im Batteriebetrieb 0 s
Innentemperatur 28,8 °C
Last 11,7 %
Letzter Selbsttest Unbekannt
Netzfrequenz 50,0 Hz
Online-Status Ein
Scheinleistung der Last 11,7 %
Selbsttest-Ergebnis NO
Status ONLINE
Verbleibende Zeit 45 Min.
Zeit im Batteriebetrieb 0 s



From:
https://wiki.mahlen.biz/ - **Smart-Home Wiki**

Permanent link:
https://wiki.mahlen.biz/doku.php?id=home_assistant:home_assistant_rezepte:apcusv_einbinden&rev=1793946996

Last update: **06.11.2026**