9. Synology NAS als Netzwerk-Client einbinden

Da das DSM-eigene USV-Menü der Synology starr auf das NUT-Protokoll setzt, binden wir die Synology NAS über ein Community-Paket direkt an das native `apcupsd`-Netzwerkprotokoll (Port 3551) des Raspberry Pi an. Dies läuft ohne Docker-Container, übersteht DSM-Updates und schont die Ressourcen.

Schritt 1: SynoCommunity im Paketzentrum freischalten

1. Im DSM das **Paketzentrum** öffnen. 2. Oben rechts auf **Einstellungen** klicken und zum Reiter **Paketquellen** wechseln. 3. Auf **Hinzufügen** klicken und folgende Daten eintragen:

- **Name:** `SynoCommunity`
- **Ort:** `https://synocommunity.com`

4. Mit **OK** bestätigen.

Schritt 2: apcupsd installieren und konfigurieren

1. Im Paketzentrum nach **apcupsd** suchen und installieren. 2. Nach der Installation per SSH auf der Synology NAS einloggen. 3. Die Konfigurationsdatei öffnen:

```
 <code>sudo nano
/var/packages/apcupsd/target/etc/apcupsd/apcupsd.conf</code>
```

4. Die Parameter für den Client-Betrieb anpassen (alte USB-Einträge auskommentieren):

```
UPSCABLE ether
UPSTYPE net
DEVICE 192.168.10.248:3551
POLLTIME 30
BATTERYLEVEL 20
MINUTES 10
```

5. Die Datei speichern (Strg+O, Enter, Strg+X). 6. Im Paketzentrum das Paket **apcupsd** einmal kurz **Stoppen** und wieder **Starten**.

Die Synology fährt nun bei einem Stromausfall vollautomatisch herunter, sobald der Akku unter 20 % sinkt oder weniger als 10 Minuten Restlaufzeit verbleiben.

10. UniFi Dream Machine (UDM) via Home Assistant abschalten

Die UniFi Dream Machine (UDM / UDM Pro / SE) erlaubt aufgrund ihres geschlossenen Betriebssystems

keine dauerhafte Installation von USV-Software. Da Home Assistant den USV-Status in Echtzeit kennt, nutzen wir Home Assistant, um der UDM im Ernstfall einen sicheren Abschaltbefehl per **SSH** zu senden.

Schritt 1: SSH auf der Dream Machine aktivieren

1. Die UniFi-Oberfläche im Browser öffnen. 2. Zu den **Console Settings** (Konsolen-Einstellungen) navigieren. 3. Den Punkt **SSH** aktivieren, einen Benutzernamen (Standard: `root`) und ein sicheres SSH-Passwort vergeben.

Schritt 2: SSH-Befehl in Home Assistant hinterlegen

Damit Home Assistant den Befehl absetzen kann, nutzen wir die `shell_command`-Integration. Dazu muss `sshpass` auf dem Home Assistant System installiert sein (bei HAOS standardmäßig der Fall).

Die `configuration.yaml` von Home Assistant öffnen und folgenden Block hinzufügen:

```
shell_command:
  shutdown_udm: "sshpass -p 'IHR_SSH_PASSWORT' ssh -o
StrictHostKeyChecking=no root@192.168.10.1 'ubnt-systool poweroff'"
```

(Hinweis: Die IP `192.168.10.1` bei Bedarf an die Gateway-IP Ihrer UDM anpassen. Der Befehl `ubnt-systool poweroff` ist zwingend erforderlich, da ein normales `shutdown` die UniFi-Datenbank beschädigen kann).

Home Assistant danach einmal neu starten bzw. die YAML-Konfiguration neu laden.

Schritt 3: Automation erweitern

Erweitern Sie den Block `batterie_leer` der bestehenden iPhone-Kritik-Automation (Kapitel 6) um den Aufruf des neuen Shell-Commands:

```
  # Dieser Block wird unter den Actions der bestehenden Automation
eingefügt:
  - choice:
      - conditions:
          - condition: template
            value_template: "{{ trigger.id == 'batterie_leer' }}"
        sequence:
          # 1. iPhone benachrichtigen
          - service: notify.mobile_app_iphone_17_von_kolja
            data:
              title: "🚨 UDM SHUTDOWN!"
              message: "USV-Akku kritisch niedrig! Die UniFi Dream Machine
wird heruntergefahren."

          # 2. UDM per SSH ausschalten
```

```
          - service: shell_command.shutdown_udm
```

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

Permanent link:
https://wiki.mahlen.biz/doku.php?id=blackout_shutdown_doku_bei_kolja&rev=1793974005

Last update: **06.11.2026**