Autoladen und Hausakku

Was sich mit EVCC 0.316.1 geändert hat · Stand 28.09.2026

> **Der Modus „Min+PV“ existiert nicht mehr.** EVCC kennt nur noch **Aus**, **Smart** und **Schnell**. Was früher Min+PV war, ist heute **Smart mit „Dauerhaft laden“** — in der Oberfläche erkennbar an „Smart ∞“. Technisch heißt das `alwaysCharge`, mit den Werten *aus*, *ein* und *einmalig*. Eine Automatisierung, die noch auf den alten Modusnamen prüfte, lief dadurch monatelang ins Leere.

> **Nicht verwechseln: „Dauerhaft laden“ und „Batterie Boost“ sind zwei verschiedene Dinge.** Das kostet leicht einen Abend. **Dauerhaft laden** (`alwaysCharge`) erzwingt Laden mit Mindeststrom, ohne Rücksicht auf Überschuss — und **ohne jede Untergrenze**. Es ist immer einstellbar, auch ohne Auto. **Batterie Boost** (`batteryBoost`) gibt zusätzlich die volle Akkuleistung frei und hat mit `batteryBoostLimit` einen eigenen Boden. Sein Knopf erscheint nur, wenn ein Fahrzeug angesteckt ist. **Das Limit von 55 % gilt nur für den Boost, nicht für „Dauerhaft laden“.** Genau deshalb braucht es dort die Automatisierung aus Home Assistant.

Die vier Ebenen der Batterienutzung

EVCC entscheidet gestaffelt, wann der Hausakku dem Auto helfen darf. Die Werte stehen in der Oberfläche unter „Batterienutzung“.

`unter 75 %`	**Hausspeicher hat Vorrang.** Überschuss geht zuerst in den Akku, das Auto bekommt nichts. (`prioritySoc`)
`ab 80 %`	**Akku darf das Laden stützen.** Oberhalb dieser Marke darf Akkuenergie das Laden tragen und verzögert das Abschalten im Smart-Betrieb. (`bufferSoc`)
`Boost`	**„Batterie Boost“ greift unabhängig davon.** Er nutzt bewusst die volle Leistung des Hausakkus — und **die Vorrang-Grenze von 75 % gilt für ihn ausdrücklich nicht**. Im Quellcode steht dazu: der Boost entleere die Batterie ja absichtlich. Sein Boden ist `batteryBoostLimit`.
`ohne Boden`	**„Dauerhaft laden“ kennt keine Untergrenze.** Es lädt mit Mindeststrom weiter, bis der Akku leer ist. Diese Lücke schließt die Automatisierung aus Home Assistant — siehe unten.

> **Die wichtigste Falle:** Der Wert **100** bedeutet bei diesem Limit *nicht* „bis 100 %“, sondern **gar kein Boden** — im Quellcode als „disabled“ geführt. Er ist der Auslieferungswert. Solange er so stand, hätte „Dauerhaft laden“ den Hausakku ungebremst leerziehen können. Ein Wert zwischen 1 und 99 ist dagegen ein echter Boden: Wird er unterschritten, hält der Boost an (das Auto bleibt bevorzugt, bis es abgesteckt wird — das verhindert ständiges Hin und Her).

Was jetzt eingestellt ist

EINSTELLUNG	WERT	WIRKUNG
`batteryBoostLimit`	**55 %**	Harter Boden. Wirkt im Ladepunkt selbst, auch wenn Home Assistant steht.
`prioritySoc`	75 %	Hausspeicher-Vorrang (gilt nicht für den Boost)
`bufferSoc`	80 %	ab hier darf der Akku das Laden stützen
`bufferStartSoc`	0	0 = PV-Ladung startet nur mit echtem Überschuss
`batteryDischargeControl`	True	keine Akku-Entladung im Schnell-Modus und bei geplantem Laden
`minCurrent`	13 A	Mindeststrom, wird automatisch gesetzt (siehe unten)
`phasesConfigured`	0	0 = EVCC schaltet die Phasen selbst

Was Home Assistant dazu tut

Mindeststrom nach Betriebsart

Ohne „Dauerhaft laden“ lädt der Ladepunkt nur mit Überschuss — dann soll er fein herunterregeln können, also **6 A**. Mit Boost kommt ein Teil aus dem Hausakku; dann lieber zügig laden mit **13 A**, weil der Eigenverbrauch der Umwandlung bei kleiner Leistung relativ stärker durchschlägt. Das erledigt `automation.evcc_mindeststrom_nach_betriebsart`. Die Phasenzahl bleibt unberührt.

Die Untergrenze folgt der Sonnenprognose

> **Es ist eine Untergrenze, kein Ladeziel.** Die Zahl sagt nicht, wie voll der Hausakku wird — das macht die Sonne. Sie sagt, **wie weit das Auto ihn leerfahren darf**, solange „Dauerhaft laden“ läuft. Unterschreitet der Akku sie, schaltet `automation.evcc_min_pv_zuruck_auf_pv_wenn_hausakku_unter_puffer` das Dauerhaft-Laden ab; der Ladepunkt lädt dann nur noch aus Überschuss weiter. **Das ist der einzige Schutz für diesen Fall**, denn EVCC hat dafür keine eigene Untergrenze — `batteryBoostLimit` gilt nur für den Boost.

Bis zum 28.09.2026 gab es dafür zwei feste Stufen: mehr als zehn Kilowattstunden erwarteter Überschuss → 50 %, sonst 80. Die zehn waren eine Faustzahl. **Der tatsächliche Akkustand kam in der Rechnung gar nicht vor**, und es wurde auch nicht geprüft, ob der Überschuss reicht, um von der Untergrenze aus wieder *voll* zu werden. An genau diesem Abend fiel es auf: 13 kWh Überschuss waren zu erwarten, von 50 % auf 100 braucht der Akku aber rund 16 kWh — er wäre bei etwa 90 % stehengeblieben.

Jetzt wird die Grenze aus der Physik gerechnet: Der Akku darf so weit leer, wie der erwartete Überschuss ihn am nächsten Tag wieder füllen kann.

```
Punkte = Überschuss × 0,85 / Kapazität × 100
Grenze = 100 − Punkte,  begrenzt auf 40 ⋯ 80 %
```

Die Kapazität wird live aus `sensor.evcc_battery_capacity` gelesen (32.15 kWh), nicht fest eingetragen. Der Abschlag von 15 % fängt ab, dass die Prognose sich irrt; die Untergrenze von 40 % schützt den Akku auch bei bester Prognose, die Obergrenze von 80 % verhindert, dass Dauerhaft-Laden nutzlos wird. Bei schwacher Prognose steigt die Grenze von selbst, bei starker sinkt sie — stufenlos, ohne willkürliche Schwelle.

Aktuell: Prognose morgen **32.7 kWh** Ertrag bei 19.7 kWh Verbrauch, also 13.0 kWh Überschuss → Untergrenze **66 %**.

Was von der Prognose übrig bleibt

Zwei Anzeigen rechnen laufend mit, im Dashboard unter **Hausakku Laden** → **PV-Prognose**:

ANZEIGE	WERT	BEDEUTUNG
`sensor.hausakku_fehlt`	9.0 kWh	Was dem Hausakku gerade bis 100 % fehlt.
`sensor.pv_ueberschuss_frei_morgen` „Übrig, wenn Akku voll wird“	2.0 kWh	Der morgige nutzbare Überschuss **minus** dem, was der Akku braucht. Wird die Zahl negativ, wird er morgen nicht voll — das sagt die Anzeige dann ausdrücklich.

Beide aktualisieren sich mit jedem Prozentpunkt. Während das Auto lädt, sieht man den freien Rest schrumpfen — am Abend des 28.09. fiel er binnen zwanzig Minuten von 4,6 auf 2,7 kWh, während der Akku von 80 auf 74 % ging.

> **Warum die Werte einen Neustart überleben.** Sie wurden ursprünglich per `POST /api/states` gesetzt — solche Werte hält Home Assistant nur im Arbeitsspeicher und stellt sie nach einem Neustart nicht wieder her. Beim Update auf 2026.9.4 waren sie schlagartig weg, und die Automatik rechnete stumm mit ihrem Rückfallwert von 80 %. Jetzt liegen sie in **Helfern** (`input_number.prog_*`), die HA selbst führt; was man sieht, sind Vorlagen-Sensoren darüber. Gerechnet wird alle 15 Minuten. Regel für künftige Skripte: `POST /api/states` nur für Wegwerfwerte — alles, woran eine Automatik hängt, gehört in einen Helfer.

> **Trifft die Prognose überhaupt?** Seit dem 28.09.2026 vergleicht `~/energie/prognose_abgleich.py` jeden Abend um 23:50 die Prognose mit dem, was die drei Wechselrichter tatsächlich geerntet haben — eine Zeile je Tag in `~/energie/prognose_abgleich.db` . Erster Datenpunkt: Prognose 35,5 kWh, geerntet 33,8, also knapp fünf Prozent zu hoch. Nach etwa zwei Wochen lässt sich damit der Sicherheitsabschlag von 15 % begründen statt raten.

Woher die Prognose kommt

Aus **EVCC selbst**, nicht aus der Cloud: `forecast.solar` liefert heute, morgen und übermorgen und rechnet einen gelernten Korrekturfaktor für die eigene Anlage mit ein (derzeit 1.19). Die Zahlen sind dieselben wie im Victron-Portal, nur eben lokal und ohne Anmeldung. Das Victron bleibt als Reserve und liefert weiterhin die Verbrauchsprognose, die EVCC nicht hat. Achtung beim Auslesen: EVCC gibt die Energie in Wattstunden an, nicht in Kilowattstunden.

Wo die Werte stehen — und warum sie einen Neustart überleben

Im Dashboard unter **Hausakku Laden → PV-Prognose**: die Grenze selbst, der erwartete Überschuss, Ertrag und Verbrauch getrennt, der Rest für heute und der Akkustand daneben. Darunter steht die Entscheidung im Klartext und ein Verlauf über drei Tage.

> **Am 28.09.2026 gelernt:** Die fünf Werte wurden ursprünglich per `POST /api/states` nach Home Assistant geschrieben. Solche Werte haben keine Integration hinter sich — HA hält sie **nur im Arbeitsspeicher** und stellt sie nach einem Neustart nicht wieder her. Beim Update auf 2026.9.4 waren sie schlagartig weg, und die Automatik rechnete mit ihrem Rückfallwert von 80 % statt der richtigen 50 — stumm, ohne Fehlermeldung, bis zum nächsten Lauf des Skripts.
>
> Jetzt liegen sie in **Helfern** (`input_number.prog_*`), die Home Assistant in seiner eigenen Ablage führt und nach jedem Neustart wiederherstellt. Was man auf der Seite sieht, sind Vorlagen-Sensoren darüber — mit denselben Bezeichnern wie vorher, damit die Automatik unverändert weiterläuft. Nebengewinn: Als echte Entitäten haben sie jetzt Einheit, Geräteklasse und **Langzeit-Statistik**, es gibt also erstmals eine Historie der Prognose.
>
> Die Regel daraus, auch für künftige Skripte: `POST /api/states` nur für Wegwerfwerte. Alles, woran eine Automatik hängt, gehört in einen Helfer oder über MQTT.

Gerechnet wird **alle 15 Minuten** (`~/energie/pv_prognose.py` , Crontab zu den Minuten 7, 22, 37 und 52 — versetzt, damit es nicht mit dem Preis-Logger zusammenfällt). Fehlt der Wert doch einmal, nimmt die Automatik 80 %; das ist die sichere Richtung, weil der Akku dann *mehr* geladen wird und nicht weniger.

Offene Punkte

> **Greift das Boost-Limit auch bei „Dauerhaft laden“?** Nach dem Quellcode nein — es ist ausdrücklich an den Boost gebunden. Am 27.09. wurde das Limit testweise auf 55 % gesetzt, während „Dauerhaft laden“ läuft. Bleibt die Ladung bei 55 % bestehen und endet erst bei 50 %, ist belegt, dass nur die Automatisierung aus Home Assistant greift. Der Verlauf wird mitgeschrieben.

> **Zwei Wege zum selben Ziel beim Schnellladen.** Unsere eigenen Automatisierungen schonen den Hausakku über den ESS-Modus des Victron. EVCC tut dasselbe inzwischen selbst über `batteryDischargeControl` , das eingeschaltet ist. Ob sich die beiden sauber ergänzen oder ins Gehege kommen, ist noch nicht gemessen — beim nächsten Schnellladen sollte mitgeschrieben werden.

Grundlagen: EVCC-Quellcode Tag 0.316.1 (`core/loadpoint.go` , `core/site.go`), EVCC-Dokumentation, eigene Messungen am Ladepunkt. Erzeugt von `~/energie/evcc_doku_build.py` · 28.09.2026