Ziel der Automatik: Ladedauern: Huawei Solar Integration: forcible_charge_soc Schonende Akkuladung **0 % bis 10 %:** 0,2C bis 0,3C Stufe 0: 0,0C (= Modus Pause) Huawei Solar Integration: stop_forcible_charge **10** % **bis 30** %: 0,3C bis 0,5C **30 % bis 80 %:** 0,5C bis 0,7C **Stufe 1: 5% bis 10 %:** 0,2C --> 25:00 Minuten **Stufe 2: 10 % bis 30 %:** 0,5C --> 40:00 Minuten **80 % bis 100 %:** 0,2C bis 0,3C **Stufe 3: 30 % bis 80 %:** 0,5C --> 100:00 Minuten **Stufe 4: 80 % bis 100 %:** 0,2C --> 100:00 Minuten - bei Erreichen von 80% SoC Ladevorgang pausieren und restliche 20% so steuern, dass erst spät nachmittags fertig geladen wird **Stufe 1: 5% bis 10 %:** 0,3C --> 16:42 Minuten **Stufe 2: 10 % bis 30 %:** 0,5C --> 40:00 Minuten **Stufe 3: 30 % bis 80 %: 0,7C*** --> 71:24 Minuten **Stufe 4: 80 % bis 100 %:** 0,3C --> 60:42 Minuten * Erst ab einem 10 kWh Akku relevant. Huawei erlaubt folgende Ladeleistungen: Main Triggers: [SoC] 5 kWh => 2.500 W 10 kWh => 5.000 W [Automatik] [Vollladung aktiv] 15 kWh => 5.000 W [Vollladung preisoptimiert] [forciertes Laden aktiv] [Battery Akkusteuerung Restladezeit Stunden] [EPEX Spot Data Rank] [Tage seit Letzter Voll-Ladung] [Wöchentliche Vollladung] [Schonungsmodus] [Soll-Ladeleistung] { Sonnenaufgang } [Wallbox ignorieren] (mit Service forcible_charge_soc nicht benötigt) (mit Service forcible_charge_soc nicht benötigt) [Soll-Ladeleistung] [Vollladung aktiv] [Vollladung Preisoptimiert] [Automatik] [EPEX Spot Data Rank] [Automatik] [Battery Akkusteuerung Restladezeit Stunden] [Vollladung aktiv] [Tage seit letzter Voll-Ladung] [Tage seit letzter Voll-Ladung] [Schonungsmodus] [Wöchentliche Vollladung] [Wallbox ignorieren] [Automatik] = ON { Sonnenaufgang } [Vollladung aktiv] = TRUE Nein [Automatik] ist "ON" ({Sonne scheint} SoC = 100% ODER Stop [Soll-Ladeleistung] nach [Vollladung aktiv] = TRUE) [Huawei Leistungslimit Laden] mit Service forcible_charge_soc nicht benötigt [Tage seit letzter Voll-Ladung] > 6 [SoC] zwischen 81% und 100% [SoC] zwischen 31% und 80% [SoC] zwischen 11% und 30% [SoC] zwischen 0% und 10% Heute in [Datum letzer Voll-Ladung] schreiben [Wöchentliche Vollladung] = TRUE [Vollladen aktiv] = TRUE Ja [Battery Akkusteuerung Restladezeit Stunden] > [Vollladung preisoptimiert] = True [Vollladung aktiv] = TRUE [EPEX Spot Data Rank] [Wallbox ignorieren] = TRUE [Ladestufe] = 1 [Ladestufe] = 2 [Ladestufe] = 3 (sollte gesetzt sein, wenn Wallbox-Ladeleistung vo Hausverbrauch abhängig und nicht vice versa!) [forciertes Laden aktiv] = FALSE)(→ [Ladestufe] = 0 [Restliche Kapazität] < (0,5 * [erw. PV-Überschuss [forciertes Laden aktiv] = TRUE [forciertes Laden aktiv] = FALSE [Vollladung aktiv] = FALSE [Restliche Kapazität] < (0,5 * [erw. PV-Überschuss]) ohne WB]) ODER [Vollladung aktiv] = TRUE [Vollladung aktiv] = TRUE - sensor: - name: "Battery Akkusteuerung Restladezeit Stunden" unit_of_measurement: "minutes" state: > Trigger: [forciertes Laden aktiv] {% set start_soc = states('sensor.battery_batterieladung') | float %} {% set end soc = 100.0 %} {% set schonungsmodus = states('input_select.battery_akkusteuerung_schonungsmodus') %} {% if schonungsmodus == 'sehr schonend' %}
{% set c_value_0_10 = 0.2 %}
{% set c_value_10_30 = 0.5 %} [forciertes Laden aktiv] [Automatik] = AUS {% set c_value_30_80 = 0.5 %}
{% set c_value_80_100 = 0.2 %} {% else %}
{% else %}
{% set c_value_0_10 = 0.3 %}
{% set c_value_10_30 = 0.5 %}
{% set c_value_30_80 = 0.7 %}
{% set c_value_80_100 = 0.3 %}
{% endif %} Nein {% set time_0_10 = (([end_soc, 10] | min) - start_soc) / c_value_0_10 if start_soc < 10 else 0 %}
{% set time_10_30 = (([end_soc, 30] | min) - [start_soc, 10] | max) / c_value_10_30 if start_soc < 30 and end_soc > 10 else 0 %}
{% set time_30_80 = (([end_soc, 80] | min) - [start_soc, 30] | max) / c_value_30_80 if start_soc < 80 and end_soc > 30 else 0 %}
{% set time_80_100 = (end_soc - [start_soc, 80] | max) / c_value_80_100 if end_soc > 80 else 0 %} {% set total_time = time_0_10 + time_10_30 + time_30_80 + time_80_100 %}
{{ (total_time * 60) | round(0) }} ([Huawei Laden aus dem Netz] = Schreibe Maximum in (Service "disable_forcible_charge") [Huawei Leistungslimit Laden] - name: "Battery Akkusteuerung Restladezeit Stunden" unit_of_measurement: "hours" state: > {% set minutes = states('sensor.battery_akkusteuerung_restladezeit') | float %}
{% set buffer = 15 %} [Huawei Laden aus dem Netz] = Service "forcible_charge_soc" Aus {{ ((minutes + buffer) / 60.0) | round(0, 'ceil') }} template: - name: "Battery Akkusteuerung Soll-Ladeleistung"
unit_of_measurement: "W" state: > {% set ladestufe = states('input_number.battery_akkusteuerung_ladestufe') | int %}
{% set schonungsmodus = states('input_select.battery_akkusteuerung_schonungsmodus') %}
{% set kapazitaet = states('input_number.battery_speicherkapazitaet') | float %} # Liest die maximal zulässige Ladeleistung aus {% set max_leistung = state_attr('number.battery_maximale_ladeleistung', 'max') | float %} {% if ladestufe == 0 %} {% set c_wert = 0 %} {% elif ladestufe == 1 %} {% set c_wert = 0.2 if schonungsmodus == 'sehr schonend' else 0.3 %}
{% elif ladestufe == 2 %} {% set c_wert = 0.5 %} {% elif ladestufe == 3 %} {% set c_wert = 0.5 if schonungsmodus == 'sehr schonend' else 0.7 %} {% else %} {% set c_wert = 0 %} $\{% \text{ endif } %\}$ {% set berechnete_leistung = (c_wert * kapazitaet * 1000) %} # Ausgabe mit Deckelung auf maximal zulässige Ladeleistung {{ [berechnete_leistung, max_leistung] | min | round(0) }}