DAI – Free Energy Session Controller v3.6 r1 – Detailed Design Specification (v1.0)

Issue Date: 2025-10-26

Prepared by: ChatGPT (GPT-5 Thinking)

Design Framework: DAI + RBC Framework v5.84 (Oct 2025)

Change Control: No-Surprises Protocol – Semantics Frozen by default.

1. Functional Overview

The DAI – Free Energy Session Controller (v3.6 r1) manages Octopus Free Energy sessions within the DAI layer. It transitions the SolaX Master inverter between Feed-in Priority and Self Use modes based on Octopus calendar events. The controller preserves battery headroom before the free-energy window, ensures full grid charging during the live session, and safely returns control to planners after the event. The automation is fully Visual-Editor safe and uses only Master inverter entities.

2. Entity and Helper Map

Role	Entity ID	Direction	Description
Calendar source	calendar.octopus_energy_a_26dacffe_octoplus_free_electricity_session	Read	Free Energy session events; used for start/end triggers and start_time attribute for Pre-Hold

			detection
Override flag	input_boolean.dai_session_override_active	Write / Read	Indicates controller ownership; ON during Pre-Hold and Live
Inverter mode selector	select.solax_house_charger_use_mode	Write / Read	Primary control for SolaX Master mode (Feed-in / Self Use)
SoC cap (Master)	number.solax_house_selfuse_nightcharge_upper_soc	Write	Target set to 100% at session start
Planner SoC floor helpers	input_number.dai_min_soc_floor_cosy / dai_min_soc_floor_flux	Read by Planner only	Controller does not read or write; ownership retained by Grid Charge Planner
Timestamp helpers	input_datetime.last_mode_change / dai_last_mode_notify	Write	Logging and notification throttle
Notificati	notify.mobile_app_pixel_9_pro	Send	Mobile notificati

on target on target

3. Trigger Matrix and Expected Outcomes

Triggers and associated outcomes reflecting current live automation behaviour:

Trigger ID	Source	Event	Expected Outcome
t_session_start	Calendar	event:start	Set Self Use + SoC=100% + Override ON
t_session_end	Calendar	event:end	Clear Override OFF (guard + retry); no mode/cap writes
t_ha_start	Home Assistant	start	Evaluate Pre-Hold; assert if event today not yet live
t_heartbeat_5m	Time pattern	/5 min	Maintain Pre-Hold if conditions true; no notifications
t_watchdog_2m	Time pattern	/2 min	If calendar=off and override=on, force-clear override and log

4. Logic Flow Description

• Pre-Hold Detection: derives from calendar.start_time (localised) → if event today, not live, and now < start_time: Feed-in Priority + override ON (no SoC write). Notification only on HA start.
• Start Event: calendar fires event:start → Self Use Mode + SoC cap=100% + override ON.
• End Event: calendar fires event:end → delay 3s → override OFF (retry after 2s if still ON).
• Watchdog: every 2min → if calendar=off and override=on → clear override and log ('WATCHDOG clear'); stateless check (no lookback).
• Notification throttle: 120s minimum between Pixel alerts.
• HA restart during a live session: controller does not reassert Start logic; retains current mode until new trigger or external automation acts.

5. Interaction and Control Hierarchy

The controller coordinates with other DAI automations via override logic. While override=ON, planners and PV Clipping automations pause writes to the inverter.

Layer	Role	Takes / Yields Control	Key Behaviour
Free Energy Controller	Manages Free Energy sessions	Takes control (override ON)	Pre-Hold = Feed-in; Start = Self Use 100%; End = override OFF
Minimise PV Clipping v2.x	Normal-day optimiser	Yields when override ON; resumes at End	Idle during override
Anti-Clipping Monitor	Safety layer	Operates under override	Requests Feed-in Priority if both batteries ≥55% SoC
Grid Charge Planner v5.3+	Tariff SoC floor planner	Paused while override ON; resumes after End	Owns SoC floors via helpers

6. Safety and Recovery Mechanisms

• End-clear guard: retries override OFF if still ON after 2s.
• Watchdog: runs every 2min and clears override immediately if calendar=off and override=on.
• HA-start alignment: ensures Pre-Hold asserts after reboot if event today not yet started.
• All mode writes are idempotent with 20s confirm and 5s dwell timers.

7. Notifications and Logging

Pixel 9 Pro notifications (≥120s throttle):
• Pre-Hold: Sent only on HA start ('Event today (not live) → Override ON, Feed-in Priority set. Planner owns SoC floors.')
• Live: 'Live session ON → Override ON, Self Use set, Target SoC=100%.'
• Ended: 'Live session OFF → Override cleared. Control returns to Planner/Clipping.'
• Default-path log always records 'No matching branch executed' if no trigger condition met.

8. Acceptance Test Cases

Test Case	Pre-Condition	Action	Expected Result
A1 – Pre-Hold Detect	Calendar shows event later today	Wait for heartbeat	Feed-in Priority + override ON (no cap write)
A2 – Live Start	Calendar fires start	Observe logs	Self Use + SoC 100% + override ON
A3 – End Event	Calendar fires end	Observe override	Override OFF within ≤5s
A4 – Watchdog Recovery	Leave override ON post-event	Wait ≤2min	Override auto-cleared + WATCHDOG log
A5 – HA Restart During Live	Calendar on (live)	Restart HA	No Start action; mode persists
A6 – No Event Day	Calendar off	Observe normal ops	No controller actions; Planner active

9. Version Lineage and Governance

v3.6 r1 (2025-10-25): Parser-safe refactor; documentation aligned with live code.
v3.6 (2025-10-25): Structural change – calendar pre-hold, Feed-in Priority, Planner-owned SoC floors, End-clear guard + watchdog.
v3.5 (2025-10-24): Calendar triggers + legacy helper.
v2.6a (2025-09): Helper-based pre-hold (Self Use mode).
Linked to Design-Doc v5.84 §8.4 and Appendix C.