

DAI – Usage Daily Rolling Accumulator v1.0

Prepared by: ChatGPT (GPT-5 Thinking)
Issue Date: 01 November 2025
Design Framework: DAI + RBC Framework v5.85 (October 2025)
Change Reference: Documentation – Detailed Design Creation (No behavioural change).

1. Functional Overview

This automation computes a rolling **Daily Usage (kWh)** by integrating instantaneous power reported by `sensor.brenchley_total_load_w` once per minute. Each run adds an increment to `input_number.dai_usage_daily_kwh_rolling` and performs a self-reset at local midnight so the next day starts from 0.00 kWh. All maths occurs in Jinja variables; the output is rounded to 2 dp and written only when the sample is non-negative. Visual-Editor-safe YAML structure is used throughout.

2. Entity Map

Entity ID / Var	Purpose	Direction
sensor.brenchley_total_load_w	Instantaneous whole-home power (W)	Read
input_number.dai_usage_daily_kwh_rolling	Rolling accumulator of energy (kWh)	Read/Write
inc_kwh = p_w / 60000	Per-minute energy increment (kWh) using rectangular rule	Compute
is_midnight	True at (hour==0 and minute==0) to reset accumulator	Compute

3. Trigger Matrix

Trigger ID	Condition	Purpose
t_every_min	Time Pattern: minutes: /1	Sample once per minute and update accumulator

4. Logic Flow

1. Compute `is_midnight` from local time (hour==0 and minute==0).
2. Read `p_w_raw` from sensor and coerce to numeric `p_w` (unknown/unavailable→0).
3. Calculate `inc_kwh = p_w / 60000` (since 1 minute = 1/60 h and 1 kWh = 1000 W·h).
4. Read current accumulator `cur_kwh` from helper.
5. If is_midnight → reset helper to 0.00 kWh and logbook reset message.
6. Else if $p_w \geq 0$ → write (cur_kwh + inc_kwh) rounded to 2 dp to the helper.
7. Else → log a no-add message (invalid/unavailable power).

5. Guards and Safety

- No inverter writes – helper only.
- Numeric coercion converts non-numeric sensor states to 0 to avoid template errors.
- Negative power values are not added ($p_w \geq 0$ condition), preventing erroneous subtraction.
- `mode: single` ensures the accumulator update does not overlap on minute boundaries.
- Midnight reset uses local time (hour==0 and minute==0) to start each day at zero.

6. Scheduling

Runs every minute on the minute. The minute-aligned trigger ensures that each increment models a 60-second window.

7. Energy-Conversion Note

For a sample interval $\Delta t = 1 \text{ minute} = 1/60 \text{ h}$ and instantaneous power $P \text{ (W)}$:
Energy increment (kWh) $\approx (P / 1000) \times (1/60) = P / 60000$. This is the rectangular rule (sample-and-hold). It is accurate when power does not vary rapidly within the minute; for rapidly changing signals, HA's Integration Sensor (trapezoidal rule) can improve fidelity if needed.

8. Acceptance Tests

Test ID	Scenario	Expected Result
T1	Normal minute tick with valid P	Accumulator increases by $P/60000$ kWh (rounded to 2 dp).
T2	Midnight boundary	Accumulator resets to 0.00 kWh and logs reset

		message.
T3	Unavailable/unknown power	No addition; optional log indicates no add.
T4	Negative power reading	No addition occurs (guard $p_w \geq 0$).

9. Version Lineage & Governance

v1.0 – Initial rolling accumulator based on 1-minute sampling of total load. Uses rectangular integration $P/60000$.

Governance: Protected Architecture Mode v5.9; Two-Phase Code-Change Gate v2.2. This document records the live behaviour; no code changes introduced.

10. Compliance Checklist

- ✓ Entity ↔ ID alignment confirmed against live YAML: `brenchley_total_load_w`, `dai_usage_daily_kwh_rolling`.
- ✓ Trigger has ID and uses minute pattern ``/1``.
- ✓ Visual-Editor-safe structure; no anchors/includes.
- ✓ No inverter writes; Master-only rule unaffected.
- ✓ Energy conversion $P/60000$ rationale documented; rounding to 2 dp noted.