

SRNE ASP48120SH3 Safe Settings Write: design, registers, and commissioning (#53)

Purpose and scope

This page documents the SRNE ASP48120SH3 safe settings write feature (story #53): the ability to remotely change a bounded set of inverter parameters over Modbus, with every write guarded, verified by read back, and never retried. It records the design, the exact registers, the commissioning procedure, and, importantly, the current verification status so the work stays traceable and defensible.

Read this alongside the validated register map (page 126) and the mbpoll command reference (page 125).

What can and cannot be written

The feature can change eight settings only. Each is disabled by default (sentinel -1), bounded, applied only when the configured target differs from the live reading, and verified by a fresh read back.

#	Setting	Register (FC16)	Bounds	Units / raw multiplier
0	Discharge cutoff SoC	0xE00F	0 to 100	percent, x1
1	Stop charge current	0xE01C	0 to 100 A	amps, x10 (deci-amp raw)
2	Stop charge SoC	0xE01D	0 to 100	percent, x1
3	Low SoC alarm	0xE01E	0 to 100	percent, x1
4	Switch to line (grid) SoC	0xE01F	0 to 100	percent, x1
5	Switch to battery SoC	0xE020	0 to 100	percent, x1
6	AC charge current limit	0xE205	0 to 100 A	amps, x10

#	Setting	Register (FC16)	Bounds	Units / raw multiplier
7	Max total charge current limit	0xE20A	0 to 100 A	amps, x10

The current ceiling is `MAX_BATTERY_CURRENT_A` = 100 A, the continuous rating. The datasheet peak is 120 A for 3 seconds, so the continuous value is the correct, conservative cap.

It explicitly cannot: change charge or discharge time windows, command a specific charge or discharge power, change operating or source priority modes, start or stop the inverter, or reset or clear data.

Safety design

Every write passes through the same defended path (component `SrneBatteryInverterImpl`, helper `SafeWriteHandler` in the `io.openems.edge.ess.srne` bundle):

1. Default off. Nothing writes unless the operator sets `controlEnabled` true in the gateway config. Default is false.
2. Machine state gate. A write is only queued when the inverter reports a verified stable state. Register 0x0210 value 2 (on-grid, mains bypass) is the only verified state; every other state, including off-grid (3), transitioning, fault, and undefined, blocks all writes.
3. Bounds. Each setting is range checked against the table above. An out of range target is rejected before any Modbus write and marked failed (terminal).
4. One shot per activation. The write state machine is IDLE, then QUEUED, then AWAITING_READBACK, then VERIFIED or FAILED. A setting is queued at most once per component activation; VERIFIED and FAILED are terminal, so there is no re-write and no storm even though reconciliation now runs every cycle.
5. Read back verification. After the FC16 write executes, a fresh read of the same register must equal the target to reach VERIFIED. A mismatch goes to FAILED.
6. Bounded wait, no retry. The read back is allowed up to 30 Edge cycles; on timeout the state is FAILED. A failed or mismatched write is never retried automatically.
7. Thread safe. The handler methods are synchronized so the cycle thread (reconcile) and the Modbus worker thread (execute and read back) transition state atomically.

The aggregate `SAFE_WRITE_STATE` channel surfaces the highest severity handler state so an operator can watch commissioning progress.

The standalone topology fix (openems PR #23)

During #53 commissioning, gateway config v18 enabled control and targeted Low SoC alarm 15 to 16, but no FC16 write occurred and the read back stayed at 15. Root cause: gw-aaron runs a standalone Srne.Ess topology, so the generic ManagedSymmetricBatteryInverter run() path was never invoked, and reconciliation was hung off run(). The gateway was restored to the healthy v17 read only profile; the inverter setting did not change.

The fix moves the reconcile trigger to the Edge TOPIC_CYCLE_AFTER_PROCESS_IMAGE event, which fires every cycle in all topologies. It drives both the read back timeout advance and the queue, so the failure path works too. All guards above are unchanged; only the trigger moved.

Commissioning procedure (staged)

The commissioning is deliberately low risk and staged, with a pre staged rollback:

1. v17: read only baseline, control disabled.
2. v18: enable control, target exactly one setting (Low SoC alarm 15 to 16), leave the other seven at -1 (disabled). Verify a fresh read back of 16 and SAFE_WRITE_STATE VERIFIED, with stable telemetry and no other writes.
3. v19: restore Low SoC alarm 16 to 15, keep control enabled, all other targets at -1.
4. Rollback profile (microgrid PR #33): a break glass config that targets only Low SoC alarm 15, used if v18 changes the inverter but verification fails.

Gate: the v19 enable step (microgrid PR #34) must not be deployed until v18 has been deployed and verified with SAFE_WRITE_STATE VERIFIED on the real inverter.

Verification status (as of 2026-08-08)

Be precise here, because the live unit has not yet confirmed a write.

- Code reviewed at source level, including an adversarial second pass and mutation testing of the safety gates.
- Unit and integration tested in Java on JDK 21. Coverage includes: the write queues via the new event path in the standalone topology; control disabled never queues; a non verified machine state never queues (both mutation proven, that is, removing the guard makes the test fail with a real queued write); the one shot property, no re-queue across extra cycles; and, at unit level, the read back mismatch and timeout paths with no retry.
- Not yet verified on the live inverter. No setting has been successfully written and read back on the physical unit. The only live result so far is the v18 non write above, which exposed the topology bug. Live verification is pending: merge openems #23, build a new openems-edge image, then repeat the v18 15 to 16 commissioning gate on the real inverter.

- There is no Python live test harness for this. Writes are exercised through OpenEMS plus the gateway commissioning config and verified by Modbus read back, not by a standalone script.

Traceability

- Story: nfe-modbus-energy-logger issue #53.
- Code: openems PR #21 (guarded write infrastructure, merged), openems PR #23 (standalone topology fix plus gate tests, in review).
- Deployment: microgrid-gateway PR #32 (v18 commission, merged), #33 (rollback pre stage), #34 (v19 enable). Gated as above.
- Registers: SRNE ASP48120SH3 Modbus Register Map, validated (page 126). mbpoll command reference (page 125).

Live read test results (2026-08-08)

First live read of the eight settings on the pilot inverter (gw-aaron), via direct Modbus (mbpoll, FC03) with the OpenEMS edge and the nfe-monitor process stopped to free the RS485 bus. Read only, no writes. Accessed over Raspberry Pi Connect plus ZeroTier (nfetestpiaaron, 10.135.127.37).

Setting	Register	Raw	Decoded
Discharge cutoff SoC	0xE00F	5	5 percent
Stop charge current	0xE01C	30	3.0 A
Stop charge SoC	0xE01D	100	100 percent
Low SoC alarm	0xE01E	15	15 percent
Switch to line (grid) SoC	0xE01F	10	10 percent
Switch to battery SoC	0xE020	90	90 percent
AC charge current limit	0xE205	1200	120.0 A
Max total charge current limit	0xE20A	1000	100.0 A

Context registers: machine state 0x0210 = 3 (off-grid, inverter-powered), SoC 0x0100 = 92 percent.

Findings:

1. Low SoC alarm reads 15, matching the gateway config. This confirms the v18 15 to 16 commissioning write did not change the inverter (the standalone topology bug fixed in openems #23); the live value is still 15. This is the live evidence behind that finding.

2. Machine state is 3 (off-grid). The write feature only permits writes in the verified state (2, on-grid mains bypass), so a write attempted right now would be correctly blocked by the machine-state gate. A live write test must wait for the inverter to be on-grid.
3. Current scaling is x0.1 (deci-amp): AC charge limit 120.0 A, max charge 100.0 A, stop-charge current 3.0 A. SoC and percent settings are x1.
4. Bus collision confirmed live: both the OpenEMS edge and the nfe-monitor (F3) process were found mastering /dev/ttyUSB0 at the same time. The bus only freed after stopping both. F3 must not run a second Modbus master on this Pi.
5. The block read of 0xE001 to 0xE02D (count 45) returns Modbus exception 0x0A (gateway path unavailable) on this unit even with the bus free; read the settings individually or in small blocks.

Raw mbpoll output:

```
[57359]  5      (0xE00F discharge cutoff SoC)
[57372] 30      (0xE01C stop charge current)
[57373] 100     (0xE01D stop charge SoC)
[57374] 15      (0xE01E low SoC alarm)
[57375] 10      (0xE01F switch to line SoC)
[57376] 90      (0xE020 switch to battery SoC)
[57861] 1200    (0xE205 AC charge current limit)
[57866] 1000    (0xE20A max charge current limit)
[528]    3      (0x0210 machine state)
[256]    92     (0x0100 SoC)
```

Status: reads are now live-verified as above. The live WRITE test (Low SoC alarm 15 to 16, verify by fresh readback, restore to 15) is still pending, and requires both openems #23 deployed and the inverter in the verified on-grid state (it was off-grid at read time).

Revision #2

Created 2026-08-08 04:29:36 UTC by hillary.arinda

Updated 2026-08-08 05:04:07 UTC by hillary.arinda