INTERNAL DOC TORUS-PROG-001 REV 2026-07 PacketFive

TORUS • PROJECT DOCUMENT

INTERNAL

TORUS-PROG-001

Implementation Status & Roadmap

TORUS is an Unattended Facility Sensing platform for AI/HPC data centers. This document tracks the whole system: hardware, firmware, CCISRT, deployment rehearsal, and emulation.

As of 2026-07-07 • repository `/opt/repo/TORUS` • branch `feat/facility-sensing-docs-sim-ccisrt`, under active development.

1 Status board

Area	State	Status	Detail
Hardware design	TORUS-SN rev pre-prod	CONVERGED	Fully generated KiCad 10 board, routed & polished
Schematic / ERC	0 violations	CLEAN	Netlist mirrors PCB by construction
PCB / DRC	19 violations, 3 unconnected	HAND-FINISH	Mostly waivable; ∼10 min GUI cleanup remains
Firmware	Targets previous hardware rev	PORT NEEDED	Zephyr app written for ESP32-C3 / SX1268 board
MEG gateway	Concept + datasheet	DESIGN STUDY	Clustering / HA design target; node protocol is gateway-ready
CCISRT layer	Runnable, phases 1-2.7	RUNNABLE	Roles, login, telemetry, BKC, controllable sim, maps
Deployment rehearsal	Isaac Sim starter	RUNNABLE	Portable planner + Isaac scene in `torus-isaac-sim`
Facility emulation	Virtual RF medium	RUNNABLE	`torus-emu`: nodes + medium + MEG bridge into CCISRT
Variant hardware	SN-V2/V3/V4	DESIGN BRIEF	Architecture + BOM briefs under `hw/variants/`; no PCB yet

2 Implemented hardware (TORUS-SN)

The current board supersedes the original ESP32-C3/SX1268 concept in the original README. It is an 80 × 55 mm 4-layer node with ESP32-S3-WROOM-1 host, LR1110 (LoRa 433 MHz + GNSS + Wi-Fi-scan), OPA2134/MCP4017 geophone AFE with LM393 hardware wake, 2× INMP441 I2S microphones, DS3231 RTC, TP4056 + AMS1117 power, 3× TPS22918 gated rails. 91 parts, 68 nets.

- **Single source of truth** - `hw/torus_design.py` defines parts, nets, footprints, GPIO→pad mapping, net classes, and an explicit engineering-flags list.
- **Custom footprints** - LR1110 QFN32 (function-named pads, thermal via array), INMP441 LGA6 (sound-port hole), FXP73 antenna land, keepout-stripped WROOM variant.
- **Generated schematic** - `gen_sch.py`; ERC clean. ERC caught a real design bug (missing VBAT monitoring divider), fixed with R29/R30/C54.
- **Generated PCB** - placement with courtyard-collision avoidance (0 overlaps), GND/3V3 inner planes, RF keepouts, stitch vias, silkscreen pass.
- **Automated routing** - headless Freerouting; `release.sh` runs N builds and keeps the best (router is non-deterministic), `polish.py` post-processes (fab-layer refs, redundant via removal, silk fixes, island removal).
- **Fabrication outputs** - gerbers, drill, pick-and-place, BOM CSV, schematic PDF, top/bottom SVG, populated STEP (5.8 MB), optional GLB export. In `hw/gerbers/`, `hw/fabrication/`, `hw/3d_models/`.
- **KiCad 10 toolchain** - migrated from KiCad 8; AppImage wrappers (`setup-kicad10.sh` → `kpython` / `kcli.sh`), all API breaks handled.

Remaining DRC picture (release board)

Finding	Count	Assessment
Copper-to-edge clearance	5	Edge-launch SMA / USB-C pads, by design, waivable
Clearance	11	Sub-0.2 mm squeezes near the LR1110 fanout, review, likely acceptable
Solder-mask bridges	2	Cosmetic; check fab's minimum mask sliver
Shorting items	1	Zone-pair bookkeeping artifact
Unconnected items	3	`LR1110_IRQ`, short `VCC_LR1110` stub, one zone pair; ∼10 min of hand routing

3 Implemented firmware and docs

- **Zephyr application skeleton** - `src/main.c`: duty-cycled main loop, watchdog, interrupt-driven wake (LoRa DIO1, RTC alarm), power-gate management.
- **LoRa handler** - `src/lora_handler.c` on the Zephyr LoRa API (SX126x driver): config, TX, sleep/wake.
- **GPS handler** - `src/gps_handler.c`: UART NMEA sentence parser.
- **Peripheral handlers** - power gates, geophone ADC sampling, SPI digipot gain.
- **Custom Zephyr board** - `boards/riscv/torus_iot_edge` (device tree, Kconfig, CMake) with user & developer guides; builds with `west build -b torus_iot_edge`.
- **Documentation** - design document (README), firmware structure notes, PCB design & routing blueprints in `docs/`.

The firmware/hardware gap is the project's main debt

Everything in `src/` and `boards/` targets the earlier ESP32-C3 + SX1268 + AB1805 + LC76G + SPI-camera design. The built board is ESP32-S3 (Xtensa) + LR1110 + DS3231 + I2S mics + I2C digipot, with no camera. The application architecture carries over; the BSP and every driver binding must be ported. *A first-pass ESP32-S3 port (`boards/xtensa/torus_sn`) now exists and is documented in §4, though it has not yet been compiled against a Zephyr workspace.*

4 Implemented system docs, CCISRT, deployment rehearsal, and emulation

- **Product documentation set** - brochure plus datasheets for TORUS-SN, node variants, TORUS-MEG, and TORUS-CCISRT, cross-linked as a static site with both HTML and LaTeX-built PDF sources (this document included).
- **TORUS-MEG gateway concept** - architecture and candidate BOM captured in a datasheet, including cluster / high-availability federation.
- **TORUS-CCISRT app (runnable)** - `torus-ccisrt/` ships a shared event schema, ingest adapter, persistent store with replay, correlation/track fusion, capability-based roles (viewer, field-operator, operator, analyst, admin, root, gateway), interactive login, per-node metadata and BKC firmware upgrade/downgrade, a controllable facility simulator, OpenTelemetry telemetry, configurable maps (OSM/Mapbox/Google-proxied/offline), and an Electron desktop shell. Verified end-to-end by `npm run smoke` (43 checks).
- **Isaac Sim deployment rehearsal** - `torus-isaac-sim/` ships a dependency-free planner (ring placement, seismic detection model, SVG output) with campus perimeter, utility-yard, and four object-class scenarios (human, vehicle, drone, animal), plus an NVIDIA Isaac Sim scene script for the full twin.

- **Facility emulation fabric** - `torus-emu/` emulates a virtual RF/network medium (LoRa, SDR, wired, Wi-Fi link models with range and contention), node and MEG gateway stand-ins, and a bridge into the real CCISRT app; verified end-to-end.
- **Node variant hardware design briefs** - `hw/variants/` documents host architecture, block diagrams, candidate BOM, and open flags for SN-V2 (SDR carrier), SN-V3 (wired optical), and SN-V4 (transducer interlock daughterboard).

5 Pending TODOs

P0 - before sending the board to fab

#	Task	Effort
0.1	Hand-finish the 3 unconnected nets (`LR1110_IRQ`, `VCC_LR1110` stub, zone pair) in the KiCad GUI	∼10 min
0.2	Verify the 6 engineering FLAGS covering MINI-1→WROOM-1 substitution impact; USB on GPIO35/36 vs native GPIO19/20; MCP4017, SKY13350, SPF5189Z pin orders; LR1110 pad geometry vs production datasheet	hours
0.3	Decide the USB programming path, move D± to GPIO19/20 or add a USB-UART bridge	design call
0.4	Review AMS1117 quiescent current against sleep-current targets (original spec targeted ∼8-9 μA; an LDO swap may be needed)	design call
0.5	Review the 11 LR1110-area clearance violations against the chosen fab's rules	∼1 h

P1 - firmware port to TORUS-SN

#	Task	Notes
1.1	New Zephyr board for ESP32-S3 (Xtensa)	First-pass `boards/xtensa/torus_sn` exists; needs Zephyr-workspace compile and pinctrl macro verification
1.2	LR1110 driver integration	LoRa + GNSS scan + Wi-Fi scan; `src/lr1110.c` is an SPI skeleton, opcodes to be completed from the LR11xx manual

#	Task	Notes
1.3	I2S stereo capture for 2× INMP441	`src/acoustic.c` exists; feeds TinyML classification
1.4	DS3231 RTC support	`src/ds3231.c` exists; wire 32 kHz output usage for LR1110 GNSS
1.5	MCP4017 gain control over I2C	`src/seismic.c` exists; replaces SPI MCP41010 driver
1.6	Wake-on-seismic path and TPS22918 rail gating	`src/power.c` + `src/seismic.c` exist; verify against real hardware
1.7	Retire camera support	TORUS-SN carries no camera; legacy Arducam code path removed
1.8	Battery telemetry via ADC_VBAT divider	`src/power.c` exists; new on this rev (R29/R30)

P2 - system and housekeeping

#	Task	Notes
2.1	TORUS-MEG gateway detailed design & build	Take the datasheet concept to schematic/PCB; implement cluster membership, leader election, and HA failover
2.2	Enclosure design	Autodesk Fusion, importing `hw/3d_models/TORUS-SN.step`
2.3	Update README to the TORUS-SN architecture	Rewritten as the platform overview; legacy design preserved under `docs/`
2.4	`gen_bom.py` with manufacturer part numbers	Planned in `torus_design.py` header; current BOM is value/footprint only
2.5	Push the development branch and open a pull request	86 commits on `feat/facility-sensing-docs-sim-ccisrt`, pushed; PR not yet opened (no `gh`/token in this environment)
2.6	Bring-up & validation plan execution	Power, peripherals, AFE calibration, LoRa range, GNSS TTFF, sleep-current measurement
2.7	Isaac Sim twin with terrain import + physics seismic	GIS/heightmap terrain, PhysX-based detection, visual-mast sensors

#	Task	Notes
2.8	SN-V2 SDR carrier hardware design	Design brief done (`hw/variants/sn-v2-sdr.md`); Zynq + AD9363 phase-1 carrier stacked on the V1 board; no schematic/PCB yet
2.9	SN-V3 wired optical hardware design	Design brief done (`hw/variants/sn-v3-optical.md`); host swap for RMII EMAC + 100BASE-FX PHY, or ADIN1110 T1L fallback
2.10	SN-V4 transducer interlock daughterboard	Design brief done (`hw/variants/sn-v4-transducer.md`); hardware three-condition interlock, fail-safe de-energized
2.11	CCISRT production hardening	Real identity provider (OIDC/mTLS), database-backed persistence, packaged desktop/mobile installers, code signing
2.12	Native `native_sim` node in `torus-emu`	Build the real Zephyr firmware onto the emulated virtual medium via `radio_virtual.c` (sketch, unbuilt)

6 Toolchain reference

Piece	Location / command
KiCad 10 (AppImage)	`/opt/kicad`, wrappers: `kpython`, `kcli.sh` (`setup-kicad10.sh`)
Full hardware build	`hw/build.sh` (`REUSE_SES=1` skips ∼3 min routing)
Best-of-N release	`hw/release.sh`
Schematic + BOM	`hw/build_sch.sh`
Firmware build	`west build -b torus_sn .`
CCISRT app	`cd torus-ccisrt && npm start`
Isaac Sim planner	`cd torus-isaac-sim && python run_sim.py --scenario ...`
Facility emulation	`cd torus-emu && npm run demo`

Pipeline gotchas already solved

Encoded in the scripts, don't fight them. Zone fill must run as a subprocess on the saved board; `kicad-cli` exports without refilling zones; Freerouting hangs on stacked pre-placed vias; several library footprints ship board-blanketing antenna keepouts that must be stripped.

TRADEMARKS & THIRD-PARTY NOTICE

TORUS is an independent platform. Company names and product model numbers referenced in this document (including but not limited to Semtech, Teledyne FLIR, R.T. Clark, InvenSense, Texas Instruments, Microchip, Espressif, Analog Devices / Maxim Integrated, and NVIDIA) are used solely for engineering and bill-of-materials identification and imply no affiliation with or endorsement by those companies. Supply of any such third-party product or component to the designers, manufacturers, integrators, or evaluators of the TORUS platform remains at the sole discretion of the respective owning company, organisation, or legal entity. All trademarks are the property of their respective owners.