Project SHIELD Generation 2.0

Purpose. Evolve SHIELD from TRL-3 concept to TRL-6 field-validated prototype for universal, embedded sensor prognostics (on-device RUL + UQ + XAI) under tight MCU constraints.

Target hardware classes.

A: STM32H7 (≤1MB SRAM, ≤200MHz) • B: ESP32-S3 (≤512KB SRAM+PSRAM) • C: STM32F4 (≤128KB SRAM)

---

1) Problem & Goals (What "universal" means)

- **Universal** = sensor-agnostic across classes (mechanical, inertial, electrochemical) with ≤20-shot adaptation per new sensor instance.
- **Embedded** = INT8 inference on MCUs with peak SRAM ≤400KB (A) / ≤96KB (C), latency ≤10ms/event-window.
- **Trusted** = calibrated UQ (coverage within ±5%), attention-based XAI, graceful failover when uncertainty is high.

Primary deliverable (TRL-6). SHIELD firmware + tools that detect degradation early, forecast RUL with uncertainty, explain key contributors, and expose a clean API to host applications.

---

2) System Architecture (Five Layers)

1. **Input/Buffer (C/C++):** interrupt-driven circular buffer; change-point trigger (CUSUM/Page-Hinkley) for event-driven inference.
2. **Physics-Informed Residuals:** grey-box observers (ARX + KF/UKF) by default; PINN path when ODEs are known. Outputs residual stream + model params (bias/noise/latency, AR coeffs, state estimates).
3. **Self-Supervised Encoder:** frozen SSL encoder (contrastive) that maps residual windows → low-dim "degradation embeddings".

4. **Temporal Forecaster:** iTransformer‑NAS for Class‑A; TCN‑Attention‑NAS fallback for Class‑C. Both export attention for XAI.
5. **Probabilistic Decision:** QUTE early‑exit ensemble heads → mean RUL + CI; OOD / shift flags from uncertainty.

   Adaptation (background): replay‑buffered few‑shot updates (CL‑lite), gated by persistent high UQ.

Unified API. predict(window) → {rul_mean, rul_ci, attention, uq, flags}

---

3) Acceptance Metrics (used at every milestone)

Forecasting: RUL RMSE ≥15% better than baselines on C‑MAPSS; ≤25% loss post‑QAT.

Early Warning: detect slow drift ≥10–20 s before fault in HIL.

Uncertainty: Expected Calibration Error (ECE) ≤0.05; coverage ±5%.

OOD/Shift: AUROC ≥0.9 using UQ signals.

Embedded: peak SRAM ≤400KB (A) / ≤96KB (C); latency ≤10ms; ≤1mJ/inference.

Explainability: top‑k contributor stability under ±10% noise.

---

4) Work Packages (90‑day plan → TRL‑6)

WP‑0: Program Setup (Week 1)

- Metrics dashboards; repo layout; data governance; coding standards (C/C++, PyTorch).

WP‑1: TRL‑3 Foundations (Weeks 1–4)

- Residual Layer v1 (ARX+KF/UKF); simulators for bias/drift/noise/latency.
- SSL encoder (SimCLR/HNCPM variant) over residual windows.

- Baselines (DLinear, TCN) on C-MAPSS, PRONOSTIA; RMSE/ECE/AUROC reporting.

WP-2: Forecaster + UQ (Weeks 5–8)

- iTransformer-base + TCN-Attention-base; integrate QUTE K∈{2,4}.
- Reliability: calibration curves, coverage tests, OOD via covariate shift.

WP-3: NAS + Embedded (Weeks 9–12)

- TinyNAS search for Class-A and Class-C budgets.
- QAT to INT8; TinyEngine compile; latency/SRAM/energy profiling on H7/F411.
- Event-driven runtime (ISR buffers + trigger) integrated.

Milestone M1 (Day ~90): TRL-4 Alpha on device matching TRL-3 accuracy within 25%; real-time capable.

WP-4: HIL Validation (TRL-5) (Months 4–6)

- DAC/SPI loop; software fault-injection battery; end-to-end detection/forecast/UQ tests.
- Robustness: non-semantic noise, domain shift; uncertainty raises “degrade mode”.

Milestone M2: TRL-5 Report (<100ms detection; calibrated UQ; resource compliance).

WP-5: Field & Aging (TRL-6) (Months 7–12)

- Accelerated aging rig (bearing/motor or pump); run-to-failure capture.
- Deploy 5× prototypes on non-critical UW assets (HVAC, lab benches, robots).
- Few-shot specialization flow (≤20 windows) for each deployment.

Milestone M3: TRL-6 Report (successful RUL on ≥1 real fault; stability of XAI/UQ).

Deliverable: SHIELD-OS (lite): services for Self-Diagnosis, Adaptive Profiling, Failover.

---

5) Toolchain & Implementation Notes

- **Learning stack:** PyTorch + Lightning; ONNX export for TinyNAS/TinyEngine.
- **Embedded stack:** C/C++ firmware; TinyEngine; CMSIS‑DSP; static allocation only.
- **Triggers:** CUSUM/Page‑Hinkley; debounce; duty‑cycling.
- **Replay buffer:** FRAM/flash ring; ≤1% memory; per‑class prototypes.

Risk Controls

R1: iTransformer too large → auto‑fallback to TCN‑Attn‑NAS; same API.

R2: SSL transfer weak → augment with classical features + CADA DA; ablate.

R3: UQ drift → periodic temperature scaling on maintenance cycles.

R4: PINN instability → gate by identifiability; prefer grey‑box observer.

Roadmap to Autonomous Modality Recognition

Phase 0 — Scope & Guardrails (Decision Gate 0)

Goal: Lock definitions, targets, and constraints so all later choices are testable.

- **What:**
    - Level-1 goal: distinguish broad domains (thermal / mechanical / optical / acoustic / magnetic).
    - Level-2 goal: distinguish intra-domain (e.g., mic vs. IMU) when not aliased.
    - Embedded limits: MCU-class RAM <100 kB; inference <100 ms; energy per inference tracked.

  - Evidence model: calibrated probability + “Unknown” class via density thresholding.

- **Deliverables:** 1-page spec (taxonomy, metrics, acceptance thresholds, target MCUs).
- **Pass criteria:** Everyone can say “yes/no” to whether a model meets the spec.

Phase 1 — ModalityNet-v0.1 (Decision Gate 1)

Goal: Build the minimal, high-value dataset you need to learn & test modality signatures.

1. **Sensor set & DAQ design**
   - **Sensors:** 10–15 modalities; 8–12 units per modality (cross-device generalization).
   - **DAQ:** ≥16-bit (24-bit ideal) synchronous acquisition; fs per sensor ≥5× expected bandwidth.
   - **Common AFE practices:** known, logged gain stages; consistent shielding and grounding.
2. **Standardized stimuli (applied to all)**
   - **Quiescent:** shielded chamber to expose intrinsic noise floor.
   - **Micro-excitation:** low-amplitude swept sine / PRBS for transfer hints without “content”.
   - **Perturbations:** thermal ramp, EMI pulse, vibration pink noise (controlled levels).
3. **Capture protocol**
   - **Windows:** multiple lengths per take (e.g., 50 ms, 2 s, 30 s) to study time-to-confidence.
   - **Metadata:** modality label, device ID, AFE chain, fs, temp, environment notes.
4. **Data package**

- Raw signals + standardized npz/parquet; a tiny balanced eval split; data sheet.

- **Pass criteria:** At least 1,000 clips per modality per window length; identical stimulus timeline across sensors; reproducible capture scripts.

Phase 2 — Feature Lexicon & Streaming Estimators (Decision Gate 2)

Goal: Implement the physics-driven “fingerprint” features and make them MCU-friendly.

1. **Features to implement**
   - **PSD family:** log–log slope(s), band energy ratios, spectral centroid, flatness, entropy.
   - **Distributional:** skewness, kurtosis, quantile spread; Ljung–Box portmanteau.
   - **ADEV/OADEV surrogates:** online estimators for white/flicker/random-walk parameters (avoid full $O(N^2)$); fit slopes in $\log\sigma(\tau)$ vs. $\log\tau$ over a few τ decades.
   - **Optional:** wavelet band energies; AR order & poles via Burg (low-order).
2. **Engineering**
   - Streamed FFT (overlap-add) via CMSIS-DSP; fixed-point where possible.
   - Constant-memory online stats (no ring buffers longer than needed).
   - Unit tests with synthetic processes (white, 1/f, $1/f^2$) to validate estimates.

- **Deliverables:** Python reference + C/CMSIS implementations + tests; feature cards (stability vs. window length, SNR sensitivity).

- **Pass criteria:** Feature compute fits RAM/CPU budgets; correlations match theory on synthetic data.

Phase 3 — Baseline Classifier: AutoModality-Lite (Decision Gate 3)

Goal: Establish a strong, interpretable TinyML baseline before deep/SSL.

1. **Models:** RF, SVM, and GMM (for unknown detection).
2. **Training:** 5×CV, device-held-out folds (ensure cross-device generalization).
3. **Thresholding:** Calibrated probabilities (Platt/temperature scaling); set OOD threshold on GMM density or SVM margin.
4. **Ablations:**
    - Feature group drops (PSD-only vs. PSD+stats vs. +ADEV).
    - Window length (50 ms vs. 2 s vs. 30 s).
    - Quiescent vs. micro-excited vs. perturbed.

- **Deliverables:** Confusion matrices (Level-1/Level-2), ROC for “Unknown”, feature-importance plots.
- **Pass criteria:** Meet Level-1 accuracy & macro-F1 targets on device-held-out split; OOD AUROC ≥ 0.9 on synthetic “novel” modality mixes.

Phase 4 — SSL Encoder: Sensor-IDNet-SSL (Decision Gate 4)

Goal: Learn a compact, general representation from unlabeled data and keep TinyML viability.

1. **Pretraining (offline)**
    - **Backbone:** small 1D-CNN with depthwise separables; or 2D-CNN on log-mel spectrograms (if memory allows).
    - **SSL:** contrastive (SimCLR-style) and/or masked autoencoding; augmentations = time-crop, small time-warp, mild noise/EMI injection; **no** augmentations that erase noise color.

    - **Embedding:** 32–64 dims.

2. **Heads & evaluation**
    - Linear probe (softmax), kNN, and GMM over embeddings.
    - t-SNE/UMAP visuals for cluster separability across modalities and devices.
3. **Compression**
    - Post-training quantization + layer fusion; prune channels; distill into a micro-backbone.

- **Deliverables:** Embedding benchmarks vs. Phase-3; TinyML-ready quantized model.
- **Pass criteria:** ≥5–10 pp macro-F1 improvement over baseline under domain shift; “Unknown” detection maintained.

Phase 5 — Embedded Integration & Profiling (Decision Gate 5)

Goal: Put both paths on target MCUs and prove real constraints are met.

1. **Pipelines**
    - **Path A (Lite):** acquire → features (C/CMSIS) → GMM/SVM.
    - **Path B (SSL):** acquire → tiny encoder (int8) → SVM/kNN/GMM head.
2. **Profiling**
    - RAM/flash usage, inference time, end-to-end time-to-answer (dominated by acquisition).
    - Energy per inference (shunt or on-board PMIC telemetry).
    - Stress tests: temperature drift, voltage sag, EMI.
3. **Robustness**

    - Noise sweeps, sampling-rate drift, missing data, clipping; adversarial alias cases.

- **Deliverables:** Profiling tables; failure/edge-case report; optimization commits.

- **Pass criteria:** Spec met for RAM/latency/energy; graceful “Unknown/Abstain” in alias cases.

Phase 6 — Ambiguity Resolution Experiments (Decision Gate 6)

Goal: Empirically chart when the problem is unsolvable and how to mitigate.

- **Aliased pairs:** e.g., piezoresistive mic vs. accel with similar AFEs.

- **Mitigations:** require quiescent windows; inject standardized micro-excitation; multi-window voting; confidence-aware abstention.

- **Output:** Modality separability map (by SNR, window length, environment).

- **Pass criteria:** Clear guidance for when the system must abstain or request a micro-excitation.

Phase 7 — Validation, Packaging, and Paper Kit (Decision Gate 7)

Goal: Make it reproducible, review-proof, and ready to ship.

- **Artifacts:**

    - ModalityNet-v0.1 (cleaned, documented), data card, and ethics note.

    - Open-source feature library (Python + C) with tests.

    - TinyML firmware demos for both paths.

    - Benchmark report (accuracy, F1, OOD AUROC, RAM/latency/energy).

    - Negative results on aliasing (transparent and valuable).

- **Pass criteria:** One-click repro for figures/tables; internal replication on a second MCU.