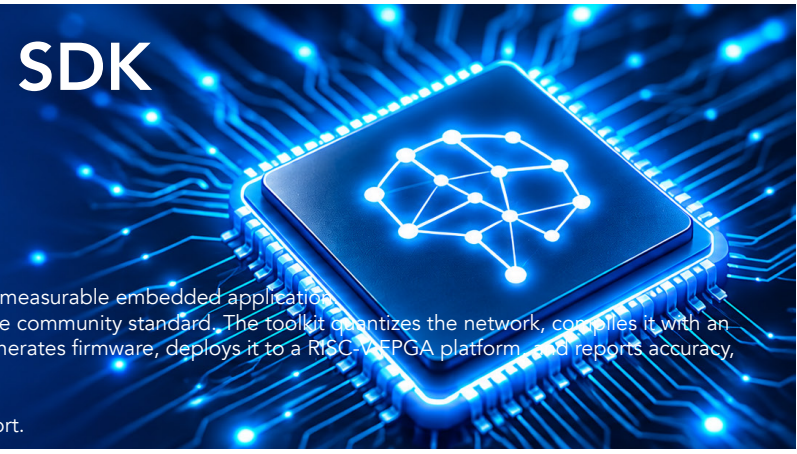
INTERA
SEAMLESS AI

INTERA Neuromorphic SDK

Product Brief

From trained spiking neural networks to measured RISC-V hardware results



The toolkit turns a trained spiking neural network (SNN) into a deployable, measurable embedded application. Models enter through NIR (Neuromorphic Intermediate Representation), the community standard. The toolkit quantizes the network, compiles it with an MLIR-based compiler, maps it onto INTERA neuromorphic accelerators, generates firmware, deploys it to a RISC-V FPGA platform, and reports accuracy, latency, and memory use.

It is a compiler, not a runtime interpreter: there is no inference engine to port.

From spiking neural network to validated hardware deployment

The INTERA Neuromorphic Toolkit: an open, automated pathway from SNN development to embedded neuromorphic deployment.

- Open standards (NIR)
- Open-source core
- Generated firmware
- Automatic optimization

SNN Development

- snnTorch
- LAVA-DL
- Other NIR-compatible frameworks

INTERA Neuromorphic Toolkit
One automated workflow

1. NIR-based workflow
2. Automatic quantization
3. SNN-MLIR compilation
4. Build automation

Deployment Targets

- RISC-V platforms
- FPGA Platform
- Neuromorphic accelerators: Eagle and Eagle Spiking

Validation and Benchmarking

- Accuracy analysis
- Performance metrics: latency and cycles
- Memory footprint
- Interactive HTML reports

- Hardware-ready deployment
- One-command build, flash and run
- Reproducible benchmarking
- Research-to-product workflow

INTERA provides a complete, open, and automated pathway from SNN to embedded neuromorphic deployment.

From manual integration to a single flow

Manual Workflow	With the INTERA Neuromorphic Toolkit
Framework-specific, proprietary model formats	Portable models. NIR-native input: SNNTorch, LAVA-DL, NENGO, NORSE, and any other NIR-compatible frameworks enter without a proprietary exporter.
Manual quantization and fixed-point tuning	Identical numerics on CPU and accelerator. Automatic INT8 and fixed-point quantization with bit-exact integer execution.
Hand-written neuron kernels and firmware	No runtime to port. The compiler generates bare-metal C and RISC-V objects, with no runtime interpreter. Each layer is placed automatically on the CPU or the accelerator.
Hand-tuned memory maps and linker scripts	Faster board bring-up. Memory layout, linker script, and firmware are generated per board and per model from a hardware description card.
Efficient accelerator memory use by hand	Automatic buffer assessment: Compilers manage automatically memory buffer and data allocation efficiently.
Complex FPGA deployment flows	Simple deployment. One-command build, flash, and run with OpenOCD over JTAG. No vendor IDE.
Ad-hoc benchmarking	Measured evidence. Spike-level accuracy against the reference model, cycle counts per stage (input copy, compute, neurons), accelerator-versus-CPU speed-up measured on hardware, and a memory footprint map from the real linker output.

C/Sabino de Arana 14-16, 2nd - Barcelona, Spain - 08028 info@intera-group.com www.intera-group.com

INTERA
SEAMLESS AI

INTERA Neuromorphic SDK

Product Brief



Supported models and targets

SNN model support

Feed-forward and recurrent SNNs
Neuron models: LIF, CubaLIF, LI, CubaLI, IF, I
Convolutional SNN layers (Conv1D/2D, pooling)

SNN model support

RISC-V, aarch64 (ARM) and native x86 targets
FPGA boards: PYNQ-Z2 and ICV32DK (SPICA)
Board-agnostic: a hardware description card defines each target

Supported INTERA AI Accelerators

AI Accelerator	Architecture	Operation Accelerated	Processing elements (PE) and scalability	What runs where
Eagle	Near-memory computing	MxV / MxM(1)	4, 16 PEs; tested up to 32 banks/ PEs. Configurable 2 to 32-bits	Synaptic matrix-vector products run in accelerator memory; neuron dynamics and spike generation run on the CPU.
Eagle Spiking*	Near-memory computing	MxV / MxM(1), LIF/CubaLIF activation	4, 16 PEs; tested up to 32 banks/ PEs. Configurable 2 to 32-bits(2)	Near-memory neuromorphic AI accelerator. The full layer (synapses, bias, neuron dynamics, spikes) runs in hardware, with neuron state kept in accelerator memory across timesteps.

(1) Indirect MxM multiplication by software arrangement.
(*) Under development.
(2) The toolkit currently quantizes weights to INT8 and uses fixed-point neuron state.

The INTERA Neuromorphic SDK currently supports Eagle and, and Eagle Spiking (under development). Falcon, Hawk Advance, Hawk Pro and Harrier are supported by the INTERA TinyML toolkit.

Open standards, optimized hardware

INTERA believes neuromorphic computing grows on open standards, builds the toolkit on them, and publishes its core.

- **NIR-native:** NIR is backed by the neuromorphic research community. INTERA trains its own models with SNNTorch, one of the most widely used open SNN frameworks, and exports them through NIR.
- **Open-source SNN-MLIR:** The SNN dialect, the NIR-to-MLIR front end, and the CPU reference lowering are released under Apache-2.0 (GitHub, PyPI, full documentation, arXiv paper 2606.09213). The compiler is built on LLVM/MLIR, the infrastructure used across the AI and HPC industry.
- **Open where it matters, optimized where it counts:** The open core gives portability and transparency; INTERA's accelerator-specific backends add the hardware performance optimization on top.

Typical use cases

Neuromorphic edge prototyping

Go from an SNNTorch model to spiking inference on real RISC-V hardware in one reproducible flow.

Accelerator evaluation

Compare INTERA neuromorphic accelerators against a CPU-only baseline on the same model and board.

SNN benchmarking

Measure spike accuracy, latency, cycles, and memory footprint per model.

Research to product

Move a model from an open research format (NIR) to deployable firmware and validation reports without rewriting it.

C/Sabino de Arana 14-16, 2nd - Barcelona, Spain - 08028 info@intera-group.com www.intera-group.com