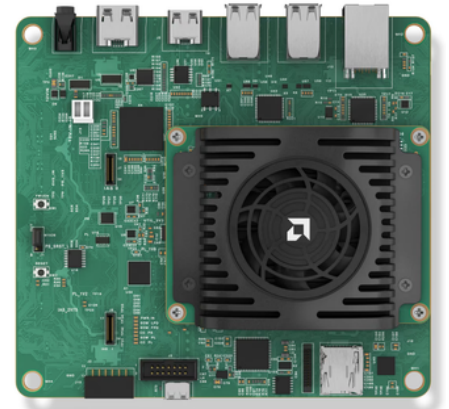


VL-KRIA

Zynq UltraScale+ MPSoC Based Vision AI Starter Kit



FEATURES

- The AMD Kria™ portfolio combines leadership AMD hardware and software for physical AI and other advanced embedded applications at the edge. The portfolio includes production-deployable SOMs and comprehensive evaluation kits
- The AMD Kria™ AI SOM provides the intelligent compute foundation for the next frontier of physical AI, combining CPU, GPU, NPU, and unified memory on a compact COM-HPC module optimized for robotics and autonomous machines. Available as a production-ready SOM or partner-validated SOM with carrier card, it enables scalable deployment and rapid system development. Built on the Kria AI SOM, the AMD Kria™ AI Robotics Developer Platform extends this foundation into the industry's first open, fully integrated platform for autonomous robotics, empowering developers to prototype, customize, and deploy real-time AI systems with accelerated
- Vision Ready
- Multi-camera Support: Up to 8 interfaces
- 3 MIPI sensor interfaces, USB cameras
- Built-in ISP component (OnSemi)
- HDMI™, DisplayPort™ outputs
- Flexible Connectivity
- 1 Gb Ethernet
- USB 3.0 / 2.0
- Expandable
- Extend to any sensor or interface
- Access Pmod ecosystem
- Accessible
- Low cost, enabling design exploration
- Available from AMD and distributors
- System Logic Cells : 256K
- Block RAM Blocks : 144
- UltraRAM Blocks : 64
- DSP Slices : 1.2K
- Ethernet Interface : One 10/100/1000 Mb/s
- DDR Memory : 4 GB (4 x 512 Mb x 16 bit) [non-ECC]
- Primary Boot Memory : 512 Mb QSPI
- Secondary Boot Memory : SDHC card
- Device Security : Zynq UltraScale+ MPSoC hardware root of trust (RoT) in support of secure boot. Infineon TPM2.0 in support of measured boot.
- Image Sensor Processor : OnSemi AP1302 ISP
- IAS MIPI Sensor Interfaces : x2 (supports auto-focus)
- Raspberry Pi Camera Interface : x1
- Pmod 12-pin Interface : x1
- USB® 3.0/2.0 Interface : x4
- DisplayPort™ 1.2a : x1
- HDMI™ 1.4 : x1
- 12V AC/DC wall mount adapter for the Kria™ KV260 Vision AI Starter Kit.

SOFTWARE

- **Ubuntu 22.04 (KV260 Image)** : Runs on KV260; Purpose: OS with Vitis AI runtime pre-installed.
- **Python 3 + OpenCV** : Runs on KV260; Purpose: AI experiments and image/data processing.
- **VS Code + Remote SSH** : Runs on Student PC; Purpose: Remote development environment for the KV260.
- **Vivado WebPACK Edition** : Runs on Student PC; Purpose: FPGA synthesis, implementation, and bitstream generation.
- **Vitis AI Docker (v3.5)** : Runs on Student PC; Purpose: Model quantization and compilation for DPU deployment.
- **Google Colab** : Runs on Browser; Purpose: All Stage 1 and Bridge notebook experiments.
- **Labellmg** : Runs on Student PC; Purpose: Dataset annotation for custom image experiments.

SPECIFICATIONS

- Device : Zynq™ UltraScale+™ MPSoC
- Form Factor : SOM + Carrier Card + Thermal Solution
- Starter Kit Dimensions : 119 mm x 140 mm x 36 mm
- Thermal Cooling Solution : Active (Fan + Heatsink)

EXPERIMENTS

GOOGLE COLAB — MASTER NOTEBOOK LIST

- REGRESSION
- CLASSIFICATION
- CLUSTERING
- DIMENSIONALITY REDUCTION
- DEEP LEARNING
- CLASSICAL AI (OPTIONAL)
- Hardware Colab Bridge
- Stage 2 experiments run directly on the AMD Kria KV260 board. They are delivered as step-by-step hardware lab sheets, not Colab notebooks. Each experiment references the corresponding Stage 1 concept so students see the direct connection between software and hardware.

Description & Stage 1 Link

H1. Platform Setup

H1.1

- KV260 orientation & SSH access
- DPU overlay, PS/PL architecture, verify Python and OpenCV are available.

H1.2

- Camera setup
- USB camera pipeline using OpenCV: capture, display, and save frames. Prerequisite for all vision-based experiments.

H2. Beginner AI Deployment

H2.1

- Image pre-processing with OpenCV
- Gaussian blur, Canny edge detection, and Sobel filters applied to live camera frames. Teaches how images are processed as arrays of numbers. Links to Stage 1 CNN concepts (notebook 5.2).

H2.2

- Live image classification on DPU (MobileNetV2)
- Run a pre-compiled MobileNetV2 model via the VART Python API. Live camera inference on the DPU in real time. Students learn how VART, XIR, and DPU subgraphs work. Direct link to the Bridge B1–B4 pipeline.

H2.3

- Static image inference (MobileNetV2)
- Run MobileNetV2 inference on a single static image. Print top predictions with confidence scores. Understand the full VART input/output pipeline without camera complexity.

H3. Intermediate — Benchmarking & Systems

H3.1

- CPU vs DPU inference benchmark
- Same model, compare latency and throughput on ARM Cortex-A53 vs. the on-chip DPU. Establishes why hardware acceleration matters.

H3.2

- Power consumption analysis
- On-chip SYSMON power analysis via Linux sysfs. Measure idle vs. CPU vs. DPU inference power draw. Tabulate and compare performance per watt.

H3.3

- YOLO multi-object detection
- Use a pre-compiled YOLOv5/v8 model from the AMD Vitis AI Model Zoo. Real-time detection via USB camera.

H3.4

- PMOD HYGRO temperature and humidity acquisition over I2C
- Read temperature and humidity data from the PMOD HYGRO sensor. Log and plot data on the ARM processor. Foundation for H5 applications.

H3.5

- PMOD ACL2 vibration anomaly detection on ARM
- Read accelerometer data from PMOD ACL2. Apply a threshold classifier to detect abnormal vibration patterns. Predictive maintenance use case. Links to Stage 1 notebook 5.3 (RNN/LSTM).

4. AI-Accelerated Hardware (Verilog + DPU)

H4.1

- LED blinker in Verilog
- First Vivado design flow end-to-end: write Verilog, synthesize, generate a bitstream, load it onto the board. Proves the PL fabric works while the DPU overlay runs simultaneously on the PS.

H4.2

- AXI-Lite GPIO counter and PS–PL communication
- A Verilog counter exposed via AXI-Lite GPIO. An ARM Python script reads the counter value over the AXI bus, proving PS and PL can communicate.

H4.3

- Hardware confidence thresholding
- Camera captures a frame and the DPU produces a classification confidence score. Python writes the score to a Verilog comparator over AXI-GPIO; the comparator checks it against a threshold register and drives an LED — confident predictions light it, uncertain ones don't. Completes the sensor → AI → hardware-decision loop in a single bitstream alongside the DPU.

H4.4

- Grayscale conversion IP
- Pixel-level Verilog IP packaged as AXI-Stream. Live camera frames pass through the Verilog grayscale IP and output to HDMI.

H4.5

- DPU inference result drives PWM via Verilog IP
- The DPU confidence score is sent to a Verilog PWM IP via AXI. The PWM duty cycle changes based on the AI output — closing the sensor-to-AI-to-actuator loop.