Open Standards Powering the Future of Embedded Compute and Vision Applications

Embedded World
April 2024

Laurent Pinchart
Founder & CEO, Ideas on Board
Kamaros Working Group Member
Laurent Pinchart is the founder and CEO of Ideas on Board, a company specialized in delivering camera and display solutions for Linux across all markets.

Making cameras work

With 20 years of experience as a Linux kernel developer and maintainer, Laurent has driven the design of the Linux kernel camera API and has participated in multiple industry working groups to standardize camera protocols. Most recently, he has started the libcamera® project to give Linux a full camera stack in collaboration with silicon vendors and OEMs.
Topics

• Introduction to Khronos and its family of open standard APIs
• Overview of Khronos APIs for Parallel Computation
• Introduction to the Kamaros Embedded Camera API in development
• How to get involved!
Khronos Connects Software to Silicon

Founded in 2000
~ 200 Members | ~ 40% US, 30% Europe, 30% Asia
The Need for Parallel Processing

Single Processor
Simple to program
but
May not provide enough performance
especially
as Moore’s Law frequency/power scaling is slowing

Multi-Processor
Additional processors can process expanded workloads
but
Adds complexity to system design and programming, as there is the need to:
(i) Distribute workload across processors
(ii) Synchronize use of compute and memory resources
(iii) Communicate intermediate data and results

Can open standards help solve this complexity?
Khronos Active Standards

3D Graphics
Desktop, Mobile and Web
- Vulkan
- ANARI
- OpenGL
- OpenGL ES
- WebGL

3D Assets
Authoring and Delivery
- COLLADA
- glTF
- KTX

Portable XR
Augmented and Virtual Reality
- OpenXR

Parallel Computation
Vision, Camera, Inferencing, Machine Learning
- OpenCL
- OpenVX
- NNEF
- SYCL
- SPIR

Open Standard APIs in Embedded Markets
- Cross-platform software reusability
- Decoupled software and hardware for easier development and integration of new components
- Cross-generation reusability and field upgradability
- Industry-wide ecosystem of tools and libraries

Safety Critical APIs
Khronos Compute Acceleration Standards

Higher-level Languages and APIs
Streamlined development and performance portability

Single source C++ programming with compute acceleration
Neural Network Exchange Format Trained Networks
Graph-based vision and inferencing acceleration
Third party vision, streaming and inferencing libraries

Applications, libraries, and higher-level languages and APIs use lower-level Khronos standards to access hardware acceleration

Lower-level Languages and APIs
Explicit hardware control

GPU rendering + compute acceleration
Intermediate Representation (IR) language compiler target supporting parallel execution and graphics
Heterogeneous compute acceleration

Multiple programming abstractions to meet the needs of diverse software stack architectures

Shaders
Kernels

GPU

CPU
GPU
FPGA
DSP
AI/Tensor HW
Custom Hardware
The Need for a Camera System API Standard

Increasing Sensor Diversity
Including camera arrays and depth sensors such as Lidar

Multiple Sensors Per System
Synchronization and coordination become essential

The cost and time to integrate and utilize sensors in embedded systems has become a major constraint on innovation and efficiency in the embedded vision market

Increasing Sensor Processing Demands
Including inferencing. Sensor outputs need to be flexibly and efficiently generated and streamed into acceleration processors

Proprietary APIs Hinder Innovation
Vendor-specific APIs to control cameras, sensors and close-to-sensor ISPs prevent access of full camera capabilities
Primary Kamaros Design Influences

Kamaros will fill an ecosystem gap for a cross-vendor embedded camera system API
May complement existing APIs
e.g., be used in their implementation, or be implemented over them

- Android Camera2 Framework API
- GenICam
  Hosted by EMVA including GenTL, SFNC (Standard Features Naming Convention) and PFNC (pixel format naming convention)
- libArgus API for acquiring images and associated metadata from cameras
- Open-source camera stack and framework for Linux, Android, and ChromeOS

This work is licensed under a Creative Commons Attribution 4.0 International License
What is Kamaros?

EMBEDDED CAMERA SYSTEM API

Jointly promoted by Khronos and the European Machine Vision Association (EMVA), the Kamaros API Working Group is now developing an open, royalty-free standard for controlling camera system runtimes in embedded, mobile, industrial, XR, automotive, and scientific markets.
Need for Embedded Camera API Standard

An effective open, cross-vendor open standard for camera, sensor and ISP control is a major ecosystem gap compared to processing APIs

Benefits of a Cross-vendor Embedded Camera API

- Portability of camera/sensor code for easier system integration of new sensors
- Preservation of application code across multiple generations of cameras and sensors
- Sophisticated control over sensor stream generation increases effectiveness of downstream accelerated processes

An effective camera API abstraction will enable sensor vendors to expose fuller hardware capabilities without exposing proprietary implementation details while gaining access to a larger ecosystem of libraries and applications
Typical Kamaros Software Stack

- **Frameworks & Middleware**
  - GStreamer
  - OpenVX
  - Libraries

- **Kamaros API**
  - Kamaros System Runtime

- **Transport**
  - CSI-2
  - USB
  - Ethernet

- **Physical Devices**
  - Sensors
  - Lenses
  - Lights
  - Processors

- **OS**
  - Windows
  - Linux
  - Android
  - Chrome
  - RTOS

**API in scope**

**Some libraries may be in scope**

*Names of transport layers, framework and operating systems are illustrative examples*
Kamaros Portable Application Structure

1. Discover Camera
   - Select Camera via Kamaros Loader
     - Data provided on physical or geographic location, sensor and lens etc.
   - Query Pipeline Templates for selected Camera
     - Select Pipeline that has Controls and capability limits that satisfy use case

2. Discover Pipeline
   - Request Pipeline Configuration
     - Controls set in priority order
   - Best Attempt Pipeline Configuration
     - Achievable Control Values, Defaults and Constraints

3. Create Pipeline
   - Create Pipeline with initial Controls
     - Allocate any needed resources e.g., buffers

4. Capture Frames
   - Sets Controls per Frame
     - Synchronize for Frame availability
   - Process Returned Frame
     - Per frame metadata provides applied settings and errors
   - Stop Pipeline
     - Free resources

Pipeline Templates enable flexible and scalable exposure of camera capabilities

Application

This work is licensed under a Creative Commons Attribution 4.0 International License
Kamaros Working Group Organization

Any company is welcome to join Khronos to participate in any Khronos Working Groups.

EMVA Members can reach out to EMVA for details on joining the Kamaros Advisory panel at no charge.
Get Involved!

Any company is welcome to join Khronos to influence standards development

https://www.khronos.org/members/ or email memberservices@khronosgroup.org

More information on any Khronos APIs

https://www.khronos.org/

Khronos members can participate in the Kamaros Camera Working Group
EMVA Members can join the Kamaros Advisory panel

https://www.khronos.org/kamaros

Khronos is developing a growing family of open, royalty-free API standards relevant to embedded and safety-critical markets