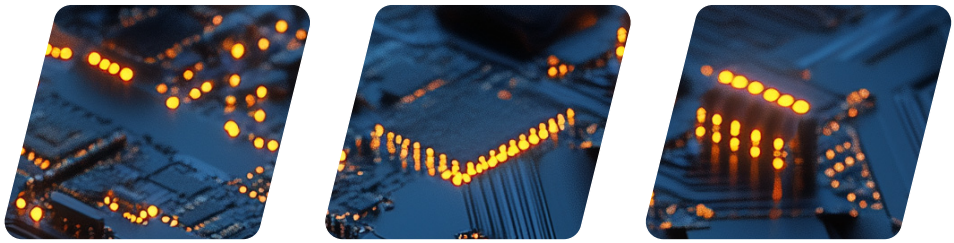
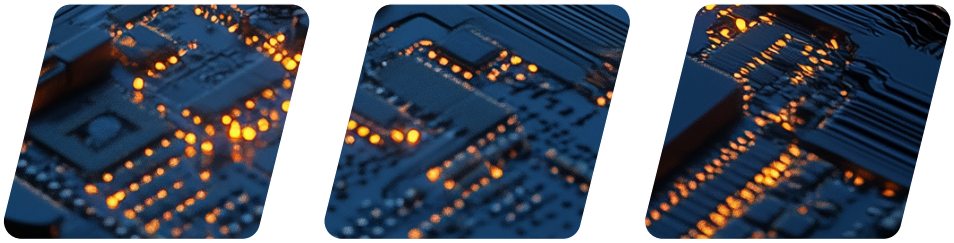


EEVideo: Open-Source Embedded Ethernet Video Protocol for Lattice FPGAs



White Paper

Authors:

Barry Walker, VP of Engineering, Tecphos
Karl Wachswender, Senior Principal System Architect, Industrial, Lattice Semiconductor

DISCLAIMERS

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. Lattice products and services are not designed, manufactured, or tested for use in life or safety critical systems, hazardous environments, or any other environments requiring fail-safe performance, including any application in which the failure of the product or service could lead to death, personal injury, severe property damage or environmental harm (collectively, "high-risk uses"). Further, buyer must take prudent steps to protect against product and service failures, including providing appropriate redundancies, fail-safe features, and/or shut-down mechanisms. Lattice expressly disclaims any express or implied warranty of fitness of the products or services for high-risk uses. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

INCLUSIVE LANGUAGE

This document was created consistent with Lattice Semiconductor's inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice's inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

ABSTRACT

Embedded vision systems increasingly require efficient, low-latency video transport as camera resolutions, frame rates, and edge AI workloads continue to grow. This white paper introduces EEVideo, an open-source Embedded Ethernet Video protocol developed by Tecphos to deliver real-time video and sensor data over standard Ethernet/IP networks while maintaining a compact footprint suitable for low power FPGA implementations. Built on Ethernet, IP, UDP, and CoAP, EEVideo combines a lightweight streaming protocol with a compact management layer to support device discovery, configuration, monitoring, and low-latency video transport. The paper describes the protocol architecture, host-side software tools, and implementation guidance for Lattice FPGA platforms, including resource utilization, latency, throughput, and application examples across industrial machine vision, robotics, automotive ADAS, medical imaging, and emerging edge AI systems. It also highlights how the EEVideo open-source model and FPGA-efficient design can help developers accelerate embedded vision system development while reducing protocol complexity, latency, and resource usage.

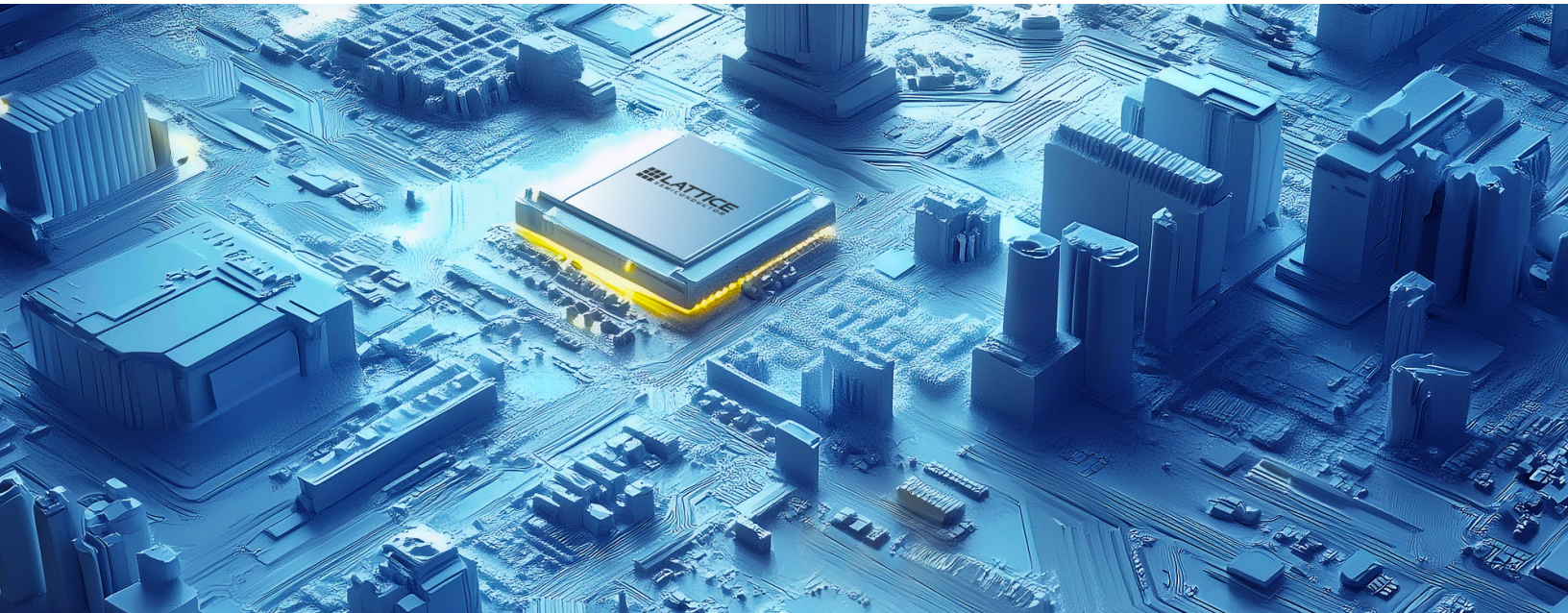


TABLE OF CONTENTS

Introduction

Machine vision continues to drive innovation across industrial automation, robotics, automotive Advanced Driver-Assistance Systems (ADAS), medical imaging, and edge AI applications. As camera resolutions and frame rates increase, the need for efficient, low-latency video transport from embedded sensors to processing systems has never been greater. Traditional video interfaces such as MIPI CSI-2 or LVDS excel at short-range, point-to-point connections but become impractical for distributed systems or longer cable runs. This is leading to widespread adoption of Ethernet-based solutions that leverage ubiquitous, standards-compliant networking infrastructure. Embedded Ethernet Video (EEVideo) is an open-source protocol developed by Tecphos Inc. specifically to address the unique demands of embedded machine-vision applications. It enables real-time delivery of video and sensor data over standard Ethernet/Internet Protocol (IP) networks while maintaining an exceptionally small footprint suitable for low power FPGA implementations.

Design Goals

Several Ethernet video protocols are available today, each optimized for different priorities and use cases. Understanding how they compare helps identify the best fit for resource-constrained edge environments. See Table 1.

Table 1: Key Differences Among Ethernet Video Protocols

	PROTOCOL		
Feature	**GigE Vision**	**Holoscan**	**RTP/RTSP**
Latency	Low	Low	Medium to High (requires frame buffering)
Protocol Complexity	High	High	High
FPGA Resource Usage	High (10k LUTs)	High (>10k LUTs)	High
Management Interface	Complex (GeniCam)	Complex (Holoscan)	Limited
Overhead	Medium to high	Medium to high	Medium
Standard Licensing/ Licensor	Licensed by A3	Licensed by NVIDIA	Open Source
Hardware	Flexible	NVIDIA only	Mostly PCs
Host Software	GenICam proprietary	NVIDIA/InfiniBand	NA
Other	NA	NA	Compression, motion artifacts

Proprietary solutions are also available. Well-documented, standards-based options offer distinct advantages, including flexible licensing, broad interoperability, and freedom from vendor lock-in. Together, these benefits accelerate innovation and help manage development costs. These advantages are especially valuable in Lattice FPGA deployments. There, designers balance high-performance video pipelines with efficient use of power, logic, and memory. This balance is essential on platforms such as Lattice CrossLink™-NX, Lattice CertusPro™-NX, Lattice Avant™, and Lattice Nexus™ 2.

The EEVideo Solution

EEVideo was created to overcome these barriers through a deliberate focus on simplicity, efficiency, and openness. The protocol delivers raw or low-latency compressed video with minimal processing overhead on the device side, built on proven standards: Ethernet (IEEE 802.3), IP, User Datagram Protocol (UDP), and the IETF Constrained Application Protocol (CoAP).

Key advantages for Lattice FPGA users include:

- Ultra-low latency through on-the-fly packetization with no mandatory frame buffering.
- Minimal resource usage, freeing FPGA fabric for additional processing such as Lattice sensAI™ inference, image enhancement, or custom control logic.
- Flexible deployment: devices can operate in a simple “headless” streaming mode or take full advantage of CoAP-based discovery, configuration, monitoring, and security.

- Security via CoAP's Datagram Transport Layer Security (DTLS).
- True open source, MIT-licensed specification and reference implementations accelerate adoption and community contributions.

By combining a lightweight streaming protocol with a powerful yet compact management layer, EEVideo provides a modern, future-proof alternative that aligns perfectly with Lattice's leadership in high-efficiency, low power FPGAs for edge vision applications.

This white paper details the EEVideo protocol architecture and demonstrates its practical implementation on Lattice platforms, including integration guidance, resource benchmarks, and real-world use cases.

Overview

EEVideo is designed to deliver low-latency video and sensor data streams over standard Ethernet networks. It targets embedded machine-vision applications where simplicity, efficiency, and minimal FPGA resource usage are critical. By leveraging widely available Ethernet hardware and standard Internet protocols, EEVideo enables straightforward integration into Lattice FPGA-based camera and sensor systems while maintaining compatibility with modern IP networks.

The latest version of the full specification is available at eevideo.tecphos.com.

Design Goals

The protocol was developed with the following priorities, which make it especially well-suited for Lattice low power FPGA platforms:

- **Standards-based:** Built on existing standards (Ethernet, IP, UDP, CoAP) to accelerate adoption and interoperability.
- **Low Complexity:** Simplified operation on the sensor/FPGA side, assuming the receiving host has significantly more processing capability.
- **High Efficiency:** Very low protocol overhead to maximize usable bandwidth for video payload.
- **Low Latency:** Support for raw or lightly compressed video with no mandatory frame buffering. Data moves from sensor to network with minimal processing on the device side.
- **Flexible Formats:** Broad support for varying resolutions, pixel formats, frame rates, and color spaces common in machine vision.

Structure

EEVideo is deliberately split into two largely independent components, the Management Protocol and the Streaming Protocol. This separation allows implementers to use only the parts required for their application, keeping the footprint small on resource-constrained devices.

Management Protocol

The Management Protocol enables device discovery, configuration, monitoring, and control. It is built on the IETF CoAP (RFC 7252), providing a lightweight RESTful interface suitable for embedded systems. CoAP transactions use UDP by default, prioritizing speed and efficiency.

Key characteristics of the Management Protocol include:

- **Flexible Deployment:** A minimal EEVideo device can begin streaming without any management interaction on a pre-configured network. More advanced use cases leverage CoAP for runtime control, parameter adjustment, and status reporting.
- **Register-Based Interface:** All configuration and monitoring occurs through registers. This unified model simplifies both hardware implementation in the FPGA and software interaction on the host.
- **Low Bandwidth Usage:** Management traffic is negligible compared to the video stream and does not impact real-time performance.

- **Device Discovery:** Device discovery requests are multicast by the host, and the responses are generated automatically, using a more IP-native model than the mechanisms used by some other video transports.
- **Feature Discovery:** Devices expose supported features through a standardized discovery mechanism. Hosts can dynamically determine capabilities without prior knowledge of the specific device. The flexibility to implement only needed features allows for easier implementation on small systems.
- **Security:** The established security protocols of CoAP DTLS provide options for encryption, authentication, and integrity.

Management Protocol Implementation

The CoAP-based management layer uses a 32-bit register bank. All configuration, status reporting, and feature discovery are handled through registers. While registers are mapped into a 32-bit address space, most implementations will not use the full range. Devices can operate in “headless” mode (pre-configured streaming with little or no management) or fully managed mode. At the minimal conformance level, only register 0 (Device Capabilities) must be implemented; full CoAP support is not required.

CoAP transactions are handled by a compact state machine or soft-core processor implementation that services GET/PUT transactions over UDP. This makes the management layer efficient and easy to implement in Lattice FPGAs.

Hosts may discover devices through a standardized discovery mechanism using the IETF Constrained RESTful Environments (CoRE) Link Format (RFC 6690). Hosts multicast a CoAP discovery request, and the responses are generated automatically by the device. The host may then read pre-configured registers to discover the implemented features of the device. This allows the host to build a map of the device’s features for monitoring and control.

The EEVideo Specification defines a set of standard features for embedded vision systems and supports custom device capabilities. Standard features include:

- **System:** device identification, register-detail embedded files and URLs, watchdog functionality, and reset and sleep controls
- **Ethernet:** interface configuration, PHY control, and packet retransmission
- **Stream:** video stream specifications and parametric pixel formats
- **Source:** MIPI monitoring information and test-pattern generation
- **Processing:** color correction control, gamma control, compression control
- **Timing:** timestamp support, including NTP and PTP, GPS synchronization, timers, and triggers
- **Peripherals:** I2C, SPI, UART, SMI, and CAN controllers, PWM and GPIO ports

While not technically part of the Management Protocol, EEVideo provides several options to establish IP device addresses: Static IPs, Link-Local assignment, DHCP, or dynamic assignment by the EEVideo host.

This protocol design maps efficiently onto Lattice’s distributed memory and small LUT resources, keeping the management overhead negligible.

Management Flow

All management functions are register transactions. Hosts always deliver register requests, and devices are only responsible for providing responses to the requests.

Host-device interactions typically follow this sequence:

- Host discovers device and uses the response to read register 0 (Device Capabilities).
- Following register 0 is a binary list of features available in the device along with address pointers to interact with the features.
- Based on the feature list, the host builds a register map for the device.
- Host configures stream parameters via register writes.

- Host starts the stream.
- The host may read status registers or trigger actions anytime during the setup and/or streaming.

Streaming Protocol

The Streaming Protocol is the high-performance core of EEVideo, optimized for transporting video data with minimal overhead and latency.

It uses a lightweight, custom packet format carried over IP/UDP. The protocol defines three primary packet types to efficiently handle video frames:

- **Start of Frame (SoF) packets:** Announce the beginning of a new frame and carry format metadata.
- **Pixel Data packets:** Deliver the bulk of the image data (supports variable-length payloads).
- **End of Frame (EoF) packets:** Signal the completion of a frame and provide checksum or status information.

Multiple header variants allow trade-offs between overhead and feature richness depending on the application. This design enables efficient packing of video data while supporting essential machine-vision requirements such as timestamping, frame sequencing, and error detection.

The streaming protocol is intentionally stateless and unidirectional for maximum efficiency. It does not require complex handshaking during active streaming, further reducing latency and FPGA resource utilization.

EEVideo devices may optionally support legacy streaming formats. This hybrid approach allows compatibility with existing tools (such as legacy GigE Vision clients), while still benefiting from EEVideo's powerful management layer.

Streaming Protocol Implementation

The streaming path is designed for maximum efficiency and minimal latency:

- **Packet Engine:** Converts incoming pixel streams into SoF, Pixel Data, and EoF packets using short, medium, or long header variants as configured via registers.
- **Header Generation:** Supports variable-length headers to balance overhead and feature support (e.g., timestamps, sequence numbers, checksums).
- **No Required Frame Buffering:** Pixels are packetized on-the-fly as they arrive from the sensor interface, enabling sub-frame latency.
- **Ethernet Integration:** Packets are handed directly to an internal Ethernet MAC. Current designs support standard 1 G/2.5 G Ethernet and are extensible to 10 G+ on higher-speed FPGAs.
- **Ethernet Infrastructure:** EEVideo can support any Ethernet bandwidth or Physical interconnect. Streams may be routed across off-the-shelf Ethernet hardware using any interface from Single Pair Ethernet automotive chipsets to long-range fiber optic interfaces.

Streaming Protocol Packet Formats

The streaming engine implements the three core packet types defined in the EEVideo specification with flexible header variants (Short, Medium, Long) to optimize overhead vs. feature support. All packet headers start with a 1-byte packet format type, followed by optional flag byte(s), Frame and Packet IDs. SoF packet headers have additional fields for video format information. Flags are used to signal packet status and device-specific information. See Table 2.

Table 2: Packet Header Overhead Comparison

Packet Type	Variant	Frame ID (Bytes)	Packet ID (Bytes)	Additional Fields Usage	Total Header Size (Bytes)	Typical Use Case
SoF	Short	3	4	Pixel Format, Timestamp, Resolution, Offset	24	Minimal-overhead streams
	Medium	6	8	+ Flags, increased field ranges	40	Standard machine vision
	Long	8	8	+ Additional Flags, increased field ranges	52	High-feature/ debug modes
Pixel Data	Short	3	4	None	8	N/A
	Medium	6	8	+ Flags	16	N/A
	Long	8	8	+ Additional Flags	20	N/A
EoF	Short	3	4	None	8	N/A
	Medium	6	8	+ Flags	16	N/A
	Long	8	8	+ Additional Flags	20	N/A

Streaming Protocol Alternative Formats

By modifying a small number of header fields, EEVideo can support other legacy and emerging packet formats when the application requires cross-standard functionality.

Host Side Software

To support cost effective embedded video systems, Tecphos released several EEVideo Host software resources under the MIT open-source license to simplify discovery, control, streaming, and integration.

These tools are written in the Go language, so they can be built and run with high performance on a wide range of platforms, including:

- Windows 10/11 PCs
- Linux AMD64 PCs
- Linux ARM64 (Including Jetson and RPi)
- macOS (Intel and ARM)

All source code is available at gitlab.com/eevideo; the tools can be ported to other languages for specific applications.

Protocol Library (goEEVideo)

The foundation of all host-side software is the goEEVideo protocol library, available at goeevideo.tecphos.com. This library offers API functions to support the full EEVideo specification, including:

- CoAP-based management
- Device discovery via CoRE Link Format
- Stream Control
- Real-time parsing of the lightweight streaming protocol (SoF, Pixel Data, and EoF packets)
- High level SPI and I2C peripheral control
- In-system reprogramming
- Single frame image capture

Command Line Utility (eevid)

The eevid command-line tool, available at eevid.tecphos.com, offers a powerful and easy-to-use interface for interacting with EEVideo devices directly from a terminal.

Key commands include:

- **eevid disc:** Performs network discovery of EEVideo devices using multicast CoAP and generates detailed device description files.
- **eevid reg:** Provides full read/write access to the device's 32-bit register model by address or symbolic name.
- **eevid cap:** Captures video frames or streams from a specified device, supporting raw and compressed formats with flexible output options.
- **eevid prog:** System update functionality.

Python-like Script Parser (eevlark)

eevlark, available at eevlark.tecphos.com, is a powerful command-line scripting utility for configuring, controlling, and monitoring EEVideo devices. It executes scripts written in a Python-like syntax based on Starlark, providing a familiar and flexible environment for automation and testing.

Written in Go and built on the goEEVideo library, eevlark is distributed as both source code and precompiled binaries for multiple operating systems. It requires device description files generated by the eevid disc command.

Key capabilities include:

- Register read/write access using read_register() and write_register()
- I2C peripheral control (read_i2c, write_i2c, etc.)
- Streaming control with stream_start(), stream_stop(), and stream_capture()
- Additional utilities such as sleep() and keyboard input

eevlark is particularly valuable for rapid development and system integration on Lattice FPGA-based EEVideo devices.

Video Display (eeview)

eeview, available at gitlab.com/eevideo/eeview, is a lightweight Embedded Ethernet Video stream viewer utility. It configures a GStreamer pipeline to receive and decode EEVideo streams, then displays the live video in a graphical window.

eeview relies on the goEEVideo library to manage the sensor and a GStreamer installation to actually display the stream.

eeview provides a simple and effective way to visually validate video streams from EEVideo devices in real time. It is a tool for development, debugging, and quick demonstration of camera functionality on Lattice FPGA platforms.

eeview offers a starting point for developing SoC GStreamer processing pipelines.

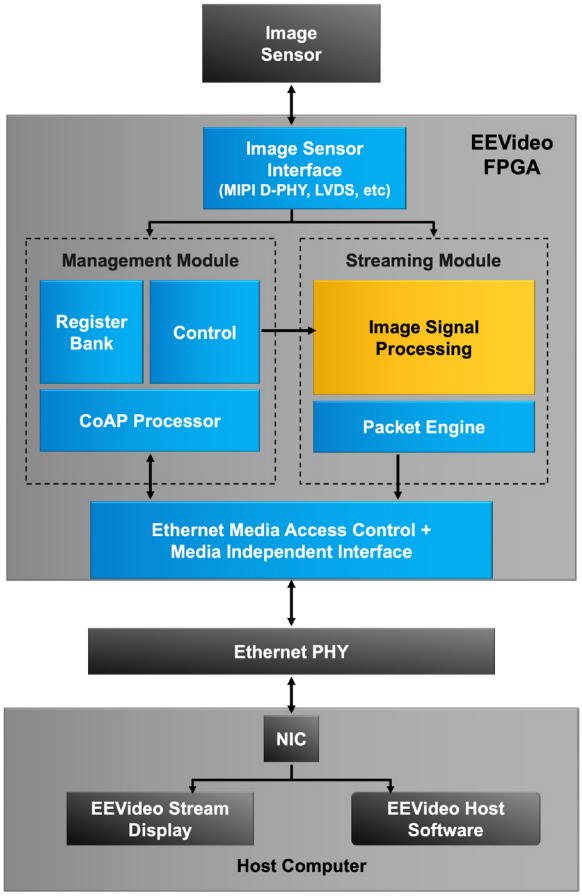
Implementation in Lattice FPGAs

EEVideo may be implemented as a modular, low-resource IP core, ideal for Lattice's low power FPGA families. The design emphasizes minimal logic utilization while delivering high-performance, low-latency video transport over standard Ethernet interfaces. See Figure 1.

A typical system-level block diagram for a Lattice FPGA implementation includes:

- Video input from a sensor interface (MIPI CSI-2, Parallel, or LVDS).
- Optional lightweight image processing pipeline (color space conversion, simple demosaic or JPEG encoding).
- EEVideo Streaming Packet Engine.
- EEVideo Management Engine (may include a processor such as Lattice's RISC-V soft core).
- Ethernet interface (EEVideo RGMII MAC)

Figure 1: EEVideo System Block Diagram



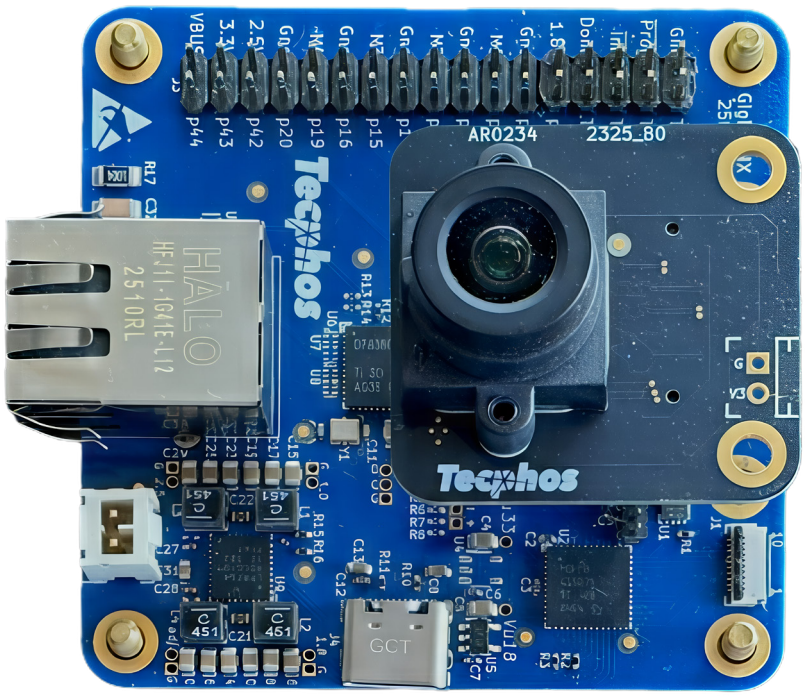
This modular approach allows designers to scale the implementation from a minimal streaming-only sensor node to a fully managed, discoverable device.

Example Implementation using Tecphos IP

Tecphos' EEVideo development board, featuring the Lattice CrossLink-NX-40 FPGA, provides a ready-to-use hardware platform for testing and demonstration. It includes a connection for an off-board MIPI sensor, Ethernet RJ45 jack, USB-C port for power, serial monitoring and control, and FPGA programming interface. The picture below shows the development board with a Tecphos-developed image sensor daughterboard. Other sensor options are available. See Figure 2.

Tecphos also licenses EEVideo FPGA Intellectual Property (IP) to implement the EEVideo standard in Lattice FPGAs. It includes management, streaming, and the Ethernet interface. This IP is available as either precompiled binaries or VHDL source code. 1 Gbps Ethernet is supported on the Lattice CrossLink-NX family, with speeds up to 10 Gbps for the Lattice CertusPro™-NX.

Figure 2: EEVideo Development Board



The development board, licensed IP, and open-source host-side software allow hardware and software engineers to jump-start the implementation of a custom embedded Ethernet video system.

Performance Results and Benchmarks

To validate the EEVideo protocol's design goals of low latency, high efficiency, and minimal resource usage, Tecphos tested a reference design based on the Lattice CrossLink-NX FPGA development board using Tecphos IP firmware. The results below demonstrate EEVideo's strong suitability for real-world embedded machine vision applications, with a protocol overhead typically less than 1%.

Resource Utilization

EEVideo's lightweight architecture results in an exceptionally small FPGA footprint, leaving ample resources for additional video processing, AI inference, or system control. See Table 3.

Table 3: Resource Utilization on Lattice CrossLink-NX-40 (1080p60 reference design)

Resource	Streaming Only	Full (Management + Streaming)	Percent Utilization (LFD2NX-40)
LUTs	<5k	<10k	~25%
BRAM (18K)	12	18	~12%
DSPs	2	4	<2%
Test Pixel Clock Frequency	125 MHz	125 MHz	-

Latency and Throughput

One of EEVideo's primary advantages is its ultra-low latency, achieved through on-the-fly packetization with no mandatory frame buffering. See Table 4.

Table 4: Performance Summary (1080p60 RAW, 1 G Ethernet)

Metric	Result	Notes
Sensor-to-network Latency	25 -125 µs	End-to-end from pixel input to Ethernet PHY
Protocol Overhead	0.5 - 5 %	Depending on header variant and packet size
Effective Payload Throughput	~995 Mbps	On 1 G link (after overhead with jumbo packets)
Power Consumption (Core Only)	125-250 mW	Depends on device and configuration

Tests were conducted using standard Ethernet NICs and common machine vision sensors. Latency is expected to scale favorably with higher resolutions and frame rates on 2.5 G/10 G links. All results are preliminary and expected to improve with further refinement.

Use Cases and Application Examples

EEVideo's lightweight design and low resource footprint make it an ideal protocol for a range of embedded vision applications where low latency, power efficiency, and Ethernet connectivity are essential. The following examples highlight practical deployments on Lattice FPGAs.

Industrial Machine Vision and Factory Automation

In smart factories, EEVideo enables high-speed, real-time camera feeds for quality inspection, defect detection, and process monitoring. An FPGA can implement the full EEVideo stack alongside image processing and sensAI inference while staying within power and logic budgets. Typical benefits include:

- Sub-millisecond latency for closed-loop control systems.
- Easy integration into existing Ethernet factory networks.
- Support for multiple cameras via network switching.

Example: A production line vision system performing OCR, part tracking, and dimensional measurement using a 1080p60 camera connected via EEVideo to a central controller.

Robotics and Vision-Guided Systems

Robotic arms and mobile robots require fresh, low-jitter video for navigation, object recognition, and precise manipulation. EEVideo's on-the-fly packetization minimizes end-to-end latency, allowing FPGAs to handle sensor fusion (camera, lidar, and IMUs) in real time.

Example: Random bin picking application where a Lattice FPGA processes structured-light 3D camera data streamed via EEVideo, enabling fast and accurate grasping decisions.

Automotive ADAS and In-Vehicle Systems

EEVideo supports camera-based ADAS and surround-view applications. Its low overhead and flexible formats work well with Lattice's automotive-grade FPGAs for sensor bridging, aggregation, and pre-processing.

Key advantages:

- Deterministic low-latency video transport critical for safety applications.
- Ability to operate alongside other Ethernet traffic in the vehicle network.

Example: Thermal or visible-light camera streams for pedestrian detection and lane-keeping assistance.

Medical Imaging and Life Sciences

Portable and fixed medical devices benefit from EEVideo's minimal resource usage and high efficiency. It enables clean, low-latency transmission of high-resolution imaging data from compact Lattice FPGA-based camera modules.

Example: Endoscopy or surgical microscopy systems where real-time video must be streamed to a processing workstation with minimal delay and power consumption.

Additional Emerging Applications

- **Drones and UAVs:** Lightweight video downlink with power-constrained payloads.
- **Smart Surveillance:** Multi-camera Ethernet-based security systems.
- **Edge AI Sensor Nodes:** Combining EEVideo with Lattice sensAI for distributed intelligent sensing.

Getting Started with EEVideo

Designed to be straightforward for both hardware engineers and software developers, the protocol's open-source nature and the Tecphos development board utilizing a Lattice CrossLink-NX FPGA implementation make it easy to evaluate EEVideo and integrate into new or existing designs. Tecphos licenses an IP package implementing the EEVideo protocol on targeted Lattice FPGAs, available as either a firmware binary module or VHDL source code to jump-start firmware development. Open source host-side software is available in a git repository.

Resources

- **Official Specification and Documentation:** Visit the EEVideo project website at eevideo.tecphos.com for the latest specification.
- **Source Code Repository:** The host side code libraries and documentation are hosted on GitLab at gitlab.com/eevideo/.
- **Lattice Support:** Guidance for Lattice FPGAs is provided through the EEVideo repository and Lattice's technical support channels.
- **EfficientTecphos EEVideo Development Board:** The development board is available through Tecphos (www.tecphos.com or info@tecphos.com).

Conclusion

EEVideo demonstrates the power of open collaboration in advancing embedded vision technology. Developed by Tecphos as a lightweight, open-source protocol, EEVideo was purpose-built to solve the real-world challenges of low-latency video and sensor data transport in resource-constrained environments. Lattice Semiconductor has embraced this protocol for its low power FPGA platforms, delivering a practical, high-performance implementation that leverages the strengths of both organizations.

Lattice Semiconductor's low power FPGA platforms, including the CrossLink-NX, CertusPro-NX, Avant, and Nexus families, are ideally suited to host EEVideo implementations. The protocol's small footprint integrates seamlessly with Lattice Ethernet MACs, SERDES, and video pipelines, leaving ample headroom for advanced features such as sensAI inference, image preprocessing, and system control, enabling designers to create highly integrated, power-efficient edge vision solutions.

Key benefits demonstrated in this white paper include:

- Sub-millisecond sensor-to-network latency with on-the-fly packetization.
- Very low resource usage (typically less than 10k System Logic Cells)
- True open-source accessibility that accelerates innovation and interoperability.
- Flexibility for a variety of levels of host management.
- Straightforward integration with existing Lattice Ethernet MACs and video pipelines.

As the demand for intelligent, networked vision systems continues to grow across industrial, automotive, robotics, and medical markets, EEVideo provides a future-proof foundation that accelerates the development of embedded video transport.

In summary, the partnership between Tecphos and Lattice brings together elegant protocol design with world-class low power FPGA implementation. EEVideo empowers engineers to develop faster, more efficient, and more innovative embedded vision systems than ever before.

READY TO LEARN MORE?

To learn more about Lattice low power FPGA-based solutions for compute, communications, industrial, and embedded applications, visit www.latticesemi.com or contact us at www.latticesemi.com/contact or www.latticesemi.com/buy. To learn more about EEVideo and the Tecphos EEVideo implementation and development board, visit www.tecphos.com or contact us at info@tecphos.com.

TECHNICAL SUPPORT ASSISTANCE

Submit a technical support case through **www.latticesemi.com/techsupport**.
For frequently asked questions, please refer to the Lattice Answer Database at **www.latticesemi.com/Support/AnswerDatabase.**

V.1 WP0057