Electronic Design

DAVID SHIN, Senior Product Marketing Manager,
Consumer Interface IP, Cadence, www.cadence.com

USB for Inside the Box: eUSB2 and eUSB2V2

This FAQ introduces and explains the new eUSB2 and eUSB2V2 standards, targeted at inside-the-box connections within AI PCs and other edge AI applications.

If eUSB2 — or its newer variant, eUSB2V2 — is new to you, you're not alone. These USB-based interfaces are still emerging, but they matter now more than ever to engineers building 5-nm and more advanced SoCs, or connecting one of these chips to a more mature-node companion. They're particularly relevant in edge AI and other embedded systems, where massive amounts of data pinball between components.

At a high level, eUSB2 is USB 2.0 reworked for the demands of modern silicon. eUSB2 was introduced through the efforts of the USB-IF to modernize the USB 2.0 standard, starting in the mid-2010s.

Referred to as eUSB2 (or eUSB2 version 1), the first version preserved USB 2.0 signaling and software compatibility but replaced the traditional PHY with a lower voltage, SoC-friendly implementation.

As the demand for greater data rates in edge AI applications grew, along came eUSB2 Version 2 (eUSB2V2) in late 2024, boosting performance up to 4.8 Gb/s while retaining USB 2.0 ecosystem backward compatibility. For many designs, eUSB2V2 is the simplest way to hit power and performance targets without jumping all the way to USB 3.x.

What Exactly is eUSB2?

Think of eUSB2 and eUSB2V2 as USB 2.0 redesigned for inside-the-box connections. eUSB2 drops the traditional 3.3-V I/O requirement down to ≤1.2 V, which is critical for advancednode SoCs. It's optimized for embedded chiptochip links rather than external links. It also supports tighter power, electromagnetic interference (EMI), and RF interference (RFI) noise budgets.

eUSB2V2 builds on that foundation. In addition to the

Spec	USB 2.0	eUSB2 V1	eUSB2 V2
I/O Voltage	3.3V	≤ 1.2V	≤ 1.2V
Data rate	480Mbps (Symmetric)	480Mbps (Symmetric)	480Mbps ~ 4.8Gbps (Symmetric / Asymmetric)
UTMI	UTMI+	Compatible to UTMI+	UTMI 2.0 (new for eUSB2V2)
Controller requirement	USB 2.0 controllers	Can utilize existing USB 2.0 Ctrl	Need eUSB2V2 controller
Operational mode	Native and external mode	Both native and repeater mode	Only native mode
Hub supported	Yes	Yes	No (scalable number of Host ports required)

1. This table highlights the key differences between eUSB2 Version 1 and eUSB2 Version 2 in relation to legacy USB 2.0. (Credit: Cadence)

480-Mb/s mode, it supports up to 4.8 Gb/s in one direction while keeping the reverse direction slower and more efficient. Depending on your system, eUSB2 can be used natively between chips or through a repeater to connect to legacy USB 2.0 devices.

Typical use cases include AI PCs, edge AI-enabled IoT devices, and wireless products pairing advanced and mature process nodes.

How Fast are These Interfaces, and What Link Modes are Available?

- eUSB2 (v1): Up to 480 Mb/s similar to legacy USB 2.0.
- eUSB2V2: Introduces HSx modes ranging from approximately 960 Mb/s up to 4.8 Gb/s.

One of the most practical advantages of eUSB2V2 is flexibility. You can run links symmetrically (the same speed in both directions) or asymmetrically. For example, a camera streaming video upstream might run at multiGb/s, while the return path stays at 480 Mb/s. This flexibility allows you to match the link to real traffic patterns rather than overengineering the interface.

What's Wrong with USB 3.x? Why Not Use It Instead?

USB 3.x is an excellent choice for external ports and very high-throughput applications. But it also comes with significantly higher PHY complexity, power consumption, and EMI/RFI noise challenges.

For many internal connections (debug links, cameras, and localized data paths), USB 3.x can be more than what is actually necessary. eUSB2 and eUSB2V2 are positioned between USB 2.0 and USB 3.x, offering multiGb/s performance with a simpler, lowerpower implementation that's better suited for embedded environments.

What are the Differences Between USB 2.0, eUSB2, and eUSB2V2?

The key differences separating these USB variants are highlighted in *Figure 1*. USB 2.0 supports data rates up to 480 Mb/s and uses a 3.3-V I/O voltage in the PHY.

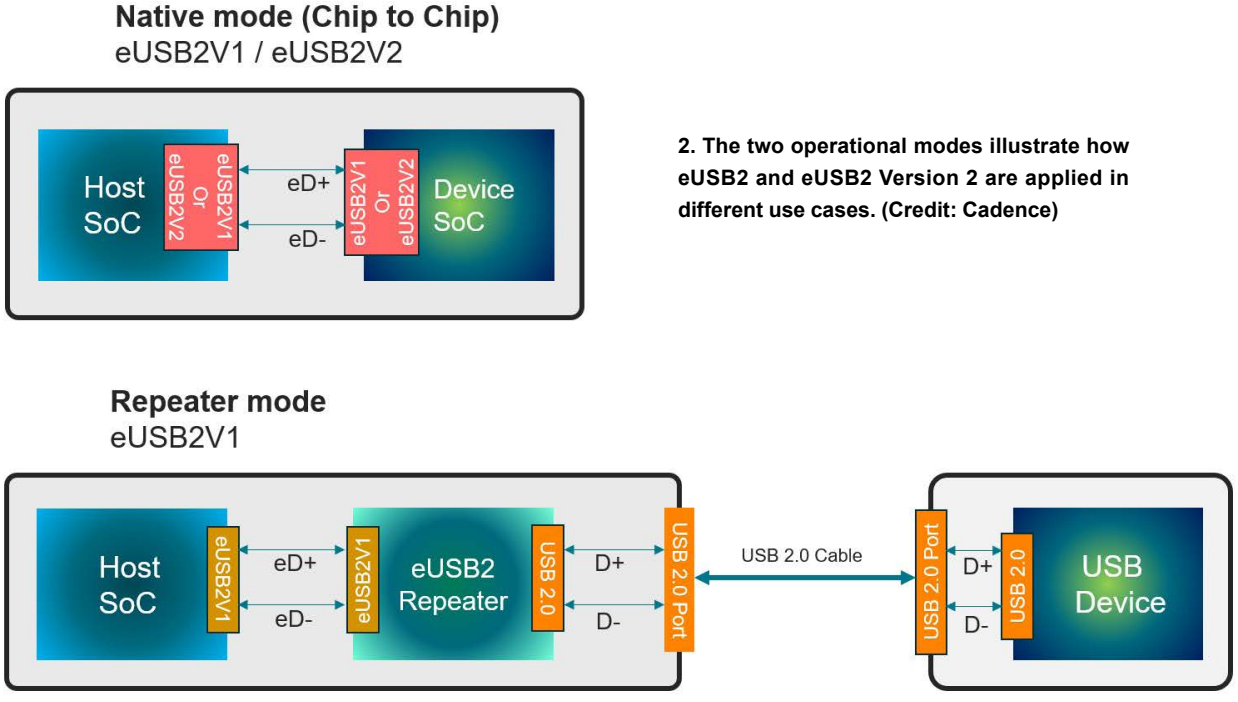
eUSB2 keeps the USB 2.0 protocol and the same 480 Mb/s data rate but shifts to a low I/O voltage (around 1.2 V or less) optimized for advanced-node SoCs. It supports both chip-to-chip connections and external USB 2.0 ports through repeater mode. eUSB2V2 further extends this concept by boosting data rates up to 4.8 Gb/s, targeting higherbandwidth, chiptochip interconnects only.

Is eUSB2 Only Targeted at Embedded Connections?

Not entirely. While eUSB2V2 is intended strictly for internal, chiptochip links, eUSB2 v1 can support both internal connections and external USB 2.0 ports *(Fig. 2)*.

Where Will eUSB2V2 Add the Most Value?

eUSB2V2 is a strong fit for new edge AI designs that need higher bandwidth but can't afford higher power or added noise. In AI PCs and edge systems, large amounts of data move continuously among sensors, high-resolution cam-



2. The two operational modes illustrate how eUSB2 and eUSB2 Version 2 are applied in different use cases. (Credit: Cadence)

eras, AI accelerators, and advanced-node SoCs. eUSB2V2 supports these faster chip-to-chip links efficiently, helping designers meet performance goals without the power, noise, and complexity tradeoffs of USB 3.x.

eUSB2V2 is particularly well-suited for high-resolution cameras in AI PCs and edge systems, as well as connecting advanced-node SoCs to mature-node companions.

How Does eUSB2V2 Help with Power and EMI/RFI Noise Issues?

Power efficiency and EMI/RFI noise reduction are key design goals for eUSB2V2, especially for modern edge AI systems. One of the primary means to achieve this is asymmetric link operation, where each direction runs only as fast as needed.

In many real-world designs, such as connecting a camera to a processor, most of the data moves in one direction. eUSB2V2 enables a high-traffic path to operate at up to 4.8 Gb/s, while keeping the return path at a much lower speed for control and status data. This reduces high-speed switching activity, thereby lowering overall power. It also limits high-frequency noise sources on the board, helping manage EMI and RFI.

Do eUSB2 and eUSB2V2 Still Use UTMI?

Yes, mostly.

eUSB2 v1 continues to use the familiar UTMI/UTMI+ interface. eUSB2V2 introduces UTMI 2.0, designed to support higher data rates and flexible link configurations beyond the 480-Mb/s limit.

Is There Working Silicon for These New Standards?

Yes. Several public demos of complete endtoend eUSB2V2 systems have taken place, including FPGAbased host and device controllers and a PHY implemented in a 3-nm test chip. This demo, shown at CES 2026, runs real workloads such as live 4K video in raw format, streaming (UVC) alongside recorded video (bulk data) transfers — demonstrating both performance and interoperability at the system level.

When Should I Consider Using eUSB2 v1 Over eUSB2V2?

The decision mainly comes down to two factors: how much bandwidth you need, and what the USB port is connecting to, namely, internal chiptochip links or external devices.

You may want to stick with eUSB2 v1 if:

- Your bandwidth needs are comfortably below 480 Mb/s. In this range, eUSB2 is simpler and more than sufficient.
- You must connect directly to external hubs or legacy USB devices, which requires standard USB compatibility.

What are Some Recommendations for System Designers?

If your design depends on USB 2.0 but targets an advanced-node SoC or needs to connect to one, and USB 3.x feels heavier than necessary, eUSB2 (especially eUSB2V2) is worth a serious look. It delivers multiGb/s internal bandwidth using lowvoltage, embeddedfriendly signaling, without the cost and complexity of a full SuperSpeed stack.

Rather than replacing external USB 2.0, eUSB2 complements it by offering a more balanced option for modern embedded and edge AI designs.

David Shin is a senior product marketing manager at Cadence, where he specializes in interface IP for the consumer segment, including ubiquitous USB IP. Before joining Cadence, David worked as a product line management professional at Cypress Semiconductor (now part of Infineon), Synaptics, and Omnivision. In these roles, he defined and developed cutting-edge device solutions such as USB products, touchscreen controllers, fingerprint sensors, and CMOS image sensors for applications including smartphones, IoT, and AR/VR/MR in the metaverse. David holds a master's degree in computer science from California State University, Chico.