Application Note

A Guide to USB2.0 High Speed Host Transmit Electrical Compliance

Shreyas Rao

Abstract

This application note describes how to perform USB 2.0 high-speed (HS, 480Mbps) host transmit electrical compliance testing on Jacinto™ SoCs (TDA4VEN/ J722S/ TDA4AEN) using Linux™ commands. The procedure covers writing the xHCI Port Status and Control (PORTSC) register to place the USB2.0 port into compliance mode to generate high speed test packet, capturing the differential D+/D− eye diagram on a high-speed oscilloscope, and post-processing the captured waveform with the USB-IF electrical analysis tool to verify compliance against the USB 2.0 specification eye masks. Additionally, the application note describes how to use the USB2 PHY TX analog front-end register to fine-tune the differential TX output amplitude.

Table of Contents

Trademarks

1 Introduction

USB 2.0 High Speed (HS) electrical compliance testing verifies that a device meets the eye-diagram, output amplitude, and data-rate requirements defined in the USB 2.0 Specification (Section 7.1.20) and the USB-IF Compliance Test Specification (CTS). The test evaluates the quality of the differential transmit signal (TX) at the host port, measured as the eye opening against the inner and outer mask specification.

This application note focuses on the Host Transmit (TX) compliance path for Jacinto SoCs implementing the xHCI host controller with a USB 2.0 UTMI+ PHY. The procedure is applicable to TDA4VEN, J722S, and TDA4AEN devices. Unlike USB 3.0 Host compliance (described in the *A Guide to USB3.1 Gen1 SuperSpeed Host Transmit Electrical Compliance* application note), which requires a SERDES link training state machine (LTSSM), USB 2.0 compliance mode is entered by programming the *Port Test Control* field in the xHCI PORTSC register and optionally adjusting the PHY analog front-end registers for TX amplitude optimization.

A practical consideration for TDA4VEN[1] is the signal integrity impact of the USB multiplexer and Type-C ICs present on the EVM board. The additional passive and active components in the signal path degrade the eye diagram, causing the inner eye mask violation at the oscilloscope input. For best compliance results, routing the USB2.0 HS signals to a type-A connector without the intermediate ICs or Type-C connectors is recommended.

[1] TDA4VEN is used thorughout the application note but equally applies to TDA4AEN

2 Acronyms and Definitions

Table 2-1. Acronyms and Definitions

Acronym	Definition
CTS	USB-IF Compliance Test Specification
DFP	Downstream Facing Port
DTS / DTB	Device Tree Source / Device Tree Blob (compiled binary)
EVM	Evaluation Module (TI reference hardware platform)
HS	High Speed (USB 2.0, 480Mbps)
HSFE	High Speed Far End
HSNE	High Speed Near End
PHY	Physical Layer Transceiver
PTC	Port Test Control (xHCI PORTPMSC field)
PORTSC	Port Status and Control register
SDK	Software Development Kit
xHCI	eXtensible Host Controller Interface (USB host controller)
USB	Universal Serial Bus
UTMI	USB Transceiver Macrocell Interface

3 USB 2.0 High Speed TX Compliance: Theory of Operation

3.1 USB 2.0 High Speed Signaling Overview

USB 2.0 High Speed (HS) operates at 480Mbps on a single differential pair (D+, D−). Data is encoded using non-return-to-zero inverted (NRZI) with bit stuffing. HS signaling uses a differential swing of approximately 400mV peak-to-peak (with single-ended, centered at 0V (SE0 = both lines at 0V for EOP, SOP, or reset signaling). The transmit eye diagram at the host port output must satisfy the normative inner and outer eye mask boundaries specified in USB 2.0 Specification Table 7-8.

The compliance test evaluates the worst case eye opening using a standardized test packet sequence: the High Speed Test Packet, defined in USB 2.0 Section 7.1.20. This fixed-length bit pattern is designed to exercise a broad range of transition densities, maximizing inter-symbol interference (ISI) and therefore producing the worst case eye diagram for mask testing. The pattern length is 53 bytes.

3.2 xHCI Port Test Control and Compliance Mode Entry

The xHCI specification defines a Port Test Control (PTC) field in bits [31:28] of the Port Power Management Status and Control (PORTPMSC) register. Writing PTC = 4 (TEST_PACKET) to the USB 2.0 HS port causes the UTMI+ PHY to continuously transmit the HS Test Packet. This is the standard mechanism used by all xHCI-based USB 2.0 host controllers to enter compliance mode.

The Jacinto xHCI implementation requires three sequential register writes to place the USB2 port into continuous test packet transmission:

- Write to the xHCI Port Status and Control register for the USB2 HS port to initialize the port state
- Write to the xHCI USB Command register to prepare the host controller
- Write PTC = 4 (TEST_PACKET) to the USB2 HS PORTPMSC register (upper nibble = 0x4, bits [31:28] = 0100)

Once the test packet is active, the PHY transmits the HS Test Packet continuously at 480Mbps. Capture the differential output on the oscilloscope through the USB Host Test Fixture SMA taps.

3.3 Eye Diagram Compliance

The oscilloscope captures multiple repetitions of the test packet on Channels 1 and 2 (D+ and D− respectively). A math trace (Channel 1 − Channel 2) represents the differential voltage waveform. This differential trace is exported as a csv file and loaded into the USB-IF Electrical Analysis Tool, which applies the normative eye mask template and reports pass/fail results.

Two test fixture modes defined by USB-IF:

- HSNE (High Speed Near End): No captive 1-meter cable between DUT and fixture. The inner eye mask for HSNE is more stringent than HSFE. Use HSNE when connecting the DUT directly to the fixture without a captive cable.
- HSFE (High Speed Far End): A 1-meter captive cable is included between DUT and fixture. The inner eye mask for HSFE is relaxed compared to HSNE. Use HSFE only when a captive cable is present in the test path.

4 Register Reference

4.1 xHCI PORTSC — Port Status and Control Register

For TDA4VEN, the USB0 xHCI controller MMIO base is at 0x31000000. USB2p0 PORTSC is at operational base + 0x400.

Table 4-1. xHCI Register Addresses (TDA4VEN USB0)

Register	Address (USB0)	Description
xHCI Operational Base	0x31000020	xHCI USB Command (USBCMD);
PORTSC — (USB2 HS)	0x31000420	Port Status and Control for USB 2.0 High Speed port
PORTPMSC — (USB2 HS)	0x31000424	Port Power management Status and Control for USB 2.0 HS port
USB2SS_PHY2_AFE_TX_REG1 (USB0)	0x0F908004	USB2 PHY TX analog front-end register; controls HS TX differential amplitude

4.2 PORTPMSC Bit-Field Description (USB 2.0 — xHCI Spec)

Table 4-2. Key PORTPMSC Bit-Field Description (USB 2.0 HS Port)

Bits	Field Name	Description
[31:28]	PTC — Port Test Control	Selects test mode. 0x0 = Normal operation. 0x4 = TEST_PACKET (continuous HS test packet transmission).

4.3 Compliance Write Value Decode: 0x40000000

Write the following 32-bit value to PORTPMSC (0x31000424) to enable continuous HS Test Packet transmission:

Write value: 0x40000000

Binary: 0100_0000_0000_0000_0000_0000_0000_0000

Bits [31:28] = 0100 = PTC = 4 = TEST_PACKET

=> xHCI instructs the UTMI+ PHY to continuously transmit the HS Test Packet

All other bits = 0 => default port state preserved

4.4 USB2 PHY TX Amplitude Register — USB2SS_PHY2_AFE_TX_REG1

The USB2 PHY analog front-end TX register controls the differential output amplitude boost. The default boost value is 0x08 (decimal 8). The register is at 0x0F908004 location for USB0 on TDA4VEN.

Table 4-3. USB2SS_PHY2_AFE_TX_REG1 Register (USB0 = 0x0F908004)

Field	Bits	Description	Default
BOOST_EN	[0]	Boost Enable; Boost code can be controlled by bits[6:1]	0x0 (Boost code= 0x8)
BOOST	[6:1]	TX amplitude boost control. Higher value equals to higher differential amplitude. Lower value equals to lower differential amplitude. Valid range: 0x00 to 0x3F.	0x08

Table 4-4. Boost Value Versus TX Amplitude Behavior

BOOST Value (hex)	TX Amplitude Effect
0x00	Minimum amplitude: use if eye exceeds outer mask (overdrive)
0x08	Default (SDK default); nominal amplitude
0x3F	Maximum amplitude: use for maximum output if inner mask violations persist

5 Compliance Procedure: TDA4VEN / J722S / TDA4AEN

5.1 Hardware Requirements

Table 5-1. Hardware Requirements

Item	Description / Part
TDA4VEN EVM	Type-C connector can show reduced eye amplitude due to signal path components- use Type-A connector.
USB 2.0 Host Test Fixture	USB-IF Host Test Fixture with D+ and D− SMA tap connectors
High-Speed Oscilloscope	≥ 4GHz bandwidth, ≥ 10GSa/s sample rate (for example, Tektronix® DPO/ MSO70000)
Matched SMA Cable Pair	Phase-matched SMA cables for D+ (Ch 1) and D− (Ch 2) connections
USB-C to USB-A Adapter	Only if required, depending on the board used.

5.2 Software Requirements

Table 5-2. Software Requirements

Item	Version	Notes
TI Processor SDK Linux (ADAS)	11.02 or later	ADAS variant for TDA4VEN; install to SD card per SDK documentation
USB-IF Electrical Analysis Tool	Latest from USB-IF	Post-process oscilloscope CSV data; available at usb.org

5.3 Hardware Setup

5.4 Physical Connections

- Plug the USB 2.0 Host Test Fixture Type-A plug connector into the USB Type-C™ port using a Type-C to Type-A adapter on the TDA4VEN EVM (USB0 port) or connect directly to the Type-A port if Type-C is not used
- Connect a pair of matched SMA cables from the D+ tap of the test fixture to Channel 1 of the oscilloscope, and the D− tap to Channel 2
- Connect the Main UART port of the EVM to the PC. Open TeraTerm and configure the serial port to 115200 baud. For TDA4VEN, use the third COM port of the four available.
- Insert the prepared SD card with TI Processor SDK Linux image. Set the EVM DIP switches to SD card boot mode.
- Verify the DIP switch is set to DFP mode for USB host operation.

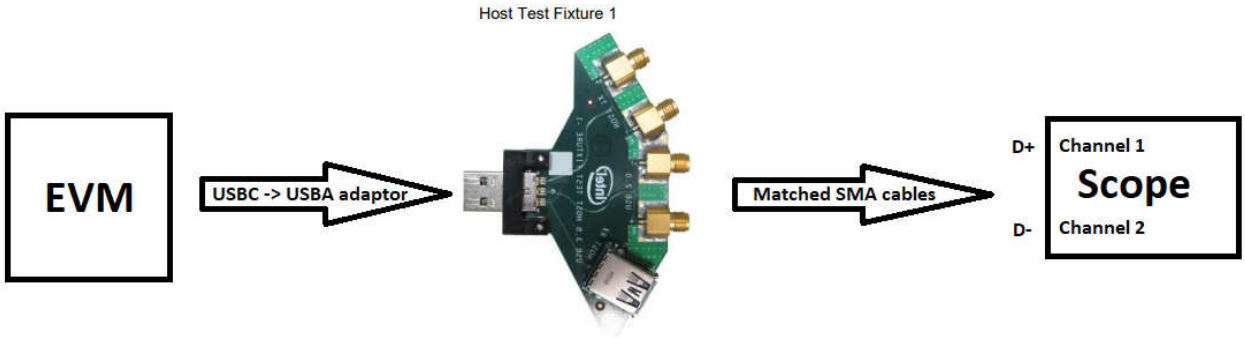

Figure 5-1. TDA4VEN EVM USB2.0 Host TX Test Setup

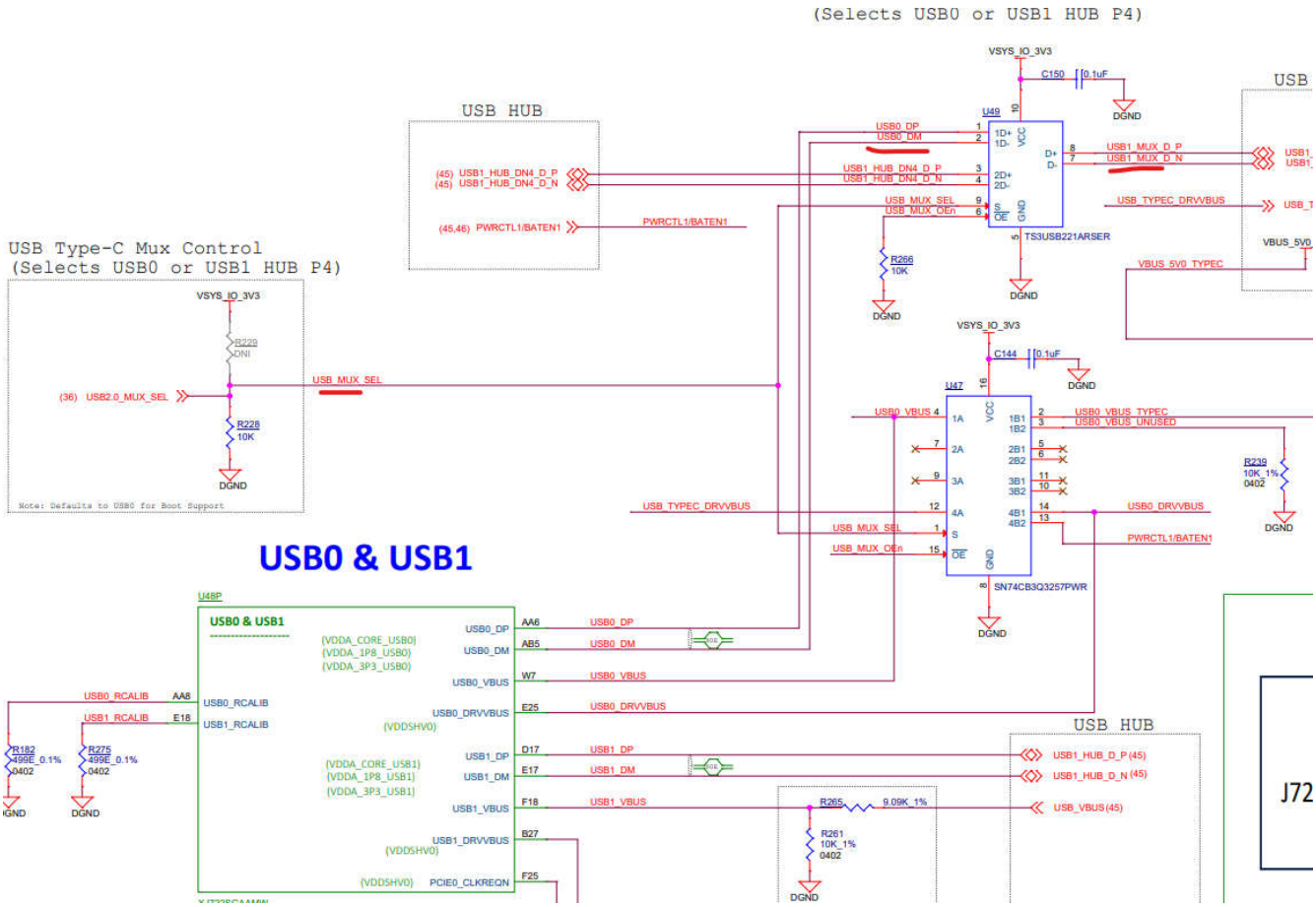

Figure 5-2. USB Mux and Type-C Path on J722SEVM

5.5 Device Tree Source (DTS) Modification for USB0

Install the TI Processor SDK Linux onto the SD card. Use the *wget* command in a Linux terminal to download the ADAS SDK installer, then make the sdk executable with chmod +x and run the installer. Follow the SDK documentation to create the SD card image.

By default, the TI Processor SDK DTS can have USB1 enabled and USB0 disabled, or have the USB mux configured for USB1 output. To route USB0 output on the EVM, modify the DTS file to enable USB0 and configure the USB mux select signal accordingly.

Go to <SDK_INSTALL_DIR>/board-support/ti-linux-kernel-<version>/arch/arm64/boot/dts/ti/ and open the relevant board .dts file for J722S. Enable the USB0 node and disable USB1 as appropriate for the board variant. See the TDA4VEN EVM schematics for USB mux select signal mapping.

```
};

&usbss0 {
        ti,vbus-divider;
        status = "okay";
};

&usb0 {
        dr_mode = "host";
        /*usb-role-switch;*/
        status = "okay";

};

&usb0_phy_ctrl {
        bootph-all;
};

&usbss1 {
        pinctrl-names = "default";
        pinctrl-0 = <&main_usb1_pins_default>;
        ti,vbus-divider;
        status = "disabled";
};

&usb1 {
        dr_mode = "host";
        maximum-speed = "super-speed";
        phys = <&serdes0_usb_link>;
        phy-names = "cdns3,usb3-phy";
        status = "disabled";          /* explicitly enable */
        ti,force_host = "true";    /* prevent OTG/device mode flip */
```

Figure 5-3. DTS with USB0 Enabled and USB1 Disabled

```
    exp1: gpio@23 {
            compatible = "ti,tca6424";
            reg = <0x23>;
            gpio-controller;
            #gpio-cells = <2>;
            gpio-line-names = "TRC_MUX_SEL", "OSPI/ONAND_MUX_SEL",
                              "MCASP1_FET_SEL", "CTRL_PM_I2C_OE#",
                              "CSI_VIO_SEL", "USB2.0_MUX_SEL",
                              "CSI01_MUX_SEL_2", "CSI23_MUX_SEL_2",
                              "LMK1_OE1", "LMK1_OE0",
                              "LMK2_OE0", "LMK2_OE1",
                              "GPIO_RGMII1_RST#", "GPIO_AUD_RSTn",
                              "GPIO_eMMC_RSTn", "GPIO_uSD_PWR_EN",
                              "USER_LED2", "MCAN0_STB",
                              "PCIe0_1L_RC_RSTz", "PCIe0_1L_PRSNT#",
                              "ENET1_EXP_SPARE2", "ENET1_EXP_PWRDN",
                              "PD_I2ENET1_I2CMUX_SELC_IRQ", "ENET1_EXP_RESETZ";
            bootph-all;

            p05-hog {
                    /* P05 - USB2.0_MUX_SEL */
                    gpio-hog;
                    gpios = <5 GPIO_ACTIVE_LOW>;
                    output-high;
                    line-name = "USB2.0_MUX_SEL";
            };
```

Figure 5-4. USB Mux DTS Configuration to Select USB0 D+/D−

Compile the DTS file to generate the .dtb file:

```
$ cd /<sdk-install-dir>/board-support/ti-linux-kernel-<version>
$ export CROSS_COMPILE=/<sdk-install-dir>/linux-devkit/sysroots/x86_64-arago-linux/usr/bin/aarch64-
oe-linux/aarch64-oe-linux-
$ export ARCH=arm64
$ make dtbs
```

Copy the compiled .dtb file to the SD card:

```
$ sudo cp arch/arm64/boot/dts/ti/k3-j722s-evm.dtb /media/<username>/root/boot/dtb/ti/
$ sudo eject /media/<username>/root/
```

Set the EVM to SD card boot mode. Power on the EVM and wait for Linux to boot. Log in as root on the TeraTerm console.

5.6 Step 1: Verify EVM Boot and Console

Connect to the EVM console (third COM port at 115200 baud). After boot, login as root:

```
root@j722s-evm:~#
```

5.7 Step 2: Write xHCI Registers to Enable HS Test Packet Mode

The following three sequential register writes place the USB0 port into continuous HS Test Packet transmission. Do not connect the USB test fixture to the port.

```
# Step 2a: Initialize USB2 HS PORTSC (Port 1)
devmem2 0x31000420 w 0xA0
# Step 2b: Write xHCI USB Command register
devmem2 0x31000020 w 0x4
# Step 2c: Write PTC = 4 (TEST_PACKET) to PORTPMSC
devmem2 0x31000424 w 0x40000000
```

5.8 Step 3: Capture the HS Test Packet Waveform on the Oscilloscope

After the register writes, the USB0 port continuously transmits the HS Test Packet on D+ and D−. On the oscilloscope:

- Acquire a single-shot capture of the test packet to build a repeatable eye diagram
- Save the Math1 (Ch1-Ch2) trace (differential waveform) as a CSV file to an external USB drive

6 USB-IF Electrical Analysis Tool: Post-Processing Procedure

The USB-IF Electrical Analysis Tool processes the oscilloscope CSV output and applies the normative eye mask from the USB 2.0 Compliance Test Specification to generate a pass/fail report.

- Open the USB Electrical Analysis Tool on the PC
- Select the Host SQ tab at the top of the application window
- Click the file browser and navigate to the saved Math1 CSV file from the oscilloscope
- Select HSNE (High Speed Near End)
- Check the *Summary File* checkbox to generate an HTML summary report
- Click *Test* to analyze the waveform. The tool applies the eye mask, check rise and fall times, amplitude, and other HS signal quality parameters, and generates an HTML compliance report

Note

Select HSFE (High Speed Far End) only if a captive 1-meter USB cable is present between the DUT port and the test fixture input. The HSFE inner mask is relaxed relative to HSNE, accounting for cable loss.

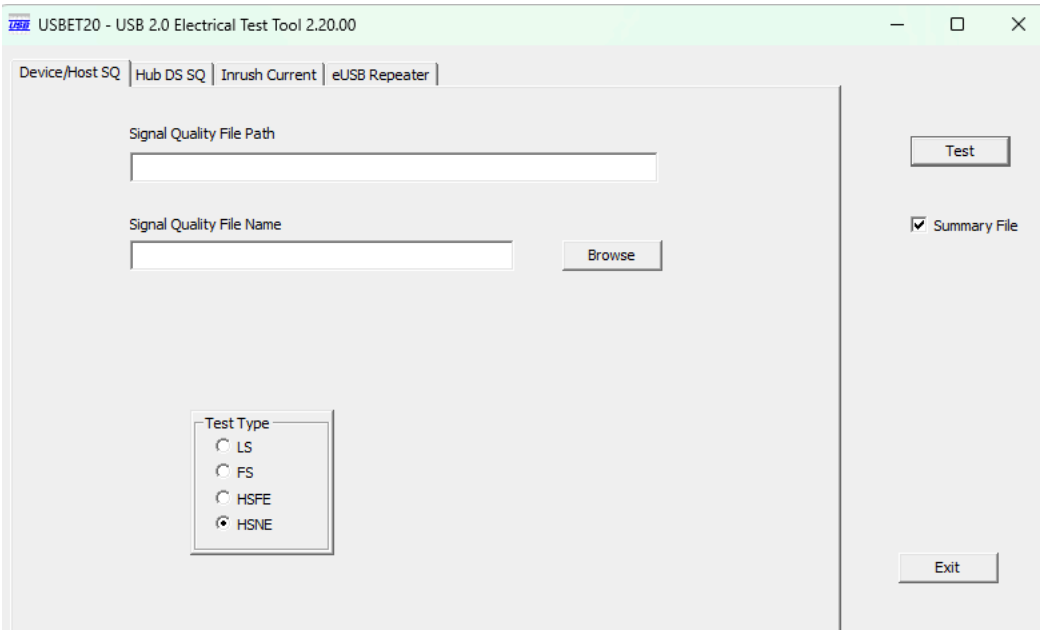

Figure 6-1. USB-IF Electrical Analysis Tool: Host SQ Tab Configuration

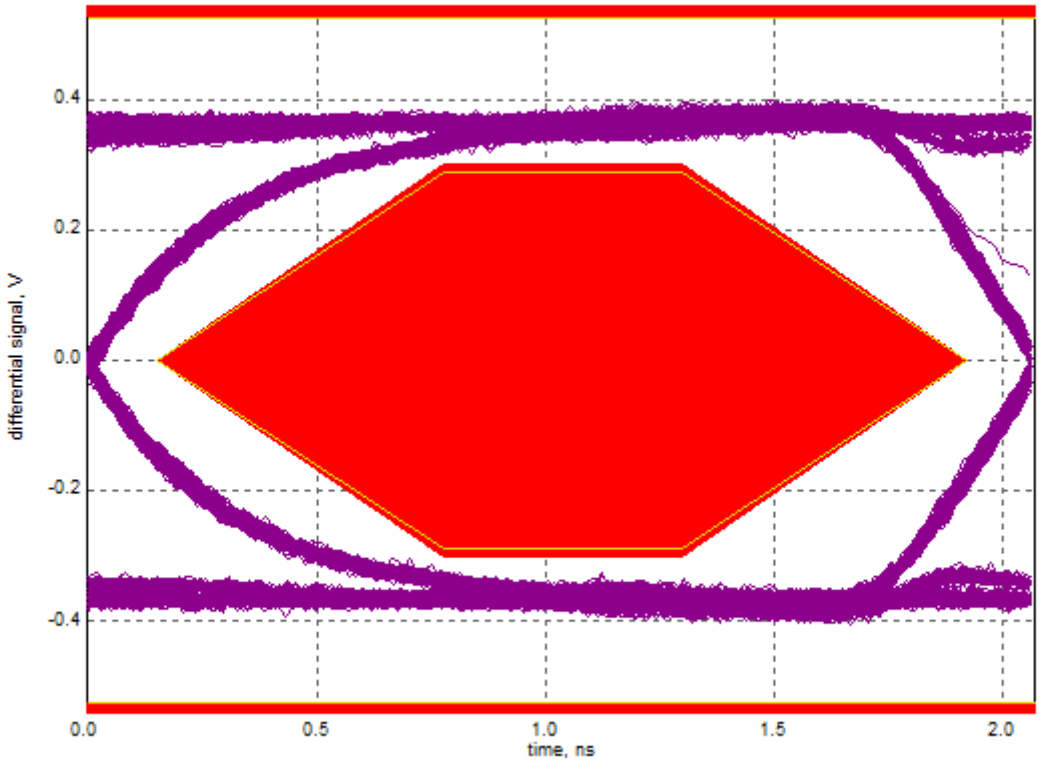

Figure 6-2. USB2.0 HS TX Eye Diagram: A Pass Example

7 Adjusting the HS TX Amplitude

Depending on the default BOOST setting, board traces, layout parasitics, and external components in the signal path, the HS TX eye diagram can clip the inner eye mask (amplitude too low) or exceed the outer eye mask (amplitude too high). The USB2 PHY TX analog front-end register (USB2SS_PHY2_AFE_TX_REG1) provides a digital boost control field that adjusts the differential TX amplitude.

The SDK default BOOST value is 0x08 (decimal 8). To increase amplitude, write a higher BOOST value; to decrease amplitude, write a lower value. The field is 6 bits wide with valid range 0x00 to 0x3F with an enable in the bit 0 field.

7.1 TX Amplitude Adjustment Procedure

- Run the compliance procedure to capture the baseline eye diagram with the default BOOST CODE= 0x8
- If the eye clips the inner mask (too small), increase the BOOST value. If the eye exceeds the outer mask (overdrive), decrease the BOOST value.
- Write the new BOOST value to register 0x0F908004 (USB2SS_PHY2_AFE_TX_REG1):

```
# Low amplitude
devmem2 0x0F908004 w 0x01
# Max amplitude
devmem2 0x0F908004 w 0x7F
```

- After writing the new BOOST value, recapture the oscilloscope waveform (the test packet continues transmitting) and re-run the USB-IF tool
- Iterate until the eye diagram passes both the inner and outer masks under all HSNE conditions

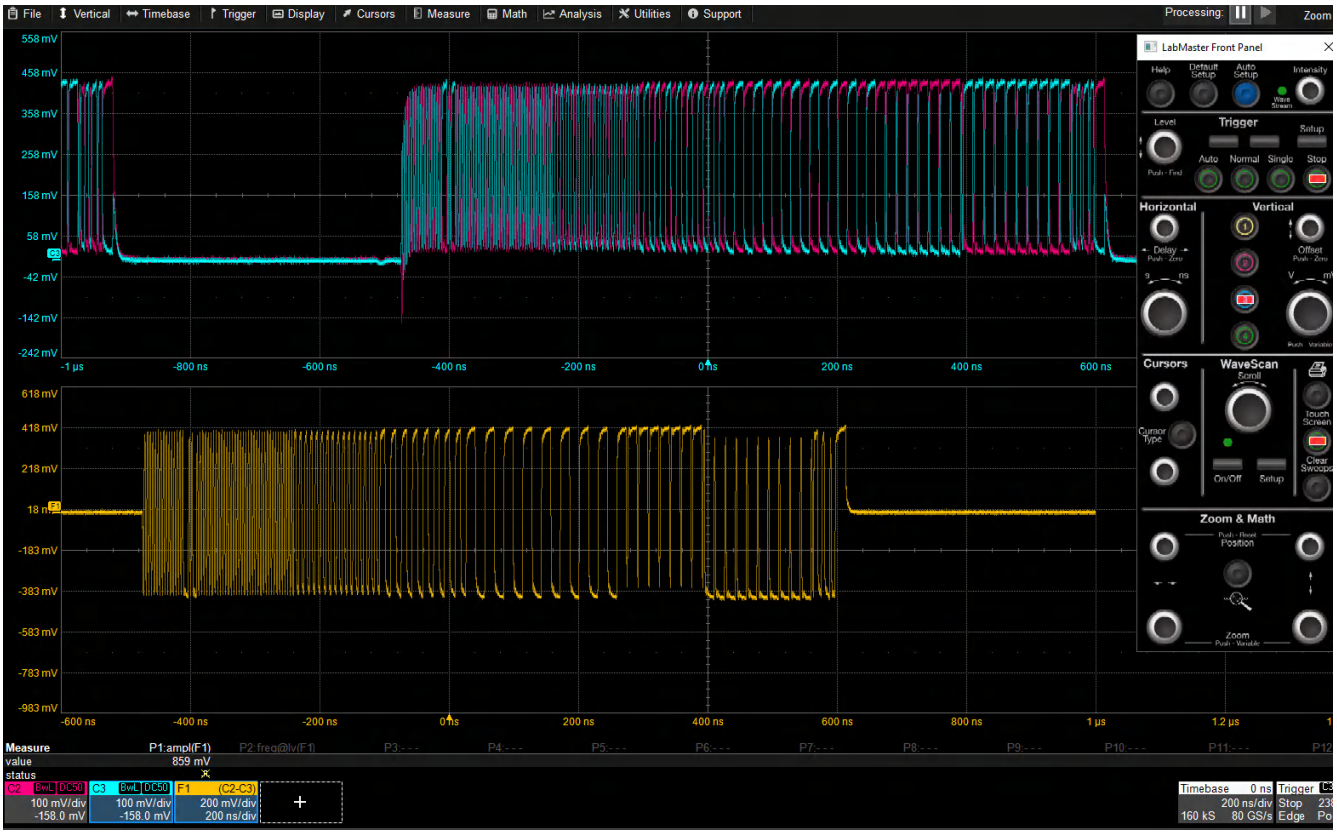

Figure 7-1. USB2.0 HS TX Eye Diagram: BOOST = 0x01 (Low Amplitude)

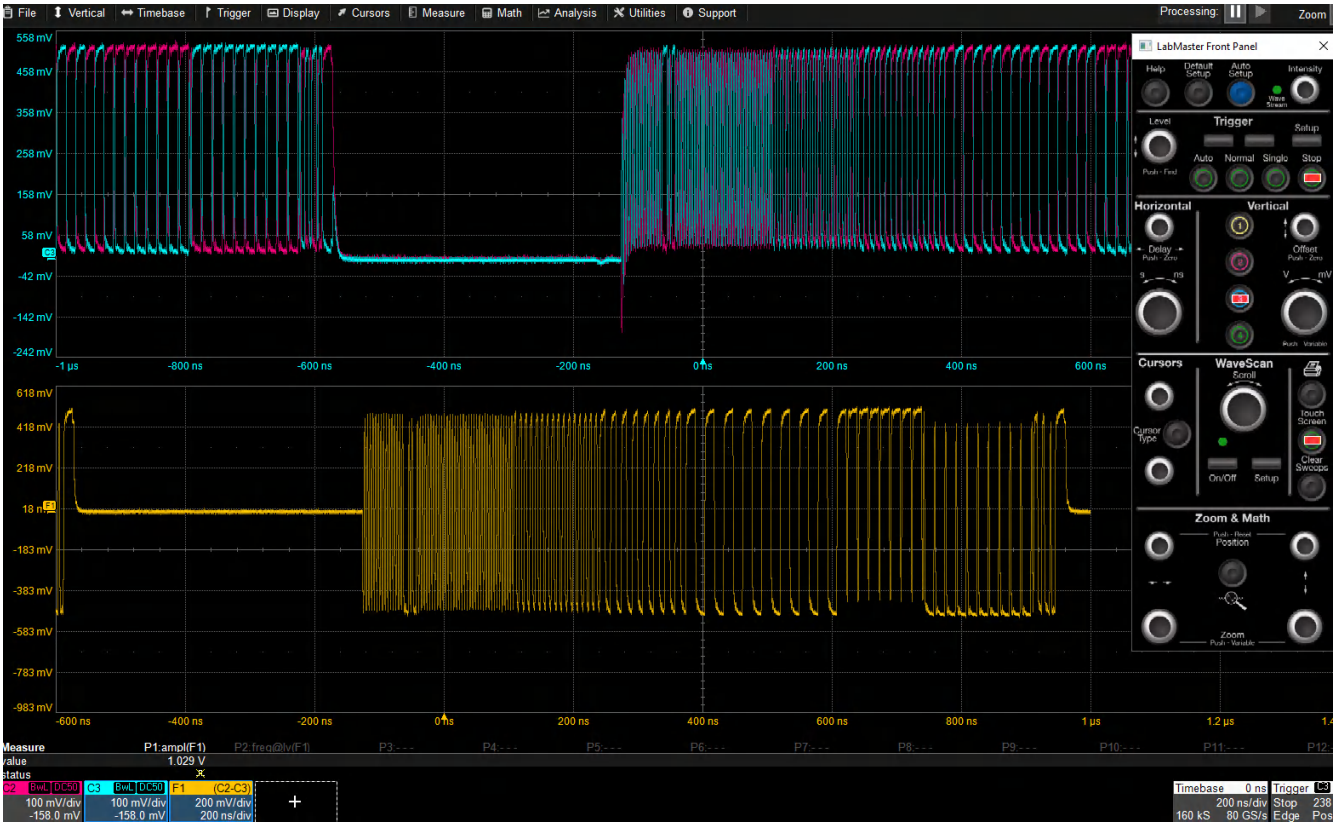

Figure 7-2. USB2.0 HS TX Eye Diagram: BOOST = 0x7F (Maximum Amplitude)

8 Compliance Procedure: J7200/J784S4/J721S2/J742S2

The J7200, J784S4, J742S2, and J721S2 SoCs have a single instance of USB0 with similar xHCI implementation but with different offset (0x80) and MMIO base address structure (0x06010000) as TDA4VEN. All the write values, and compliance procedure steps described in the TDA4VEN compliance section apply identically. See the respective device USB0 register sheet for additional information. The TX adjustment procedure is identical to TDA4VEN TX adjustment.

9 Troubleshooting

Table 9-1. Troubleshooting Guide

Symptom	Likely Cause	Corrective Action
No signal on oscilloscope after register writes	Register writes did not take effect; incorrect port selected; USB0 not enabled in DTS	Verify DTS has USB0 enabled. Check that no USB device is connected to the port. Re-issue all three devmem2 writes in sequence.
Eye diagram clips inner mask (amplitude too low)	Default BOOST setting too low; USB mux or Type-C IC in signal path	Use short SMA cables. Increase BOOST via 0x0F908004 register write.
Eye diagram exceeds outer mask (amplitude too high)	BOOST value set too high;	Decrease BOOST value through register write.
USB-IF tool reports rise and fall time failure	Excessive parasitic capacitance on D+/D− traces; cable or connector discontinuity	Inspect SMA cable integrity. Verify matched SMA cable pair. De-skew the scope channels. Check the test fixture connection.
Serial console not responding after power-on	Wrong COM port selected; baud rate mismatch	For TDA4VEN, use the third COM port of the four available. Set baud to 115200.
HSNE test fails but HSFE test passes	Captive cable in signal path	Confirm no captive cable. Select HSNE. If captive cable is present, select HSFE.

10 Summary

A comprehensive guide to USB 2.0 Host TX High Speed signal quality electrical compliance on Jacinto SoCs (TDA4VEN, J722S, and TDA4AEN) has been presented. The method described uses the native Linux devmem2 utility and the TI Processor SDK to place the xHCI host controller into continuous HS Test Packet transmission.

The procedure covers: hardware setup using the EVM; device tree modifications to enable USB0; three sequential xHCI register writes to activate continuous test packet transmission; oscilloscope differential TX capture; and post-processing with the USB-IF Electrical Analysis Tool to evaluate the eye diagram against the HSNE mask.

Additionally, the USB2 PHY TX amplitude adjustment procedure using the USB2SS_PHY2_AFE_TX_REG1 BOOST register field (0x0F908004) is described, enabling validation engineers to fine-tune the differential output amplitude iteratively until the eye diagram satisfies both inner and outer mask requirements. This approach provides a low-cost, repeatable USB 2.0 compliance validation path for custom boards and production qualification.

11 References

1. USB-Implementers Forum, *USB 2.0 Specification, Rev 2.0*, webpage
2. USB-IF, *USB-IF Compliance Test Specification (CTS)*, webpage
3. Intel, *eXtensible Host Controller Interface for Universal Serial Bus*, requirement specification
4. Texas Instruments, *A Guide to USB3.1 Gen1 SuperSpeed Host Transmit Electrical Compliance*, application note
5. Texas Instruments, *J722S / TDA4VEN / TDA4AEN Jacinto™ SoC Technical Reference Manual*, technical reference manual
6. Texas Instruments, *TI Processor SDK Linux for Jacinto ADAS*, software development kit
7. USB-IF, *USB-IF USB2.0 Forum*, webpage

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Last updated 10/2025