

USB4 CV Guide

06/04/2025

Rev2.0

Page | 1

Table of Contents

1.	Required Hardware and Software.....	5
2.	Setting Up USB4 CV On Windows System	7
2.1	Installing USB4 CV.....	7
2.2	Host Settings	7
2.3	Installing USB Protocol Suite	7
2.4	USB4 CV Settings	7
2.5	Launching USB4CV	7
2.5.1	Graphical User Interface (GUI)	7
2.5.2	Command Line Interface (CLI)	9
2.6	Manually Install USB4_Hi Driver.....	11
2.7	USB4_Hi Device & KMGL Device Version	13
2.8	USB4CV Interface Window	13
3.	Setting Up USB4CV on Linux System	15
3.1	Download & Install Linux USB4CV	15
3.2	Setting up USB4CV	16
3.2.1	USB4CV GUI Version	16
3.2.2	USB4CV Command Line Version.....	23
4.	M4x Analyzer Connection.....	28
4.1	Introduction	28
4.2	Analyzer Test Setup.....	28
4.3	Exerciser Test Setup	28
4.4	Using Ethernet Connection	29
4.4.1	Assign Static IP Adress To ARMx64 System.....	29
4.4.2	Setting a Static IP Address on Your USB Analyzer	31
4.5	Setup Ethernet connection To Run USB4CV on Linux System.....	32
4.5.1	Setup IP on Linux	33
4.5.2	Setup IP on Windows.....	35
4.6	Troubleshooting	38
5.	Running USB4 CV Tests on Linux.....	40
5.1	Linux Permissions for USB4CV.....	40
5.2	Hint for GNOME users	40
5.3	Starting LeCroy on Remote Windows Machine	41

5.4	Start Testing on Linux	43
5.4.1	Set Up LeCroy IP Address	43
5.4.2	Running cv-gui	43
5.4.3	Running cv-cli	46
5.5	LeCroy Trace	48
6.	USB4CV Test Setups	49
6.1	Chapter 4, Chapter 5, Chapter 6 & Chapter 8	49
6.1.1	UUT is a Host - Chapter 4 AN & Chapter 5 AN & Chapter 6 AN & Chapter 8 AN:	50
6.1.2	UUT is a Host - Chapter 4 EX & Chapter 5 EX & Chapter 6 EX & Chapter 8 EX:	50
6.1.3	UUT is a Device - Chapter 4 AN & Chapter 5 AN & Chapter 6 AN & Chapter 8 AN:	51
6.1.4	UUT is a Device - Chapter 4 EX & Chapter 5 EX & Chapter 6 EX & Chapter 8 EX:	51
6.1.5	UUT is a Hub - Chapter 4 AN:	52
6.1.6	UUT is a Hub - Chapter 4 EX:	53
6.1.7	UUT is a Hub - Chapter 5 AN & Chapter 6 AN & Chapter 8 AN:	54
6.1.8	UUT is a Hub - Chapter 5 EX & Chapter 6 EX & Chapter 8 EX:	55
6.2	Chapter 7	57
6.2.1	UUT is a Host - Chapter 7:	57
6.2.2	UUT is a Device - Chapter 7:	58
6.2.3	UUT is a Hub - Chapter 7:	59
6.3	Chapter 9	68
6.3.1	UUT is a Host - Chapter 9:	68
6.3.2	UUT is a Device - Chapter 9:	69
6.3.3	UUT is a Hub - Chapter 9:	70
6.4	Chapter 10	73
6.4.1	UUT is a Host – Chapter 10:	75
6.4.2	UUT is a Device – Chapter 10:	76
6.4.3	UUT is Hub – Chapter 10:	78
6.5	Chapter 11	80
6.5.1	UUT is a Host - Chapter 11:	80
6.5.2	UUT is a Device - Chapter 11:	81
6.5.3	UUT is a Hub - Chapter 11:	82
6.6	Chapter 12	84
6.6.1	UUT is a Host - Chapter 12 AN	84

6.6.2	UUT is a Host - Chapter 12 EX:.....	85
6.7	Chapter 13	85
6.7.1	UUT is a Host – Chapter 13:.....	86
6.7.2	UUT is a Device – Chapter 13:	87
6.7.3	UUT is a Hub – Chapter 13:	89
7.	Linux USB4CV Chapters Running Requirements:	92
7.1	Chapter 4	92
7.2	Chapter 5	93
7.3	Chapter 6	94
7.4	Chapter 7	95
7.5	Chapter 8	96
7.6	Chapter 10	97
7.7	Chapter 11	98
7.8	Chapter 12	99
7.9	Chapter 13	100
8.	Appendix	101
8.1	Time Measurement Equipment – Setup Procedure	101
8.2	DP Testers Setup	127
8.3	Troubleshooting CV System (MR system)	153
8.4	Chapter 7 Linux Host	157
8.4.1	Set Up Remote Windows Machine.....	157
8.4.2	Running The Test.....	158
8.5	Linux usb4_hi Driver	162
8.5.1	Load usb4_hi Driver Manually	162
8.5.2	Re-build usb4_hi Driver.....	163
8.5.3	Driver Signature for usb4_hi Driver	166

1. Required Hardware and Software

Software	Download Link and Vendor Contact
USB4 Command Verifier	Download Installer Here
USB Protocol Suite	Download Installer Here Mike Micheletti mike.micheletti@teledyne.com Teledyne Lecroy Support psgsupport@teledynelecroy.com
proFPGA-2020C-install.exe	<i>*Contact proFPGA for this software</i>
Files to use with USB4CV	https://usb.org/compliancetools#anchor_usb4tools
UCD-500 Gen2 SW Bundle	https://www.unigraf.fi/downloads/
M42De	https://www.quantumdata.com/downloads.html

Hardware	Vendor Contact
USB4 CV System Comes with USB4 Host Card installed	USB-IF
Comes with the following necessary cables: <ul style="list-style-type: none"> Teledyne Lecroy USB4 C-C 0.3m 40 Gbps, E-marked Teledyne Lecroy USB4 C-C 0.3m 40 Gbps, (Unmarked) Teledyne LeCroy USB4 C-C 0.3m 40Gbps Vconn Thru 	Voyager M4x Protocol Analyzer Mike Micheletti mike.micheletti@teledyne.com Teledyne Lecroy Support psgsupport@teledynelecroy.com
USB4 40Gbps Gen3 Type-C Cable - USB-IF Certified Accell USB4 40Gbps Male to Male Cable, 8K/5K/4K, 100W PD, Compatible with USB 3.2/2.0 and Thunderbolt 3, 2.6ft / 0.8m	https://www.amazon.com/gp/product/B08MT81KYG/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1
Belkin Thunderbolt™ 3 Cable (USB-C™ to USB-C) (100W) (6.5ft/2m)	https://www.belkin.com/us/p/P-F2CD085/
Moshi Cable	Amazon.com: Moshi USB C to DisplayPort Cable 1.5m/5ft, Support 5K@60 Hz, 4K HDR, VESA Certified, Aluminum Housing, Thunderbolt 3 Compatible (Ensure Monitors Accept DisplayPort Signal) : Electronics
HDMI cable	
DP-to-DP cable	Club 3D DisplayPort 1.4 to DisplayPort 1.4 M/M 1m/3.28ft
Satechi Hub (Satechi Thunderbolt 4 Slim Hub 5-in-1, USB C 60W Charging, Single 8k or Dual 4K Display, 4 Thunderbolt 4 Ports, USB 3.2 Gen2)	https://www.amazon.com/gp/product/B0BM1Q38T7/ref=ppx_yo_dt_b_asin_title_o07_s00?ie=UTF8&th=1

MCCI Model 3411	https://store.mcci.com/products/model-3411
HP 3YE87UT#ABA Smart Buy TB Dock 120W G2 W, Black,	https://www.amazon.com/gp/product/B07DP5QBDV/ref=ppx_yo_dt_b_asin_title_o04_s00?ie=UTF8&psc=1
HP Thunderbolt Dock 120W G4 (4J0A2AA) Docking Station Port Replicator	https://www.amazon.com/dp/B09Y71WMSP?psc=1&smid=AMSSWE236JND5&ref_=chk_typ_imgToDp
SAMSUNG X5 Portable SSD 500GB - Up to 2800MB/s -Thunderbolt 3 NVMe External Solid State Drive, Gray/Red (MU-PB500B/AM)	https://www.amazon.com/gp/product/B07GBGG68L/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&psc=1
USB-C Digital AV Multiport Adapter	https://www.apple.com/shop/product/MUF82AM/A/usb-c-digital-av-multiport-adapter
Goodway Gatkex Board with TMU CLK Out Board (DBD1210L1)	Good Way Technology Co., Ltd. - https://www.goodway.com.tw/ Contact: raymond_chan@goodway.com.tw, jo_lee@goodway.com.tw
Quad Motherboard (proFPGA)	https://www.profpga.com/products/motherboards-overview/profpga-quad
Virtex® 7 module XC7V2000T	https://www.profpga.com/products/fpga-modules-overview/virtex-7-based/profpga-xc7v2000t
UCD500 Gen2	UCD-500 Gen2 • DisplayPort 2.1 Generator & Analyzer Unigraf
M42De	https://www.teledynelecroy.com/protocolanalyzer/quantumdata-m42de.aspx?gad_source=1&gclid=EAIaIQobChMIp4LDs suYiQMvVvNHCBB3P6xrFEAAAYASAAEgLsSvD BwE
ASUS (ProArt Display PA329CRV)	https://www.asus.com/us/displays-desktops/monitors/proart/proart-display-pa329crv/

2. Setting Up USB4 CV On Windows System

2.1 Installing USB4 CV

Install the USB4 CV from usb.org website. Note, The USB4CV installer will install both the Graphical User Interface version (gui) and the Command Line Interface version (cli).

2.2 Host Settings

After installing USB4 CV, please also make sure the following Windows Settings are set on the Host:

- User Account Control >>> Set to **Never notify**
- Power & Sleep >>> Set **Never** for Screen and for Sleep
(Host should never turn off display or go to sleep automatically)

2.3 Installing USB Protocol Suite

After installing USB4 CV, go to **C:\Program Files\ USB-IF Test Suite\USB4CV\Documents** and open the **Release Notes – USB4CV** document. In this document, it will mention the LeCroy version that is required for that specific USB4 CV release. Install the USB Protocol Suite from Teledyne LeCroy.

2.4 USB4 CV Settings

Go to **C:\Program Files\ USB-IF Test Suite\USB4CV\Documents** and open the **USB4 CV Readme** document. This document has a couple of settings that users must make to make sure that USB4 CV will be running correctly. Also, it mentions the path to where the usbif.json file should be saved.

2.5 Launching USB4CV

2.5.1 Graphical User Interface (GUI)

In the windows search bar, type **USB4CV – USB4 Command Verifier** and open the software.

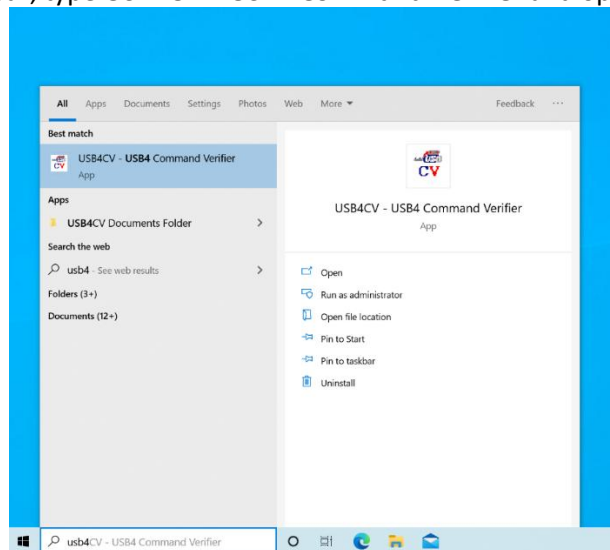


Figure 2.1: Launching USB4 CV

Select the host controller to load the Compliance Driver Stack on

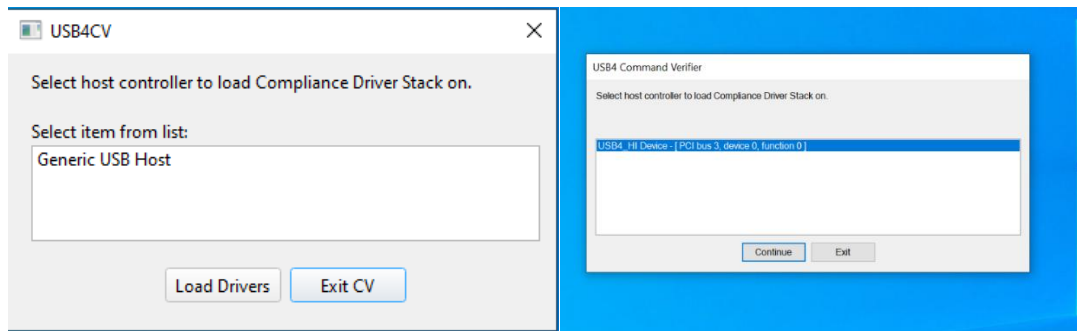


Figure 2.2: Selecting Host Controller

Go to the Device Manager and make sure the compliance driver is there. It should be automatically installed when the program runs.

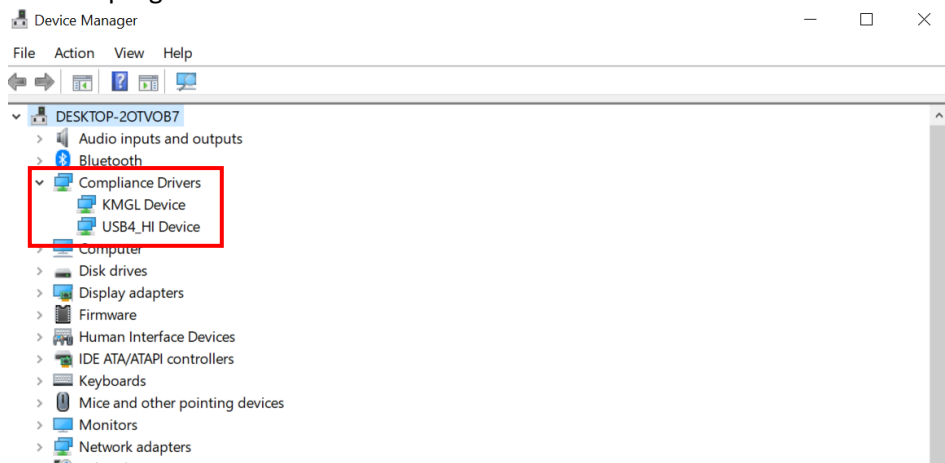


Figure 2.3: The Compliance Driver

Note, it should automatically re-install the Windows driver stack when the program exists. If USB4_Hi device is not seen in the device manager after launching USB4 CV, then user must manually install driver. Check **Manually Install USB4_Hi Driver** below.

Note, in case that the USB4 Host system has two active host controllers. If there is more than one compliance driver loaded, then users will have to exit USB4CV and disable the host controller that is not being used while running the test.

2.5.2 Command Line Interface (CLI)

Users must run command prompt as administrator

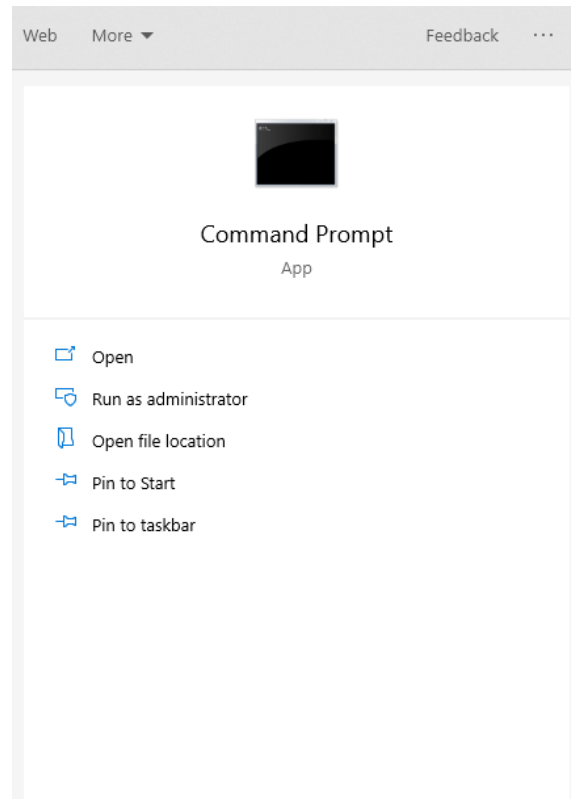


Figure 2.4: Command Line Interface

Go to the USB4CV directory: `cd C:\Program Files\USB-IF Test Suite\USB4CV`

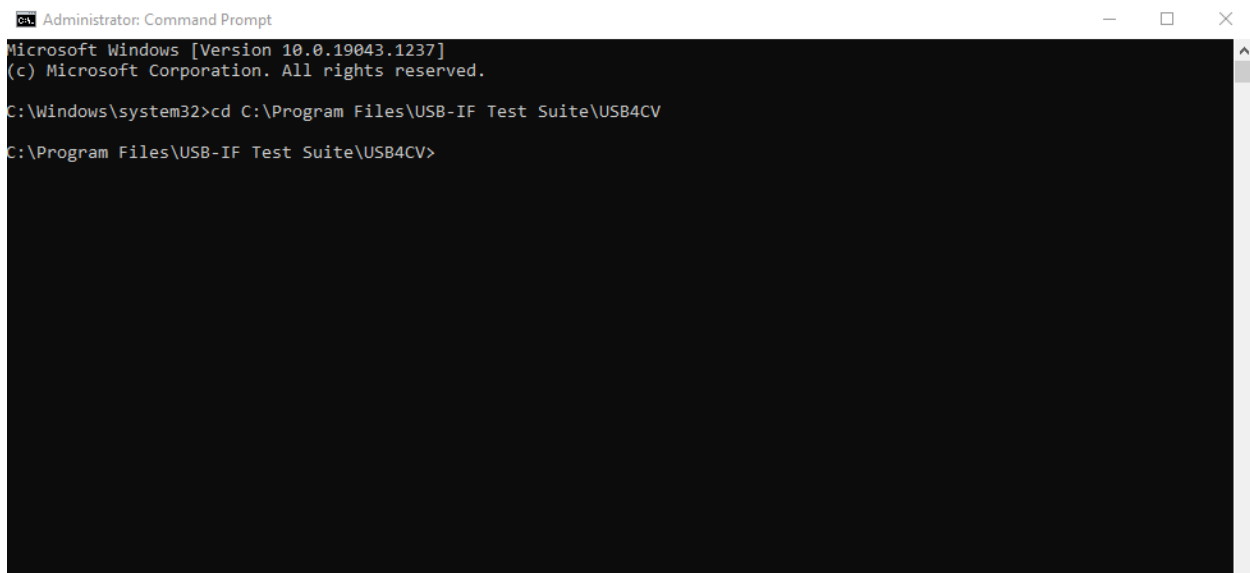
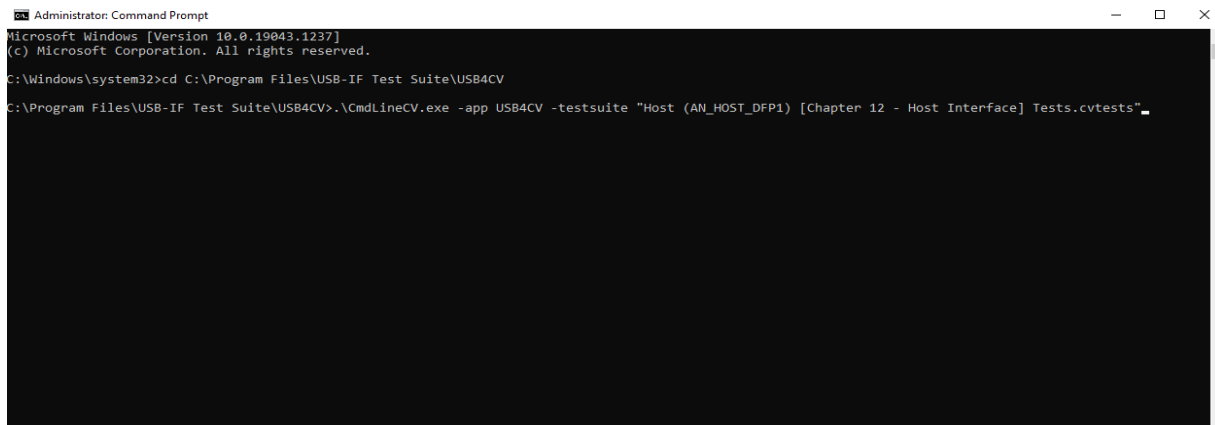


Figure 2.5: USB4CV Directory

To run the tests that don't need VIF file, run: `.\CmdLineCV.exe -app USB4CV -testsuite "Test name"`



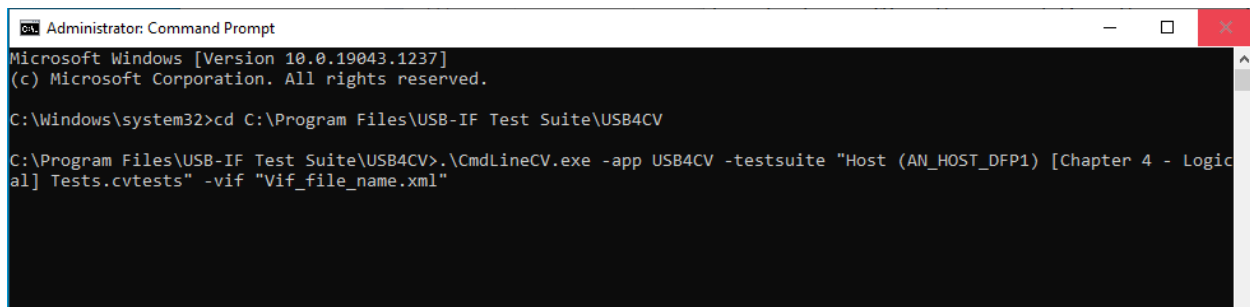
```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\USB-IF Test Suite\USB4CV
C:\Program Files\USB-IF Test Suite\USB4CV>.\CmdLineCV.exe -app USB4CV -testsuite "Host (AN_HOST_DFP1) [Chapter 12 - Host Interface] Tests.cvtests"
```

Figure 2.6: Run USB4CV Test without VIF file

To run the tests that need a VIF file, if the VIF file is in the same directory, run:

`.\CmdLineCV.exe -app USB4CV -testsuite "Test name" -vif "VIF name"`



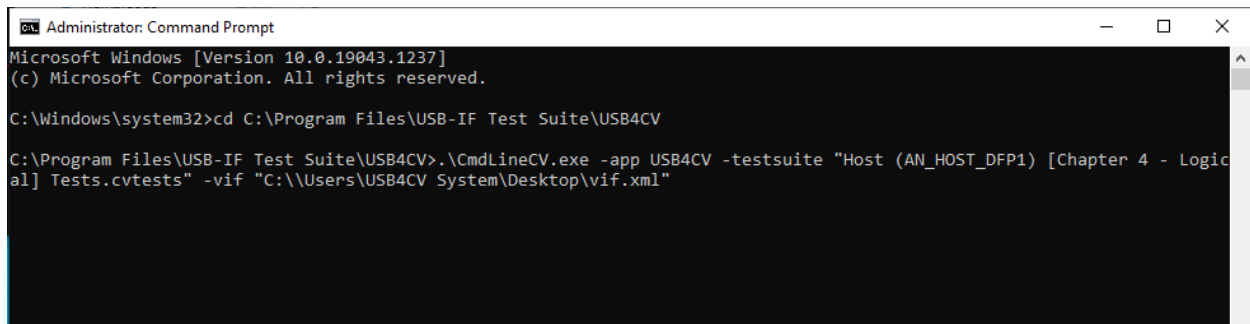
```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\USB-IF Test Suite\USB4CV
C:\Program Files\USB-IF Test Suite\USB4CV>.\CmdLineCV.exe -app USB4CV -testsuite "Host (AN_HOST_DFP1) [Chapter 4 - Logical] Tests.cvtests" -vif "Vif_file_name.xml"
```

Figure 2.7: VIF in The Same Directory

If the VIF file is in the specific directory, run:

`.\CmdLineCV.exe -app USB4CV -testsuite "Test name" -vif "\path\to\vif.xml"`



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\USB-IF Test Suite\USB4CV
C:\Program Files\USB-IF Test Suite\USB4CV>.\CmdLineCV.exe -app USB4CV -testsuite "Host (AN_HOST_DFP1) [Chapter 4 - Logical] Tests.cvtests" -vif "C:\\Users\USB4CV System\Desktop\vif.xml"
```

Figure 2.8: VIF is Somewhere Else

2.6 Manually Install USB4_Hi Driver

If you don't see the USB4_HI Device, then you must manually install the driver. Go to file explorer and select Local Disk C select then on top of the file explorer, press on "View" and check "Hidden Items." You should be able to see "ProgramData."

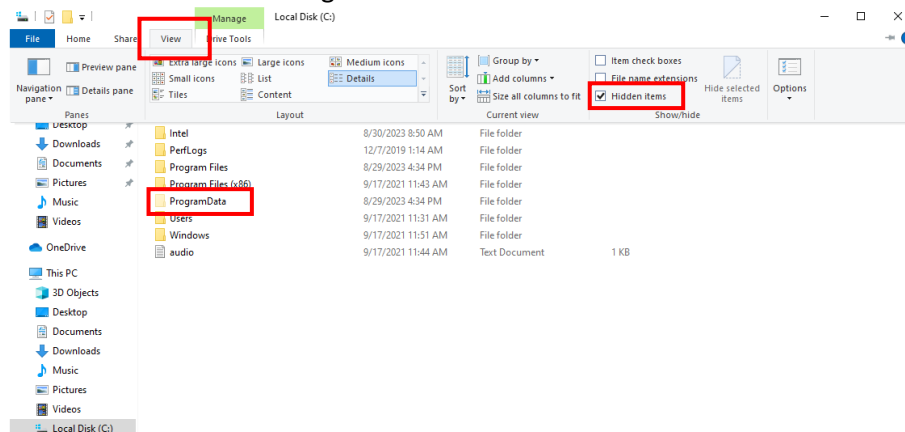


Figure 2.9: View Hidden Item

Now, go back to device manager, view → Devices by connection, scroll down and locate the controller and press on "update driver". Note this is on our MR system.

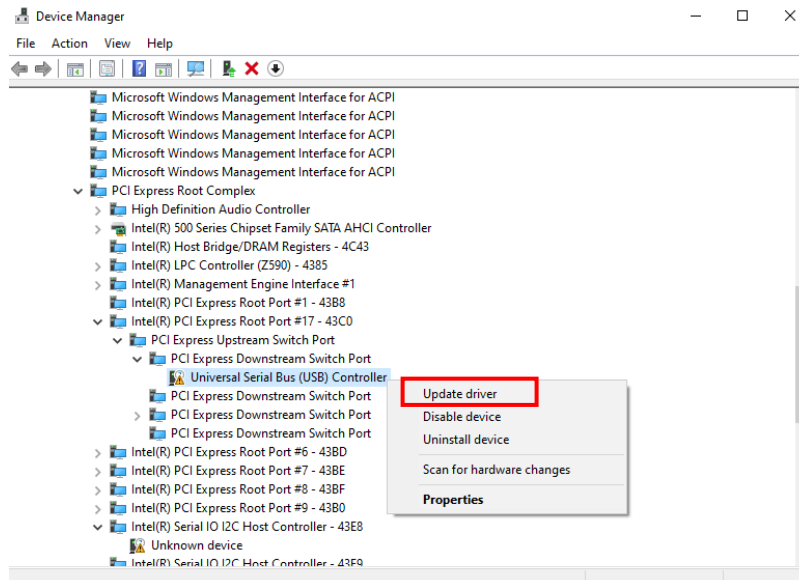


Figure 2.10: Update Driver

Select “Browse...”

Drivers are included for both x64 and ARM64 architectures. You must install the driver for the architecture you are running on. It will be one of the following directories:

C:\ProgramData\USB-IF Test Suite\USB4CV\USB4 HI Driver\ARM64

Or

C:\ProgramData\USB-IF Test Suite\USB4CV\USB4 HI Driver\x64.

Note that the previous location for the x64 driver was C:\ProgramData\USB-IF Test Suite\USB4CV\USB4 HI Driver.

This example is for an x64 system:

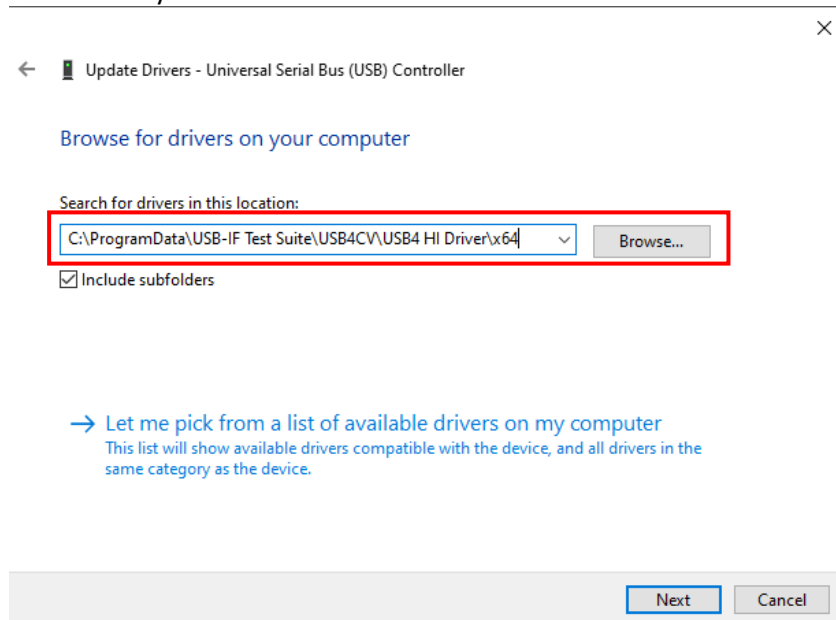


Figure 2.11: Choose The Correct Path

This shows the “USB4_HI Device” driver installed.

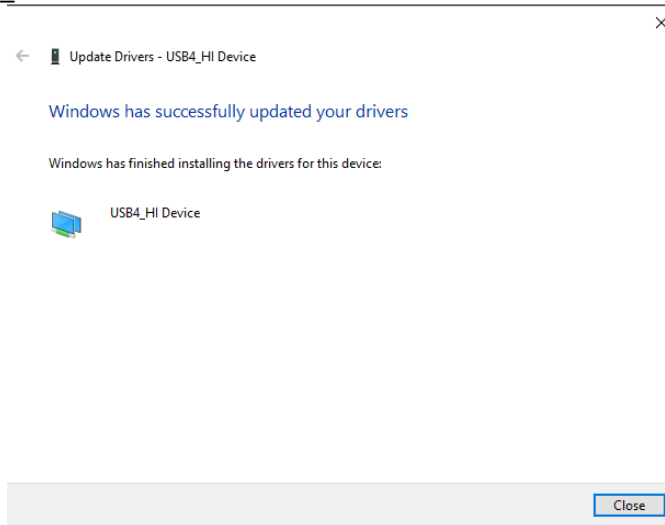


Figure 2.12: Driver is Installed

2.7 USB4_Hi Device & KMGL Device Version

While USB4 CV is being launched, go to Device Manager, and check the version for both USB4_Hi Device & KMGL Device in the compliance driver.

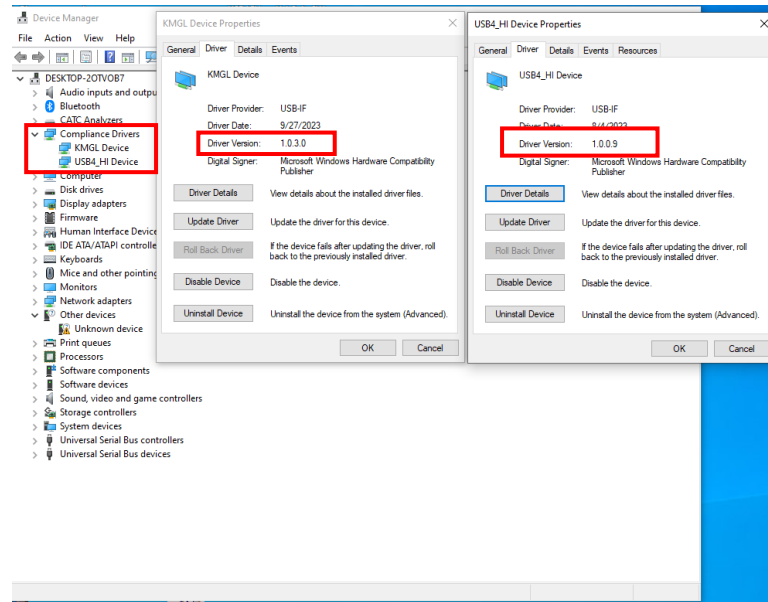


Figure 2.13: USB4_Hi Driver & KMGL Driver Versions

To confirm that it's the correct drivers' version, go to **C:\Program Files\ USB-IF Test Suite\USB4CV\Documents** and open the **Release Notes – USB4CV** document.

2.8 USB4CV Interface Window

Now, the USB4CV is ready to be used.

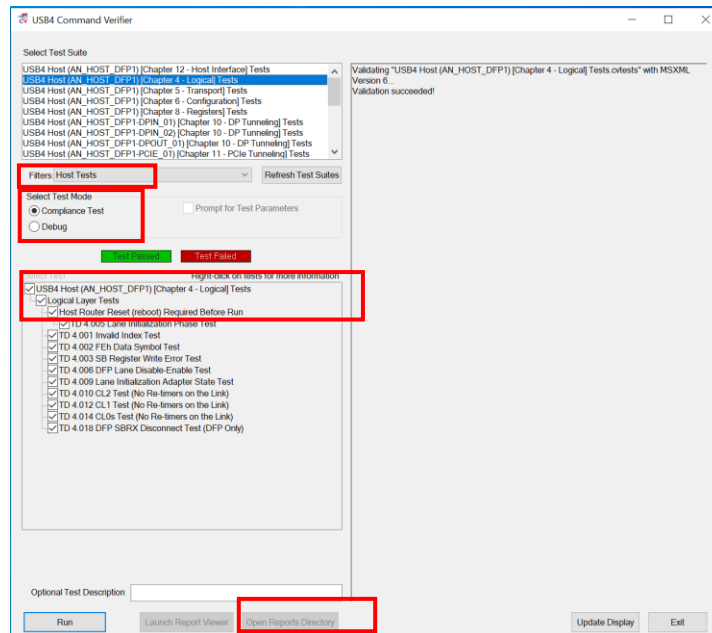


Figure 2.14: USB4 CV

1. You can filter the tests by type for Example, by host/device/hub/analyzer/exerciser.
2. For the Select Test Mode, this option lets the user either run the whole test together or only run a test selected by the user (Multiple tests from the same chapter can be selected)
3. Some tests require reboot, so users are expected to reboot the system before running this test (Figure 2.14)
4. Users can name the results log by writing in the Optional Test Description box.
5. To check the log and traces, users can click on **Open Report Directory** or go to **Document\USB-IF Test Suite\CV Reports\USB4CV**
6. Note, AN stand for Analyzer. EX stands for exerciser. DC stands for Direct Connect (No M4x needed)

Once a test is selected, for example in figure 2.14, chapter 4 analyzer host test is selected. Now, go to the CTS and check for Chapter 4 Analyzer Host setup (AN_HOST_DFP1). Once the setup is finished and everything is connected, press run in the USB4 CV.

3. Setting Up USB4CV on Linux System

3.1 Download & Install Linux USB4CV

Download either the command-line (usb4cv-cli.tar.gz) or gui (usb4cv-gui.tar.gz) archive from [here](#).

- Extract the files from usb4cv-cli.tar.gz into the directory *usb4cv-cli* in the Documents directory (this can go in any directory).
 - On the command line, run: **`tar -xf usb4cv-cli.tar.gz`**
- Extract the files from usb4cv-gui.tar.gz into the *usb4cv-gui* directory in the Documents directory (this can go in any directory).

Note, if users want to use the GUI version only, then downloading the usb4cv-cli is not necessary and vice versa. In this document, the usb4cv-cli and usb4cv-gui have been extracted in the Documents directory

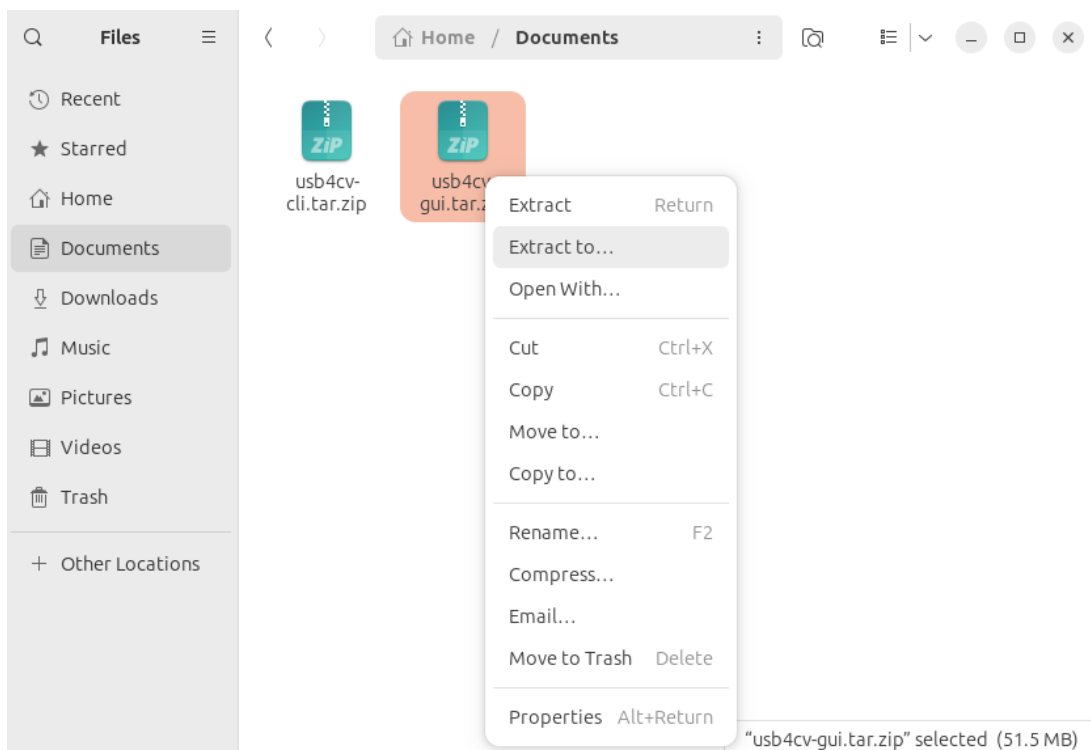


Figure 3.1: Download usb4cv

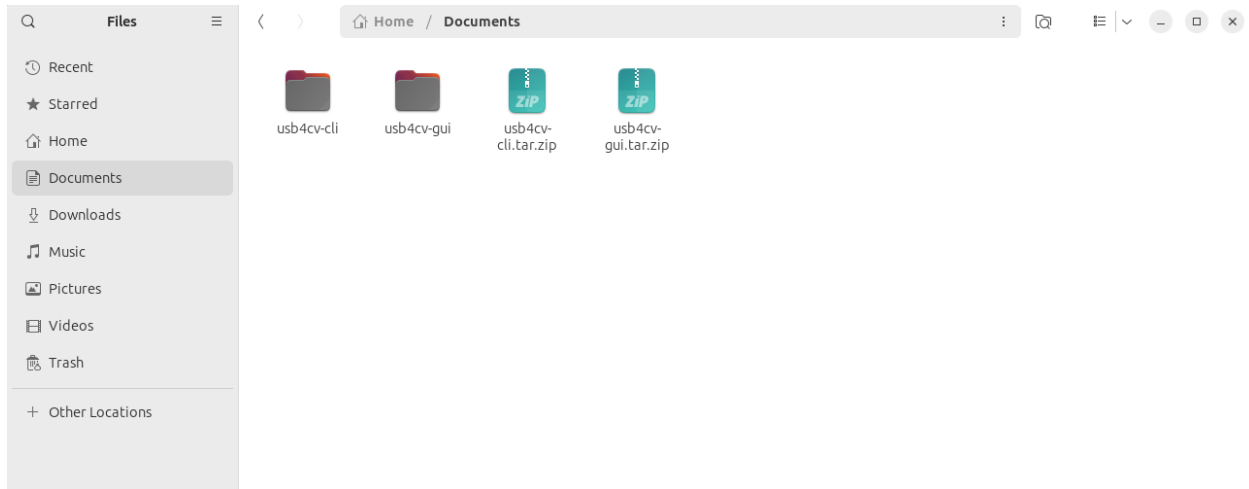


Figure 3.2: Extract The Zipped Files

3.2 Setting up USB4CV

3.2.1 USB4CV GUI Version

Before running setup:

- Make sure the system is connected to the internet.
- You will need root permission to install.
- Some test suites also require root permissions.

Open the *usb4cv-gui* folder and right click inside the folder and select “Open in Terminal”

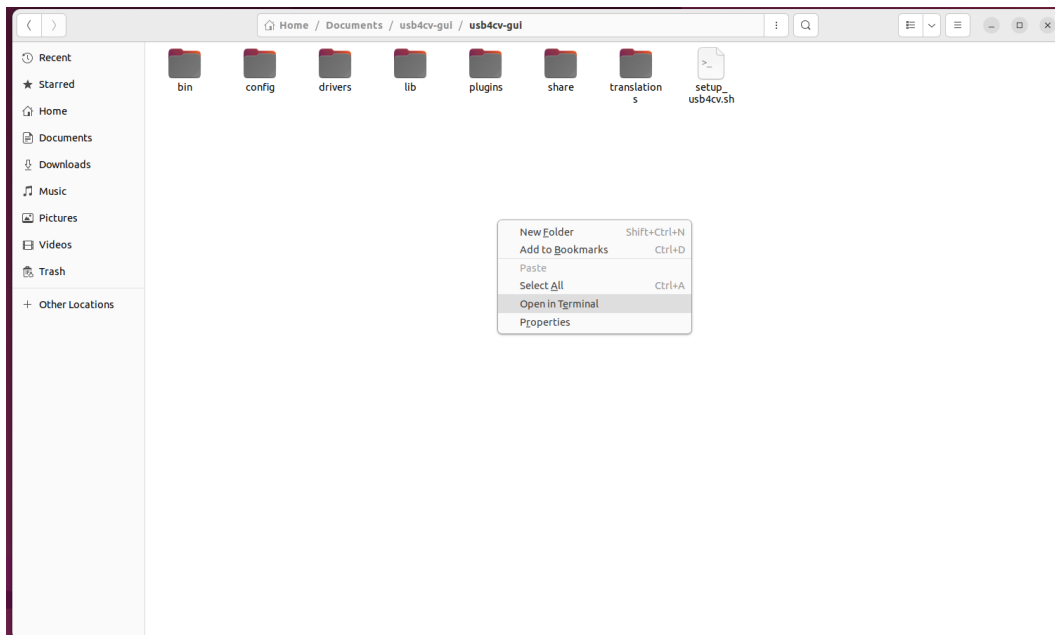


Figure 3.3: Open Terminal

Once the terminal is opened, run `./setup_usb4cv.sh`

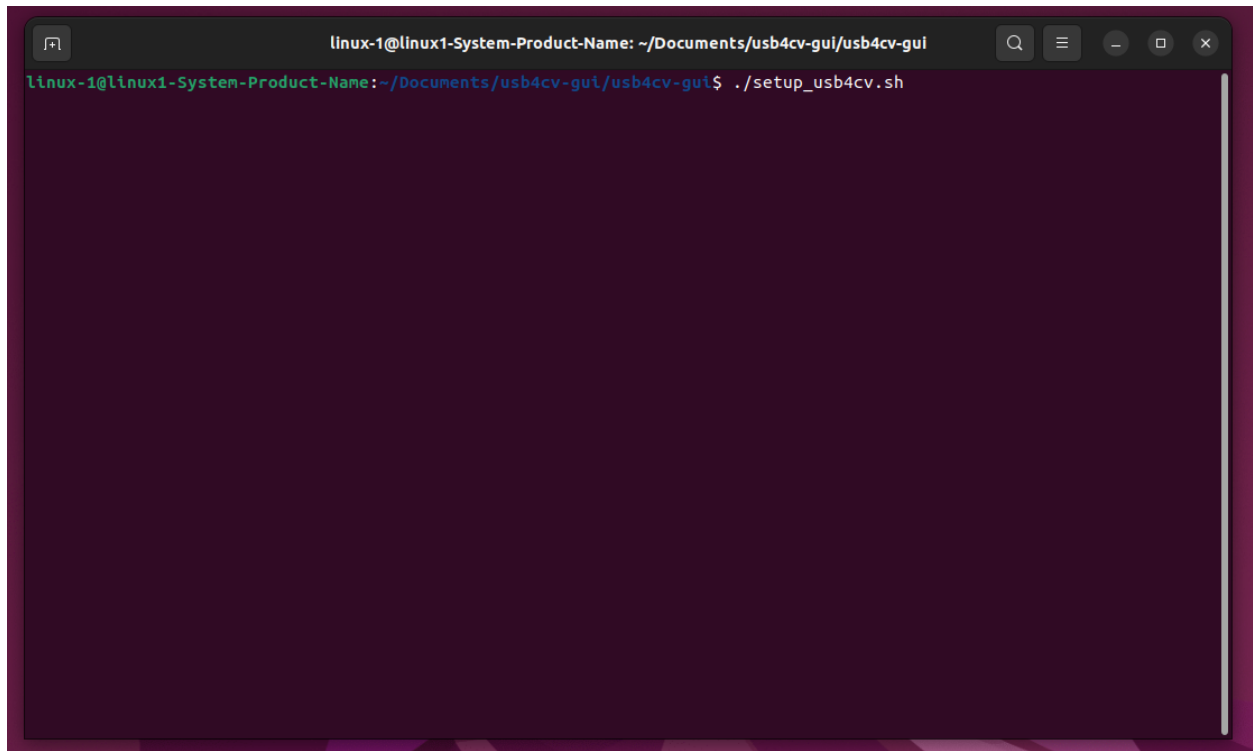


Figure 3.4: Run `./setup_usb4cv.sh`

Enter password:

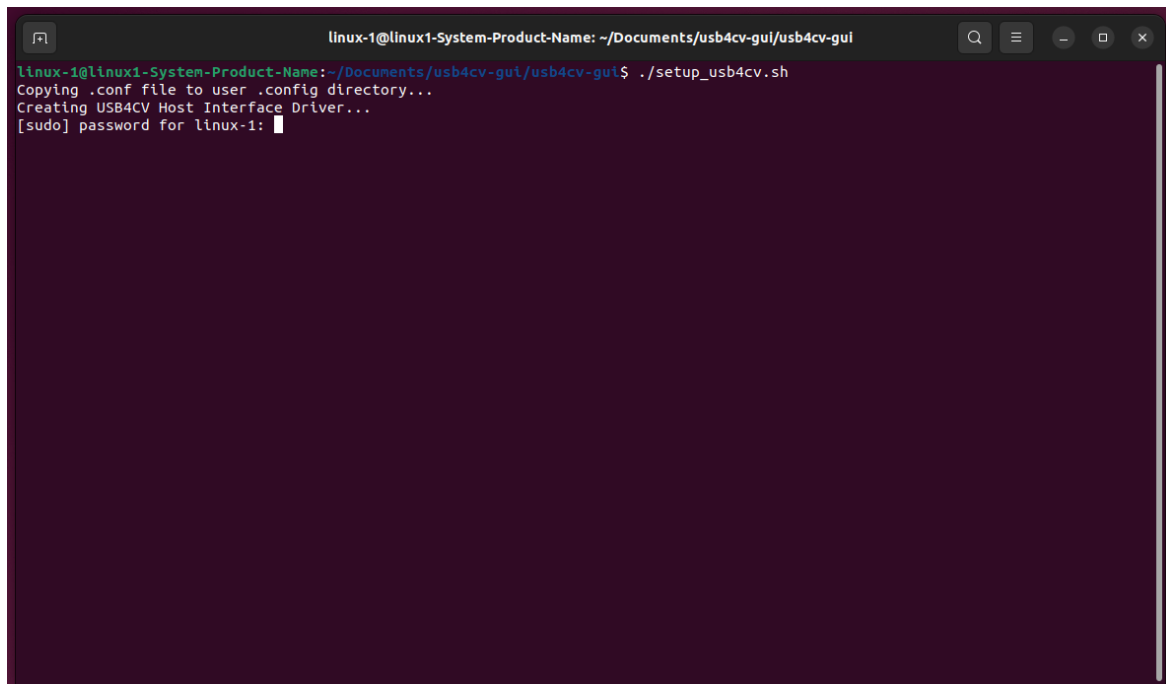
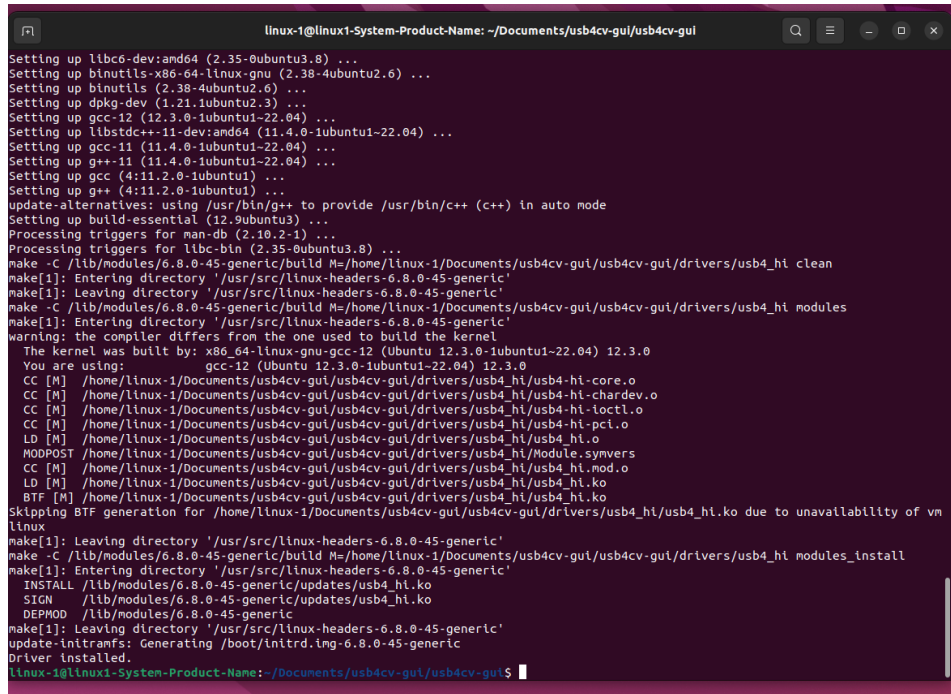


Figure 3.5: Enter password

Once the script finishes, make sure there is no error message at the end



```
linux-1@linux1-System-Product-Name: ~/Documents/usb4cv-gui/usb4cv-gui
Setting up libcc-dev:amd64 (2.35-0ubuntu3.8) ...
Setting up binutils-x86_64-linux-gnu (2.38-4ubuntu2.6) ...
Setting up binutils (2.38-4ubuntu2.6) ...
Setting up dpkg-dev (1.21.1ubuntu2.3) ...
Setting up gcc-12 (12.3.0-1ubuntu1-22.04) ...
Setting up libstdc++-11-dev:amd64 (11.4.0-1ubuntu1-22.04) ...
Setting up gcc-11 (11.4.0-1ubuntu1-22.04) ...
Setting up g++-11 (11.4.0-1ubuntu1-22.04) ...
Setting up gcc (4:11.2.0-1ubuntu1) ...
Setting up g++ (4:11.2.0-1ubuntu1) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.9ubuntu3) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
make -C /lib/modules/6.8.0-45-generic/build M=/home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi clean
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-45-generic'
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-45-generic'
make -C /lib/modules/6.8.0-45-generic/build M=/home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi modules
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-45-generic'
warning: the compiler differs from the one used to build the kernel
The kernel was built by: x86_64-linux-gnu-gcc-12 (Ubuntu 12.3.0-1ubuntu1-22.04) 12.3.0
You are using: gcc-12 (Ubuntu 12.3.0-1ubuntu1-22.04) 12.3.0
CC [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi-core.o
CC [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi-chardev.o
CC [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi-ioctl.o
CC [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi-pci.o
LD [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi.o
MODPOST /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/Module.symvers
CC [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi.mod.o
LD [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi.ko
BTF [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi.ko
Skipping BTF generation for /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-45-generic'
make -C /lib/modules/6.8.0-45-generic/build M=/home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi modules_install
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-45-generic'
INSTALL /lib/modules/6.8.0-45-generic/updates/usb4_hi.ko
SIGN /lib/modules/6.8.0-45-generic/updates/usb4_hi.ko
DEPMOD /lib/modules/6.8.0-45-generic
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-45-generic'
update-initramfs: Generating /boot/initrd.img-6.8.0-45-generic
Driver installed.
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-gui/usb4cv-gui$
```

Figure 3.6: The Script is Finished

To confirm the script ran correctly, go to “Home” and enable “Show Hidden Files”

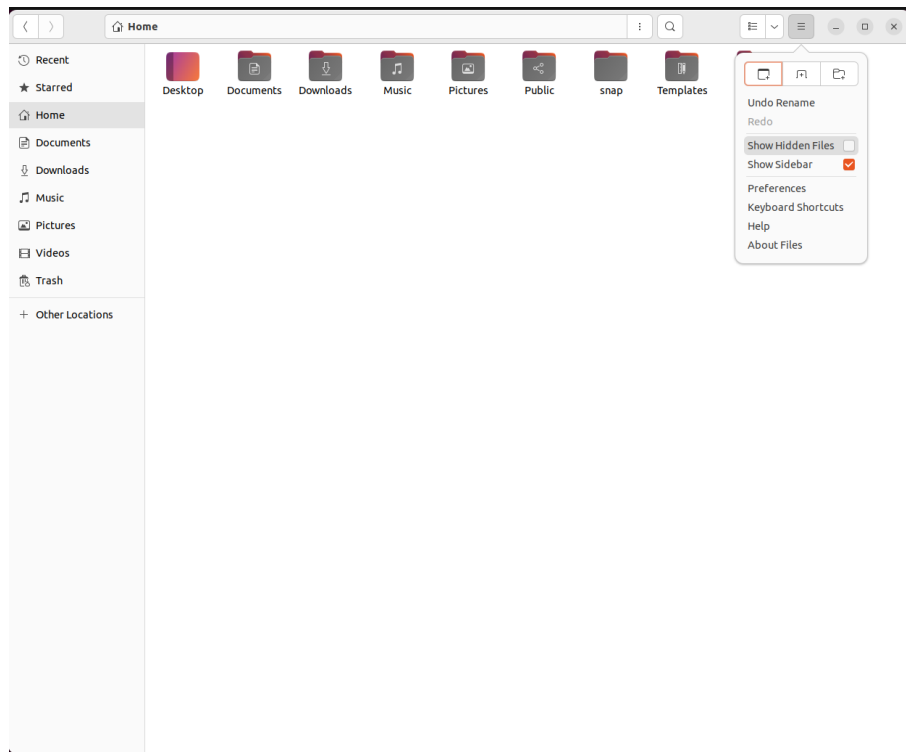


Figure 3.7: Show Hidden Files

Once the hidden files are enabled, there should be a “.config” file

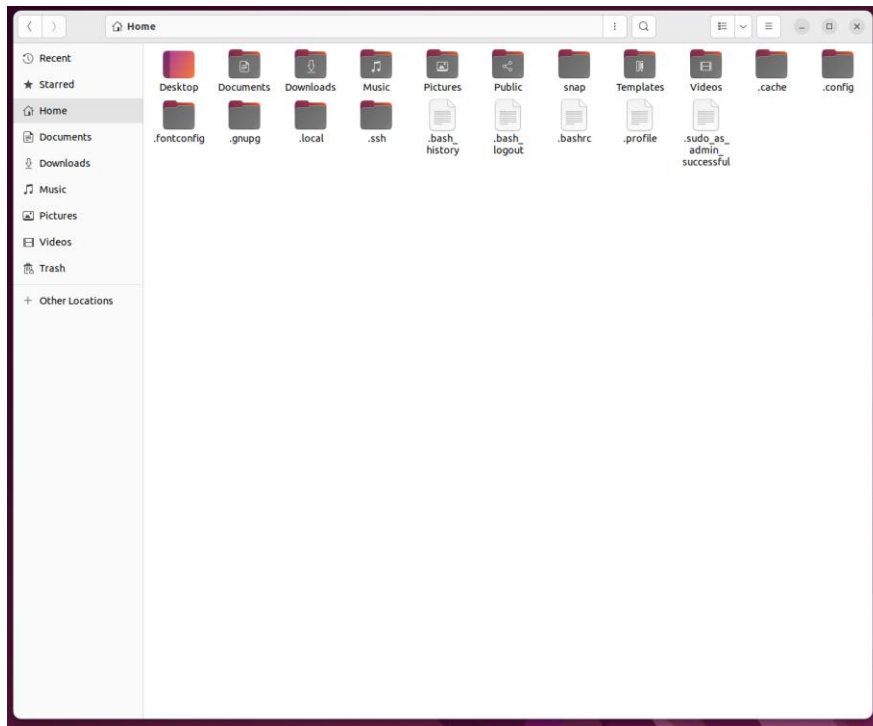


Figure 3.8: Look For .config file

In Home, go to “.config” file. There should be a *USB-IF.config* file and *USB-IF* folder

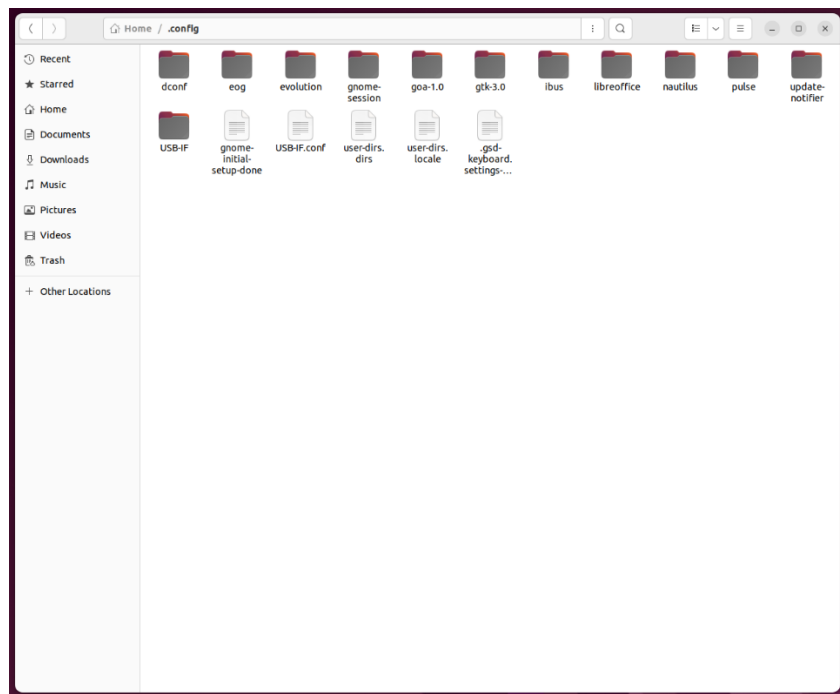


Figure 3.9: Look for USB-IF.config file

Before launching USB4CV, usb4_hi driver must be loaded.

Go to `~/usb4cv-gui/usb4cv-gui/drivers` and right click in the folder, and select “Open in Terminal”

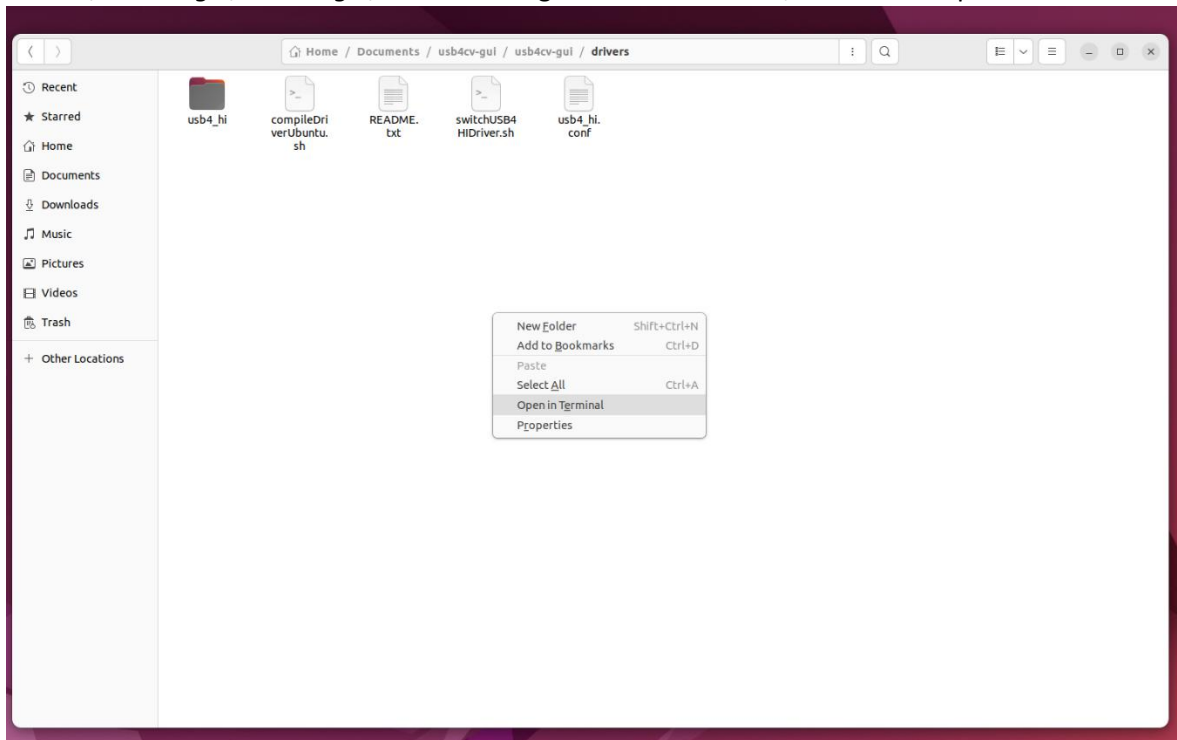


Figure 3.10: Load Driver

Run `sudo ./switchUSB4HIDriver.sh usb4_hi`

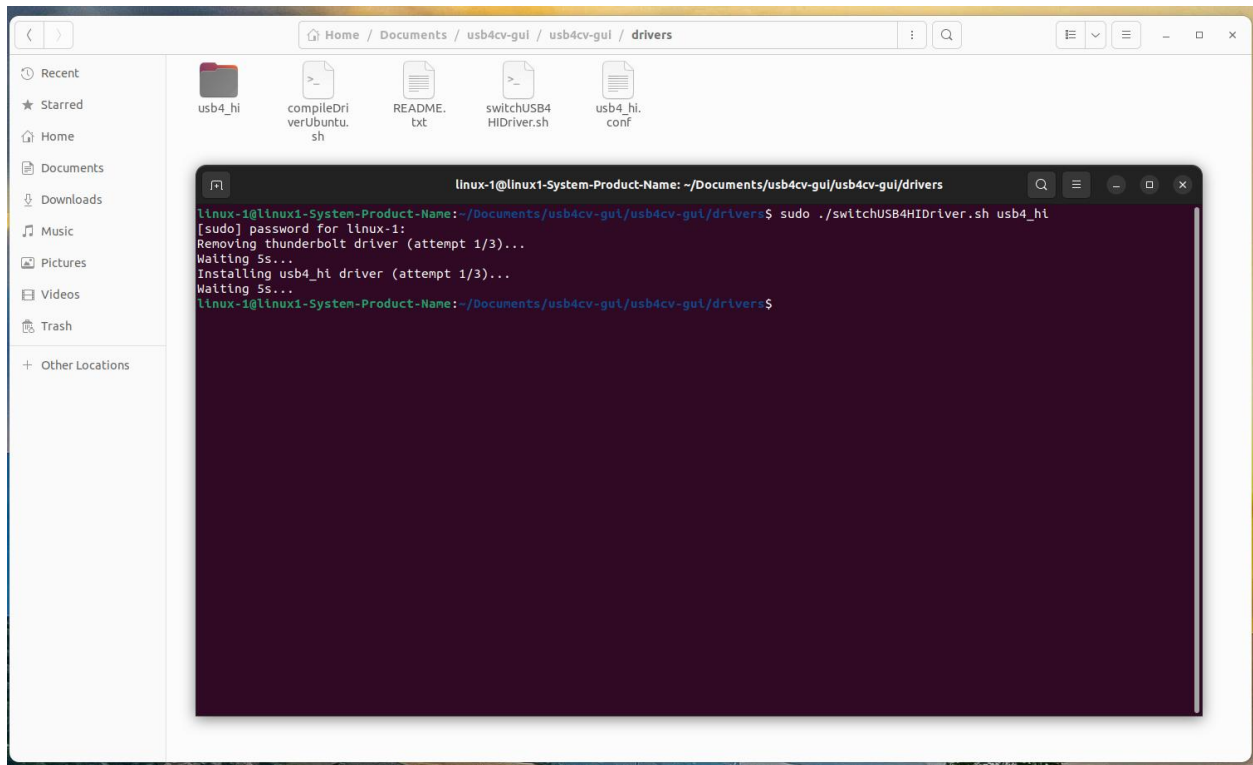


Figure 3.11: Run Script to Load Driver

To confirm the driver is loaded, run the following command: `sudo dmesg | tail`

User is expected to see the below (note, the number of paths supported is based on the host):

```
usb4_hi 0000:03:00.0: 12 paths supported
usb4_hi 0000:03:00.0: driver initialized
```

Note: Linux USB4CV does not include a KMGL driver. The KMGL driver is only present on Windows USB4CV.

```
linux-1@linux1-System-Product-Name: ~/Documents/usb4cv-gui/usb4cv-gui/drivers
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-gui/usb4cv-gui/drivers$ sudo dmesg | tail
[ 51.999275] audit: type=1400 audit(1730238980.658:63): apparmor="DENIED" operation="capable" class="cap" prof
ile="/usr/lib/snapd/snap-confine" pid=2084 comm="snap-confine" capability=12 capname="net_admin"
[ 51.999280] audit: type=1400 audit(1730238980.658:64): apparmor="DENIED" operation="capable" class="cap" prof
ile="/usr/lib/snapd/snap-confine" pid=2084 comm="snap-confine" capability=38 capname="perfmon"
[ 113.985958] xhci_hcd 0000:37:00.0: XHC error in resume, USBSTS 0x401, Reint
[ 113.985962] usb usb3: root hub lost power or was reset
[ 113.985963] usb usb4: root hub lost power or was reset
[ 113.993415] ACPI: bus type thunderbolt unregistered
[ 119.022275] usb4_hi: loading out-of-tree module taints kernel.
[ 119.022278] usb4_hi: module verification failed: signature and/or required key missing - tainting kernel
[ 119.022820] usb4_hi 0000:03:00.0: 12 paths supported
[ 119.022906] usb4_hi 0000:03:00.0: driver initialized
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-gui/usb4cv-gui/drivers$
```

Figure 3.12: Confirm The Driver is Loaded

Users must download `usbif.json` file from usb.org website. Once `usbif.json` file is downloaded, go to `usb4cv-gui/documents` and open `Linux_Readme_USB4CV` text file which will mention the path to save the `usbif.json` file.

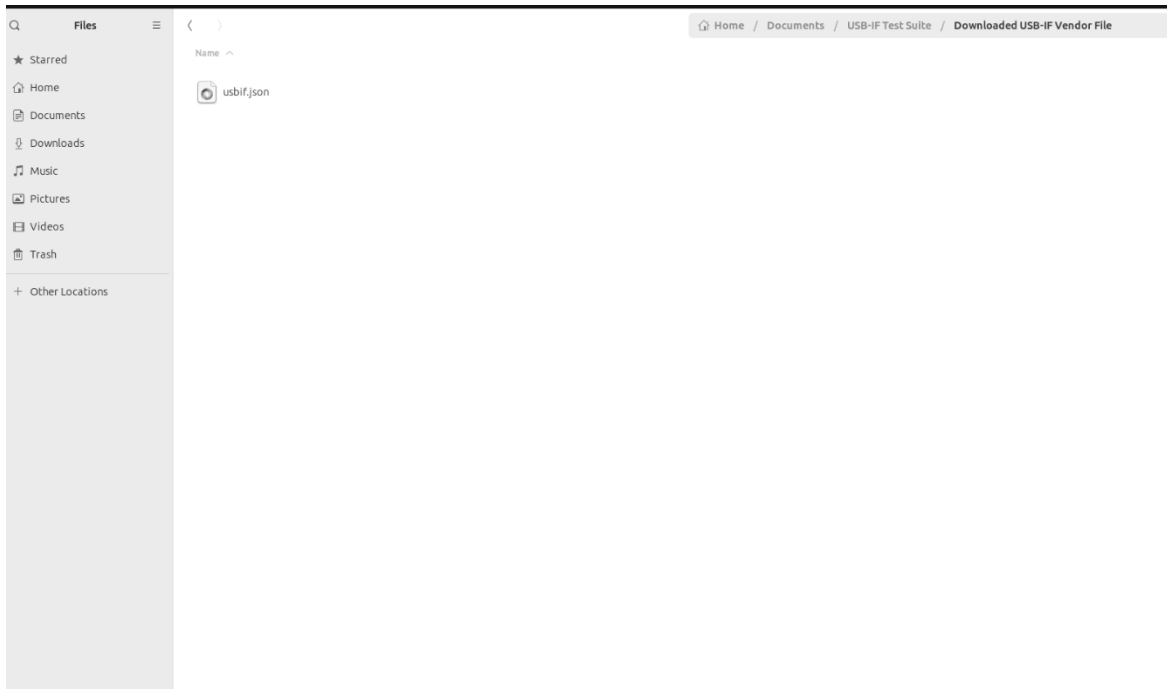


Figure 3.13: Save `usbif.json` file

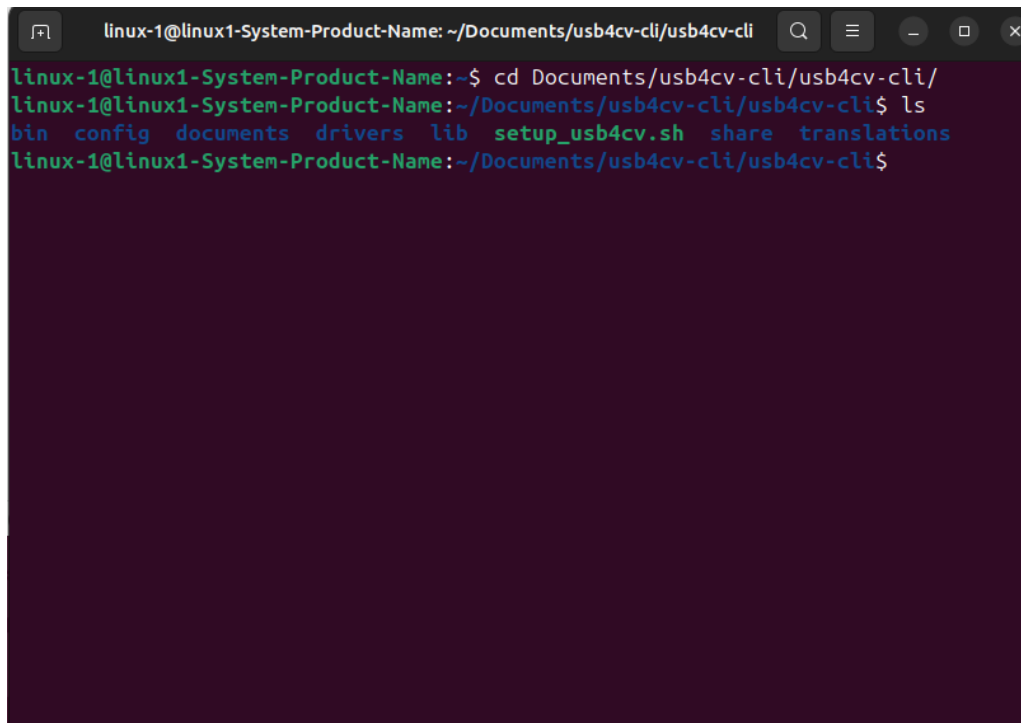
Note, go to chapter 5 to launch USB4CV.

3.2.2 USB4CV Command Line Version

Before running setup:

- Make sure the system is connected to the internet.
- You will need root permission to install.
- Some test suites also require root permissions.

To use the command line, open the Terminal and go to `/Documents/usb4cv-cli/usb4cv-cli`.

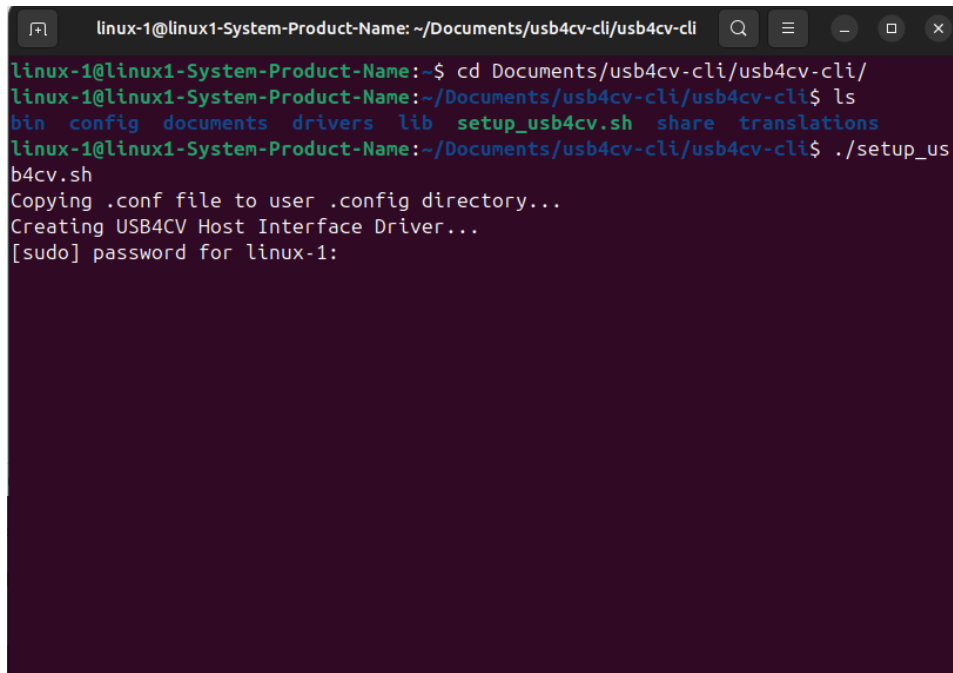
A terminal window with a dark background and light text. The title bar shows the user 'linux-1' on a system named 'linux1-System-Product-Name' at the directory '~/Documents/usb4cv-cli/usb4cv-cli'. The terminal content shows the user navigating to the directory with 'cd Documents/usb4cv-cli/usb4cv-cli/' and then listing the contents with 'ls', which outputs 'bin config documents drivers lib setup_usb4cv.sh share translations'.

```
linux-1@linux1-System-Product-Name: ~/Documents/usb4cv-cli/usb4cv-cli
linux-1@linux1-System-Product-Name:~$ cd Documents/usb4cv-cli/usb4cv-cli/
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli$ ls
bin  config  documents  drivers  lib  setup_usb4cv.sh  share  translations
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli$
```

Figure 3.15: Open usb4cv-cli folder

Run `./setupusb4cv.sh`

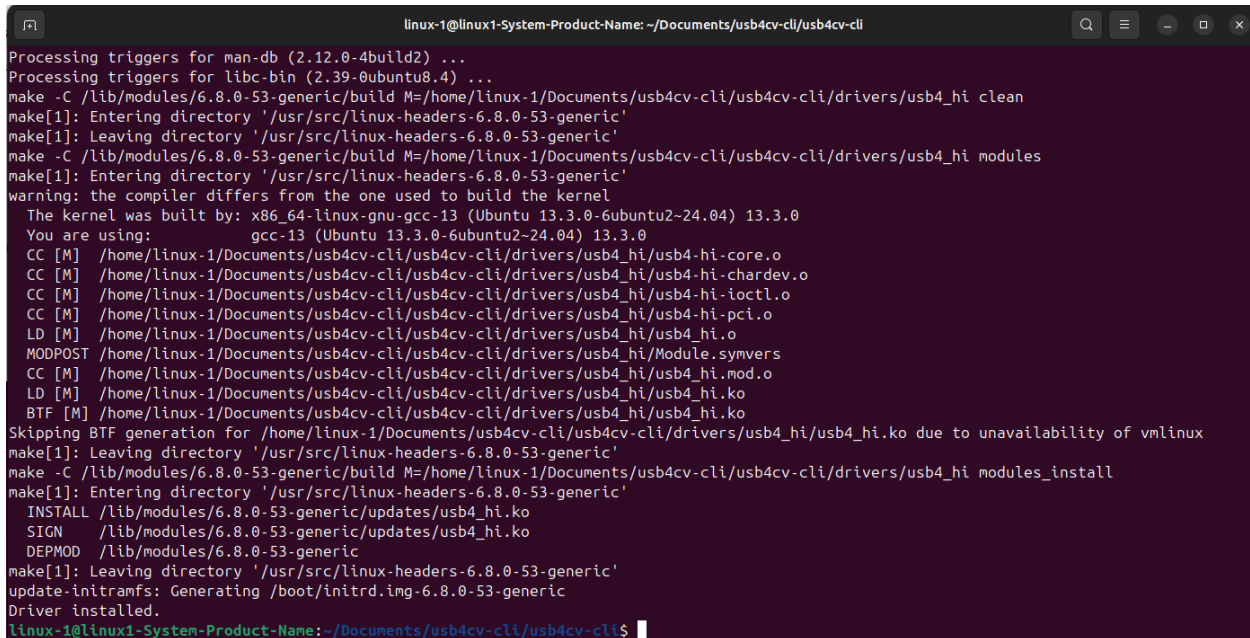
Enter the password to proceed and install the driver



```
linux-1@linux1-System-Product-Name: ~/Documents/usb4cv-cli/usb4cv-cli
linux-1@linux1-System-Product-Name:~$ cd Documents/usb4cv-cli/usb4cv-cli/
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli$ ls
bin  config  documents  drivers  lib  setup_usb4cv.sh  share  translations
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli$ ./setup_us
b4cv.sh
Copying .conf file to user .config directory...
Creating USB4CV Host Interface Driver...
[sudo] password for linux-1:
```

Figure 3.16: Enter Password

Once it's done, make sure there aren't any failures.



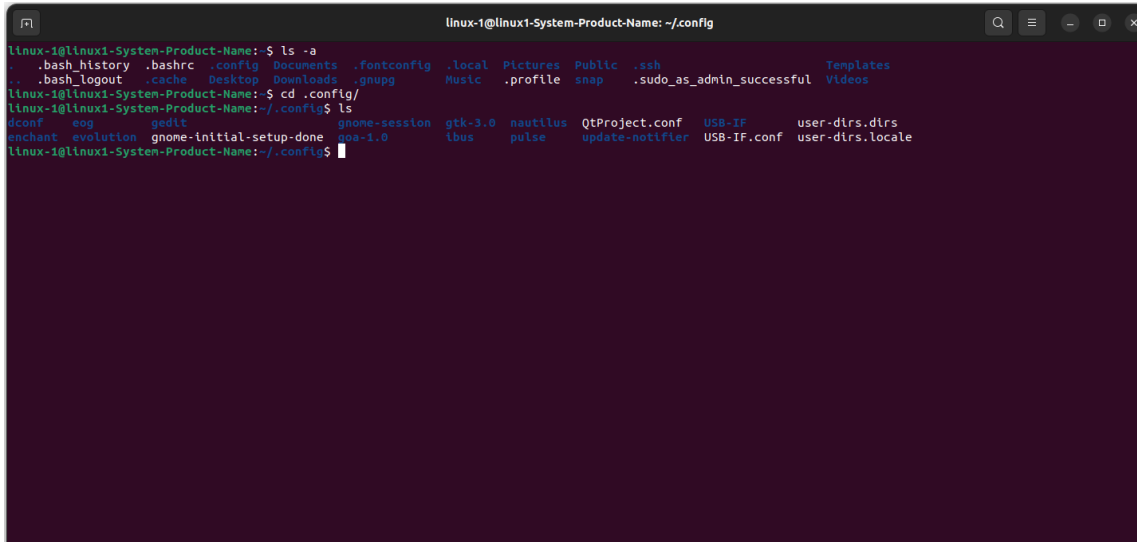
```
linux-1@linux1-System-Product-Name: ~/Documents/usb4cv-cli/usb4cv-cli
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...
make -C /lib/modules/6.8.0-53-generic/build M=/home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi clean
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-53-generic'
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-53-generic'
make -C /lib/modules/6.8.0-53-generic/build M=/home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi modules
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-53-generic'
warning: the compiler differs from the one used to build the kernel
The kernel was built by: x86_64-linux-gnu-gcc-13 (Ubuntu 13.3.0-6ubuntu2-24.04) 13.3.0
You are using:
gcc-13 (Ubuntu 13.3.0-6ubuntu2-24.04) 13.3.0
CC [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4-hi-core.o
CC [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4-hi-chardev.o
CC [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4-hi-ioctl.o
CC [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4-hi-pci.o
LD [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4_hi.o
MODPOST /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/Module.symvers
CC [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4_hi.mod.o
LD [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4_hi.ko
BTF [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4_hi.ko
Skipping BTF generation for /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4_hi.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-53-generic'
make -C /lib/modules/6.8.0-53-generic/build M=/home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi modules_install
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-53-generic'
INSTALL /lib/modules/6.8.0-53-generic/updates/usb4_hi.ko
SIGN /lib/modules/6.8.0-53-generic/updates/usb4_hi.ko
DEPMOD /lib/modules/6.8.0-53-generic
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-53-generic'
update-initramfs: Generating /boot/initrd.img-6.8.0-53-generic
Driver installed.
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli$
```

Figure 3.17: After Script Finished Running

Confirm the script ran above, copied the required file and folder to the correct directory:

- Go to Home and use `ls -a` command. This command shows the hidden files.
- Then `cd` to the `.config` folder

There will be a `USB-IF.conf` file and a `USB-IF` folder



```
linux-1@linux1-System-Product-Name: ~/.config
linux-1@linux1-System-Product-Name: ~$ ls -a
.  .bash_history  .bashrc  .config  Documents  .fontconfig  .local  Pictures  Public  .ssh  Templates
.. .bash_logout .cache  Desktop  Downloads  .gnupg  Music  .profile  snap  .sudo_as_admin_successful  Videos
linux-1@linux1-System-Product-Name: ~$ cd .config/
linux-1@linux1-System-Product-Name: ~/.config$ ls
dconf  eog  gedit  gnome-session  gtk-3.0  nautilus  QTProject.conf  USB-IF  user-dirs.dirs
enchant  evolution  gnome-initial-setup-done  gpa-1.0  ibus  pulse  update-notifier  USB-IF.conf  user-dirs.locale
linux-1@linux1-System-Product-Name: ~/.config$
```

Figure 3.18: Checking USB-IF Folder

Download `usbif.json` file from usb.org website. Once `usbif.json` file is downloaded, go to `usb4cv-cl/documents` and open `Linux_Readme_USB4CV` text file which will mention the path to save the `usbif.json` file.

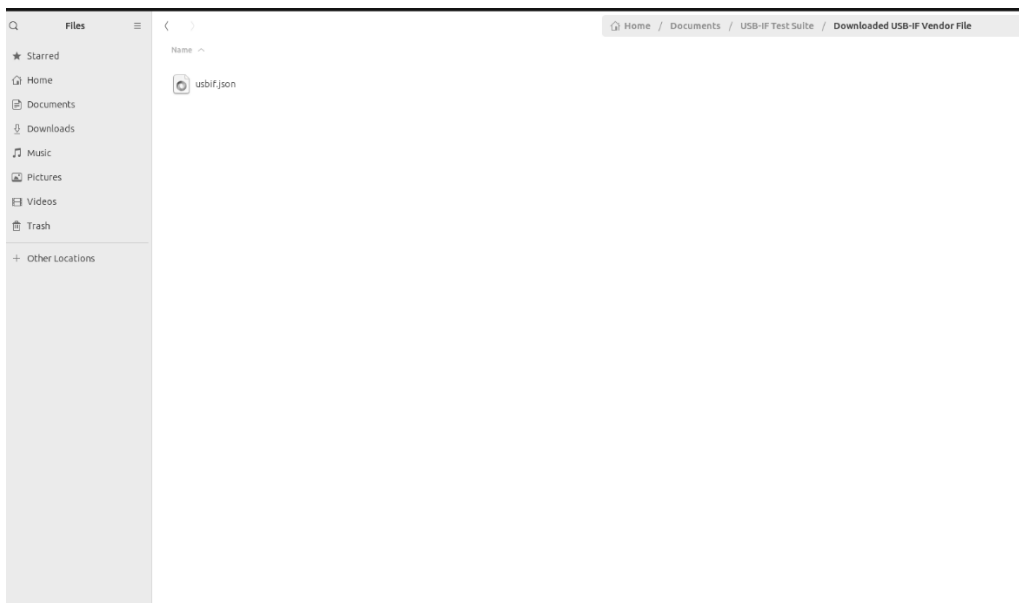
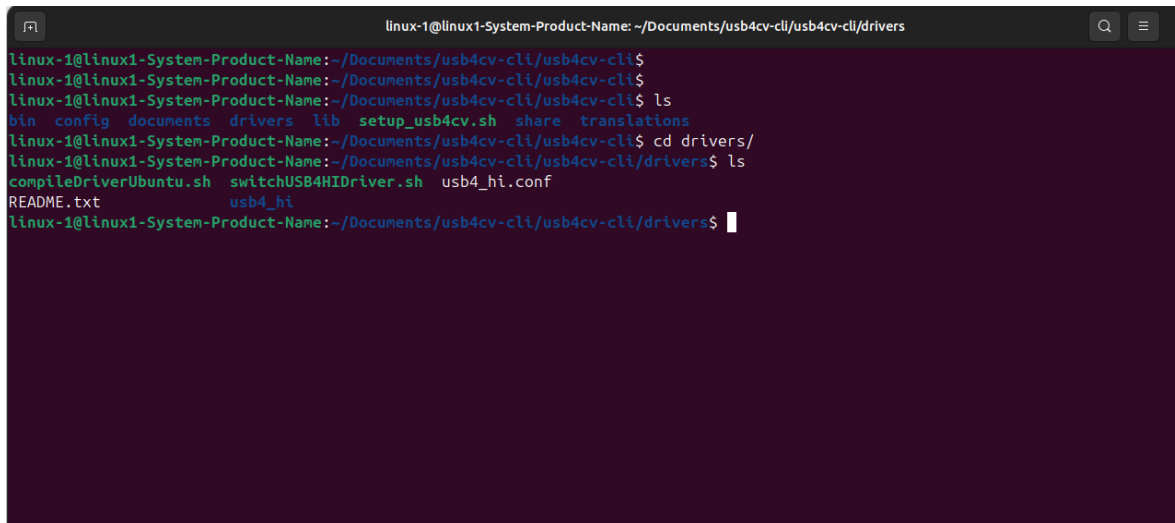


Figure 3.19: Save `usbif.json` file

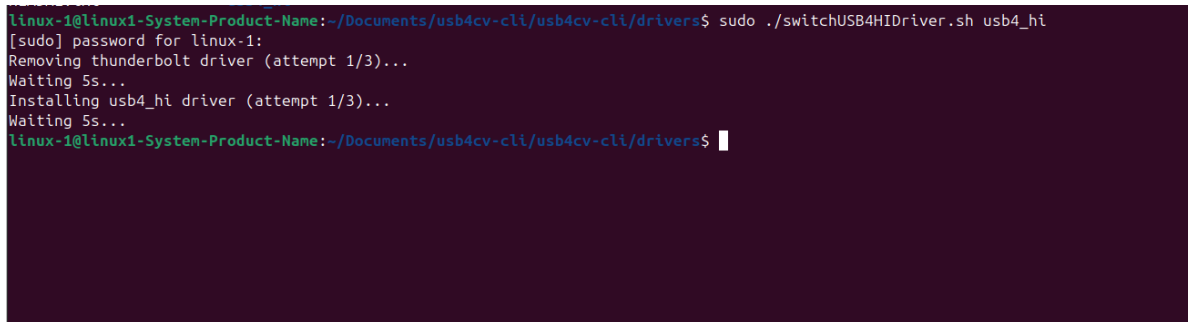
Now go to the driver folder in usb4cv-cli folder and load the usb4_hi driver



```
linux-1@linux1-System-Product-Name: ~/Documents/usb4cv-cli/usb4cv-cli/drivers
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli$
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli$
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli$ ls
bin config documents drivers lib setup_usb4cv.sh share translations
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli$ cd drivers/
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/drivers$ ls
compileDriverUbuntu.sh switchUSB4HIDriver.sh usb4_hi.conf
README.txt usb4_hi
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/drivers$
```

Figure 3.20: Driver Folder

Run `sudo ./switchUSB4HIDriver.sh usb4_hi`



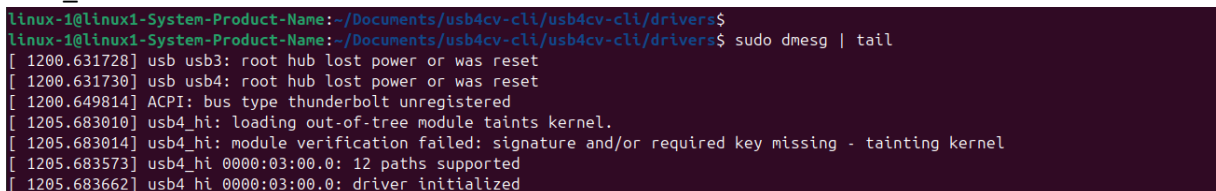
```
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/drivers$ sudo ./switchUSB4HIDriver.sh usb4_hi
[sudo] password for linux-1:
Removing thunderbolt driver (attempt 1/3)...
Waiting 5s...
Installing usb4_hi driver (attempt 1/3)...
Waiting 5s...
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/drivers$
```

Figure 3.21: Load Driver

Run `sudo dmesg | tail`

Confirm the usb4_hi driver is loaded. User is expected to see the following:

```
usb4_hi 0000:03:00.0: 12 paths supported
usb4_hi 0000:03:00.0: driver initialized
```

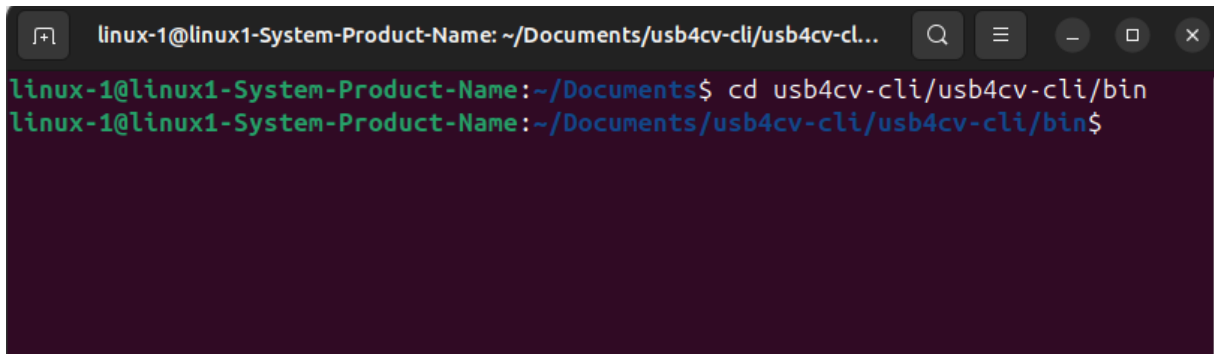


```
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/drivers$
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/drivers$ sudo dmesg | tail
[ 1200.631728] usb usb3: root hub lost power or was reset
[ 1200.631730] usb usb4: root hub lost power or was reset
[ 1200.649814] ACPI: bus type thunderbolt unregistered
[ 1205.683010] usb4_hi: loading out-of-tree module taints kernel.
[ 1205.683014] usb4_hi: module verification failed: signature and/or required key missing - tainting kernel
[ 1205.683573] usb4_hi 0000:03:00.0: 12 paths supported
[ 1205.683662] usb4_hi 0000:03:00.0: driver initialized
```

Figure 3.22: Confirm Driver is Loaded

Note, the number of paths supported is based on the host

Now the driver is loaded, go to back to usb4cv-cli and go inside “bin” folder

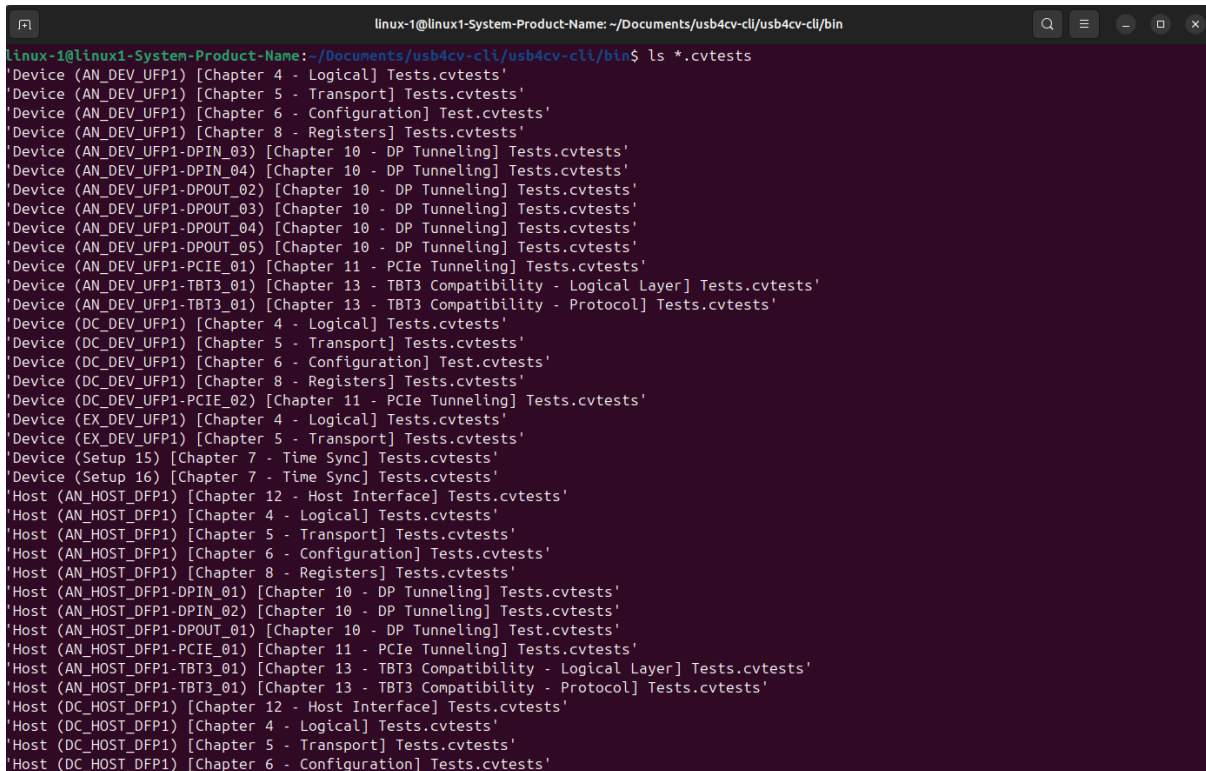


```
linux-1@linux1-System-Product-Name: ~/Documents/usb4cv-cli/usb4cv-cl...
linux-1@linux1-System-Product-Name:~/Documents$ cd usb4cv-cli/usb4cv-cli/bin
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/bin$
```

Figure 3.23: bin folder

To see the tests that can be run. Run:

`ls *.cvtests`



```
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/bin$ ls *.cvtests
'Device (AN_DEV_UFP1) [Chapter 4 - Logical] Tests.cvtests'
'Device (AN_DEV_UFP1) [Chapter 5 - Transport] Tests.cvtests'
'Device (AN_DEV_UFP1) [Chapter 6 - Configuration] Test.cvtests'
'Device (AN_DEV_UFP1) [Chapter 8 - Registers] Tests.cvtests'
'Device (AN_DEV_UFP1-DPIN_03) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Device (AN_DEV_UFP1-DPIN_04) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Device (AN_DEV_UFP1-DPOUT_02) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Device (AN_DEV_UFP1-DPOUT_03) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Device (AN_DEV_UFP1-DPOUT_04) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Device (AN_DEV_UFP1-DPOUT_05) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Device (AN_DEV_UFP1-PCIE_01) [Chapter 11 - PCIe Tunneling] Tests.cvtests'
'Device (AN_DEV_UFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Logical Layer] Tests.cvtests'
'Device (AN_DEV_UFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Protocol] Tests.cvtests'
'Device (DC_DEV_UFP1) [Chapter 4 - Logical] Tests.cvtests'
'Device (DC_DEV_UFP1) [Chapter 5 - Transport] Tests.cvtests'
'Device (DC_DEV_UFP1) [Chapter 6 - Configuration] Test.cvtests'
'Device (DC_DEV_UFP1) [Chapter 8 - Registers] Tests.cvtests'
'Device (DC_DEV_UFP1-PCIE_02) [Chapter 11 - PCIe Tunneling] Tests.cvtests'
'Device (EX_DEV_UFP1) [Chapter 4 - Logical] Tests.cvtests'
'Device (EX_DEV_UFP1) [Chapter 5 - Transport] Tests.cvtests'
'Device (Setup 15) [Chapter 7 - Time Sync] Tests.cvtests'
'Device (Setup 16) [Chapter 7 - Time Sync] Tests.cvtests'
'Host (AN_HOST_DFP1) [Chapter 12 - Host Interface] Tests.cvtests'
'Host (AN_HOST_DFP1) [Chapter 4 - Logical] Tests.cvtests'
'Host (AN_HOST_DFP1) [Chapter 5 - Transport] Tests.cvtests'
'Host (AN_HOST_DFP1) [Chapter 6 - Configuration] Tests.cvtests'
'Host (AN_HOST_DFP1) [Chapter 8 - Registers] Tests.cvtests'
'Host (AN_HOST_DFP1-DPIN_01) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Host (AN_HOST_DFP1-DPIN_02) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Host (AN_HOST_DFP1-DPOUT_01) [Chapter 10 - DP Tunneling] Test.cvtests'
'Host (AN_HOST_DFP1-PCIE_01) [Chapter 11 - PCIe Tunneling] Tests.cvtests'
'Host (AN_HOST_DFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Logical Layer] Tests.cvtests'
'Host (AN_HOST_DFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Protocol] Tests.cvtests'
'Host (DC_HOST_DFP1) [Chapter 12 - Host Interface] Tests.cvtests'
'Host (DC_HOST_DFP1) [Chapter 4 - Logical] Tests.cvtests'
'Host (DC_HOST_DFP1) [Chapter 5 - Transport] Tests.cvtests'
'Host (DC_HOST_DFP1) [Chapter 6 - Configuration] Tests.cvtests'
```

Figure 3.24: Show Every Test in bin folder

Note, go to chapter 5 for more information about running USB4CV test through Command Line.

4. M4x Analyzer Connection

4.1 Introduction

Turn on the M4x and wait until the status LED is blue (figure 9). When connecting to the M4x Analyzer, testers must use the following LeCroy cables:

- Teledyne LeCroy USB4 C-C 0.3m 40 Gbps, E-marked
- Teledyne LeCroy USB4 C-C 0.3m 40 Gbps, (Unmarked)



Figure 4.1: M4x Status LED

4.2 Analyzer Test Setup

- Analyzer Right port connects to DFP using E-marked cable.
- Analyzer Left port connects to UFP using Unmarked cable.

Make sure the **USB Protocol Suite software is closed and not running.**

Any other connections between hosts/devices/hubs that are not going to the analyzer should use a 0.8m Accell USB4 *passive* cable.

The following picture is a good indication of a stable USB4 link:

- The LED for USB4 Protocols is Green
- LEDs for Lanes 0 and 1 are Green.
- LED for Lane Speed 20G is Green. (No flickering)
- 1/Left and 2/Right are Green (*If 2/Right is red, then flip the cable from the M4x end*)



Figure 4.2: Analyzer Test Cable connection

4.3 Exerciser Test Setup

In the USB4 CV there are some tests called exerciser tests (EX). For these tests, use E-marked cable to connect to exerciser port on the M4x and PUT. Note, the orientation of the cable matters when

connecting it to the M4x. Make sure the LED next to the exerciser port is green on M4x. If the LED is red, then flip the cable from the M4x end. Also, make sure the **USB Protocol Suite software is closed and not running.**

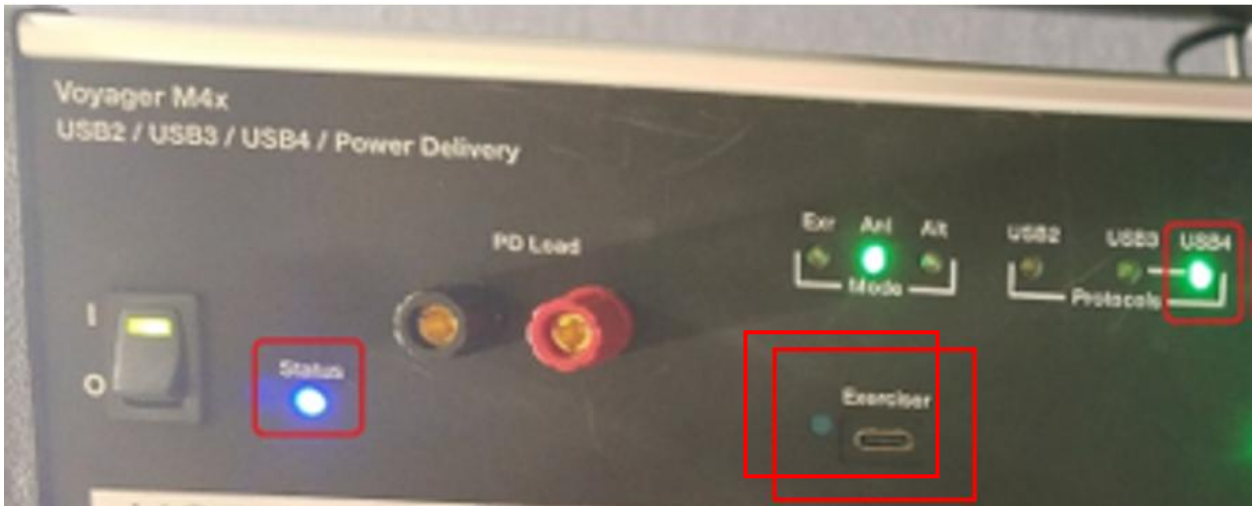


Figure 4.3: M4x Exerciser Port

4.4 Using Ethernet Connection

To use Ethernet connection between the LeCroy Voyager M4x Analyzer and Host. Follow both 4.4.1 & 4.4.2 instructions below. Note, the analyzer tends to be slower to connect on an Ethernet connection. The user might have to use the AnalyzerDelay registry key. This is discussed in the Release Note.

4.4.1 Assign Static IP Address To ARMx64 System

1. Connect ethernet cable to Type-C/Type-A port in armx64 system using USB port to ethernet adapter.
2. Open the Control Panel: Start -> Windows System -> Control Panel.
3. Select the Network and Sharing Center.
4. Click Change Adapter Settings. And choose the ethernet network connection and its properties window appear.

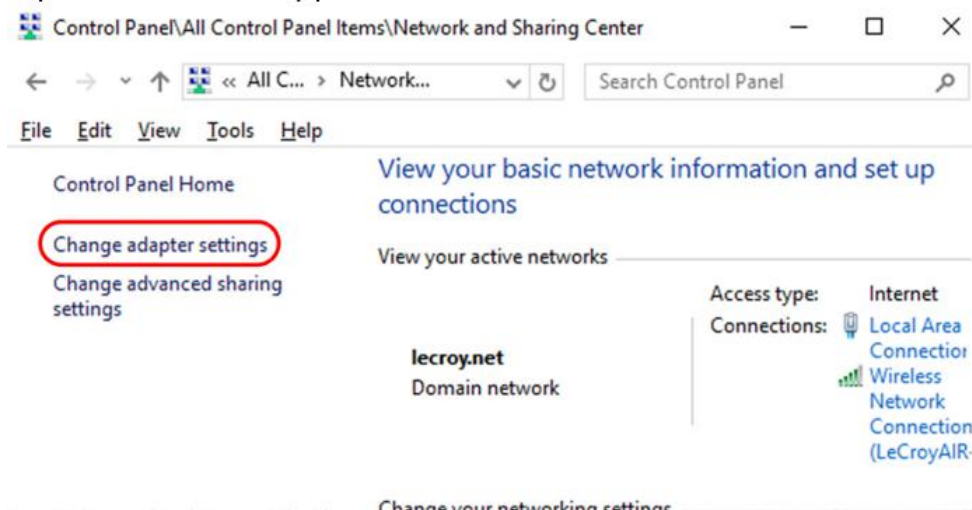


Figure 4.4: Change adapter settings

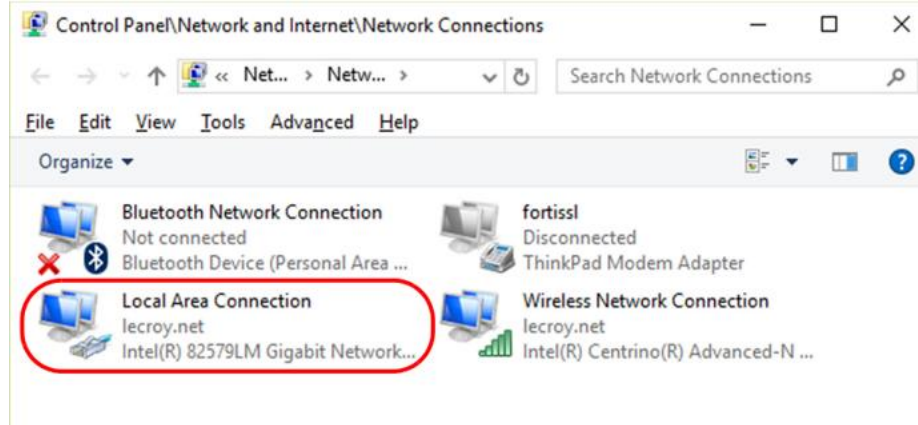


Figure 4.5: Local Area Connection Option

5. Highlight Internet Protocol Version 4 (TCP/IPv4), then click Properties. The dialog box for TCP/ IPv4 properties becomes available.
6. Enter your local static IP settings in the appropriate fields, then click OK (local static address would already be set up) – example: **IP address** – 192.168.1.2; **Subnet** – 255.255.255.0; **Gateway** - 192.168.1.3.

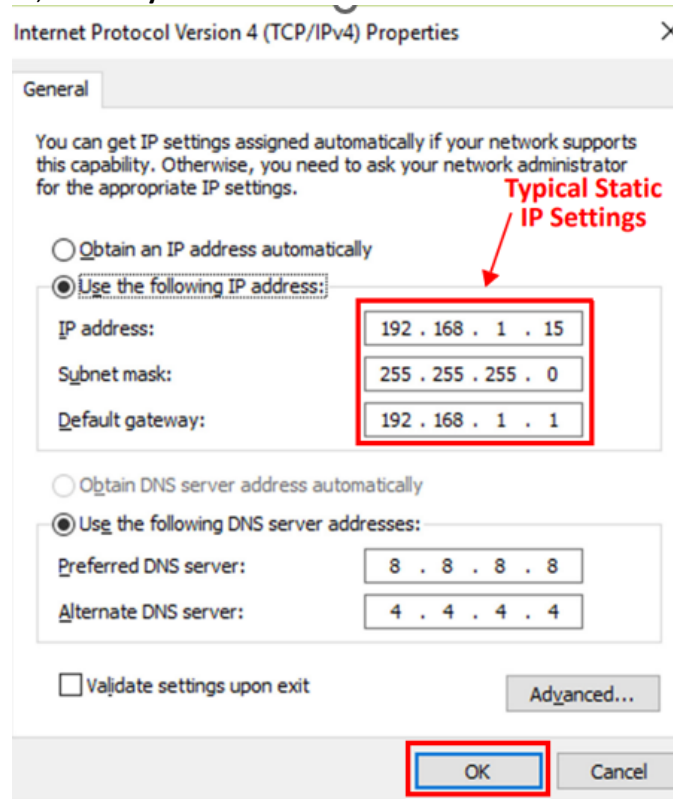


Figure 4.6: Setting Static IP Address

7. Static IP address has been assigned, and this can be verified by checking it in the command prompt -> enter **ipconfig**.

4.4.2 Setting a Static IP Address on Your USB Analyzer

1. Connect your PC to the Analyzer with a USB cable.
2. Now power on your Analyzer (with the USB cable attached).
3. After the Analyzer has initialized, launch the USB Protocol Suite software and select Setup -> All Connected Devices.

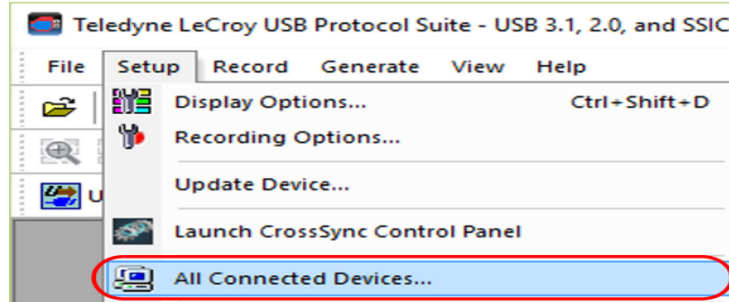


Figure 4.7: USB Protocol Suite

4. To configure the Analyzer with an appropriate IP address, select the IP Settings tab and click apply.

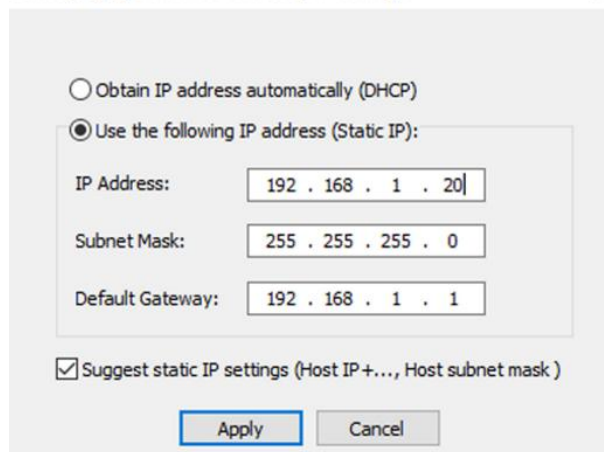


Figure 4.8: Setting Static IP Address for Analyzer

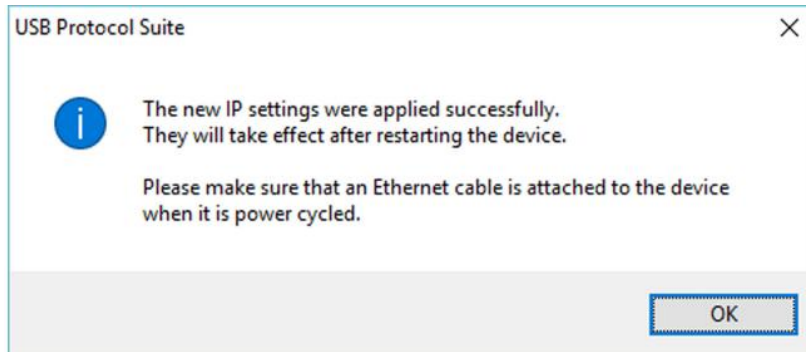


Figure 4.9: New IP Address Has Been Applied

5. Turn off the M4x.
6. Ensure the USB cable is not attached to the USB Analyzer, then attach an Ethernet cable from the host PC to the Analyzer.
7. Turn on the Analyzer.
8. Wait for the Analyzer to initialize (the blue LED comes on after initialization), then start the USB Protocol Suite software.
9. Select Setup -> All Connected Devices, then select the Analyzer you want to connect to. The Connect tab will be highlighted.

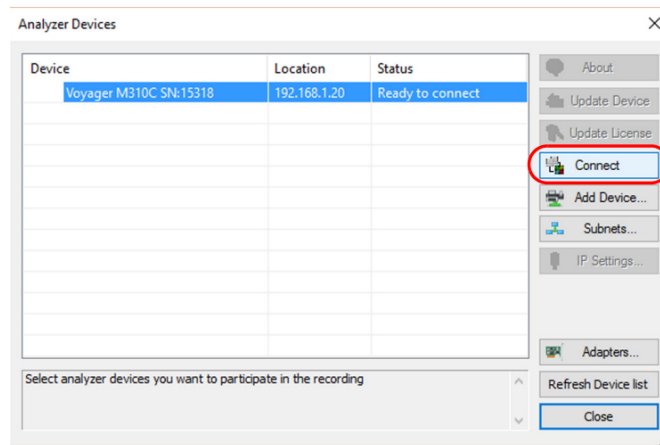
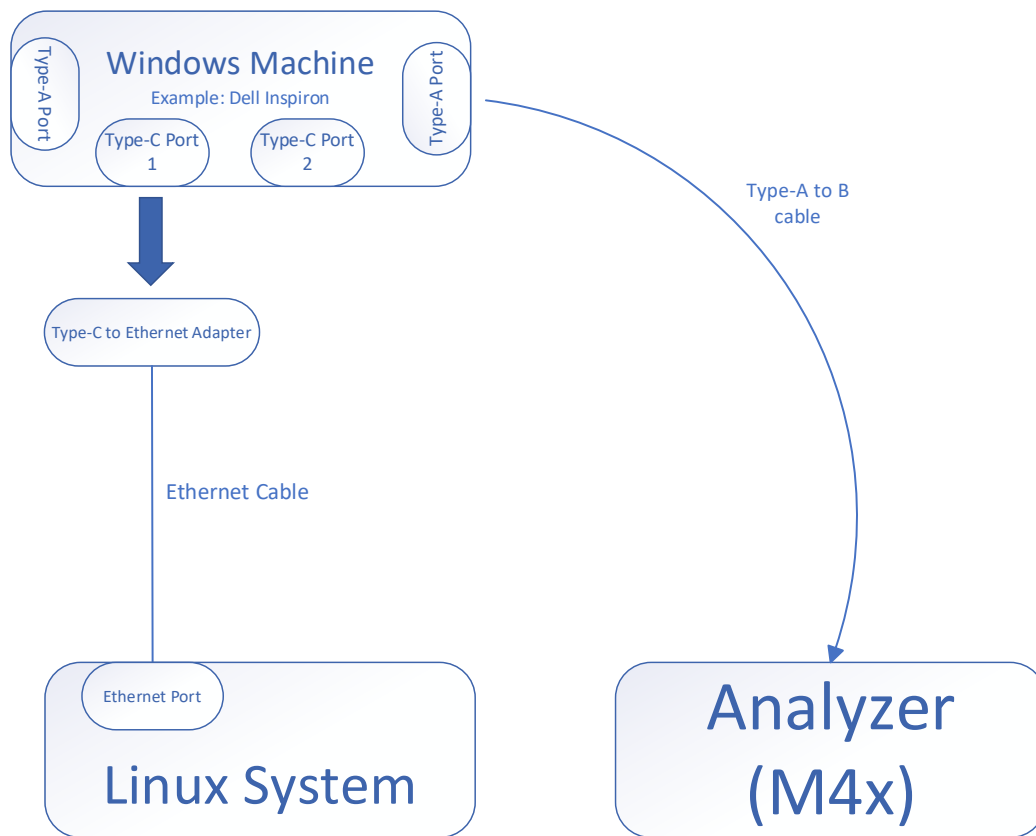


Figure 4.10: Connect Analyzer

4.5 Setup Ethernet connection To Run USB4CV on Linux System

LeCroy USB4 Protocol Suite is run remotely when running USB4CV on Linux. This is set up by connecting a Windows system to the Linux machine. The following sections will explain how to set up both machines so that Linux USB4CV can direct the Windows system to operate the analyzer/exerciser.

Below is the how the setup will look:



4.5.1 Setup IP on Linux

Make sure the Windows machine (Dell Inspiron in this example) is connected to the Linux machine through ethernet

Go to Settings → Network and enable the ethernet port

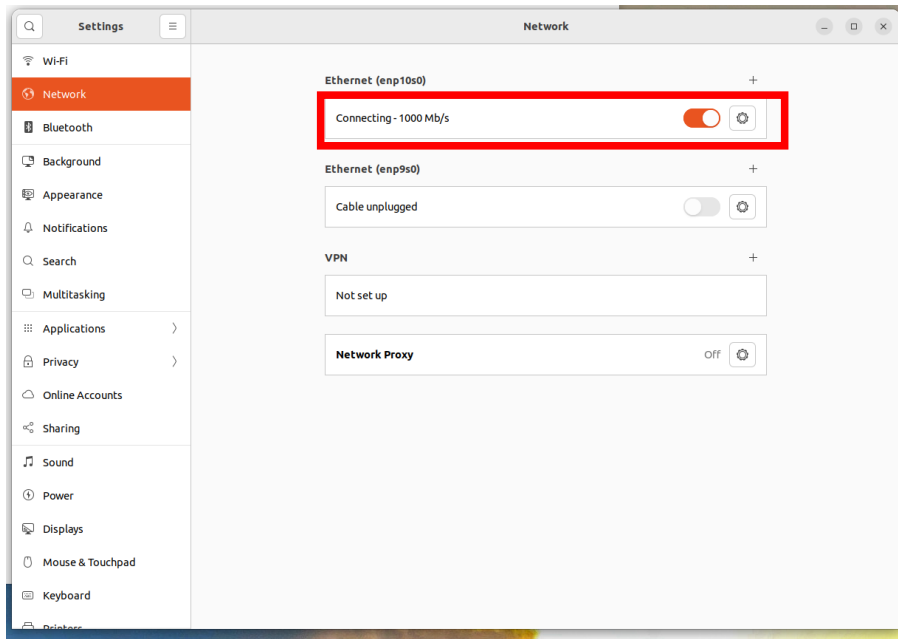


Figure 4.11: Enable The Ethernet Port

Press on the setting icon next to the network connection

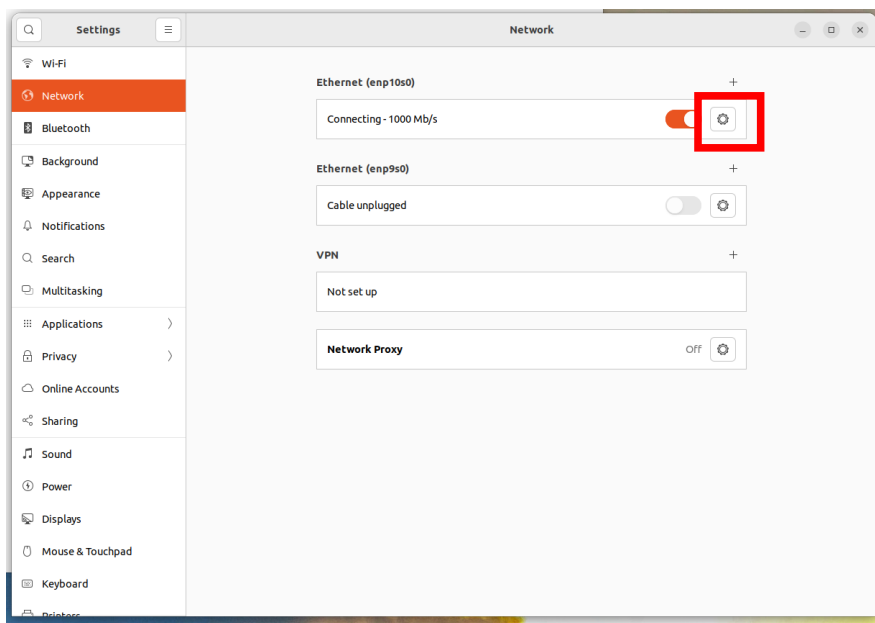


Figure 4.12: The Settings for Ethernet Port

Go to Ipv4 and choose Manual for the IPv4 Method. Then enter the following IP:

Address: 192.168.1.1 Network: 255.255.255.0

Then press Apply

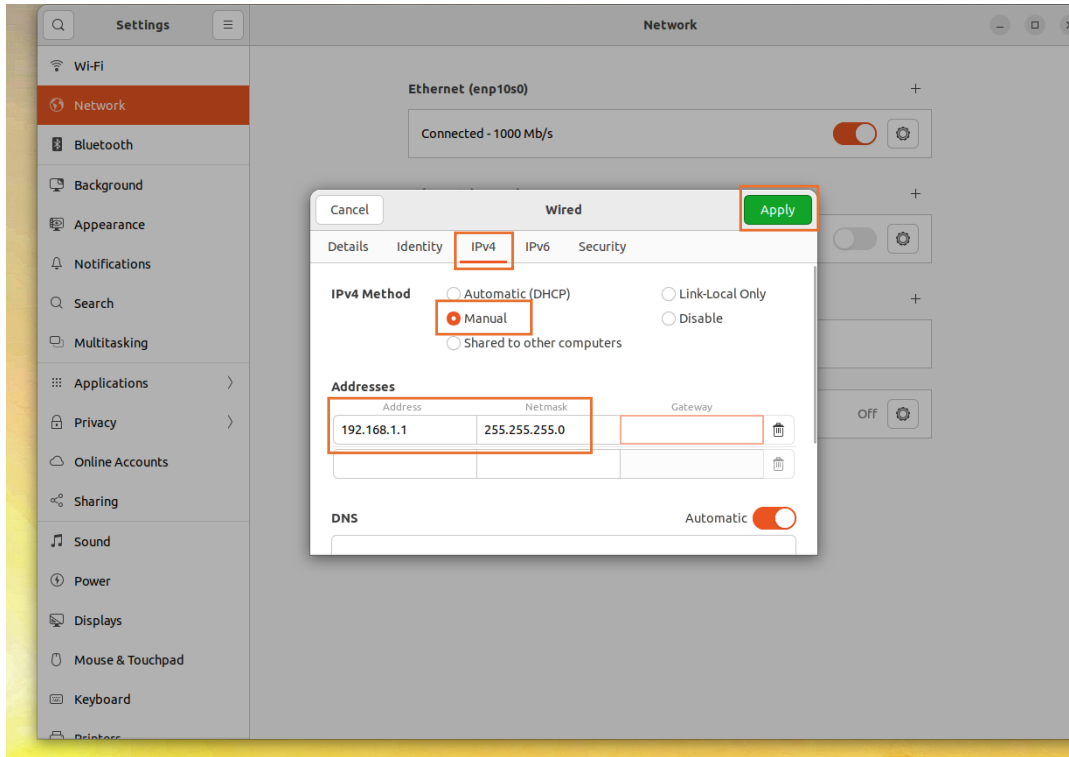


Figure 4.13: Enter IP Address

Now setup the IP on the Windows machine

4.5.2 Setup IP on Windows

In the windows machine, go to Control Panel → Network and Internet → Network and Sharing Center

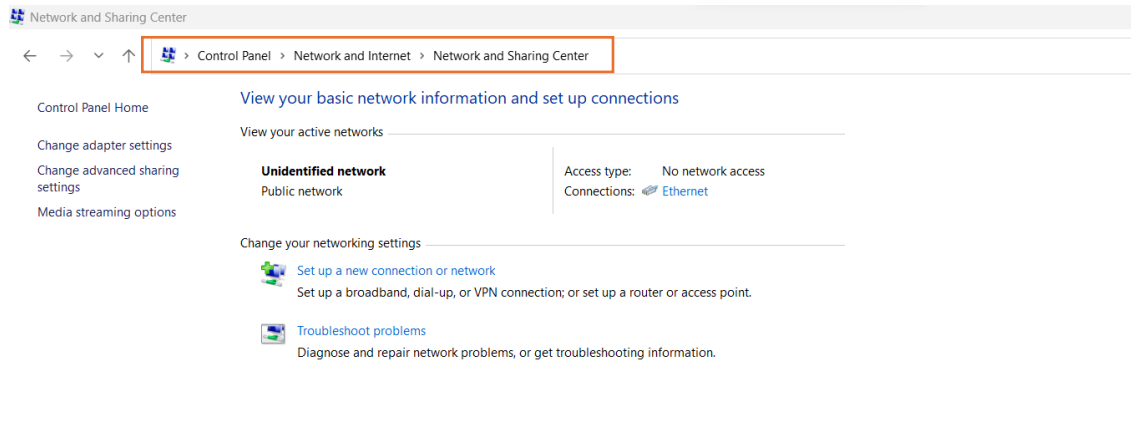


Figure 4.15: Network and Sharing Center

Select Ethernet

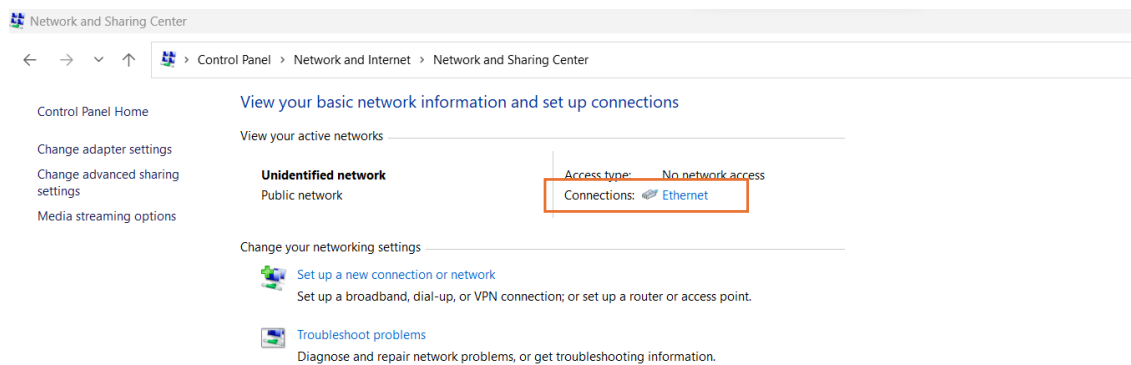


Figure 4.16: Select Ethernet

Select Properties

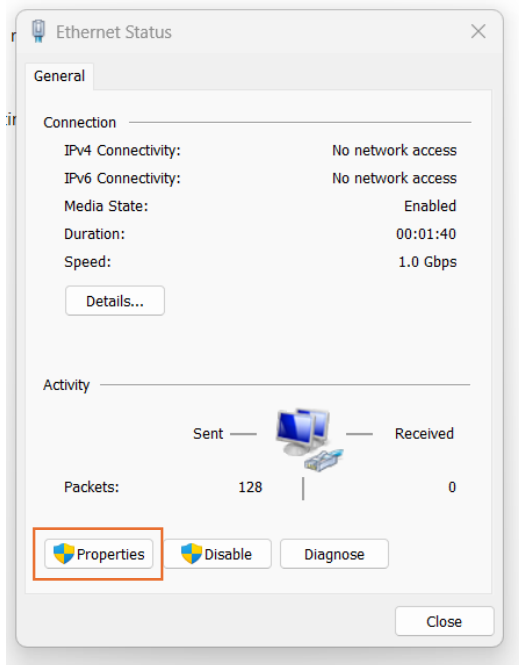


Figure 4.17: Select Properties

Select Internet Protocol Version 4 (TCP/IPv4) and click on properties

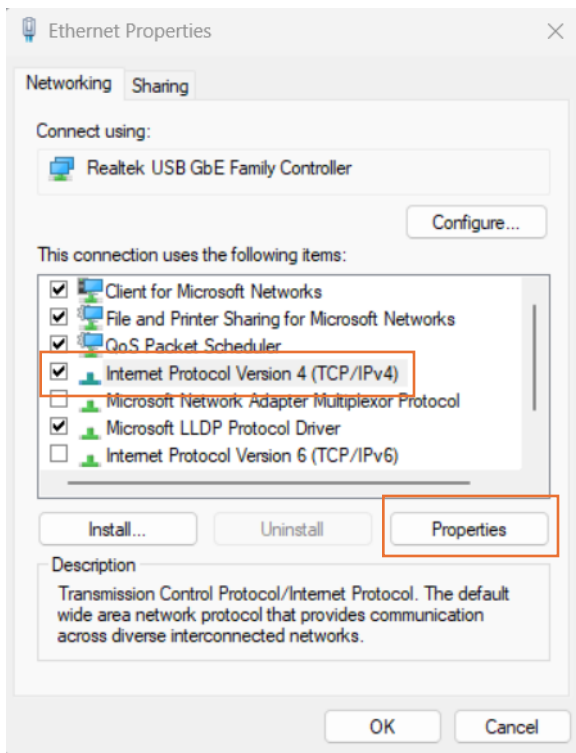


Figure 4.18: Internet Protocol Version

Select *Use the following IP address* and enter the following and then press OK

- IP address: 192.168.1.2
- Subnet mask: 255.255.255.0
- Default gateway: 192.168.1.1

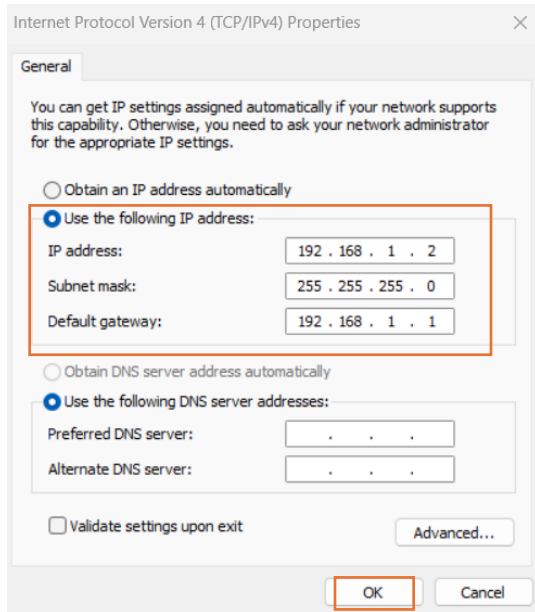


Figure 4.19: IP Address

Now both IP addresses are set up, confirm that the Windows machine can communicate with the Linux machine. To do so, in the Windows machine go to Command Prompt and run the following command:
ping 192.168.1.1

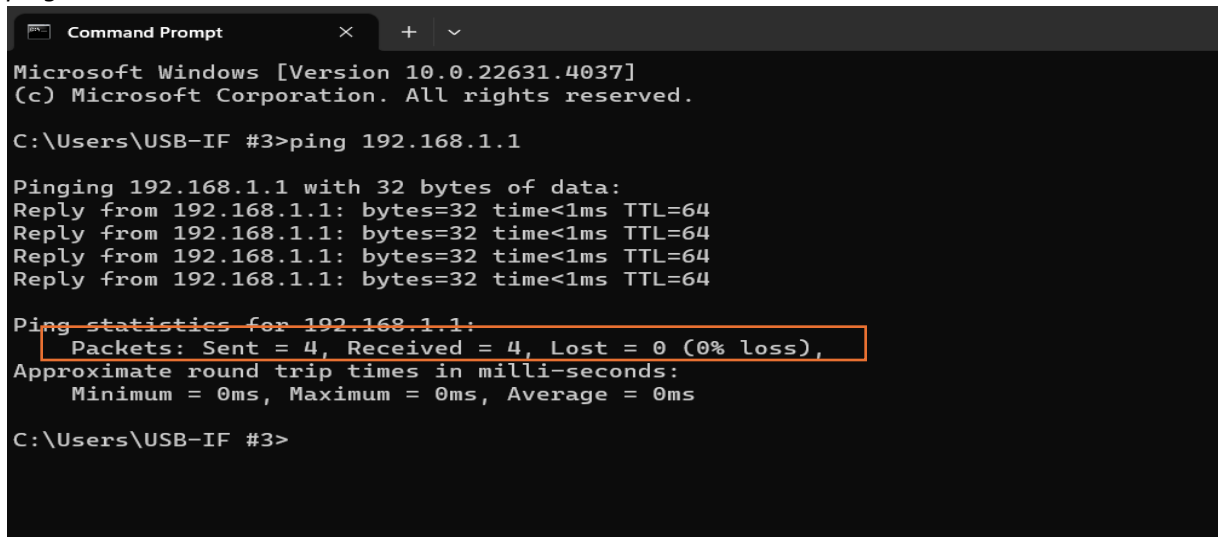


Figure 4.20: Ping IP Address

Confirm that 4 packets are sent and 4 are received.

4.6 Troubleshooting

Occasionally if you cannot get a stable link, you can try one of the following to establish a stable link:

- Make sure you have the M4x connected to the same Host used for the USB4 CV.
- Use a USB3 cable to connect the M4x to the Host that has the USB4 CV.
- Open LeCroy Protocol Suite software to connect to analyzer and close after observing stable link
- Reboot the M4x

5. Running USB4 CV Tests on Linux

5.1 Linux Permissions for USB4CV

Users are required to run USB4CV Chapter 8 tests and Chapter 11 tests using elevated permissions with `sudo -E`. The rest of the chapters can be run as a regular user.

5.1.1 Running USB4CV as Root

Example of launching USB4CV using `sudo`:

- `sudo -E ./cv_cli -app USB4CV -testsuite "USB4 Device (DC_DEV_UFP1-PCIE_01) [Chapter 11 - PCIe Tunneling]" -vif "VIF_name"`
- `sudo -E ./cv_gui -app USB4CV`

When USB4CV is run at elevated permissions, these gui items will work differently:

- *Open Report Directory* will take users the location of the logs, but users won't be able to open the logs from this view. The workaround is to manually navigate to the output directory at "`~/Documents/USB-IF Test Suite/CV Reports/USB4CV`".
- The user can't use the *Launch Report Viewer* button to access the results logs. The workaround is to manually navigate to the output directory and open the logs.

5.2 Hint for GNOME users

On GNOME, the default behavior of modal popup dialogs is to make them unmovable with respect to their parent windows, making it impossible to see what's beneath them.

We suggest that users running the gui version of USB4CV change the settings to make the dialogs movable. Do the following from the command line:

1. `sudo apt install gnome-tweaks`
2. `gnome-tweaks`
 - a. This will open a settings dialog. Select the Windows tab.
 - b. Uncheck Attach Modal Dialogs.

5.3 Starting LeCroy on Remote Windows Machine

On the windows machine, make sure the correct LeCroy Protocol Suite version is installed. See ReleaseNotes_USB4CV.txt for details about the required LeCroy Protocol Suite Version.

Open *C:\Users\Public\Public Document\LeCroy\USB Protocol Suite\Automation\TLRemoteServer*

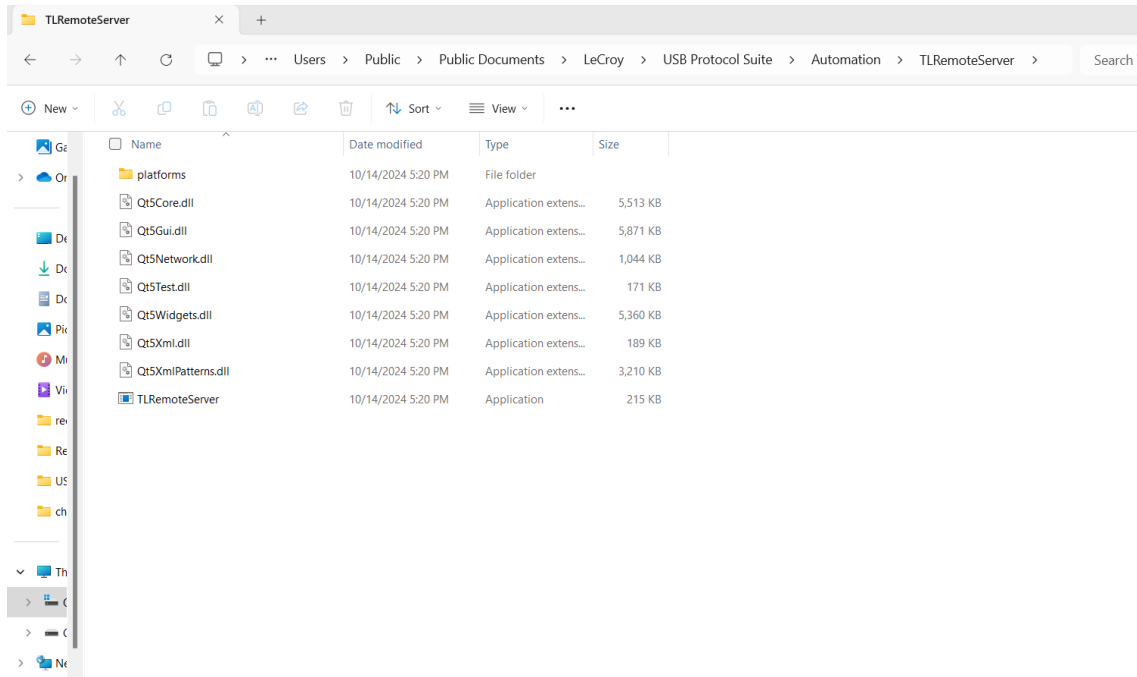


Figure 5.1: Remote Windows Machine

Run TLRemoteServer as administrator

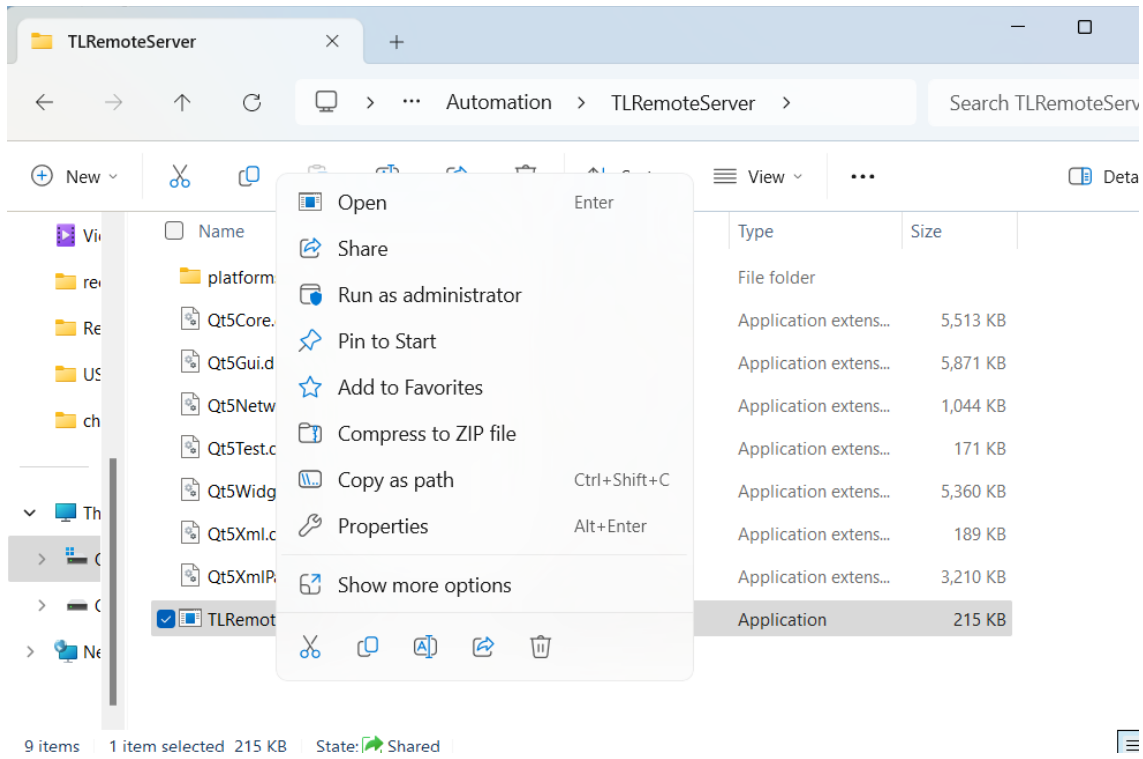


Figure 5.2: Run As Administrator

Make sure this window doesn't close while running USB4CV tests

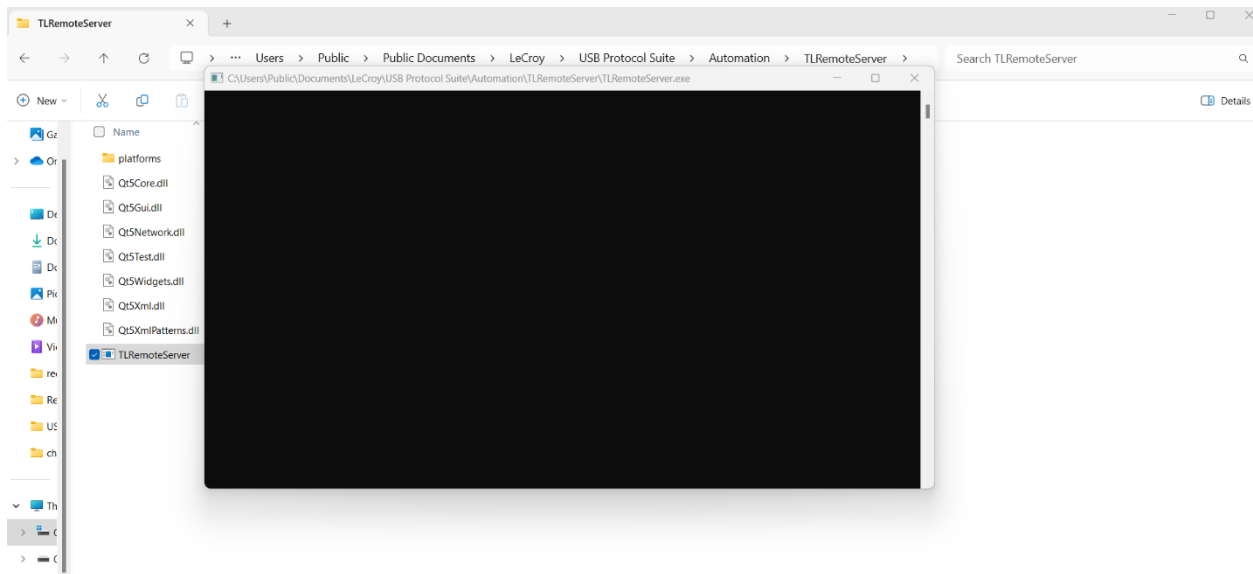


Figure 5.3: LeCory is Set

5.4 Start Testing on Linux

5.4.1 Set Up LeCroy IP Address

Go to "`~/config/USB-IF`"

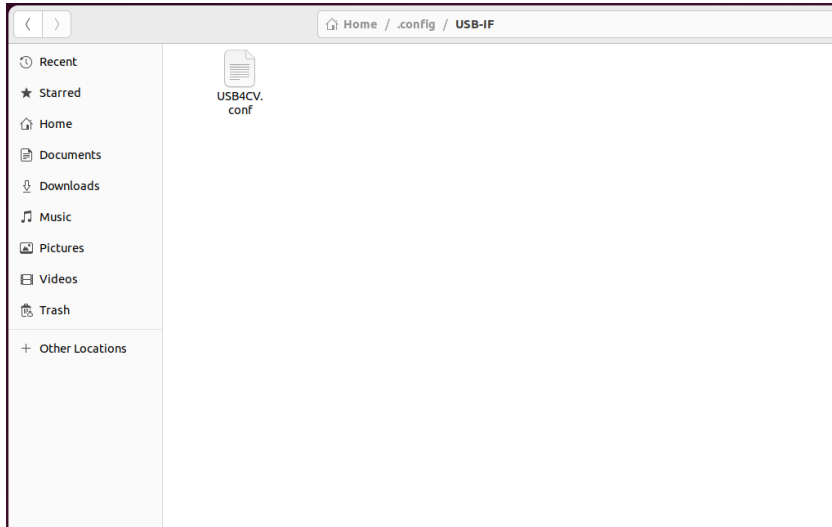


Figure 5.4: USB4CV.config File

Open `USB4CV.conf` file in a text editor and change AnalyzerIP to 192.168.1.2

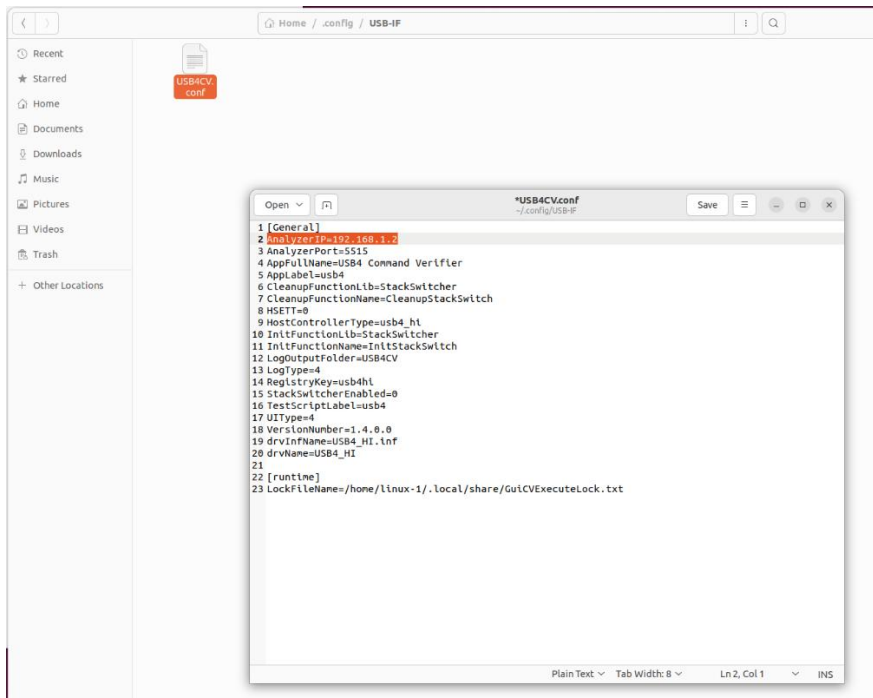


Figure 5.5 : Set IP Address on Linux Machine

5.4.2 Running cv-gui

Go to `~/Documents/usb4cv-gui/usb4cv-gui/bin`

For chapter 8 and chapter 11, users must run the following command: `sudo -E ./cv-gui -app USB4CV`

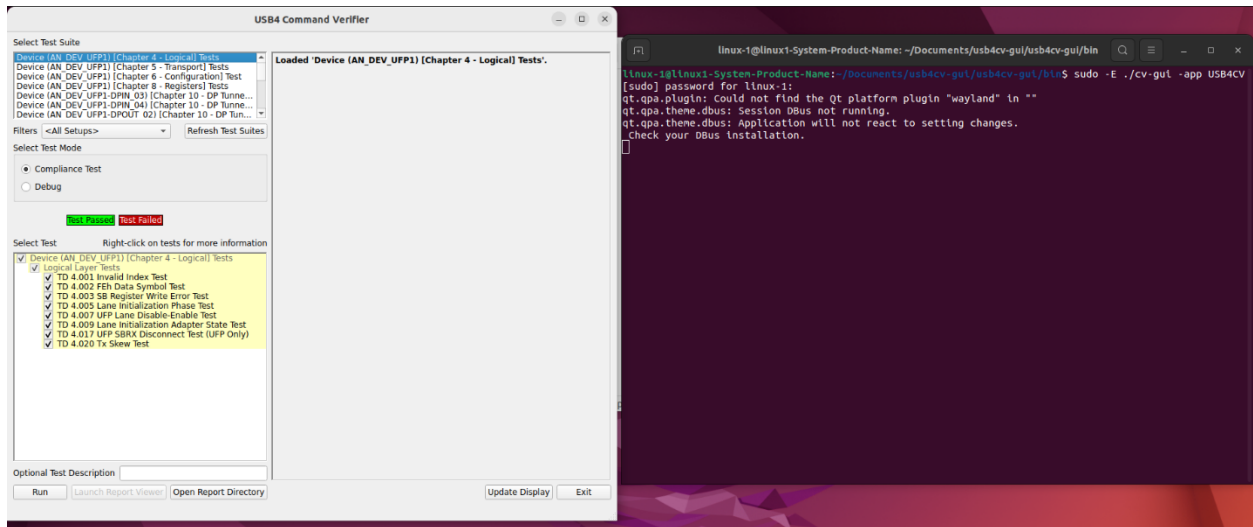


Figure 5.8: Sudo -E

After running the test, the USB4CV logs will be saved in a folder named *USB-IF Test Suite*

Go to `~/Documents/USB-IF Test Suite/CV Reports/USB4CV`

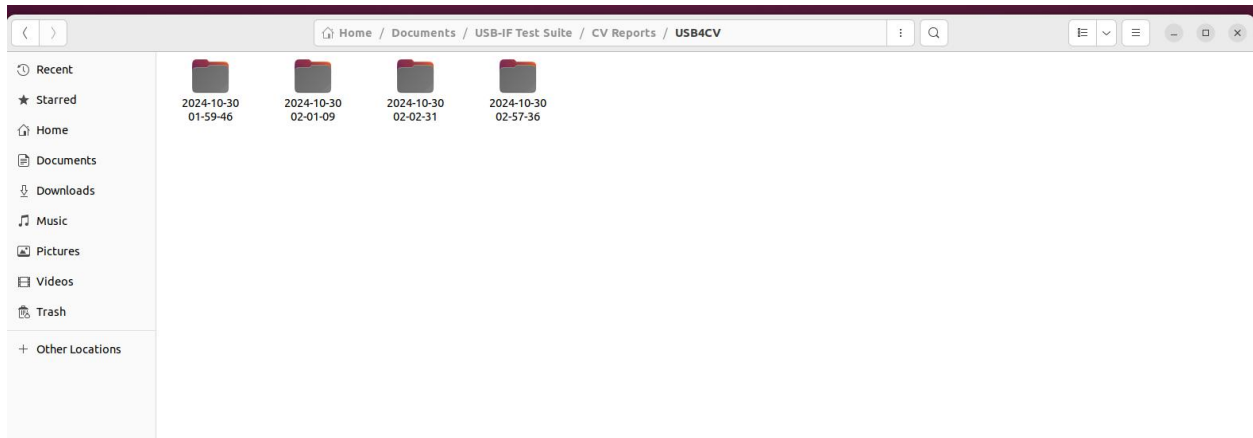


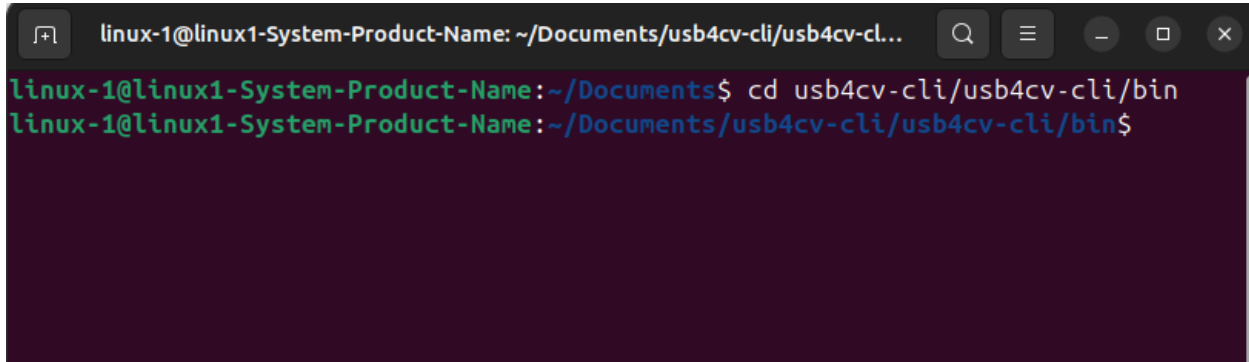
Figure 5.9: USB4CV Results Logs

Note, the folder of the test will have the date and time. Users can change the name to whatever they want once the test suite finish running. Users can press on 'Open report directory' and it will take them to the folder where the results are saved. However, users won't be able to open the logs if `sudo -E` was used to launch `usb4cv`. Users must go to the results folder manually.

5.4.3 Running cv-cli

Go to "`~/Documents/usb4cv-cli/usb4cv-cli/bin`" by running:

`cd "~/Documents/usb4cv-cli/usb4cv-cli/bin"`

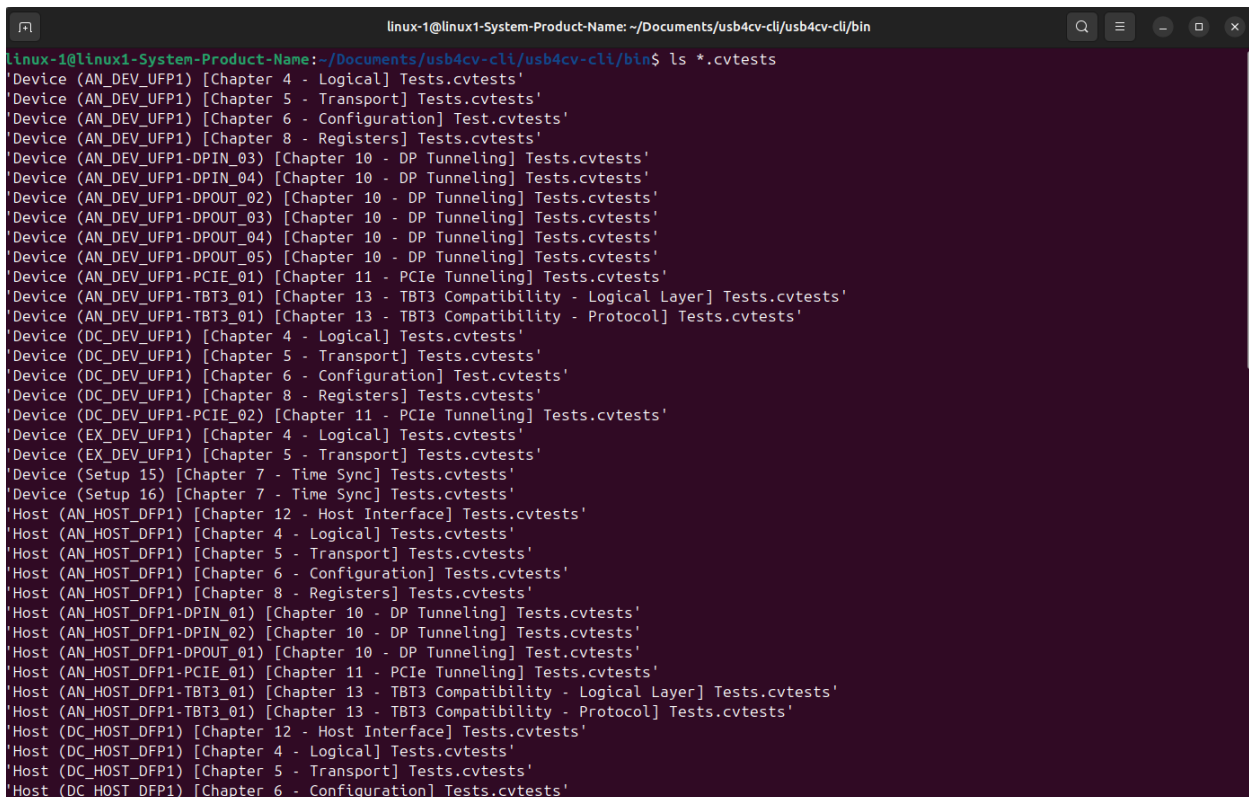


```
linux-1@linux1-System-Product-Name: ~/Documents/usb4cv-cli/usb4cv-cl...
linux-1@linux1-System-Product-Name:~/Documents$ cd usb4cv-cli/usb4cv-cli/bin
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/bin$
```

Figure 5.9: Bin Folder

To see all the tests that are available in the usb4cv, run:

`ls *.cvtests`



```
linux-1@linux1-System-Product-Name: ~/Documents/usb4cv-cli/usb4cv-cl...
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cl...$ ls *.cvtests
'Device (AN_DEV_UFP1) [Chapter 4 - Logical] Tests.cvtests'
'Device (AN_DEV_UFP1) [Chapter 5 - Transport] Tests.cvtests'
'Device (AN_DEV_UFP1) [Chapter 6 - Configuration] Test.cvtests'
'Device (AN_DEV_UFP1) [Chapter 8 - Registers] Tests.cvtests'
'Device (AN_DEV_UFP1-DPIN_03) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Device (AN_DEV_UFP1-DPIN_04) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Device (AN_DEV_UFP1-DPOUT_02) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Device (AN_DEV_UFP1-DPOUT_03) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Device (AN_DEV_UFP1-DPOUT_04) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Device (AN_DEV_UFP1-DPOUT_05) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Device (AN_DEV_UFP1-PCIE_01) [Chapter 11 - PCIe Tunneling] Tests.cvtests'
'Device (AN_DEV_UFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Logical Layer] Tests.cvtests'
'Device (AN_DEV_UFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Protocol] Tests.cvtests'
'Device (DC_DEV_UFP1) [Chapter 4 - Logical] Tests.cvtests'
'Device (DC_DEV_UFP1) [Chapter 5 - Transport] Tests.cvtests'
'Device (DC_DEV_UFP1) [Chapter 6 - Configuration] Test.cvtests'
'Device (DC_DEV_UFP1) [Chapter 8 - Registers] Tests.cvtests'
'Device (DC_DEV_UFP1-PCIE_02) [Chapter 11 - PCIe Tunneling] Tests.cvtests'
'Device (EX_DEV_UFP1) [Chapter 4 - Logical] Tests.cvtests'
'Device (EX_DEV_UFP1) [Chapter 5 - Transport] Tests.cvtests'
'Device (Setup 15) [Chapter 7 - Time Sync] Tests.cvtests'
'Device (Setup 16) [Chapter 7 - Time Sync] Tests.cvtests'
'Host (AN_HOST_DFP1) [Chapter 12 - Host Interface] Tests.cvtests'
'Host (AN_HOST_DFP1) [Chapter 4 - Logical] Tests.cvtests'
'Host (AN_HOST_DFP1) [Chapter 5 - Transport] Tests.cvtests'
'Host (AN_HOST_DFP1) [Chapter 6 - Configuration] Tests.cvtests'
'Host (AN_HOST_DFP1) [Chapter 8 - Registers] Tests.cvtests'
'Host (AN_HOST_DFP1-DPIN_01) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Host (AN_HOST_DFP1-DPIN_02) [Chapter 10 - DP Tunneling] Tests.cvtests'
'Host (AN_HOST_DFP1-DPOUT_01) [Chapter 10 - DP Tunneling] Test.cvtests'
'Host (AN_HOST_DFP1-PCIE_01) [Chapter 11 - PCIe Tunneling] Tests.cvtests'
'Host (AN_HOST_DFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Logical Layer] Tests.cvtests'
'Host (AN_HOST_DFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Protocol] Tests.cvtests'
'Host (DC_HOST_DFP1) [Chapter 12 - Host Interface] Tests.cvtests'
'Host (DC_HOST_DFP1) [Chapter 4 - Logical] Tests.cvtests'
'Host (DC_HOST_DFP1) [Chapter 5 - Transport] Tests.cvtests'
'Host (DC_HOST_DFP1) [Chapter 6 - Configuration] Tests.cvtests'
```

Figure 5.10: View Everything in Bin Folder

To run a specific USB4CV test, run the following command for tests that don't need a VIF uploaded
`./cv-cli -app USB4CV -testsuite <name of the test>`

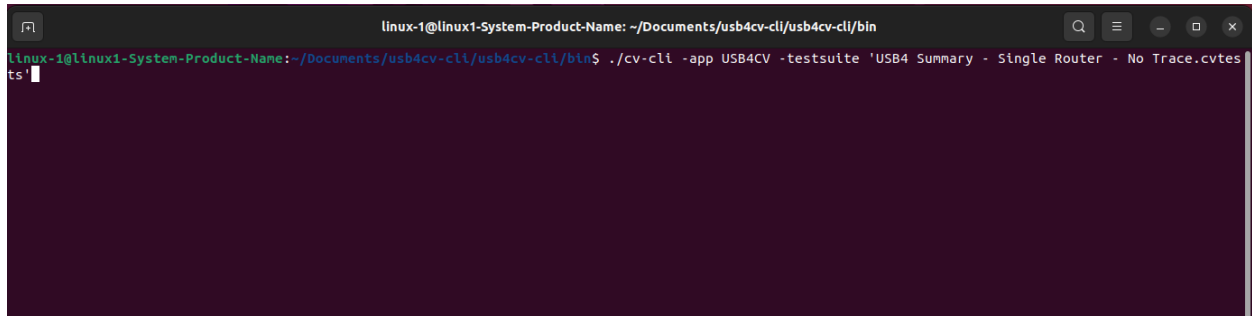


Figure 5.11: No VIF Needed

To run a specific USB4CV test, run the following command for tests that need a VIF uploaded
`./cv-cli -app USB4CV -testsuite <name of the test> -vif <path/to/vif>`

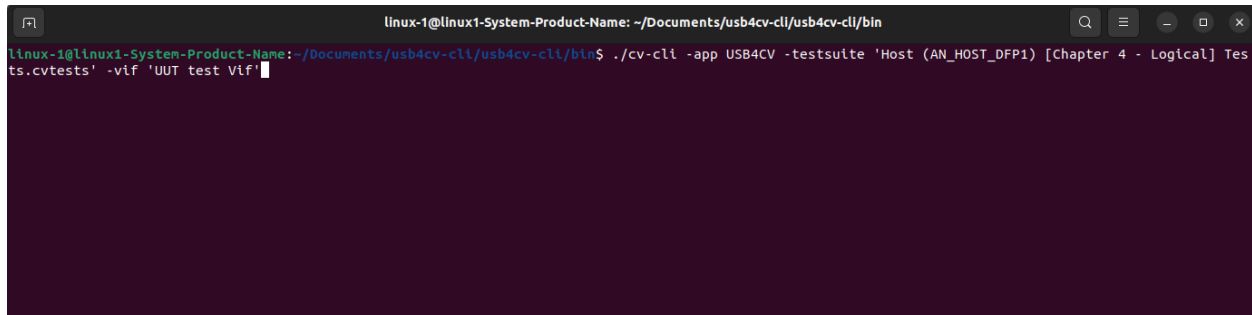


Figure 5.12: VIF Needed

For chapter 8 and chapter 11, user must run those tests with `sudo -E`:

`sudo -E ./cv-cli -app USB4CV -testsuite <name of the test> -vif <path/to/vif>`

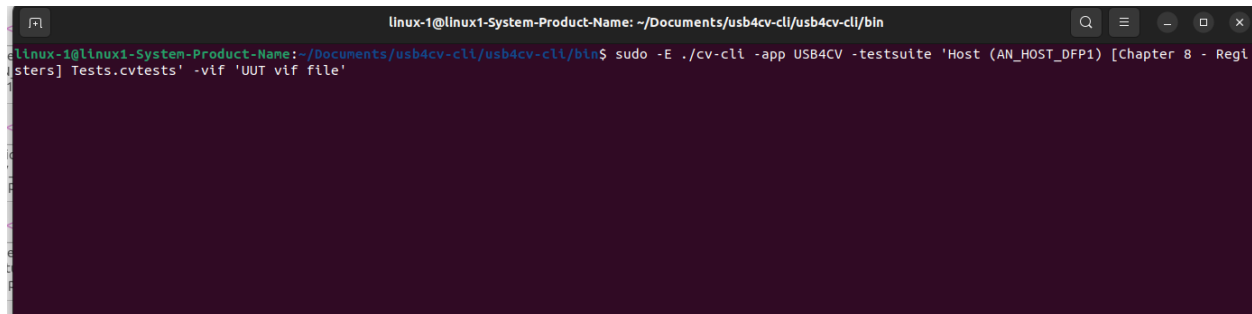


Figure 5.13: Sudo -E Required

After running the test, the USB4CV results logs will be saved in a folder named USB-IF Test Suite

Go to “~/Documents/USB-IF Test Suite/CV Reports/USB4CV”

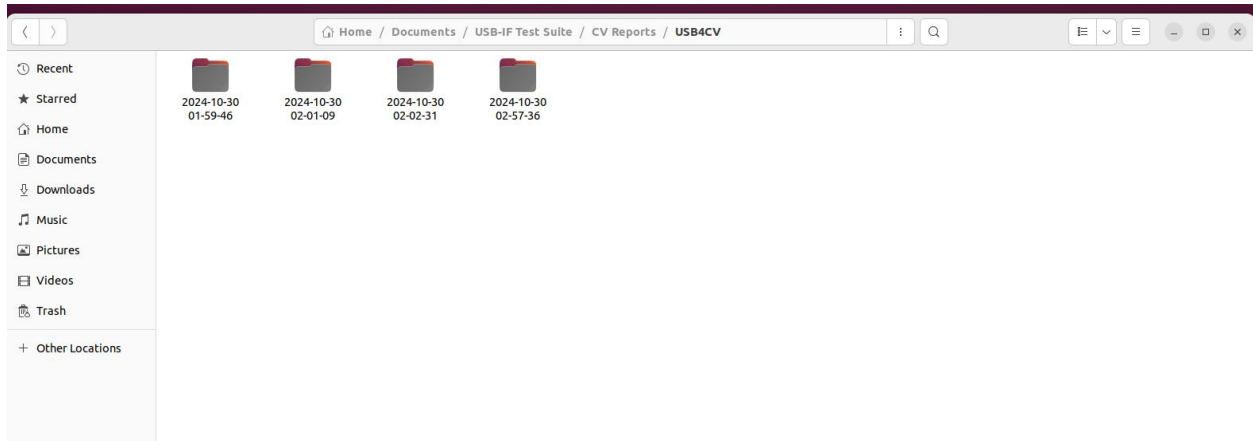


Figure 5.14: USB4CV Logs

Note: the folder of the test will have the date and time. Users are expected to rename the folder to add the corresponding chapter name.

5.5 LeCroy Trace

After running USB4CV test, LeCroy traces will be saved on the windows machine

Go to **C:\Users\Public\Public Documents\LeCroy\USB Protocol Suite\Traces**

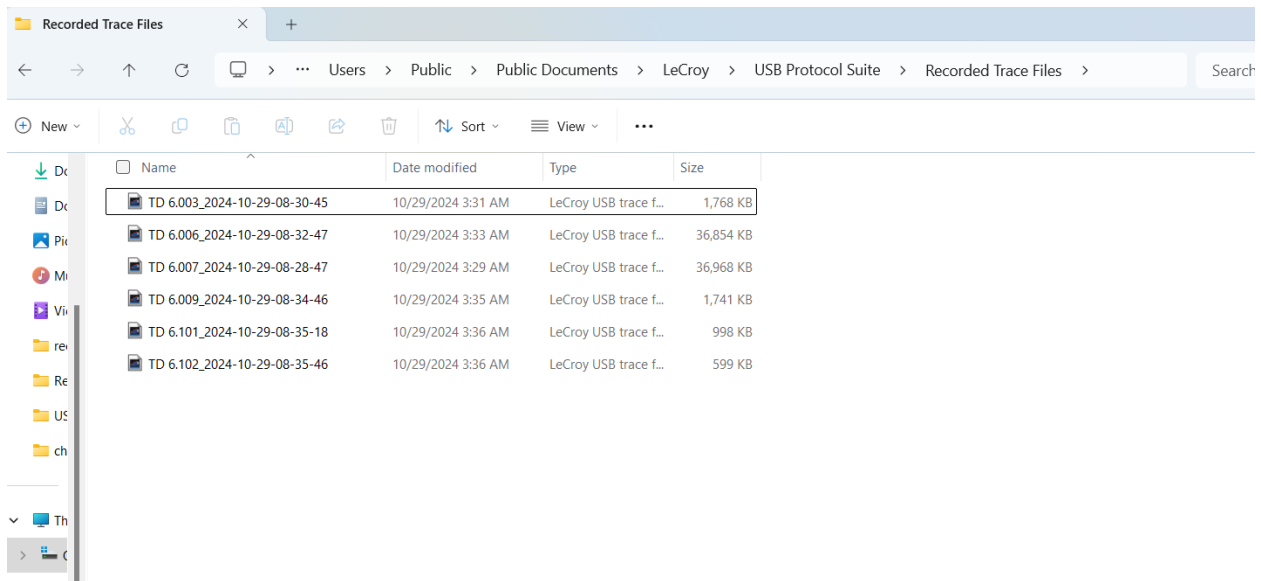


Figure 5.15: USB4CV Test Results Traces

Users are required to save these traces and include them in the results.

6. USB4CV Test Setups

Download the USB4 CV CTS from [usb.org](https://www.usb.org) website.

<https://www.usb.org/document-library/usb4cv-compliance-test-specification>

The CTS should include every setup that needs to be run for every chapter in the USB4 CV. For example, if you are running chapter 4 Analyzer then open USB4 Logical Layer document in the USB4 CTS. This document should include all USB4 setup for chapter 4. Each document states the associated chapter in the USB4 Specification which is being tested.

Below is each chapter with its corresponding CTS.

- Chapter 4 - USB4 Logical Layer
- Chapter 5 – USB4 Protocol
- Chapter 6 - USB4 Protocol
- Chapter 7 – USB4 TMU
- Chapter 8 - USB4 Protocol
- Chapter 9 – USB4 USB3 Tunneling
- Chapter 10 – USB4 DP Tunnel
- Chapter 11 – USB4 PCIe Tunneling
- Chapter 12 – USB4 Host Interface
- Chapter 13 - USB4 Logical Layer & USB4 Protocol

6.1 Chapter 4, Chapter 5, Chapter 6 & Chapter 8

In the CTS, there is a different setup depending on what the UUT is (*Host/Device/Hub*) and what chapter is being run (*See USB4 CV Tests' Setups for each chapter with its corresponding CTS*). The USB4 CV test should have the exact name as the setup in the CTS. (*Note, if the USB4 CV has a test name and the setup is nowhere to be found in the CTS for that specific test, please contact us*).

Note, if the setup below doesn't match the setup in the Compliance Test Specification (CTS), then users must use the setup in the CTS and ignore the setups in this document.

6.1.1 UUT is a Host - Chapter 4 AN & Chapter 5 AN & Chapter 6 AN & Chapter 8 AN:

AN_HOST_DFP1

- A Compliance Device is connected to the DFP of the UUT
- USB4 CV is installed on the Host System with the UUT
- A USB4 Analyzer is connected between the UUT and Compliance Device

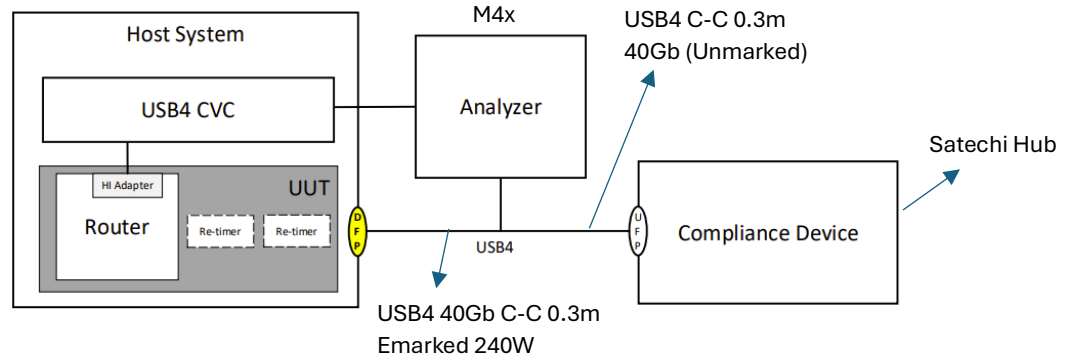


Figure 6.1: Analyzer Host Setup In The CTS

6.1.2 UUT is a Host - Chapter 4 EX & Chapter 5 EX & Chapter 6 EX & Chapter 8 EX:

EX_HOST_DFP1

- The Exerciser is connected to the UUT
- Unless specified otherwise, the Exerciser is in USB4 device mode and presents as UFP
- USB4 CV is installed on the Host System with the UUT

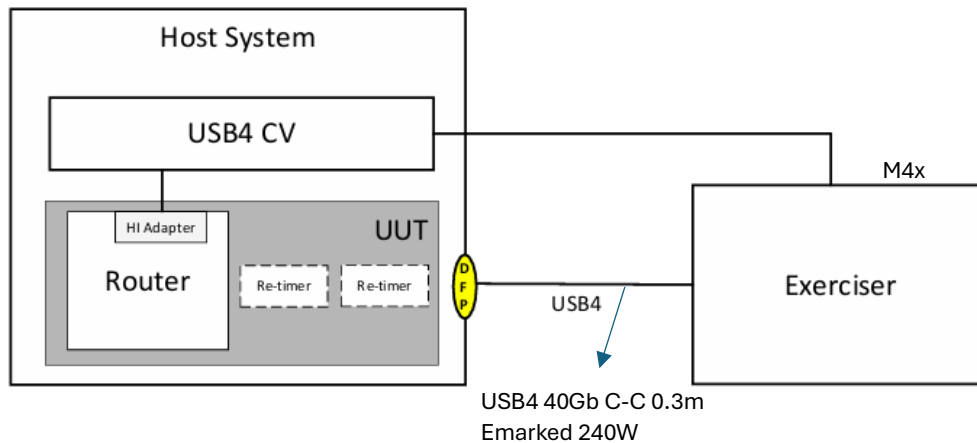


Figure 6.2: Host Exerciser Setup in The CTS

6.1.3 UUT is a Device - Chapter 4 AN & Chapter 5 AN & Chapter 6 AN & Chapter 8 AN:

AN_DEV_UFP1

Description:

- DFP on KG USB4 Host connects to UFP of the UUT
- Analyzer is connected between the KG USB4 Host and the UUT

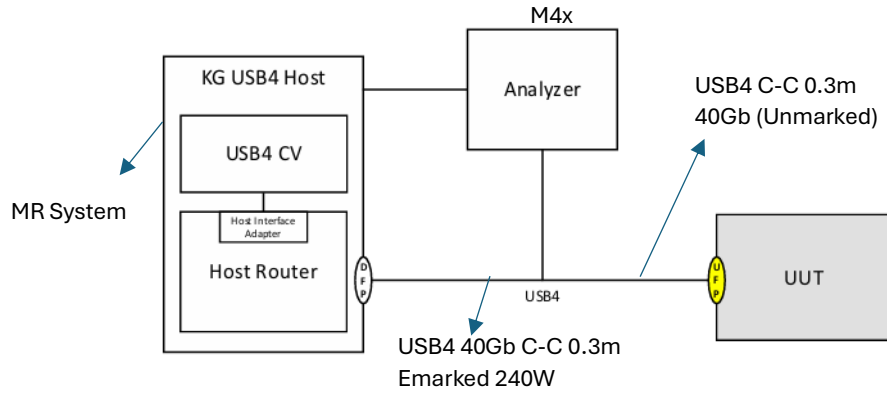


Figure 6.3: Device Analyzer Setup in The CTS

6.1.4 UUT is a Device - Chapter 4 EX & Chapter 5 EX & Chapter 6 EX & Chapter 8 EX:

EX_DEV_UFP1

- The Exerciser is connected to the UUT
- Unless specified otherwise, the Exerciser is configured as a USB4 DFP

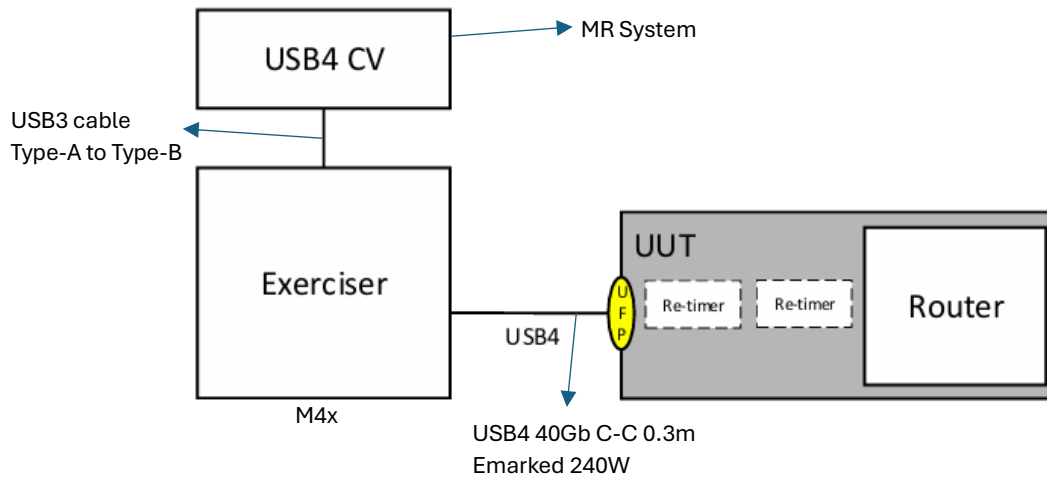


Figure 6.4: Device Exerciser Setup in The CTS

6.1.5 UUT is a Hub - Chapter 4 AN:

When a hub is the UUT then both UFP and DFP are supposed to be tested:

UUT is a Hub (testing the UFP)

AN_HUB_UFP1

- A KG USB4 Host is connected to the UUT
- USB4 CV is installed on the KG Host System
- A USB4 Analyzer is connected between the UUT and Compliance Device

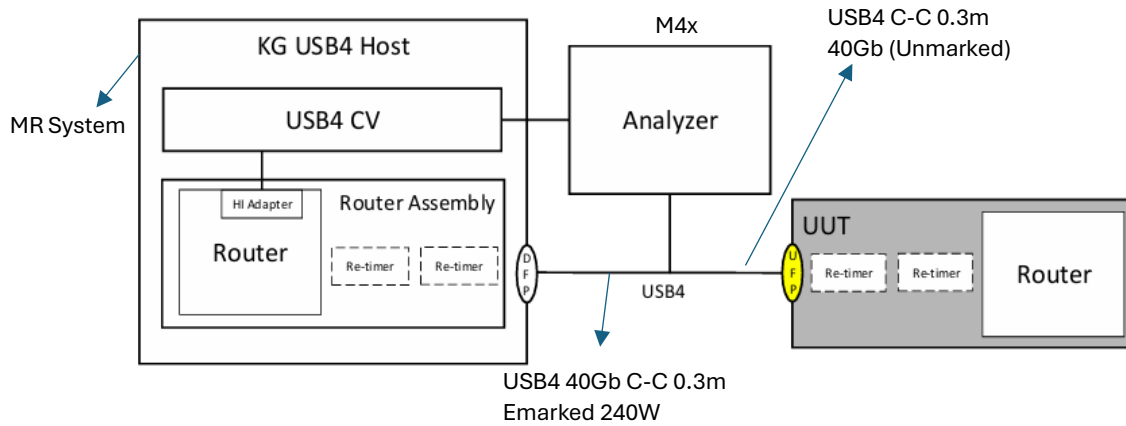


Figure 6.5: Hub UFP Analyzer Setup in The CTS

UUT is a Hub (testing the DFP)

AN_HUB_DFP1

- A KG USB4 host is connected to the UFP of the UUT
- A Compliance Device is connected to the DFP of the UUT
- USB4 CV is installed on the KG Host System
- A USB4 Analyzer is connected between the UUT and Compliance Device

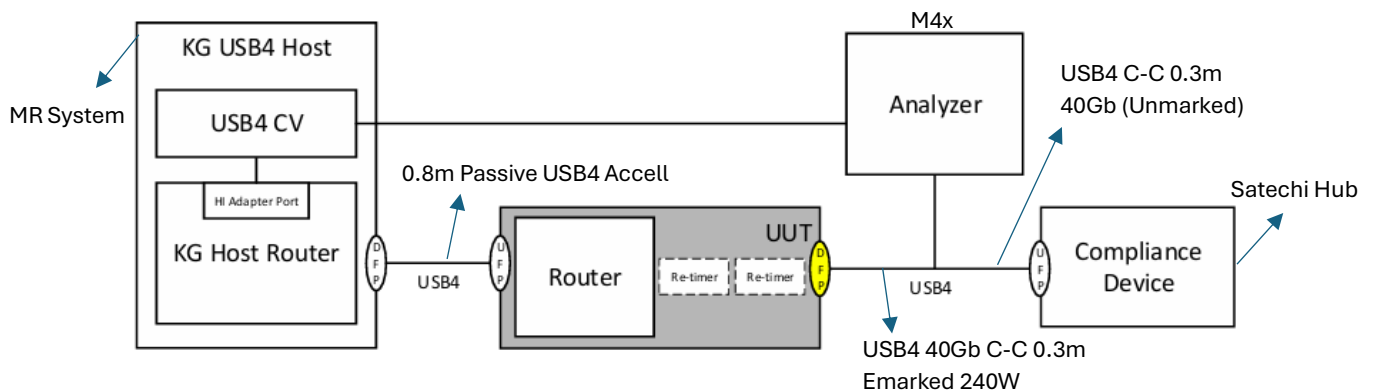


Figure 6.6: Hub DFP Analyzer Setup in The CTS

6.1.6 UUT is a Hub - Chapter 4 EX:

When a hub is the UUT then both UFP and DFP are supposed to be tested:

UUT is a Hub (testing the UFP)

EX_HUB_UFP1

- The Exerciser is connected to the UUT
- Exerciser is configured as a USB4 DFP

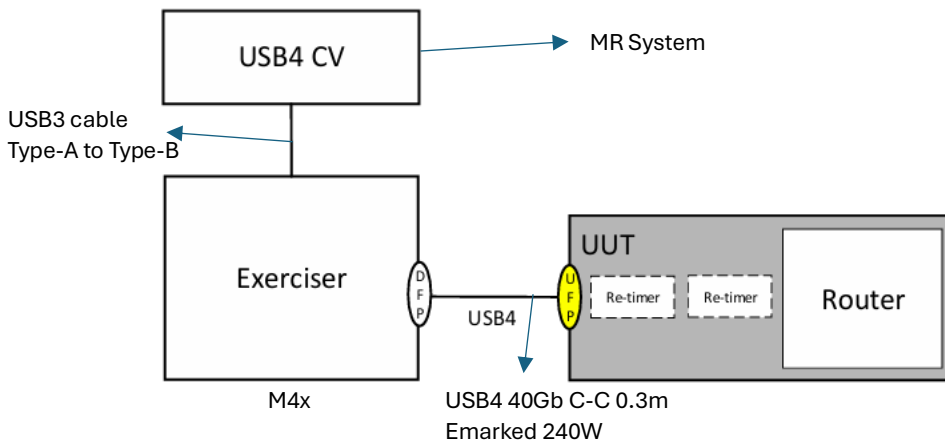


Figure 6.7: Hub UFP Exerciser Setup in The CTS

UUT is a Hub (testing the DFP)

EX_HUB_DFP1

- A KG USB4 host is connected to the UFP of the UUT
- The Exerciser is connected to the DFP of the UUT
- Unless otherwise specified, the Exerciser is in USB4 Device mode and presents as a UFP

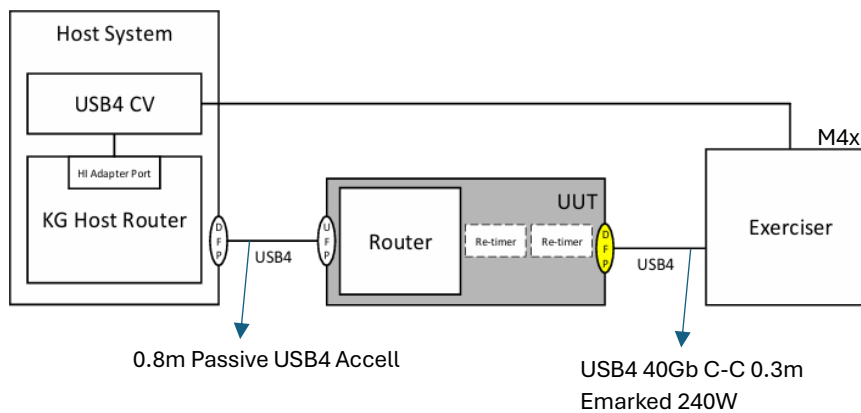


Figure 6.8: Hub DFP Exerciser Setup in The CTS

6.1.7 UUT is a Hub - Chapter 5 AN & Chapter 6 AN & Chapter 8 AN:

When a hub is the UUT then both UFP and DFP are supposed to be tested:

UUT is a Hub (testing the UFP)

AN_HUB_UFP2

Description:

- DFP on KG USB4 Host connects to UFP of the UUT
- Analyzer is connected between the KG USB4 Host and the UUT
- DFP of UUT connects to Compliance Device

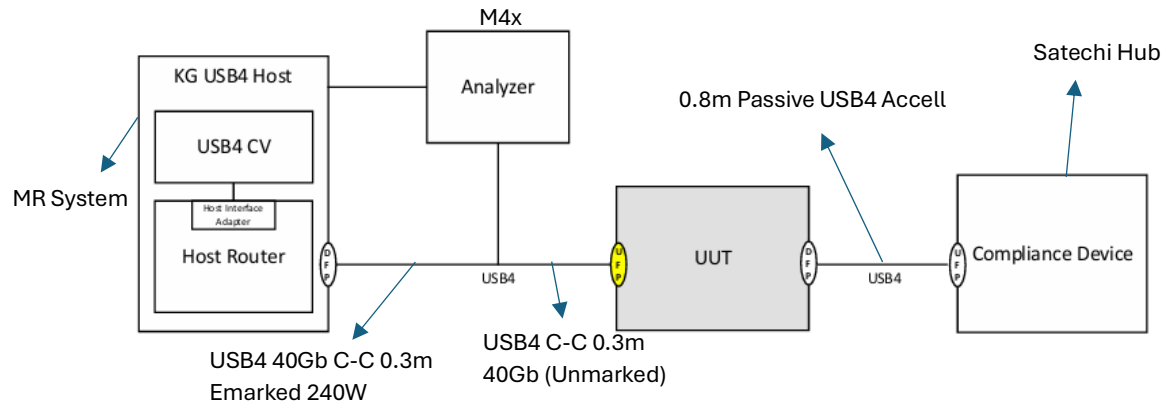


Figure 6.9: Hub UFP Analyzer Setup in The CTS

UUT is a Hub (testing the DFP)

AN_HUB_DFP1

Description:

- DFP on KG USB4 Host connects to UFP of the UUT
- UFP of Compliance Device connects to DFP of the UUT
- Analyzer is connected between the UUT and Compliance Device

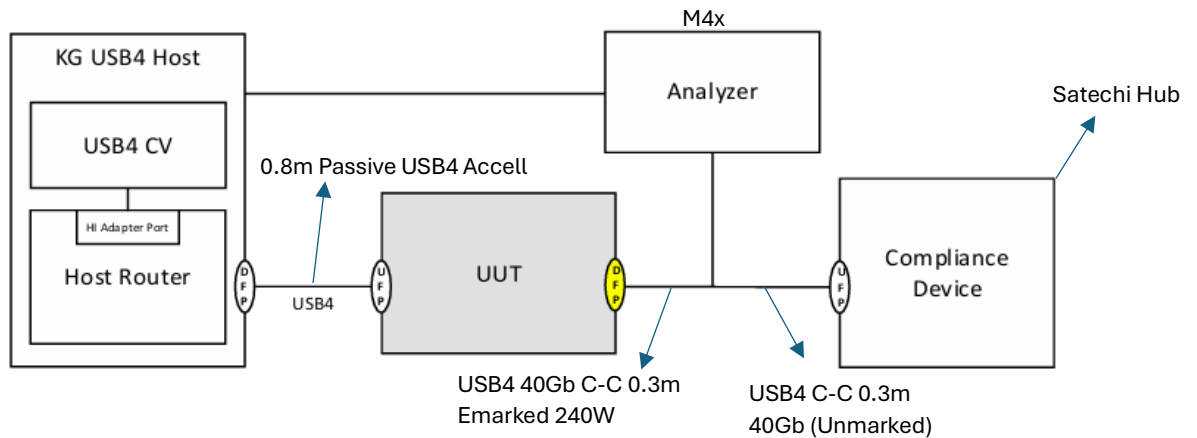


Figure 6.10: Hub DFP Analyzer Setup in The CTS

6.1.8 UUT is a Hub - Chapter 5 EX & Chapter 6 EX & Chapter 8 EX:

When a hub is the UUT then both UFP and DFP are supposed to be tested:

UUT is a Hub (testing the UFP)

EX_HUB_UFP2

Description:

- Exerciser connects to UFP of the UUT
- Unless specified otherwise, Exerciser is in USB4 Host mode and presents as DFP
- DFP of UUT connects to a Compliance Device

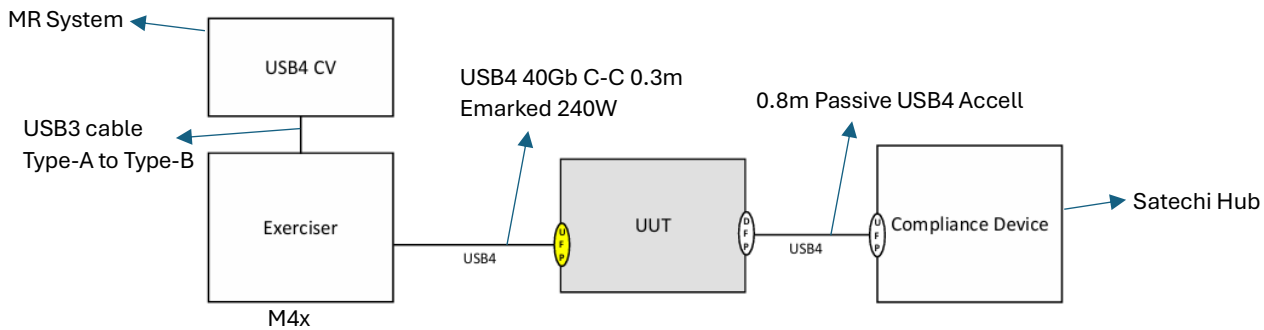


Figure 6.11: Hub UFP Exerciser Setup in The CTS

UUT is a Hub (testing the DFP)

EX_HUB_DFP1

Description:

- DFP on KG USB4 Host connects to UFP of the UUT
- Exerciser connects to DFP of the UUT
- Unless specified otherwise, the Exerciser is in USB4 Host mode and presents as DFP

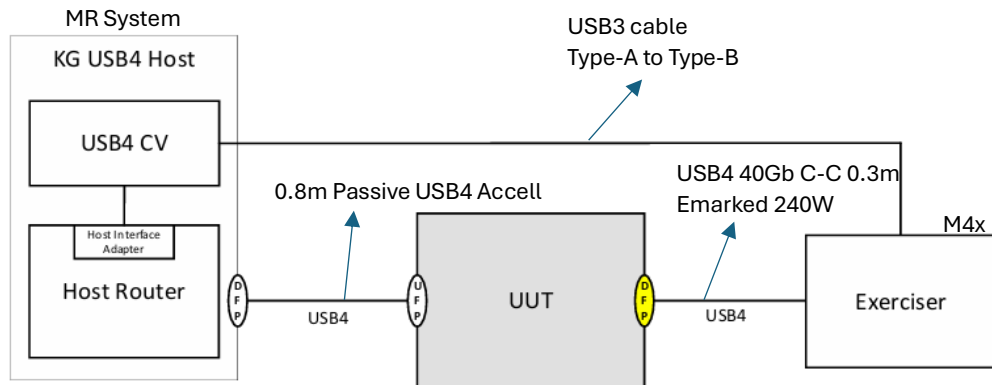


Figure 6.12: Hub DFP Exerciser Setup in The CTS

Note, before running chapter 8, user is expected to run the following test first, “**Enabled PCIe Tunneling in USB4 CV**” test. User have to filter test by “All Setups” then Enable PCIe Tunneling in USB4CV will be the first test. Once the user runs this test, a reboot is required before running chapter 8. Then once chapter 8 is finished, the user must run “Disable PCIe Tunneling in USB4CV” test and reboot the system again. Please refer to the release notes for more information on this.

USB4 CV will also ask the user to enter **PCIe Entry Mapping Segment** once the user starts running the test. If Maple Ridge system is being used, then user expected to enter 0. If the other host is being used as UUT then vendors are expected to provide that information.

Note, these setups may change. So always download the latest CTS and follow the setup in the CTS.

6.2 Chapter 7

Chapter 7, the Time Measurement Equipment needs to be connected to the Host. Read the Time Measurement Equipment - Setup Procedure (Appendix) to set up the equipment correctly. Once you have finished setting up the equipment, open the USB4 TMU document in the CTS and make sure you have the correct setup before running the test.

Note, any other connections between hosts/devices/hubs that are not going to the analyzer should use a 0.8m Accell USB4 passive cable.

Note, if the setup below doesn't match the setup in the Compliance Test Specification (CTS), then users must use the setup in the CTS and ignore the setups in this document.

6.2.1 UUT is a Host - Chapter 7:

SD_HOST_DFP_TMU_01

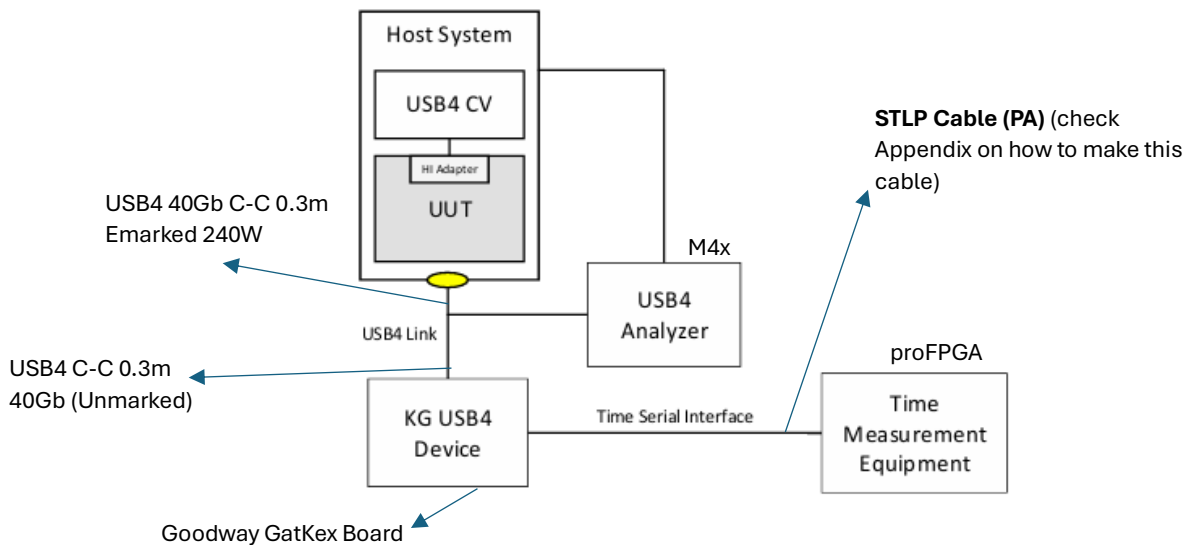


Figure 6.13: Chapter 7 Setup 1 Host Analyzer Setup in The CTS

SD_HOST_DFP_TMU_05

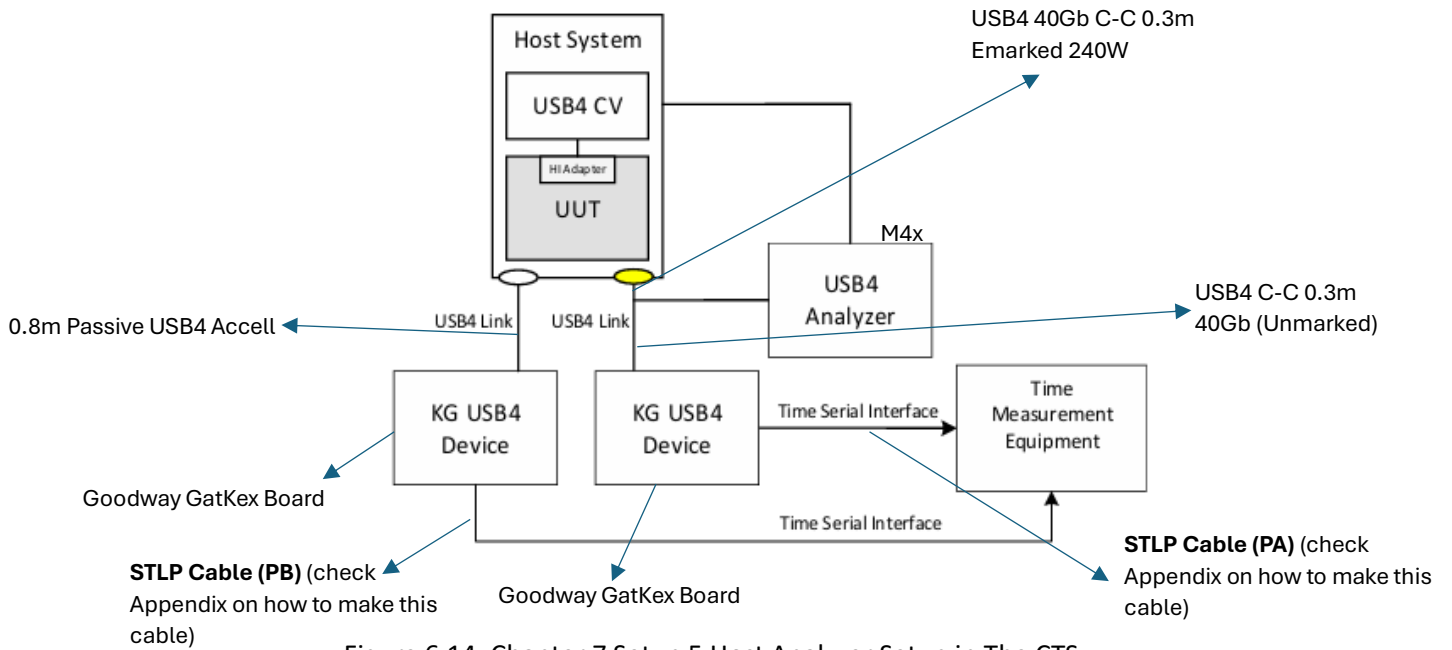


Figure 6.14: Chapter 7 Setup 5 Host Analyzer Setup in The CTS

6.2.2 UUT is a Device - Chapter 7:

SD_DEV_UFP_TMU_15

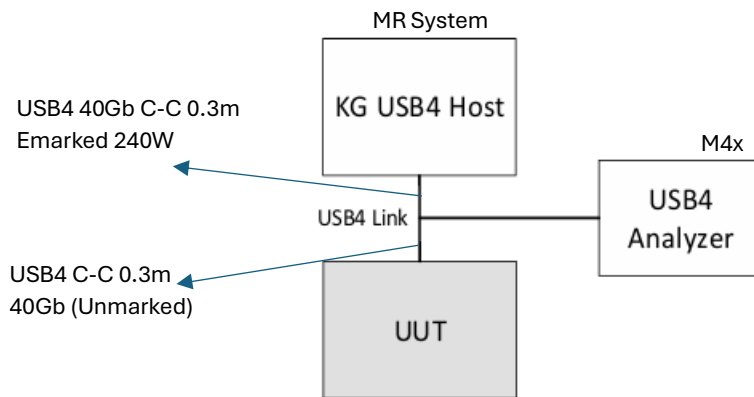


Figure 6.15: Chapter 7 Setup 15 Device Analyzer Setup in The CTS

SD_DEV_UFP_TMU_16

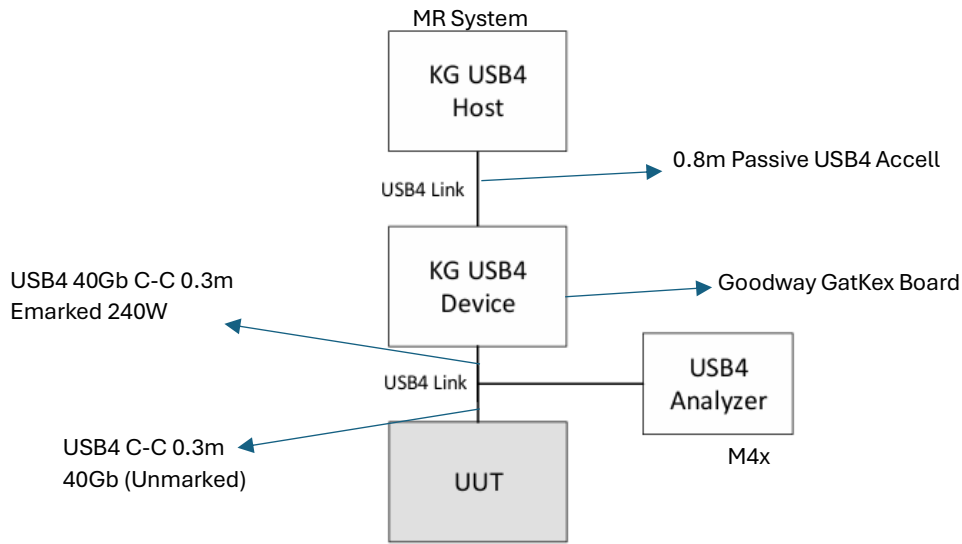


Figure 6.16: Chapter 7 Setup 16 Device Analyzer Setup in The CTS

6.2.3 UUT is a Hub - Chapter 7:

When a hub is the UUT then both UFP and DFP are supposed to be tested:

UUT is a Hub (testing the UFP)

SD_HUB_UFP_TMU_08

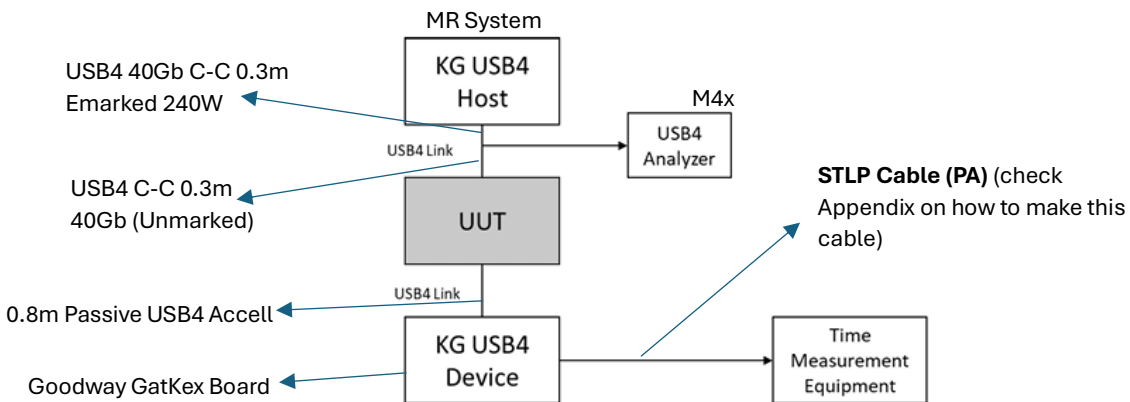


Figure 6.17: Chapter 7 Setup 8 Hub UFP Analyzer Setup in The CTS

SD_HUB_UFP_TMU_10

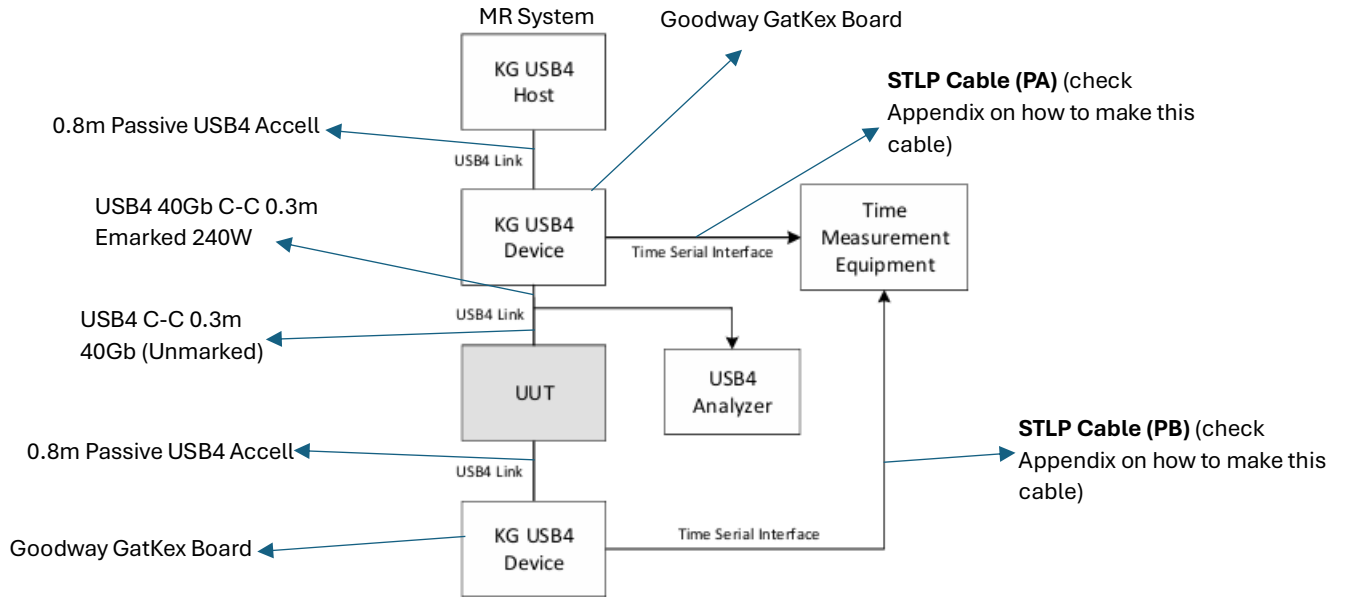


Figure 6.18: Chapter 7 Setup 10 Hub UFP Analyzer Setup in The CTS

UUT is a Hub (testing the DFP)

SD_HUB_DFP_TMU_09

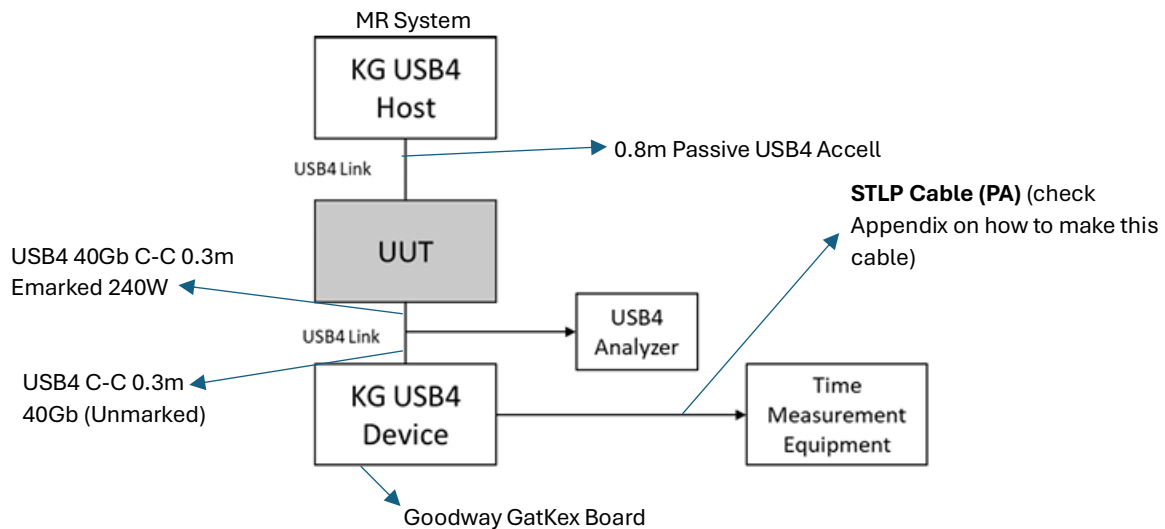


Figure 6.19: Chapter 7 Setup 9 Hub DFP Analyzer Setup in The CTS

SD_HUB_DFP_TMU_11

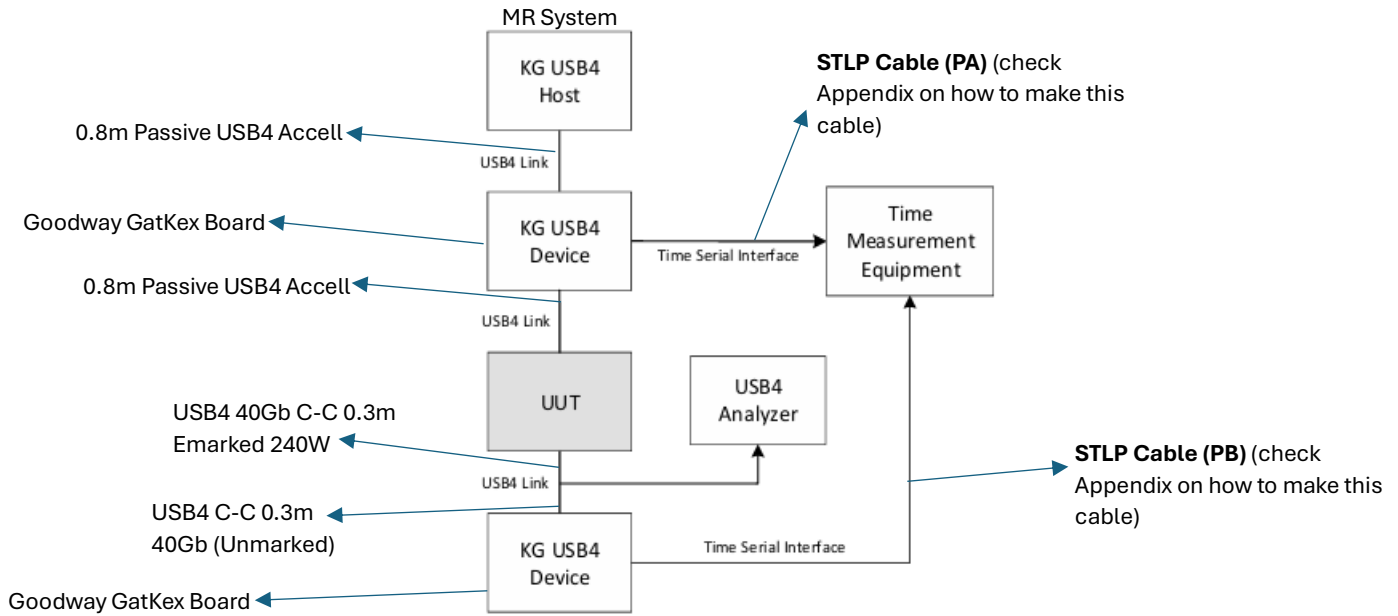
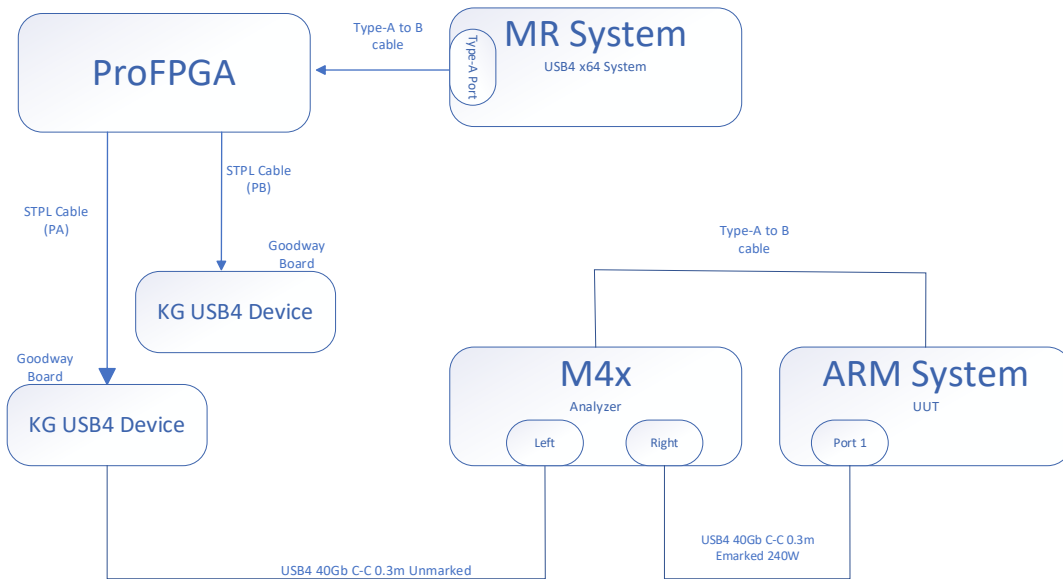


Figure 6.20: Chapter 7 Setup 11 Hub DFP Analyzer Setup in The CTS

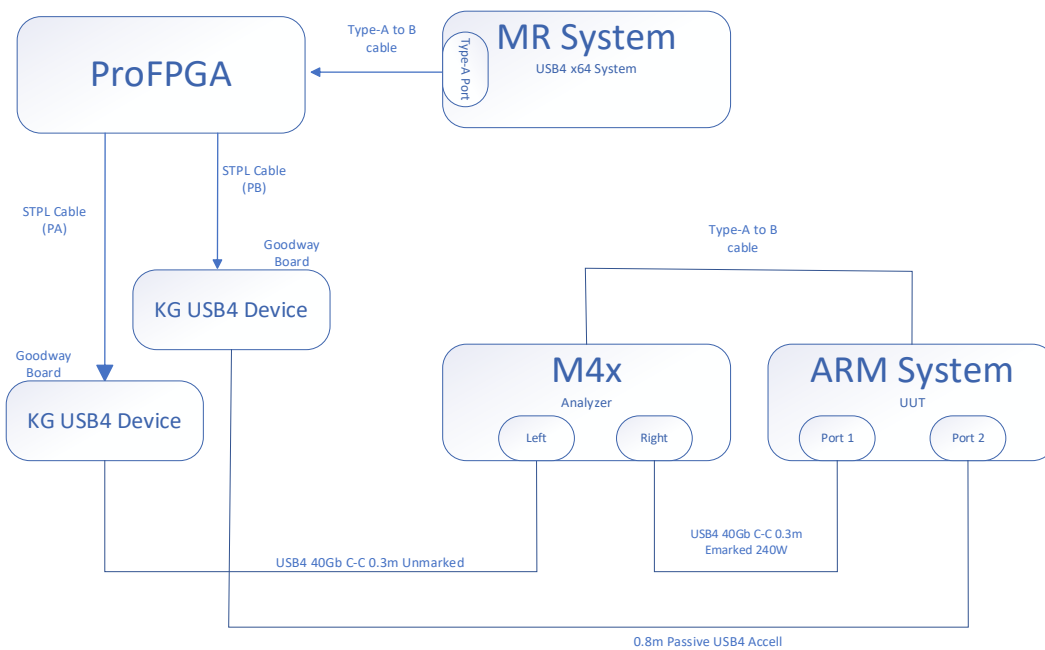
For Win-ARM system UUT (non-x64 systems), to run chapter 7 test on the Win-ARM system, two computers are needed. One is the UUT, and the second computer is the USB4 Win-x64 system. Make sure both computers have the same USB4 CV version and USB Protocol Suite.

On the USB4 Win-x64 system, connect and setup the proFPGA. Refer to Time Measurement Equipment - Setup Procedure document on usb.org website (Appendix) to set up the proFPGA. Once the setup is done, turn on the FPGA and start the proFPGA (burning the bit file).

SD_HOST_DFP_TMU_01



SD_HOST_DFP_TMU_05



Now on the Win-x64 machine, open USB4 CV and filter the tests by Host tests and make sure Debug is selected in the test mode.

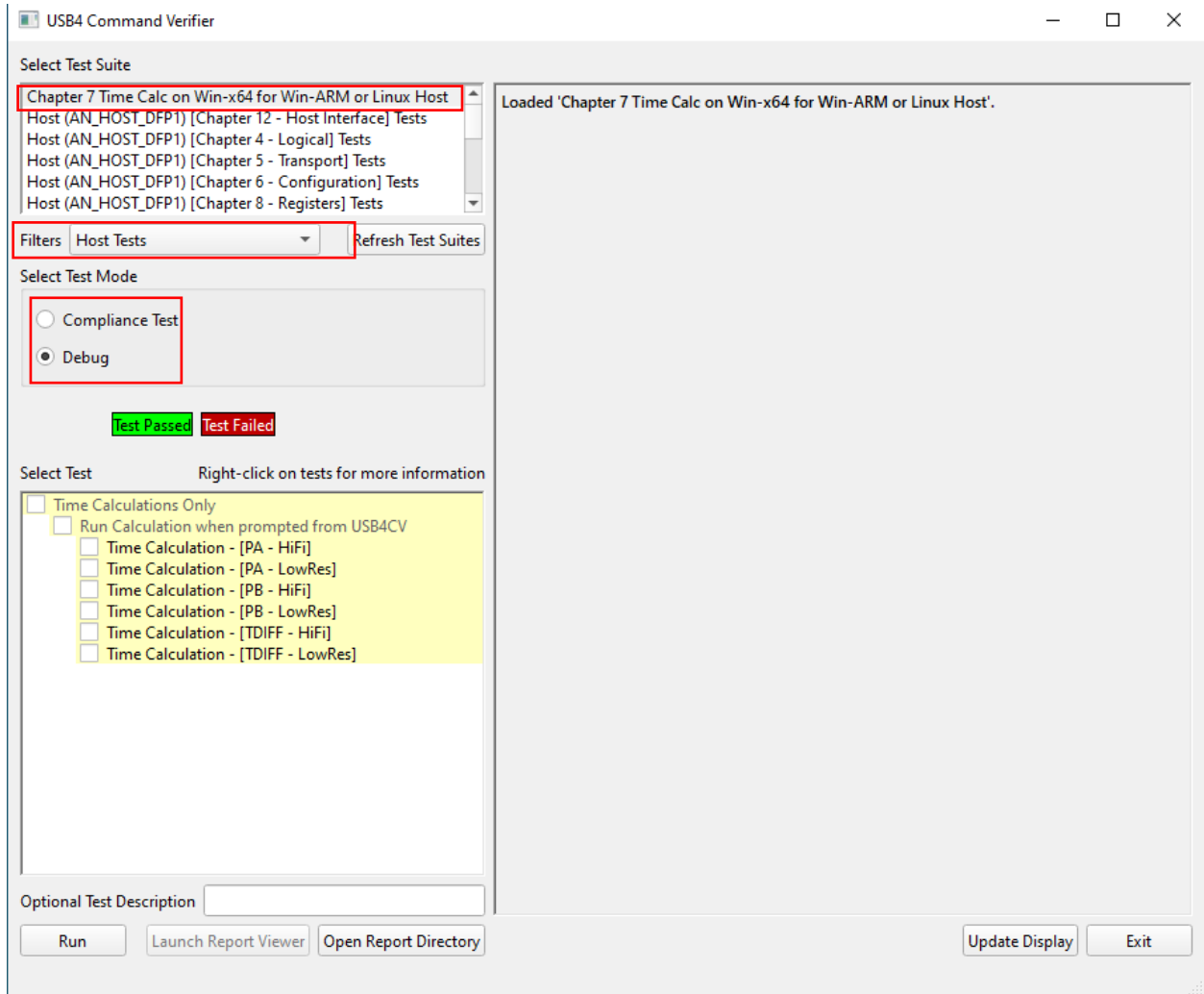


Figure 6.21: USB4 CV Window In The USB4 Win-x64 System

On the UUT system, make sure the setup for chapter 7 is connected to the UUT. Open USB4 CV and filter the tests by Host. Select USB4 Host Single-Port Win ARM or Linux (setup 1) if the UUT has one port for one router. Press Run

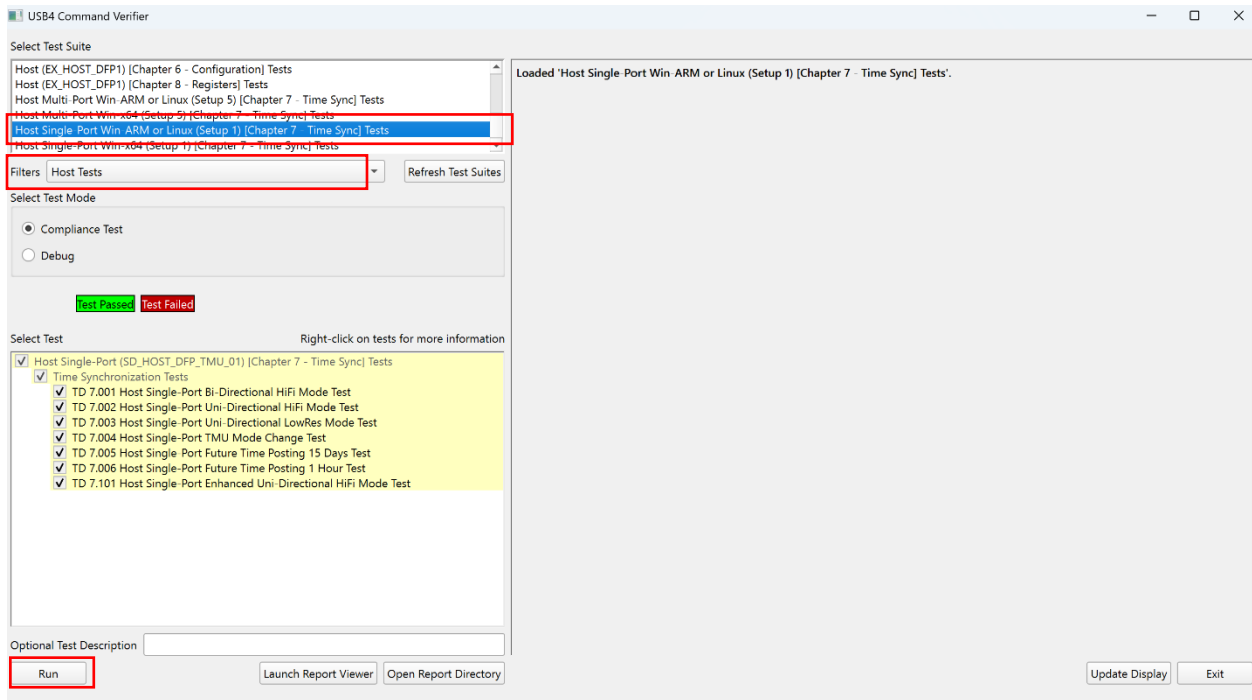


Figure 6.22: USB4 CV Window In The UUT System (Win-ARM)

Once you press run in the UUT system, a pop-up window will appear while the test is running. The pop-up window will ask the user to run a test in the USB4 Win-x64 system.

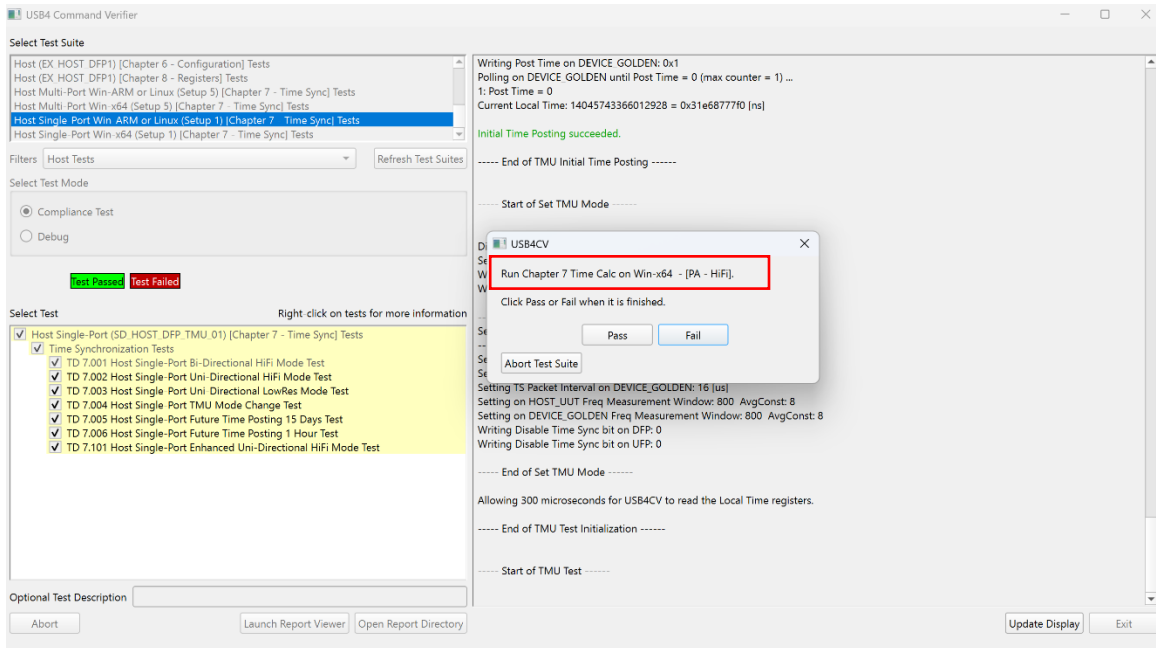


Figure 6.23: Pop-up window in the UUT system

Now go back to the USB4 Win-x64 system and look for the test in the UUT system pop-up window. Press Run

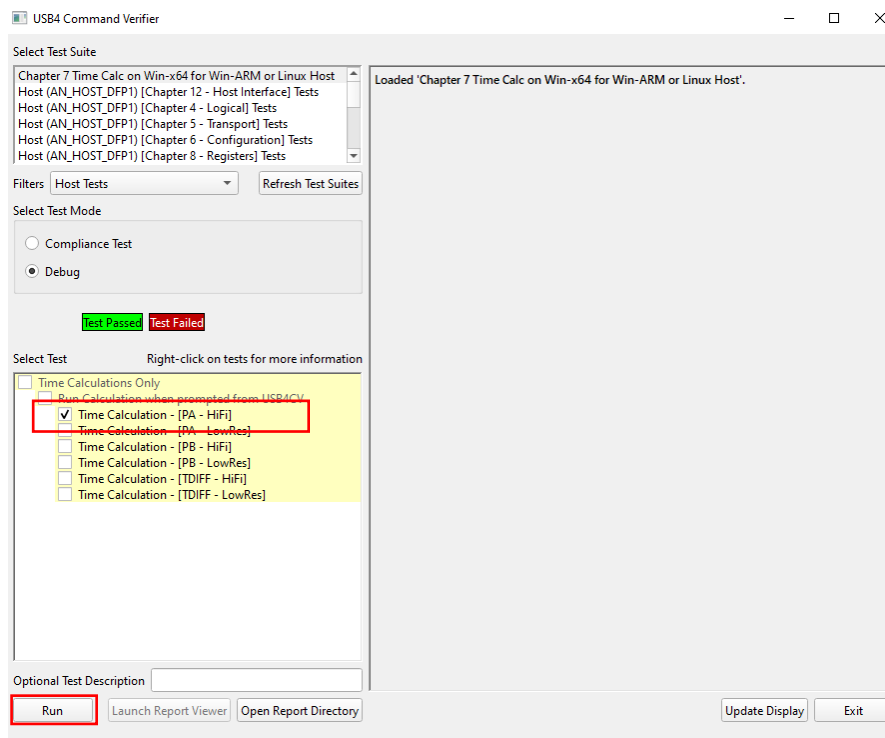


Figure 6.24: Select The Test In The USB4 Win-x64 System

Wait for the test to finish in the USB4 Win-x64 system first. Then press OK

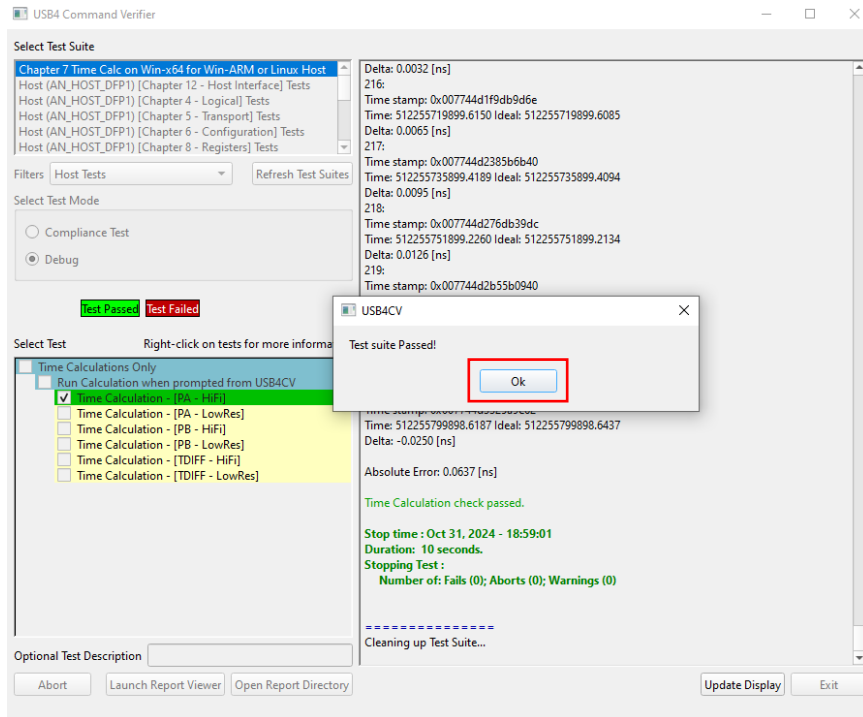


Figure 6.25: USB4 Win-x64 System Window

Go back to the UUT system, and press pass if the test passes in the USB4 Win-x64 system. Otherwise, press fail

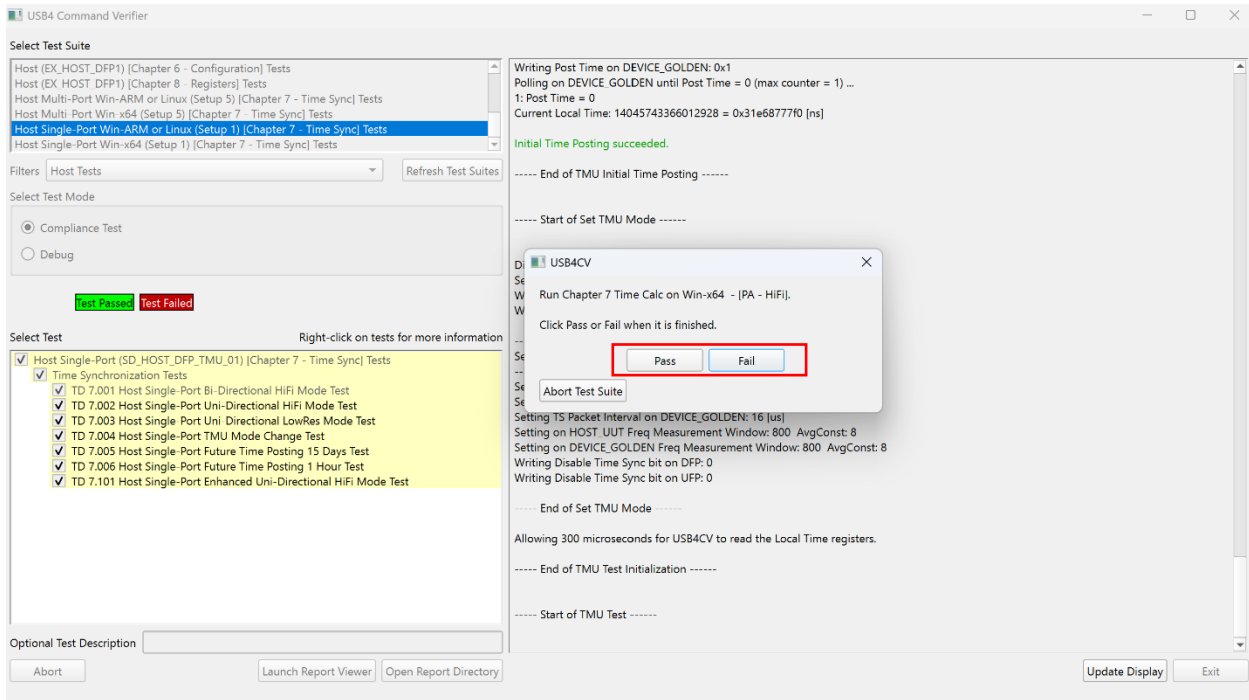


Figure 6.26: UUT System Window

Note, in some test cases, the user must run the Time calculation test in the USB4 Win-x64 system multiple times.

Once all tests are finished in the UUT system, save all logs and traces in both the UUT system and the USB4 x64 system. The user will need to submit both results.

Note, any other connections between hosts/devices/hubs that are not going to the analyzer should use a 0.8m Accell USB4 passive cable.

6.3 Chapter 9

Please refer to the CTS for chapter 9 (USB4 USB3 Tunneling) for the setup for this test. Note, for this setup, users must select the correct USB3 router. Selecting the wrong router will cause the mouse/keyboard and any usb2 to stop working. Users must make sure that the router they are selecting for the xHCI to take over doesn't have the mouse and the keyboard under it.

Note: when using the MCCI as a USB3.2 Compliance device, the MCCI device should be in a compliance mode.

Note, if the setup below doesn't match the setup in the Compliance Test Specification (CTS), then users must use the setup in the CTS and ignore the setups in this document.

6.3.1 UUT is a Host - Chapter 9:

AN_HOST_DFP1—USB3_01

Description:

- The UFP of the Compliance Device is connected to the DFP of the UUT.
- The USB4 Analyzer is connected between the UUT and the Compliance Device.

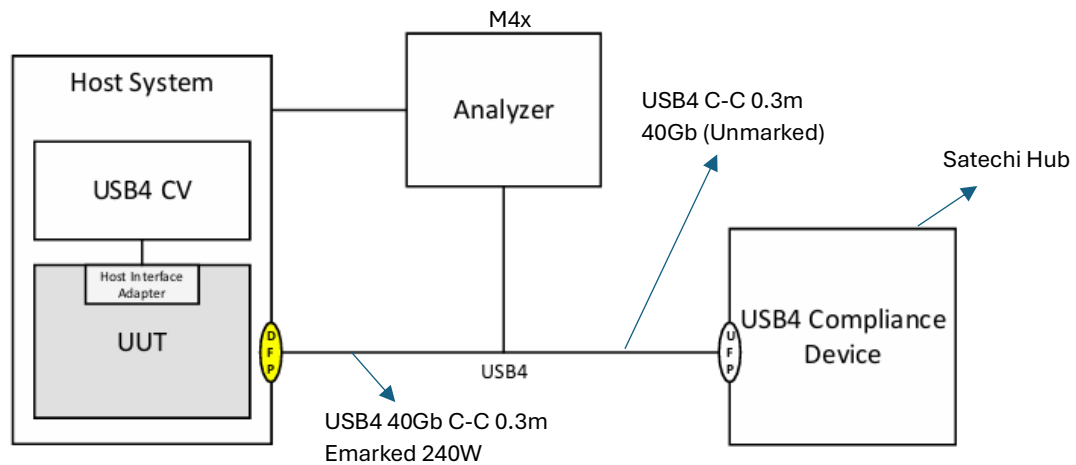


Figure 6.27: Chapter 9 USB3_01 Host Setup in The CTS

AN_HOST_DFP2—USB3_03

Description:

- The UFP of the USB4 Compliance Device is connected to the DFP of the UUT.
- The USB4 Analyzer is connected between the UUT and the USB4 Compliance Device.
- A USB 3.2 Compliance Device is connected downstream of the USB4 Compliance Device

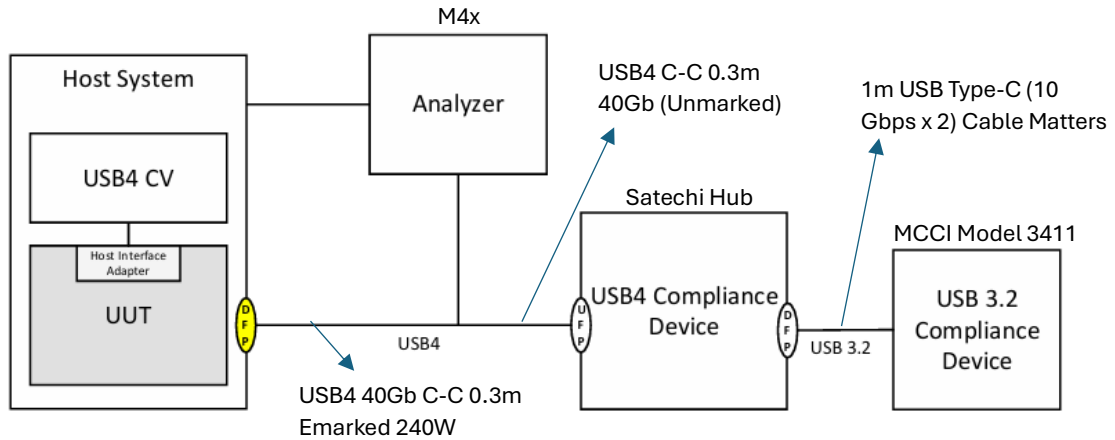


Figure 6.28: Chapter 9 USB3_03 Host Setup in The CTS

6.3.2 UUT is a Device - Chapter 9:

AN_DEV_UFP1—USB3_01

Description:

- The DFP on the KG USB4 Host connects to the UFP of the UUT.
- The Analyzer is connected between the KG USB4 Host and the UUT.

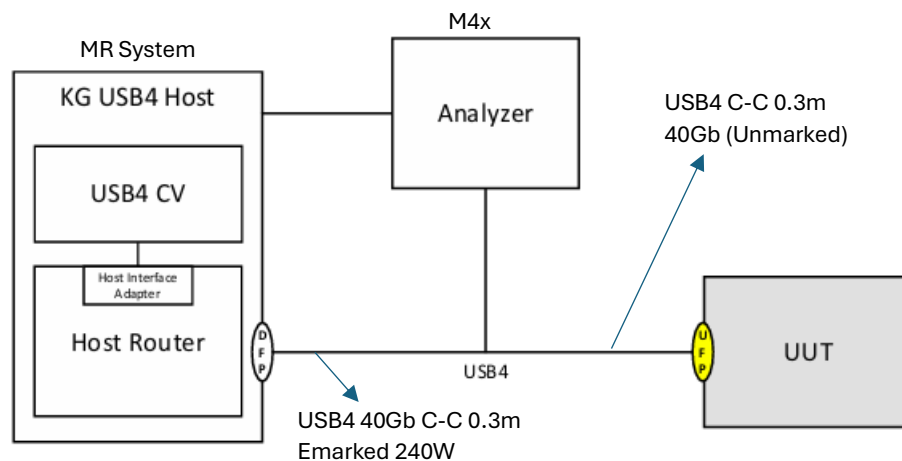


Figure 6.29: Chapter 9 USB3_01 Device Setup in The CTS

AN_DEV_UFP2—USB3_04

Description:

- The DFP on the KG USB4 Host connects to the UFP of the UUT.
- The Analyzer is connected between the KG USB4 Host and the UUT.
- A USB 3.2 Compliance Device is connected downstream of the UUT.

Note: This setup is only used for a UUT that has one or more non-USB4 DFP that support USB 3.2.

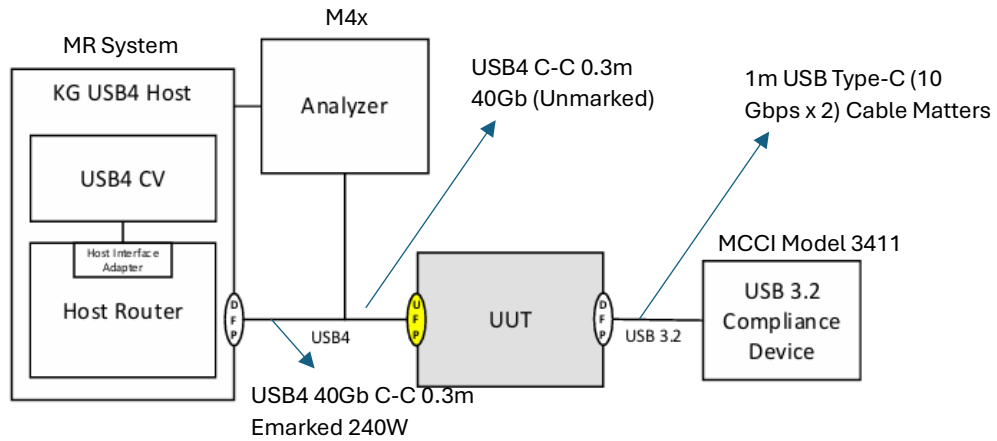


Figure 6.30: Chapter 9 USB3_04 Device Setup in The CTS

6.3.3 UUT is a Hub - Chapter 9:

When a hub is the UUT then both UFP and DFP are supposed to be tested:

UUT is a Hub (testing the UFP)

AN_HUB_UFP1—USB3_01

Description:

- The DFP on the KG USB4 Host connects to the UFP of the UUT.
- The Analyzer is connected between the KG USB4 Host and the UUT.

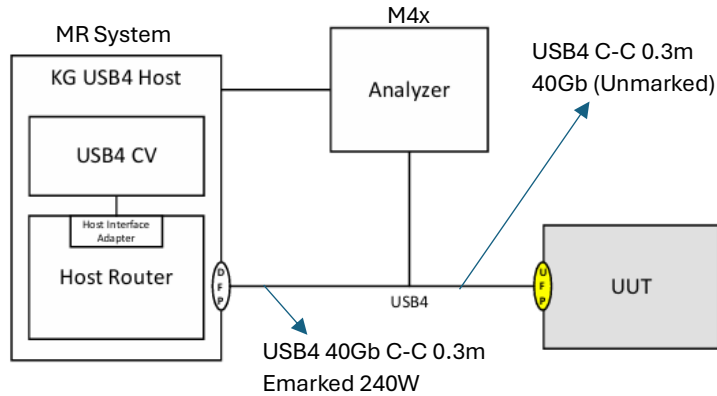


Figure 6.31: Chapter 9 USB3_01 Hub UFP Setup in The CTS

AN_HUB_UFP2_USB3_06

- The DFP on the KG USB4 Host connects to the UFP of the UUT.
- The Analyzer is connected between the KG USB4 Host and the UUT.
- A USB 3.2 Compliance Device is connected downstream of the UUT.

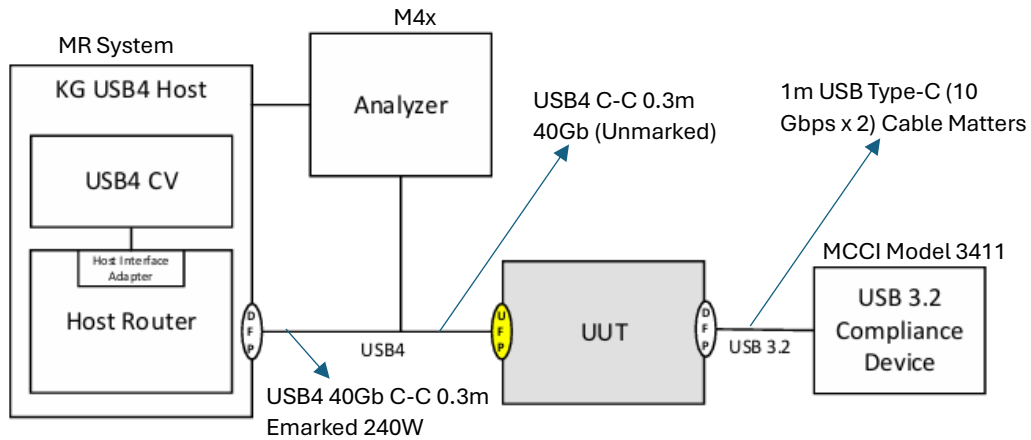


Figure 6.32: Chapter 9 USB3_06 Hub UFP Setup in The CTS

UUT is a Hub (testing the DFP)

[AN_HUB_DFP1—USB3_03](#)

Description:

- DFP on KG USB4 Host connects to UFP of the UUT.
- UFP of Compliance Device connects to DFP of the UUT.
- Analyzer is connected between the UUT and Compliance Device.

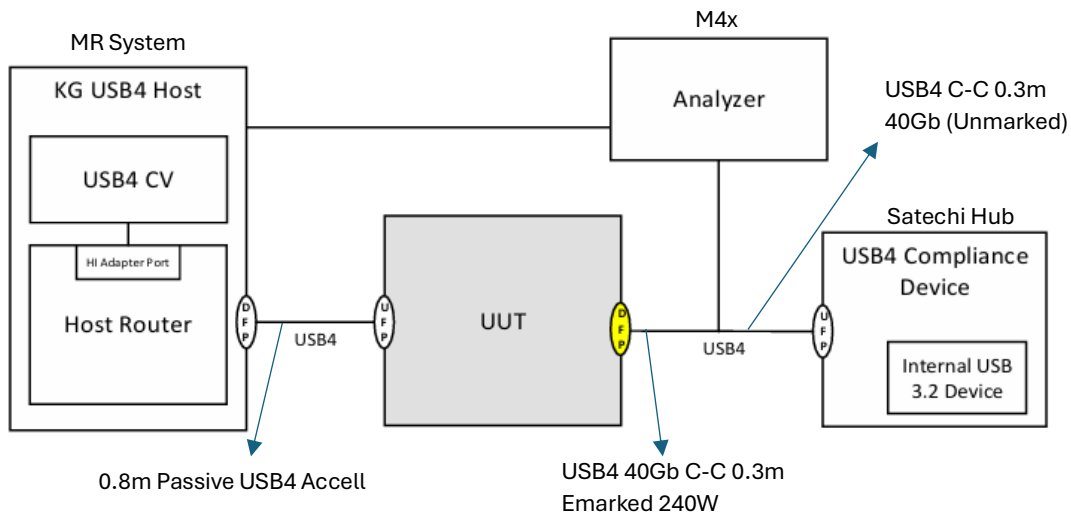


Figure 6.33: Chapter 9 USB3_03 Hub DFP Setup in The CTS

6.4 Chapter 10

Please refer to the CTS for the setup for the tests. A DP tester (UCD500 Gen2, M42De, etc..) is required for some of the tests below. **Below are just basic notes regarding some of the tests. Users must run all required chapter 10 tests**

Note, if the setup below doesn't match the setup in the Compliance Test Specification (CTS), then users must use the setup in the CTS and ignore the setups in this document.

TD10.001:

User must run AN_HOST_DFP1—DPIN_01.

TD10.002:

Even though CTS says this test does not require the use of the USB4 CV. User are still expected to use USB4 CV. User must run the correct tables mentioned in TD10.002 test CTS.

TD10.003:

Users are still expected to use USB4 CV.

TD10.004:

Below is the iteration that is supposed to be tested. **Please check the CTS for the latest iterations**

- Link: HBR2 x 4 Lanes ; Monitors: Two FHD (1080p)
- Link: HBR3 x 4 Lanes ; Monitors: Two 4K 8 pbc
- Link: HBR2 x 2 Lanes ; Monitors: One FHD (1080p)
- Link: RBR x 4 Lanes ; Monitors: One FHD (1080p)
- Link: HBR3 x 1 Lane ; Monitors: One FHD (1080p)

TD10.005 & TD10.006:

Note, users are expected to see a clear image on the monitor

TD10.007:

Check CTS for the latest notes regarding this test

TD10.008:

HDCP must be enabled on both Host and DP tester

TD10.010:

Even though CTS says this test does not require the use of the USB4 CV. User are still expected to use USB4 CV. User must run the correct tables mentioned in TD10.010 test CTS.

TD10.011:

Even though CTS says this test does not require the use of the USB4 CV. User are still expected to use USB4 CV.

TD10.012:

Below is the iteration that is supposed to be tested. **Please check the CTS for the latest iterations**

- Link: HBR2 x 4 Lanes ; Monitors: Two FHD (1080p)
- Link: HBR3 x 4 Lanes ; Monitors: Two 4K 8 pbc
- Link: HBR2 x 2 Lanes ; Monitors: One FHD (1080p)
- Link: RBR x 4 Lanes ; Monitors: One FHD (1080p)
- Link: HBR3 x 1 Lane ; Monitors: One FHD (1080p)

TD10.013 & TD10.014:

Note, users are expected to see a clear image on the monitor

TD10.015:

HDCP must be enabled on both Host and DP tester

6.4.1 UUT is a Host – Chapter 10:

AN_HOST_DFP1-DPIN_01

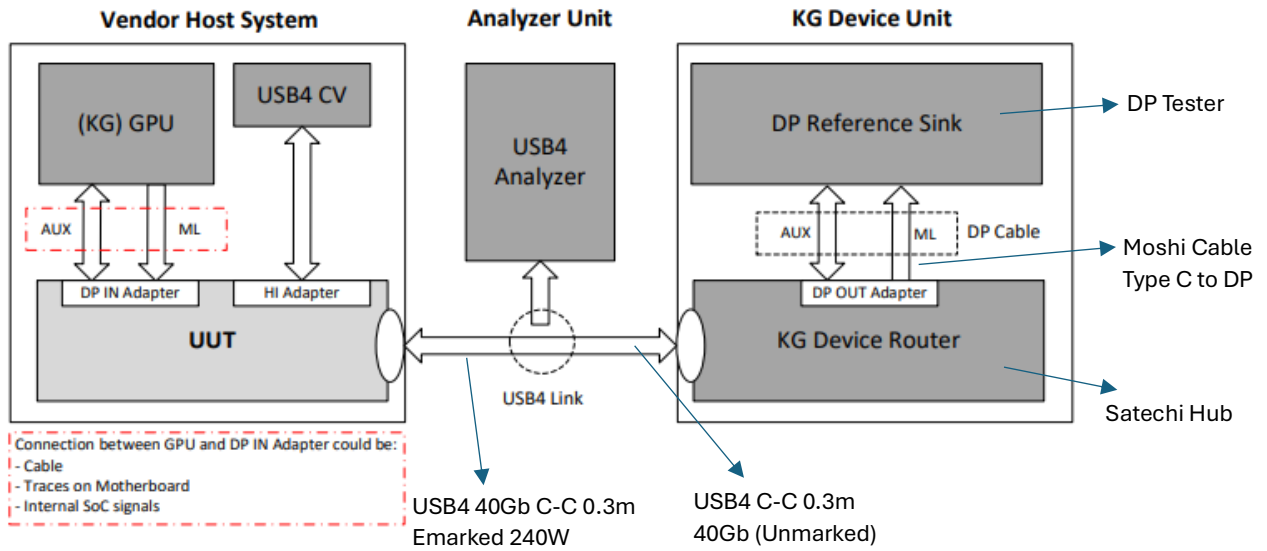


Figure 6.35: Chapter 10 AN_HOST_DFP1-DPIN_01 Setup in The CTS

AN_HOST_DFP1-DPIN_02

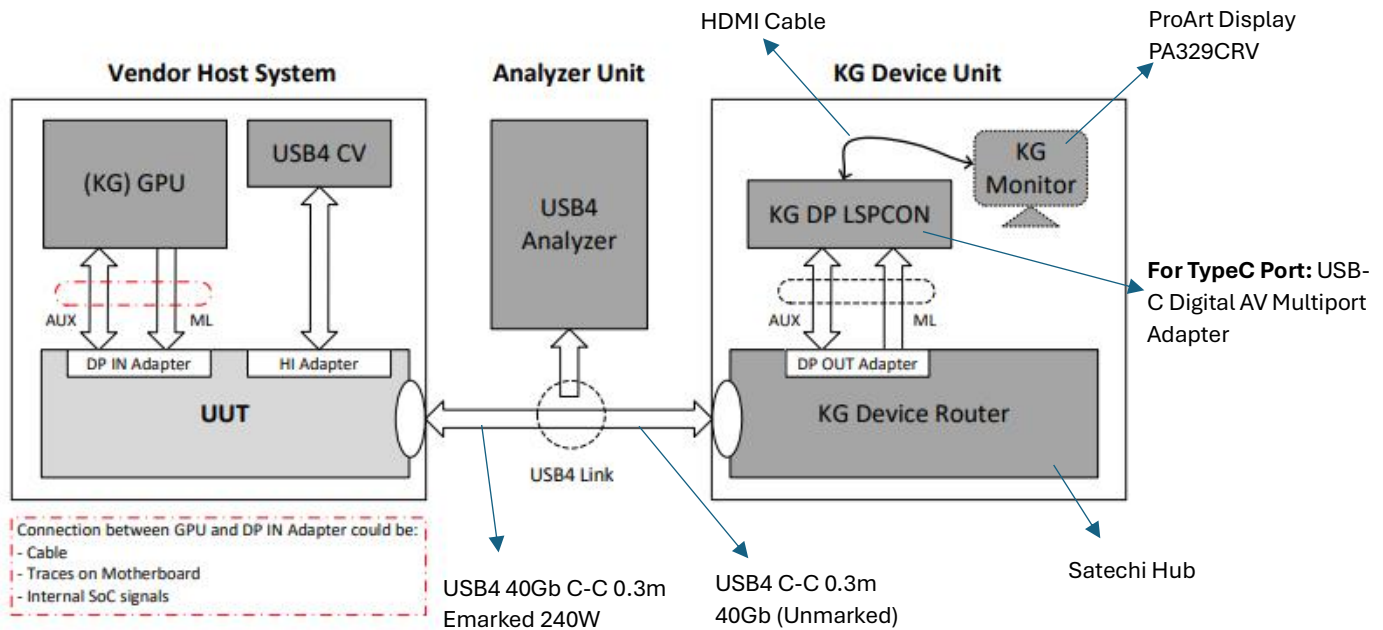


Figure 6.36: Chapter 10 AN_HOST_DFP1-DPIN_02 Setup in The CTS

AN_HOST_DFP1-DPOUT_01

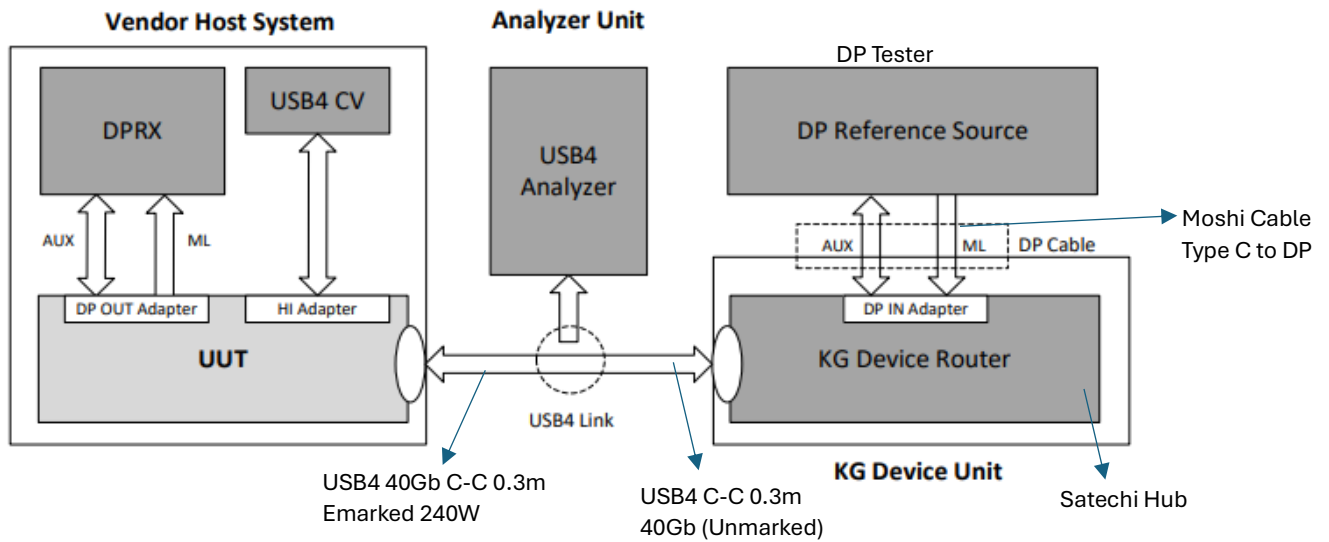


Figure 6.37: Chapter 10 AN_HOST_DFP1-DPOUT_01 Setup in The CTS

6.4.2 UUT is a Device – Chapter 10:

AN_DEV_UFP1—DPOUT_02

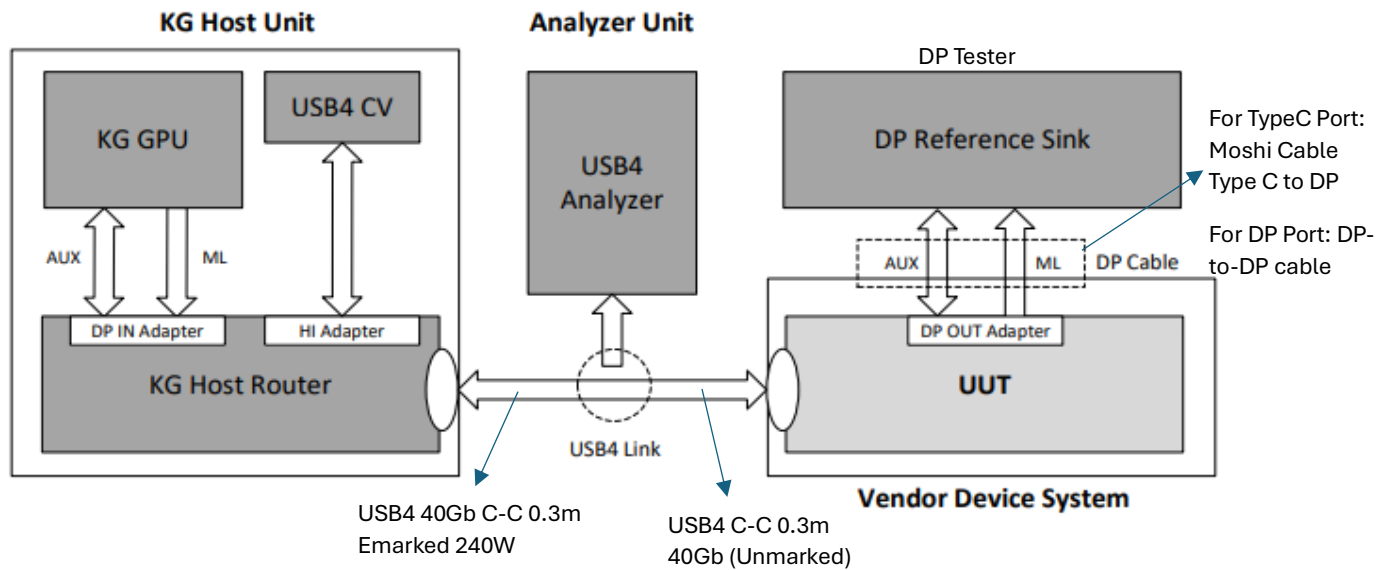


Figure 6.38: Chapter 10 AN_DEV_UFP1—DPOUT_02 Setup in The CTS

AN_DEV_UFP1—DPOUT_03

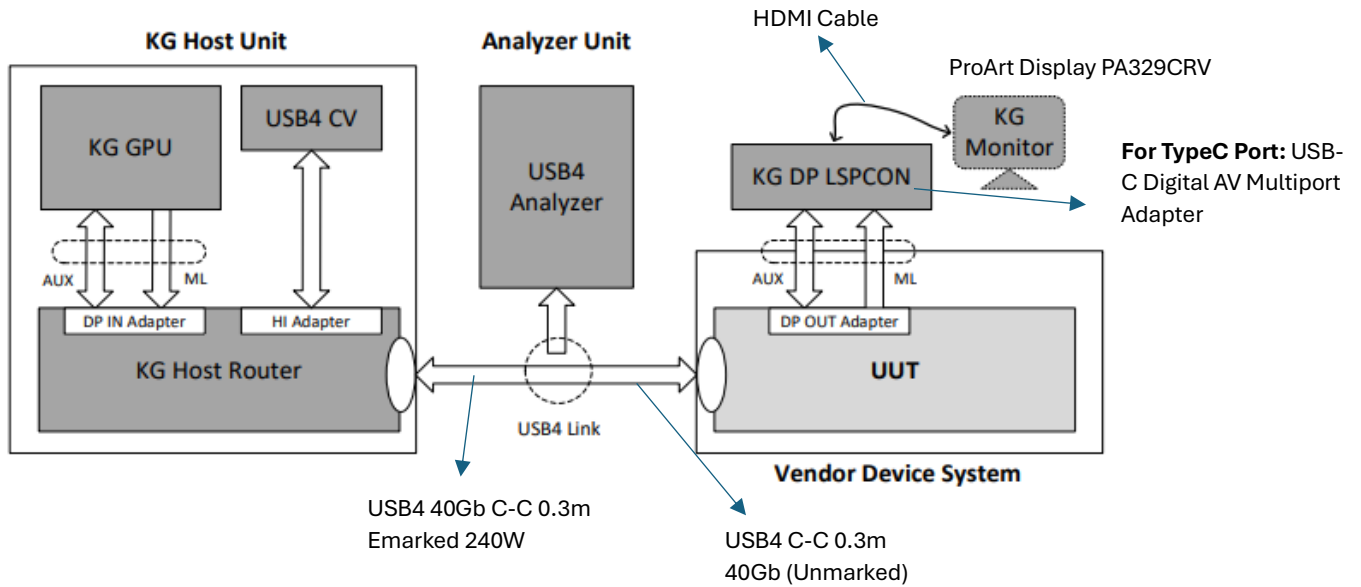


Figure 6.39: Chapter 10 AN_DEV_UFP1—DPOUT_03 Setup in The CTS

AN_DEV_UFP1—DPOUT_04

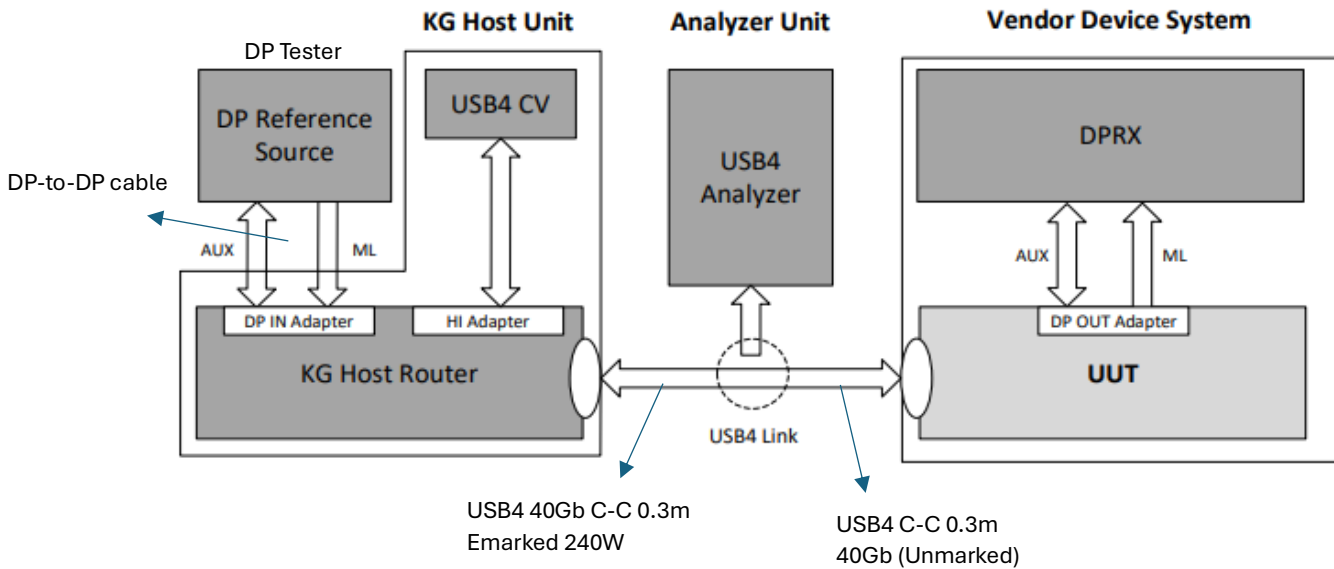


Figure 6.40: Chapter 10 AN_DEV_UFP1—DPOUT_04 Setup in The CTS

AN_DEV_UFP1—DPOUT_05

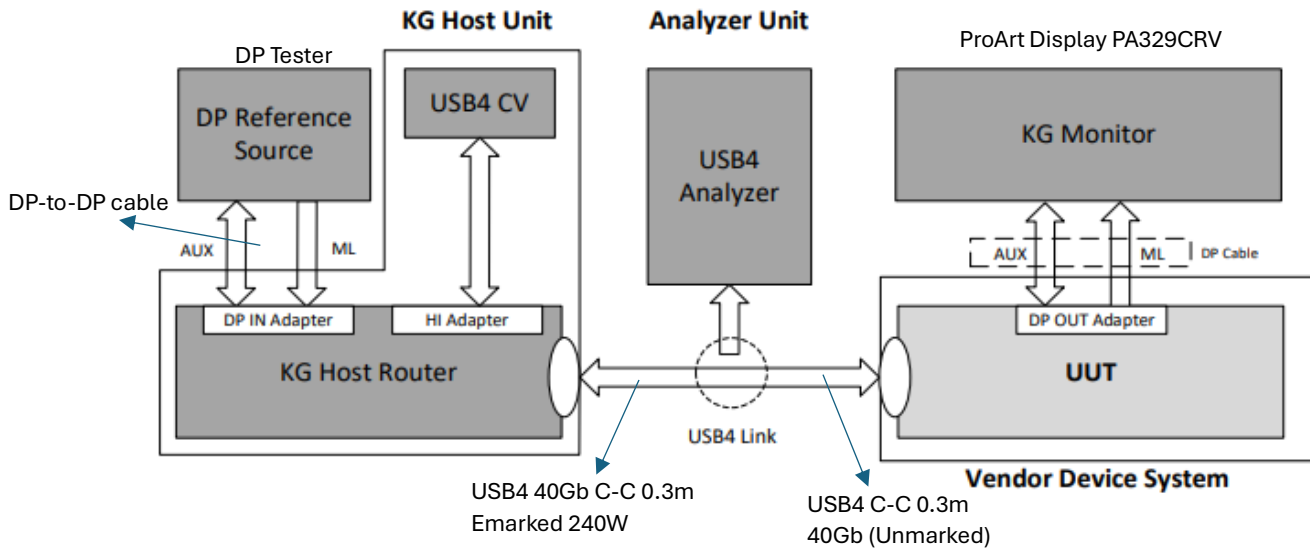


Figure 6.41: Chapter 10 AN_DEV_UFP1-DPOUT_05 Setup in The CTS

6.4.3 UUT is Hub – Chapter 10:

AN_HUB_UFP1—DPOUT_06

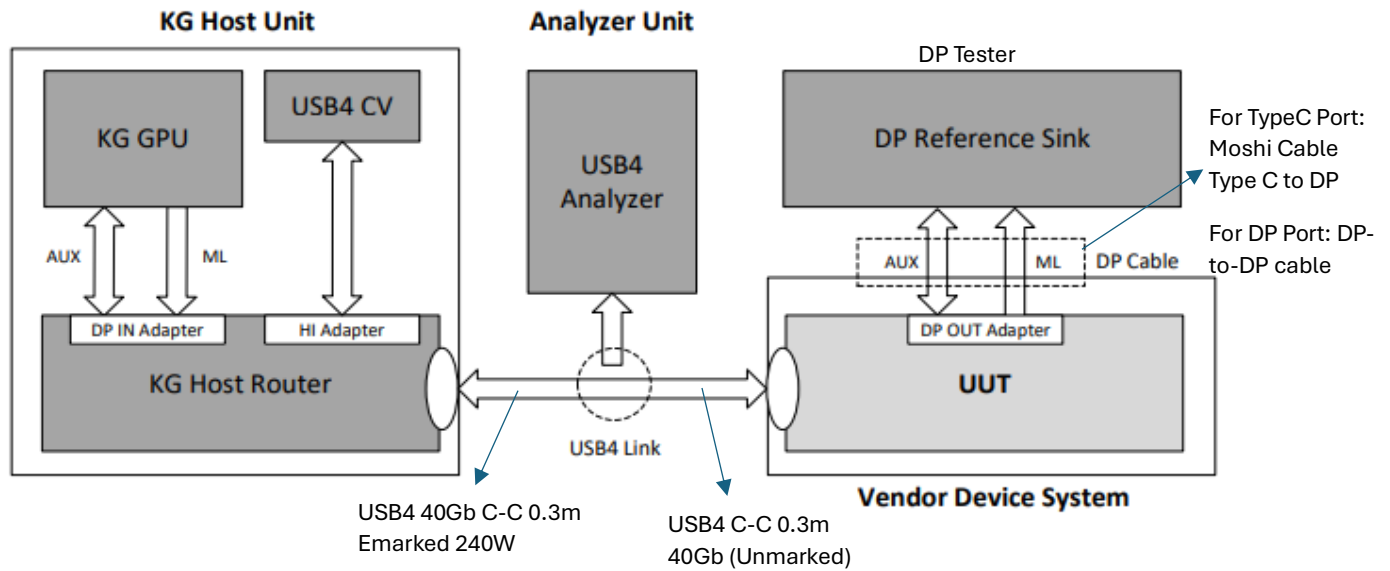


Figure 6.42: Chapter 10 AN_HUB_UFP1—DPOUT_06 Setup in The CTS

AN_HUB_UFP1—DPOUT_07

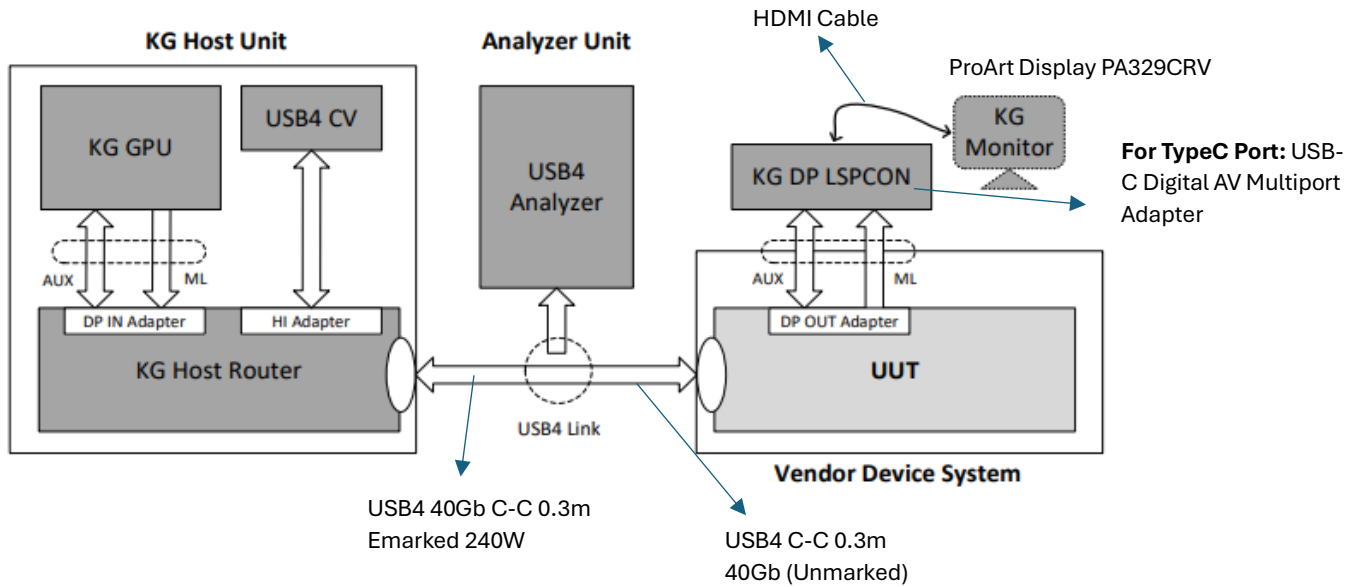


Figure 6.43: Chapter 10 AN_HUB_UFP1—DPOUT_07 Setup in The CTS

AN_HUB_UFP1—DPOUT_08

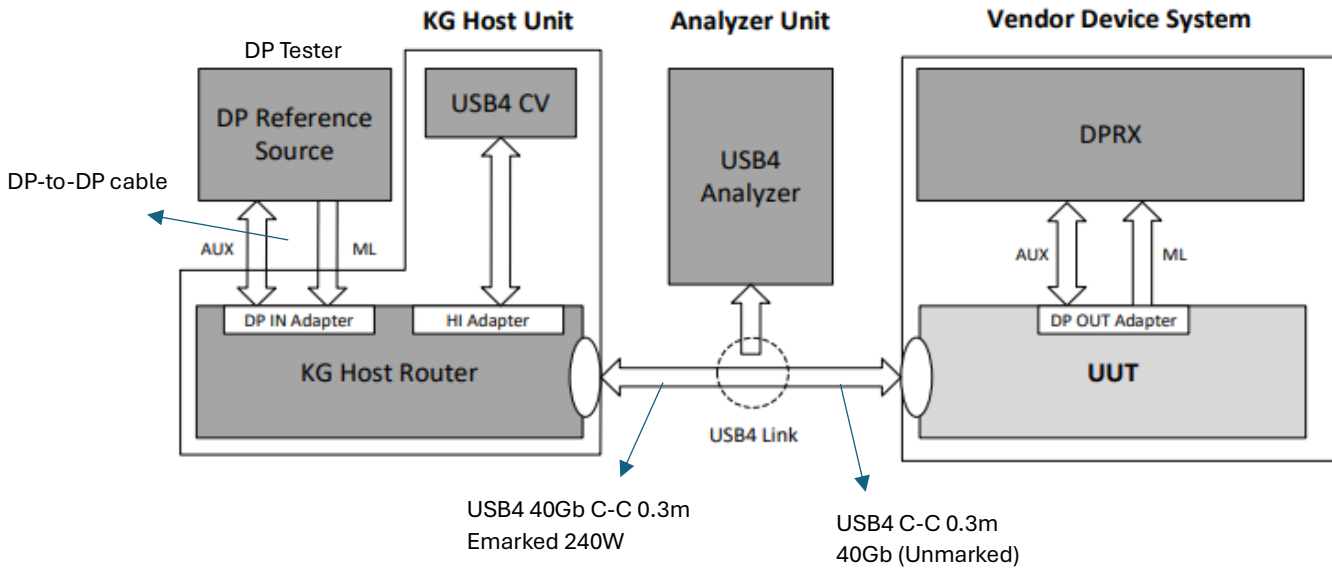


Figure 6.44: Chapter 10 AN_HUB_UFP1—DPOUT_08 Setup in The CTS

AN_HUB_UFP1—DPOUT_09

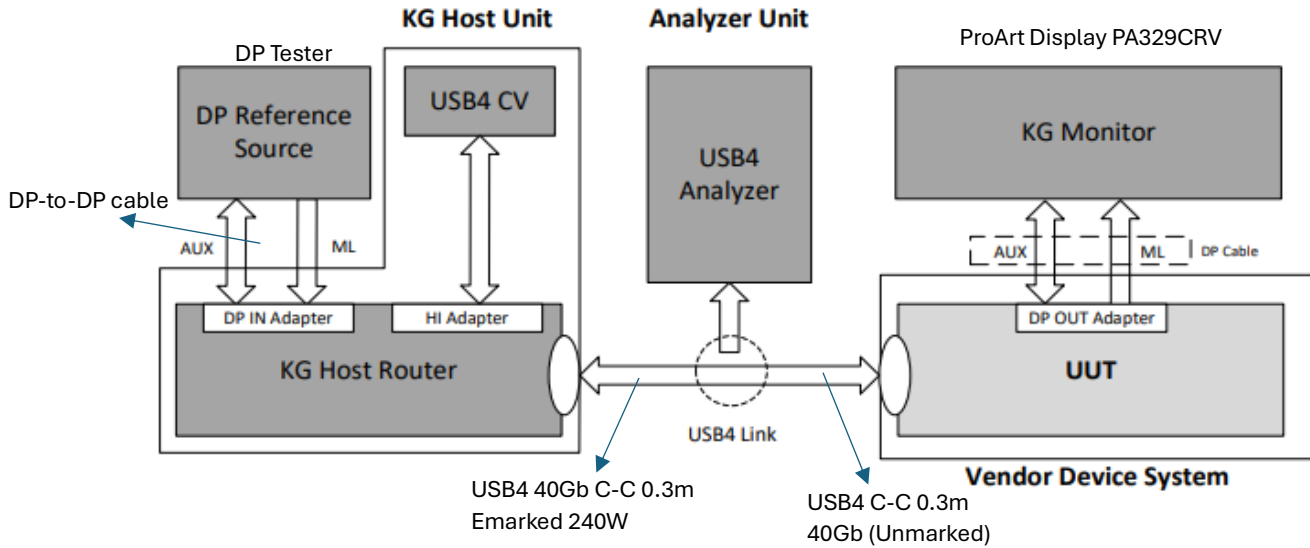


Figure 6.45: Chapter 10 AN_HUB_UFP1-DPOUT_09 Setup in The CTS

6.5 Chapter 11

Please refer to the CTS for chapter 11 (USB4 PCIe Tunneling) for the setup for this test

Note, if the setup below doesn't match the setup in the Compliance Test Specification (CTS), then users must use the setup in the CTS and ignore the setups in this document.

6.5.1 UUT is a Host - Chapter 11:

AN_HOST_DFP1—PCIE_01

- Vendor provides the host system for the UUT with:
 - USB4 CV installed
 - The relevant PCIe-related capabilities (PTM, ASPM, etc.) enabled in the BIOS/OS as required for testing
- A KG USB4 Device with a known PCIe endpoint is connected downstream.
- A USB4 Analyzer is connected between the UUT and KG USB4 Device.

Note: Some tests require the use of an active cable on the USB4 Link

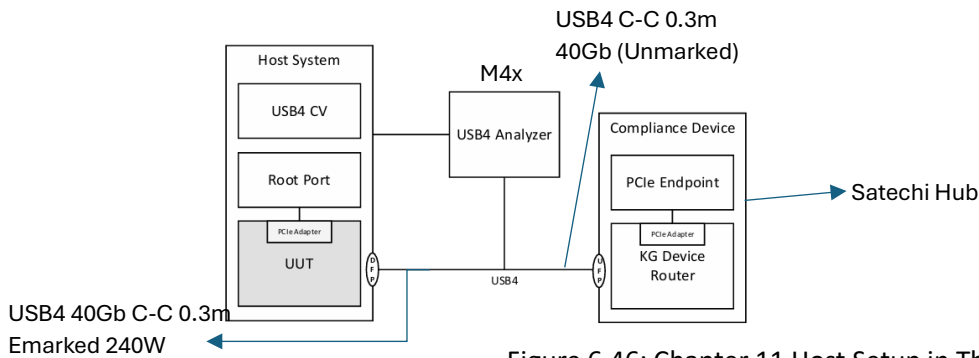


Figure 6.46: Chapter 11 Host Setup in The CTS

6.5.2 UUT is a Device - Chapter 11:

AN_DEV_UFP1—PCI_E_01

- Vendor provides device test fixture for the UUT with a x8 PCIe slot.
- For an embedded PCIe endpoint, the vendor provides a Windows driver/utility capable of running controlled Tx/Rx traffic on the device.
- A KG USB4 Host is connected to the UUT.
 - Host supports PCIe tunneling
 - Host supports ASPM L1 on all USB4 Ports
- A USB4 Analyzer is connected between the KG USB4 Host and UUT.

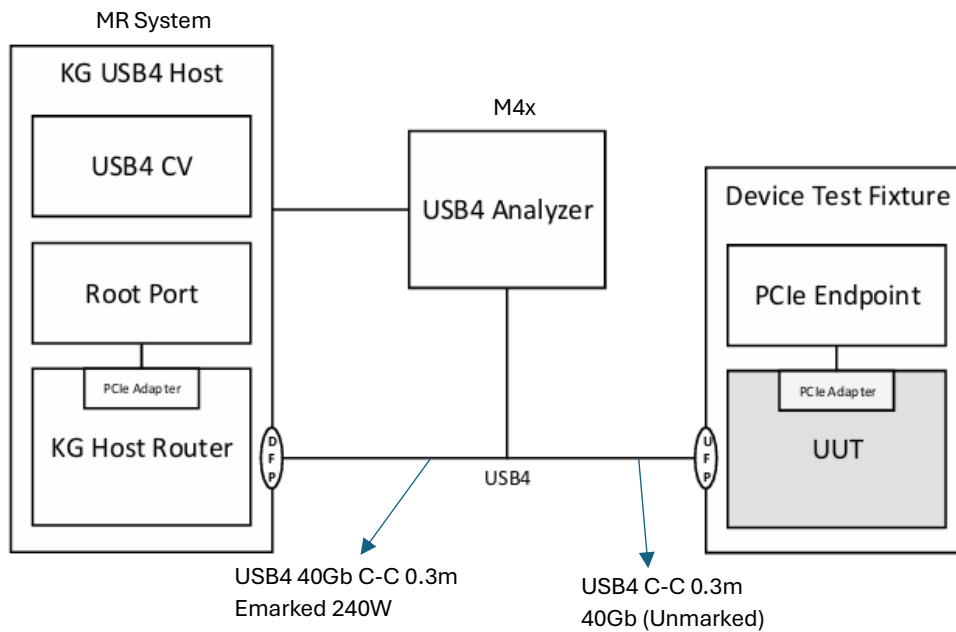


Figure 6.47: Chapter 11 Device Setup in The CTS

6.5.3 UUT is a Hub - Chapter 11:

When a hub is the UUT then both UFP and DFP are supposed to be tested:

UUT is a Hub (testing the UFP)

AN_HUB_UFP1—PCIE_01

- If the UUT includes an embedded PCIe endpoint, the vendor provides a Windows driver/utility capable of running controlled Tx/Rx traffic on the device.
- A KG USB4 Host is connected to the UUT.
 - Host supports PCIe tunneling
 - Host supports ASPM L1 on all USB4 Ports
- A USB4 Analyzer is connected between the KG USB4 Host and UUT.
- A Compliance Device with a known PCIe endpoint is connected to the DFP of the UUT.
 - The PCIe endpoint has the means to generate a PCIe wake under SW control

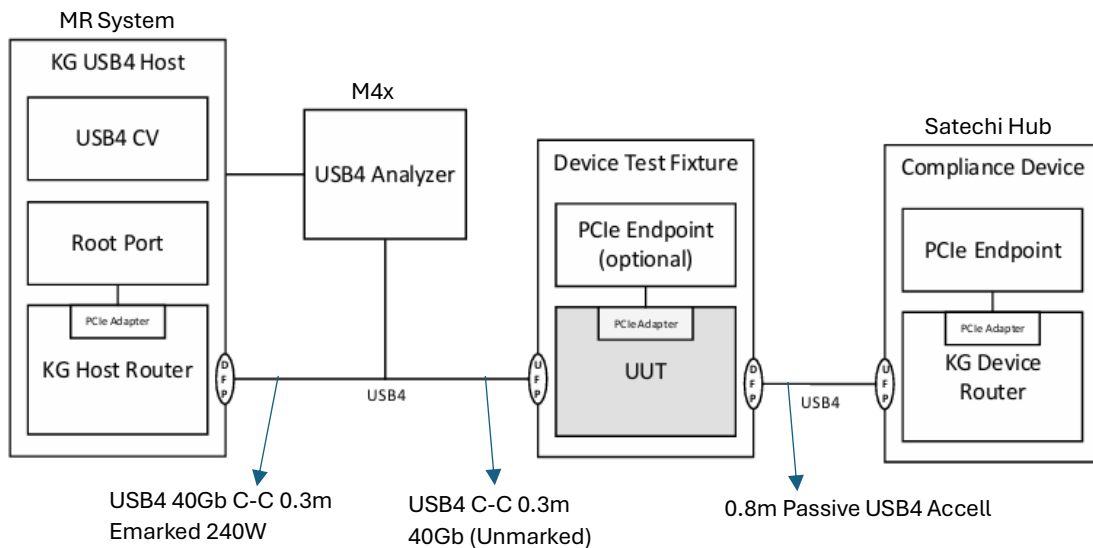


Figure 6.48: Chapter 11 Hub UFP Setup in The CTS

UUT is a Hub (testing the DFP)

AN_HUB_DFP1—PCIE_02

- If the UUT includes an embedded PCIe endpoint, the vendor provides a Windows driver/utility capable of running controlled Tx/Rx traffic on the device.
- A known host system running Windows OS with a KG USB4 Host will connect to the UUT.
 - Host supports PCIe tunneling
 - Host supports ASPM L1 on all USB4 Ports
- A Compliance Device with a known PCIe endpoint is connected to the DFP of the UUT.
 - The PCIe endpoint has the means to generate a PCIe wake under SW control

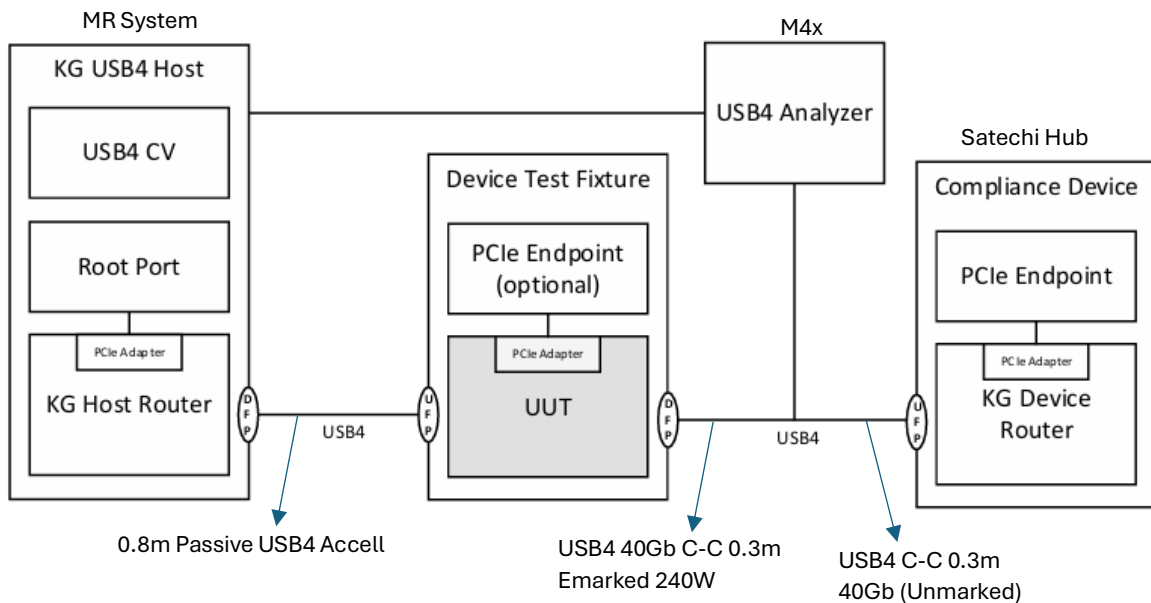


Figure 6.49: Chapter 11 Hub DFP Setup in The CTS

USB4 CV will ask the user to enter **PCIe Entry Mapping Segment** once the user starts running the test. If Maple Ridge system is being used, then user expected to enter 0. If the other host is being used as UUT then vendors are expected to provide that information.

Note, before running chapter 11, user is expected to run the following test first, “**Enabled PCIe Tunneling in USB4 CV**” test. Once the user runs this test, a reboot is required before running chapter 11. Then once chapter 11 is finished, the user must run “Disable PCIe Tunneling in USB4CV” test and reboot the system again. Please refer to the release notes for more information on this.

6.6 Chapter 12

Please refer to the CTS for chapter 12 (USB4 Host Interface) for the setup for this test

Note, if the setup below doesn't match the setup in the Compliance Test Specification (CTS), then users must use the setup in the CTS and ignore the setups in this document.

6.6.1 UUT is a Host - Chapter 12 AN

AN_HOST_DFP1

Description:

- The UFP of the Compliance Device is connected to the DFP of the UUT
- The Analyzer is connected between UUT and Compliance Device Router
- The Analyzer is invisible to UUT and Compliance Device

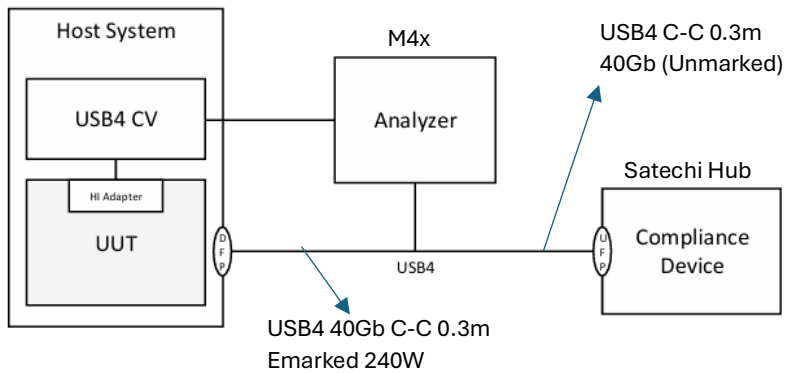


Figure 6.50: Chapter 12 AN Host Setup in The CTS

6.6.2 UUT is a Host - Chapter 12 EX:

EX_HOST_DFP1

Description:

- The Exerciser is connected to the DFP of the UUT

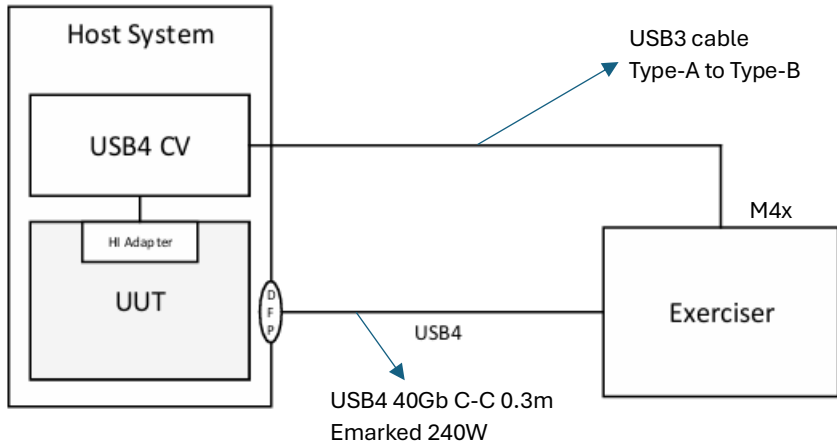


Figure 6.51: Chapter 12 EX Host Setup in The CTS

6.7 Chapter 13

Please refer to the CTS for Chapter 13 setups and make sure the below setups match the setups in the CTS

Note, if the setup below doesn't match the setup in the Compliance Test Specification (CTS), then users must use the setup in the CTS and ignore the setups in this document.

1. Logical Layer CTS for the following tests

- USB4 Host (AN_HOST_DFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Logical Layer] Tests
- USB4 Device (AN_DEV_UFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Logical Layer] Tests
- USB4 Hub (AN_HUB_UFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Logical Layer] Tests
- USB4 Hub (AN_HUB_DFP1-TBT3_03) [Chapter 13 - TBT3 Compatibility – Logical Layer] Tests

2. Protocol CTS for the following tests

- USB4 Host (AN_HOST_DFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Protocol] Tests
- USB4 Device (AN_DEV_UFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Protocol] Tests
- USB4 Hub (AN_HUB_UFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Protocol] Tests

6.7.1 UUT is a Host – Chapter 13:

USB4 Host (AN_HOST_DFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Logical Layer] Tests

1. The below setup is for TBT3-Compatibility Mode Test - Logical Layer

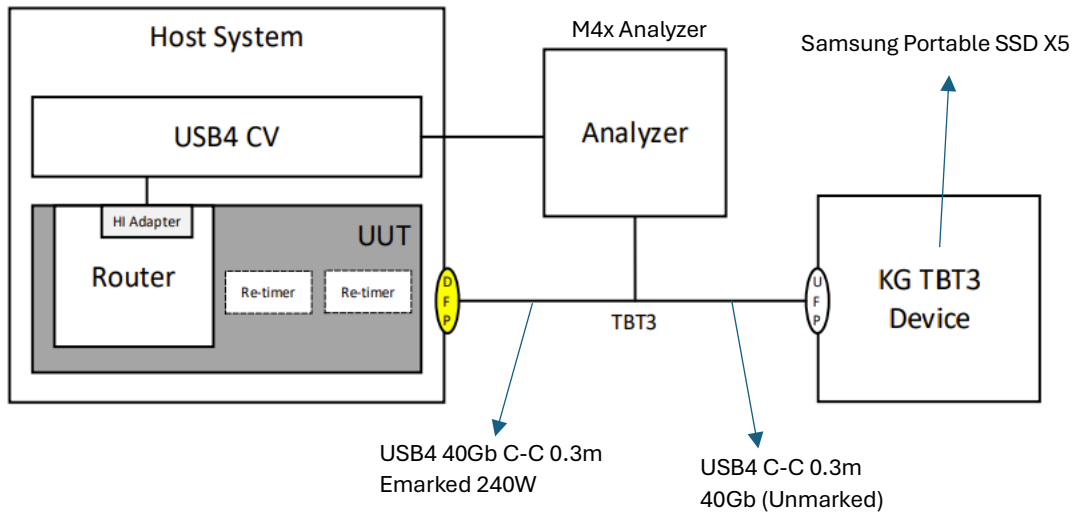


Figure 6.52: USB4 Host (AN_HOST_DFP1-TBT3_01) Logical Layer

2. The below setup is for Test that Require Active Cables

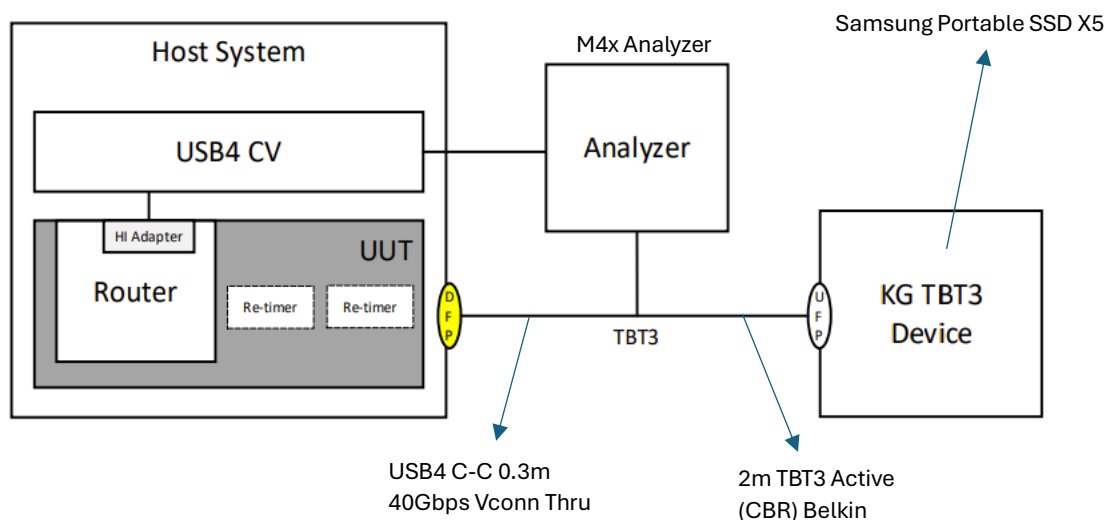


Figure 6.53: Active Cables: USB4 Host (AN_HOST_DFP1-TBT3_01) Logical Layer

USB4 Host (AN_HOST_DFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Protocol] Tests

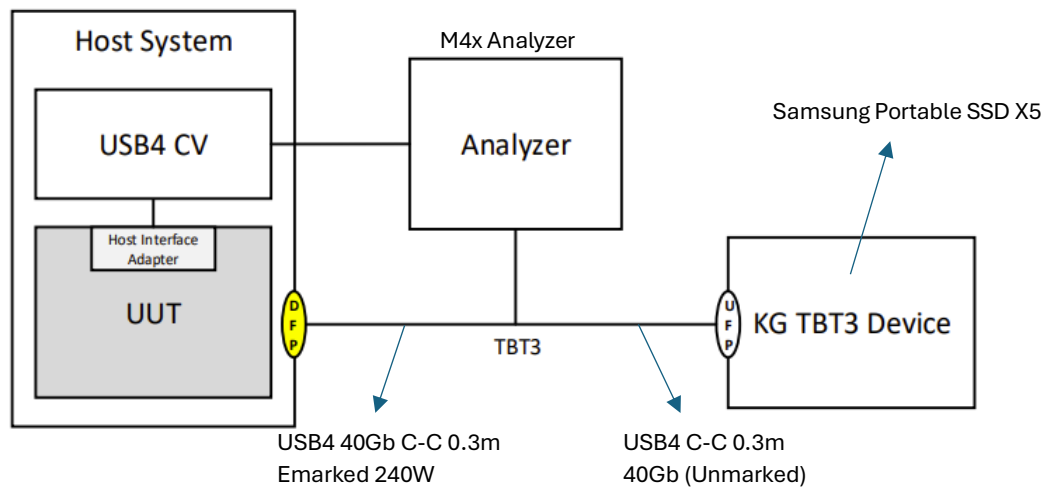


Figure 6.54: USB4 Host (AN_HOST_DFP1-TBT3_01) Protocol

6.7.2 UUT is a Device – Chapter 13:

USB4 Device (AN_DEV_UFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Logical Layer] Tests

1. The below setup is for TBT3-Compatibility Mode Test - Logical Layer

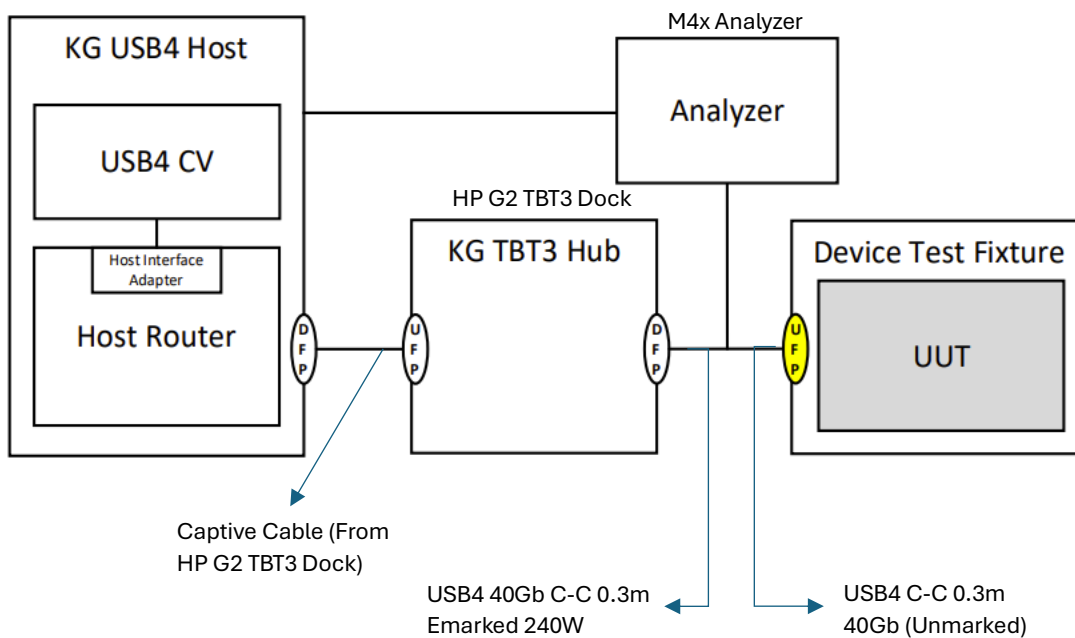


Figure 6.55: USB4 Device (AN_DEV_UFP1-TBT3_01) Logical Layer

2. The setup below is for Test that Require Active Cables

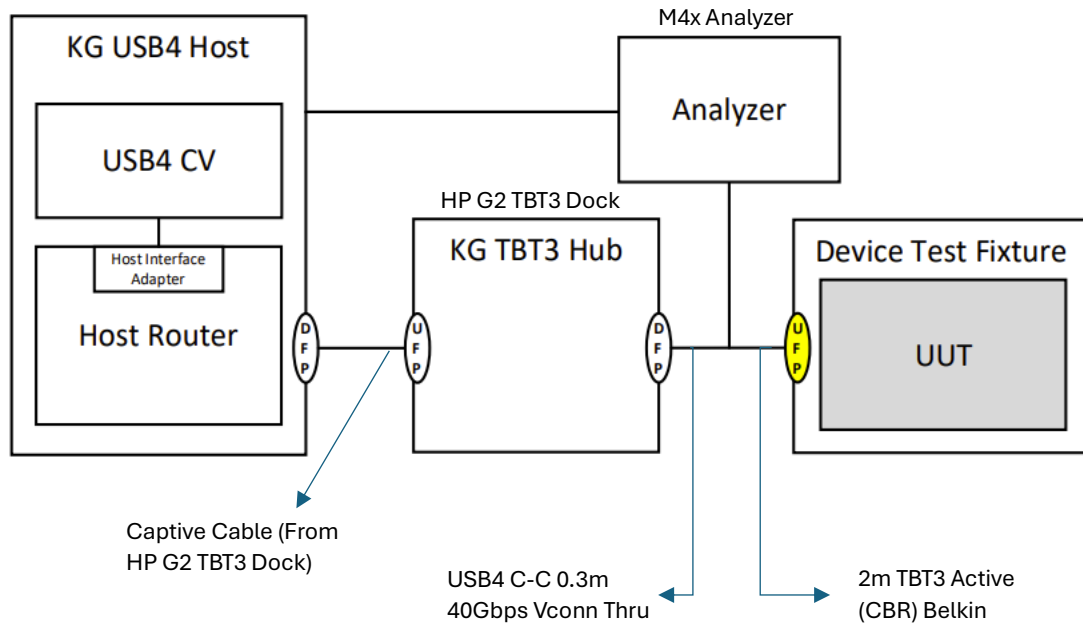


Figure 6.56: Active Cables USB4 Device (AN_DEV_UFP1-TBT3_01) Logical Layer

USB4 Device (AN_DEV_UFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Protocol] Tests

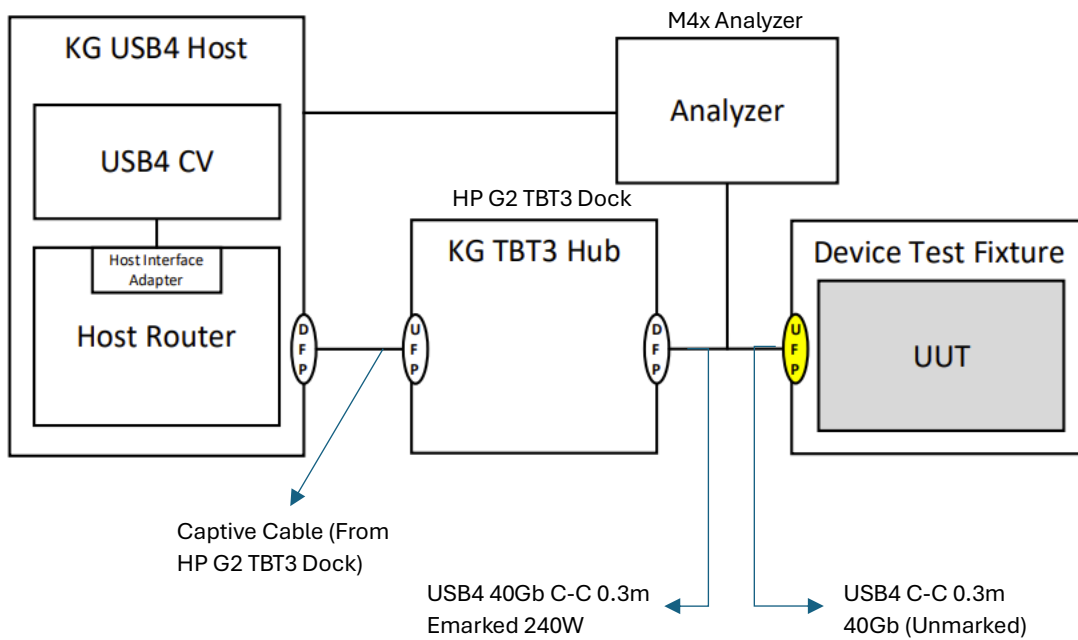


Figure 6.57: USB4 Device (AN_DEV_UFP1-TBT3_01) Protocol

6.7.3 UUT is a Hub – Chapter 13:

USB4 Hub (AN_HUB_UFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Logical Layer] Tests

1. The below setup is for TBT3-Compatibility Mode Test - Logical Layer

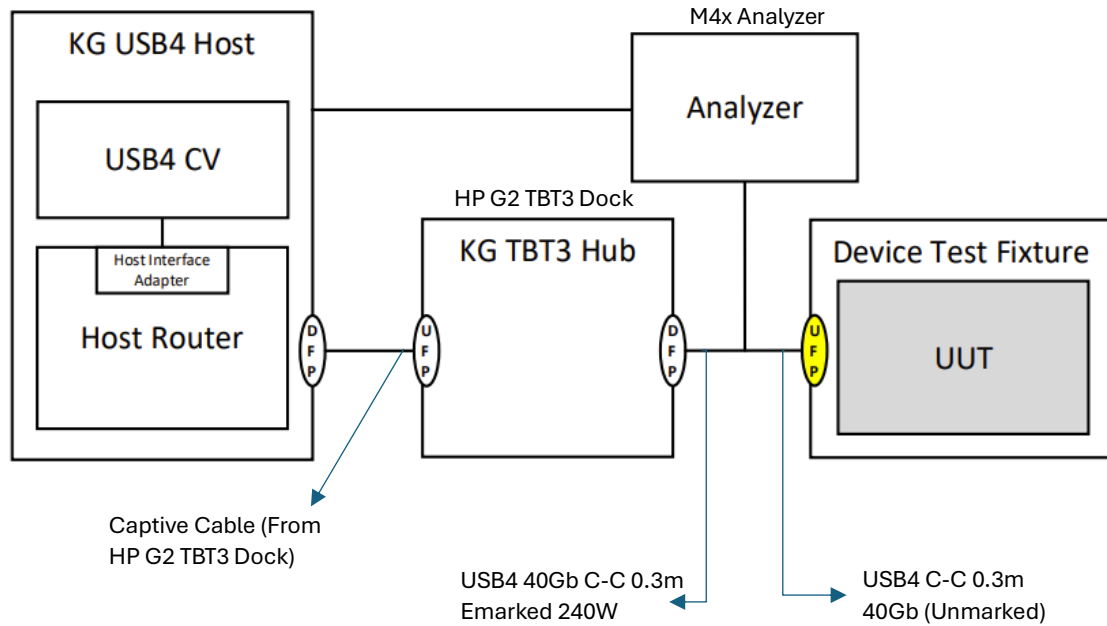


Figure 6.58: Logical Layer USB4 Hub (AN_HUB_UFP1-TBT3_01)

2. The setup below is for Test that Require Active Cables

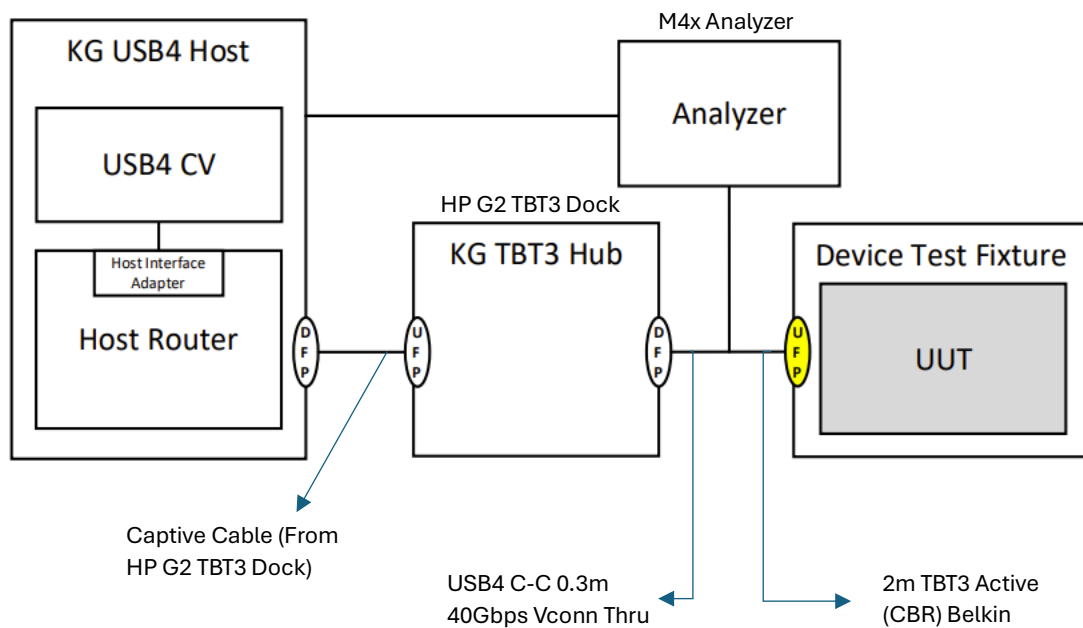


Figure 6.59: Active Cable USB4 Hub (AN_HUB_UFP1-TBT3_01)

USB4 Hub (AN_HUB_UFP1-TBT3_01) [Chapter 13 - TBT3 Compatibility - Protocol] Tests

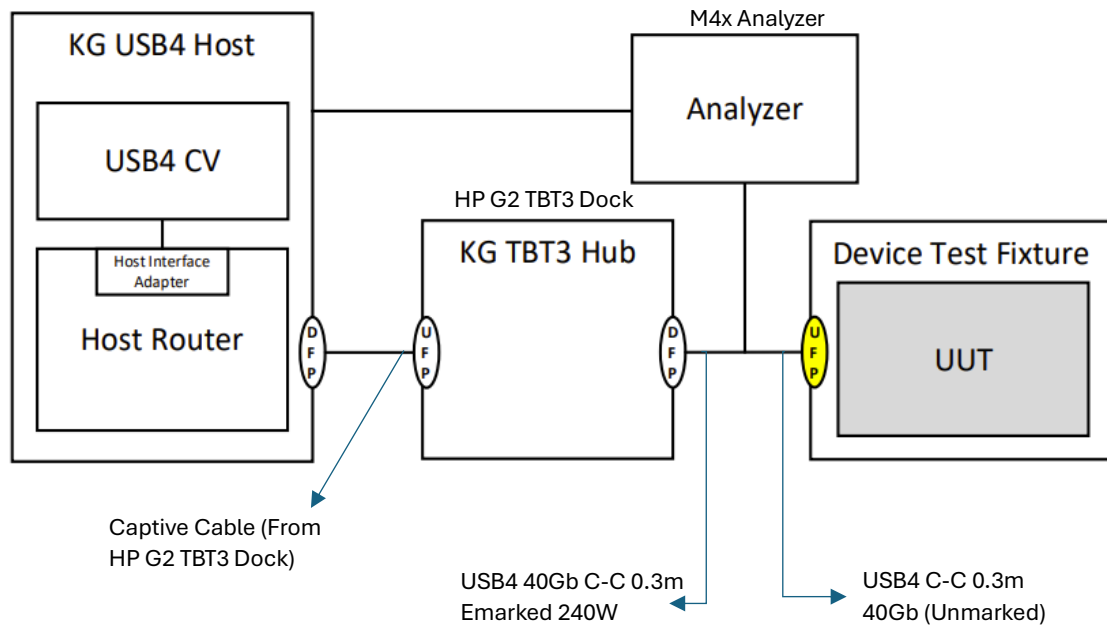


Figure 6.60: USB4 Hub (AN_HUB_UFP1-TBT3_01) Protocol

USB4 Hub (AN_HUB_DFP1-TBT3_03) [Chapter 13 - TBT3 Compatibility – Logical Layer] Tests

1. The below setup is for TBT3-Compatibility Mode Test - Logical Layer

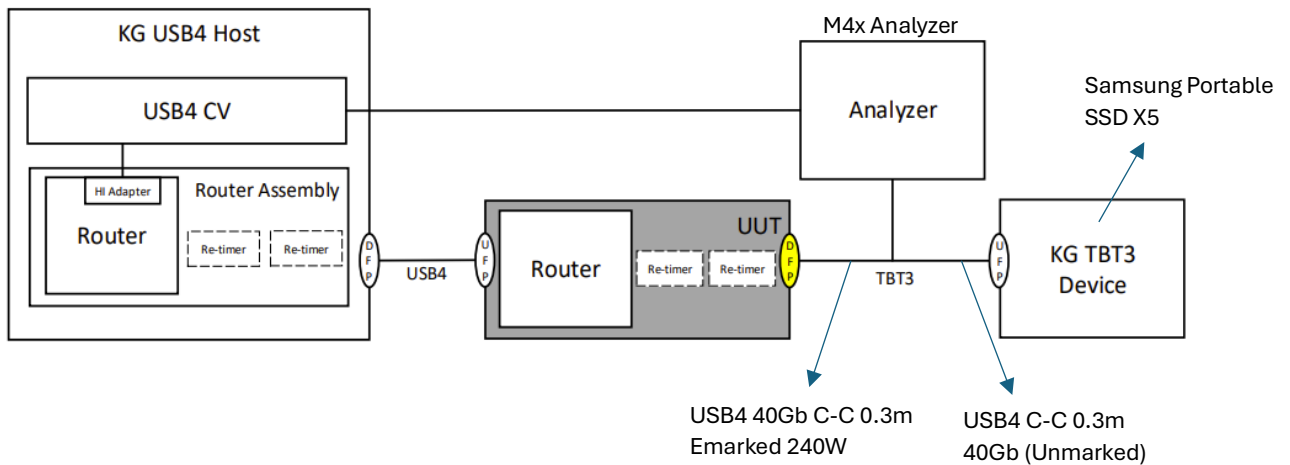


Figure 6.61: Logical Layer USB4 Hub (AN_HUB_DFP1-TBT3_03)

2. The setup below is for Test that Require Active Cables

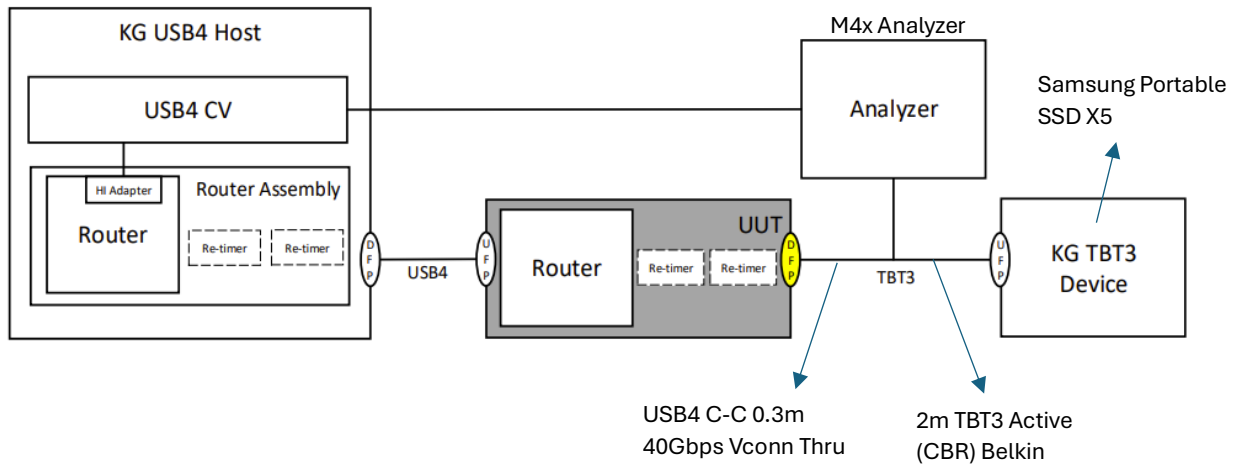


Figure 6.62: Active Cables USB4 Hub (AN_HUB_DFP1-TBT3_03)

7. Linux USB4CV Chapters Running Requirements:

7.1 Chapter 4

cv-gui command to launch USB4CV:

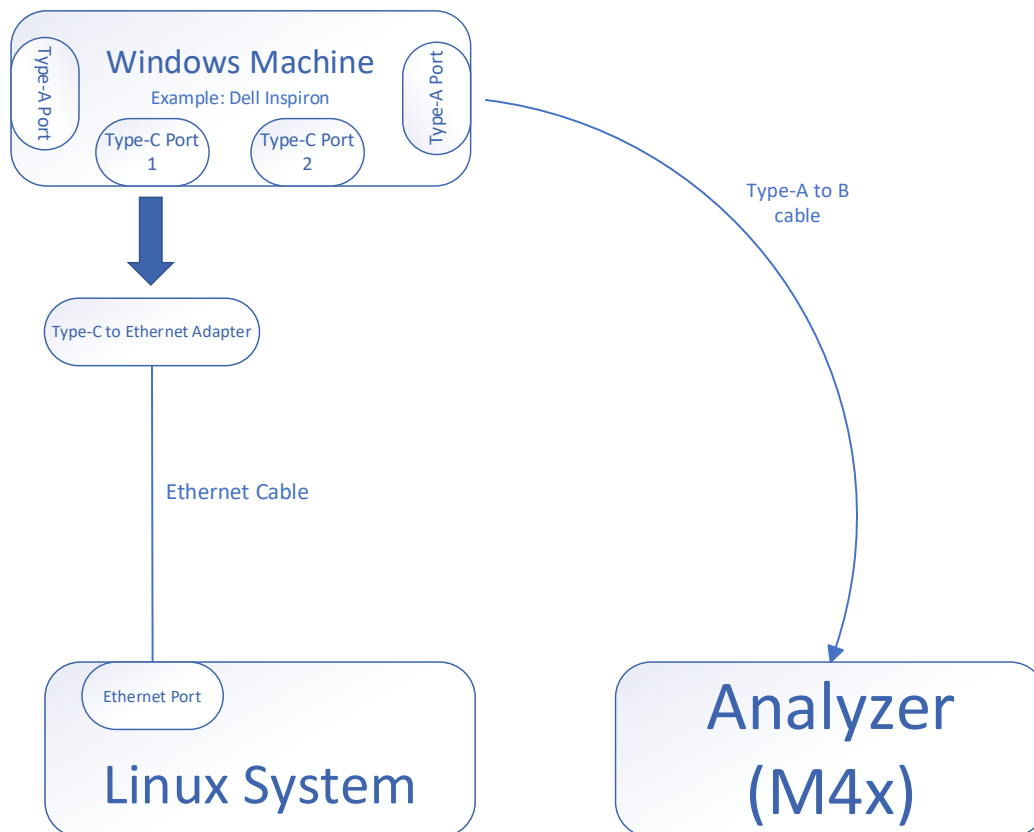
```
./cv-gui -app USB4CV
```

cv-cli command:

```
./cv-cli -app USB4CV -testsuite <name of the test> -vif <VIF file name>
```

Setup

Users must follow the CTS to connect the compliance device and M4x to the PUT (Port Under Test)



7.2 Chapter 5

cv-gui command to launch USB4CV:

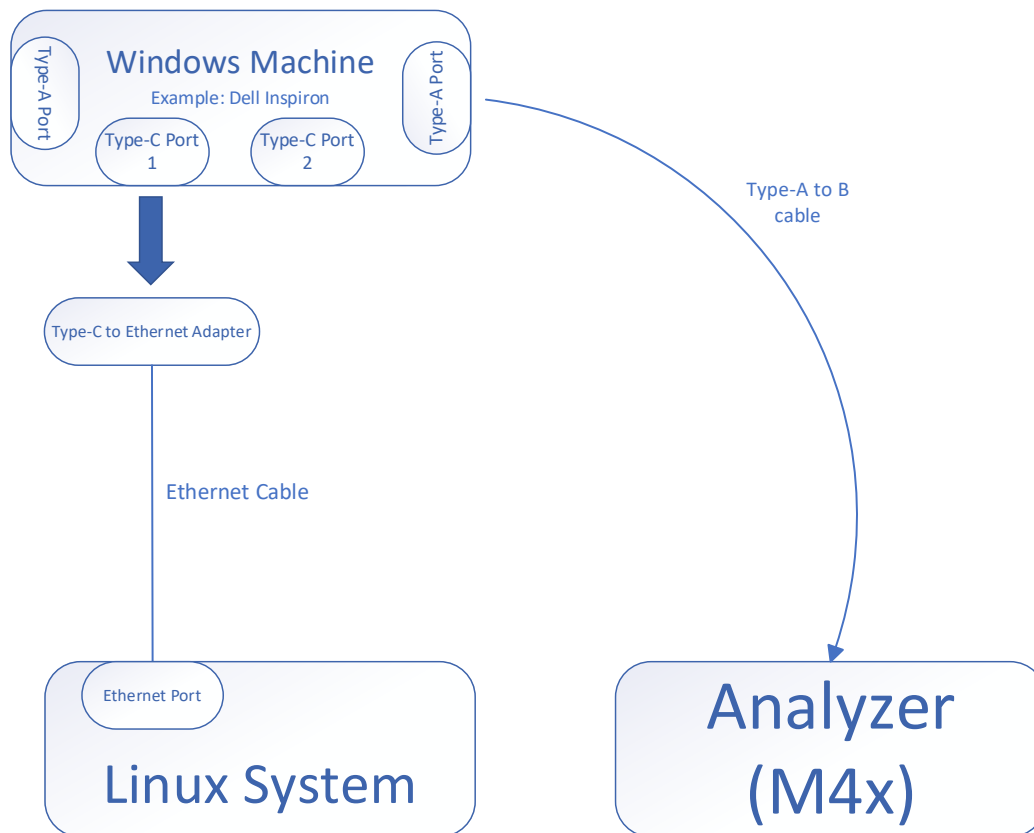
```
./cv-gui -app USB4CV
```

cv-cli command:

```
./cv-cli -app USB4CV -testsuite <name of the test>
```

Setup

Users must follow the CTS to connect the compliance device and M4x to the PUT (Port Under Test)



7.3 Chapter 6

cv-gui command to launch USB4CV:

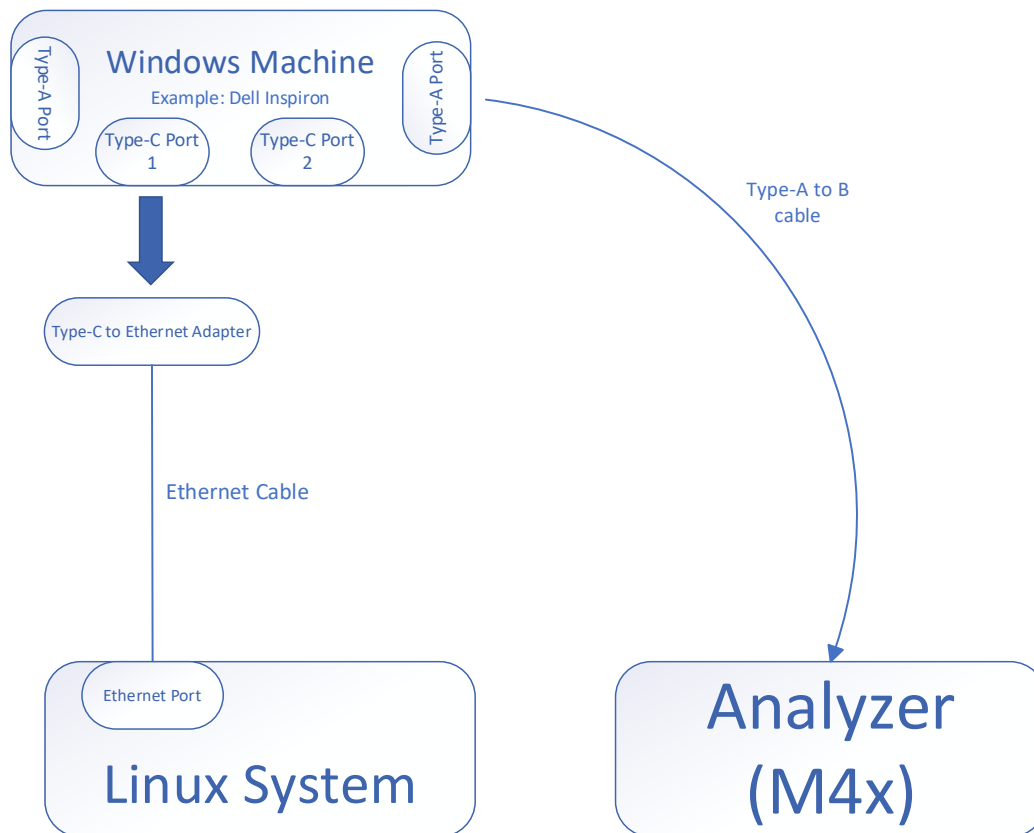
```
./cv-gui -app USB4CV
```

cv-cli command:

```
./cv-cli -app USB4CV -testsuite <name of the test> -vif "VIF file name"
```

Setup

Users must follow the CTS to connect the compliance device and M4x to the PUT (Port Under Test)



7.4 Chapter 7

cv-gui command to launch USB4CV:

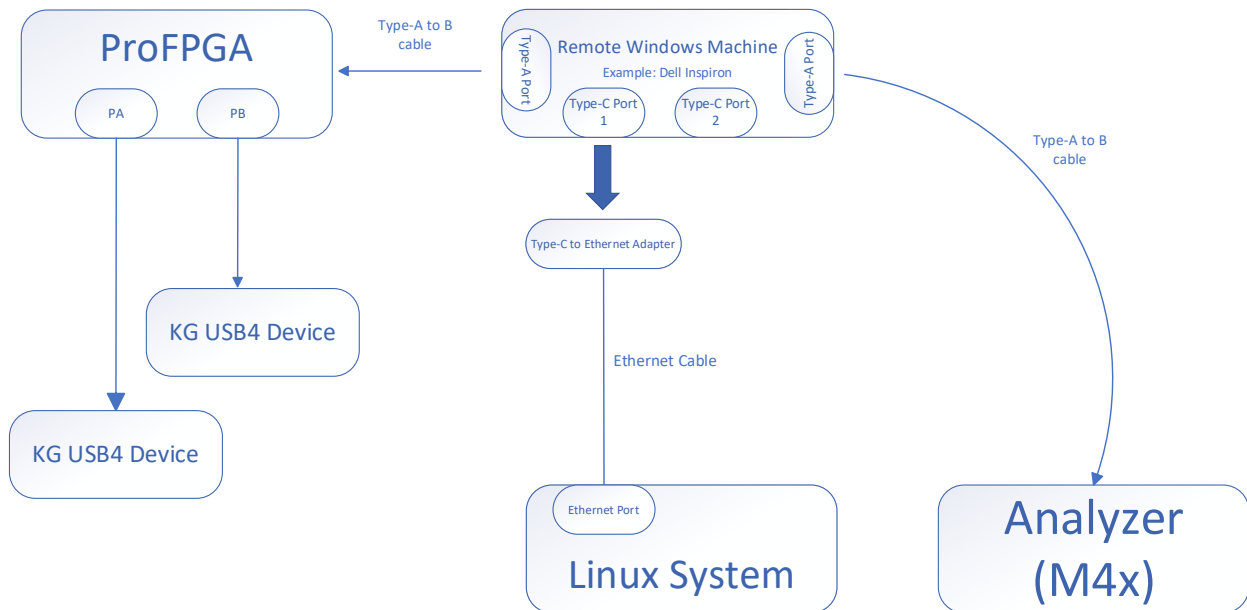
```
./cv-gui -app USB4CV
```

cv-cli command:

```
./cv-cli -app USB4CV -testsuite <name of the test>
```

Setup

Users must follow the CTS to connect the KG USB4 Device and M4x to the PUT (Port Under Test). Note, when the device is the UUT, the ProFPGA is not required. Check [Appendix](#) on how the test runs



7.5 Chapter 8

sudo -E must be used when running chapter 8 test

cv-gui command to launch USB4CV:

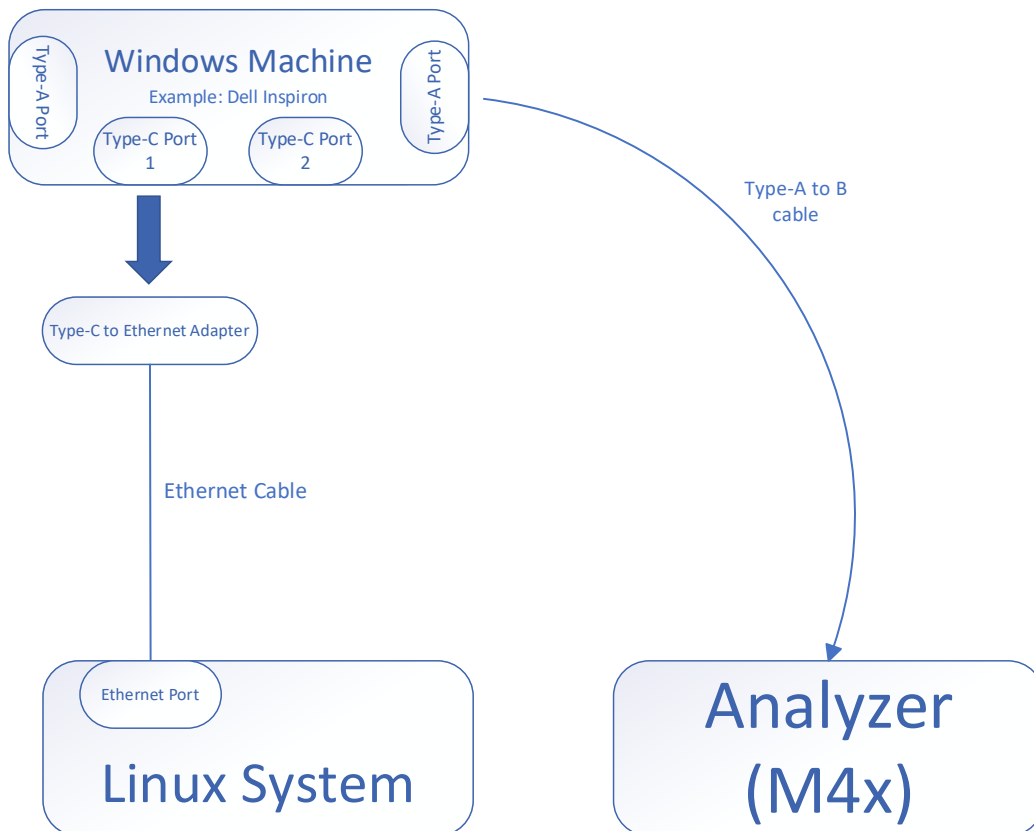
```
sudo -E ./cv-gui -app USB4CV
```

cv-cli command:

```
sudo -E ./cv-cli -app USB4CV -testsuite <name of the test> -vif <VIF file name>
```

Setup

Users must follow the CTS to connect the compliance device and M4x to the PUT (Port Under Test)



7.6 Chapter 10

GuiCV command to launch USB4CV:

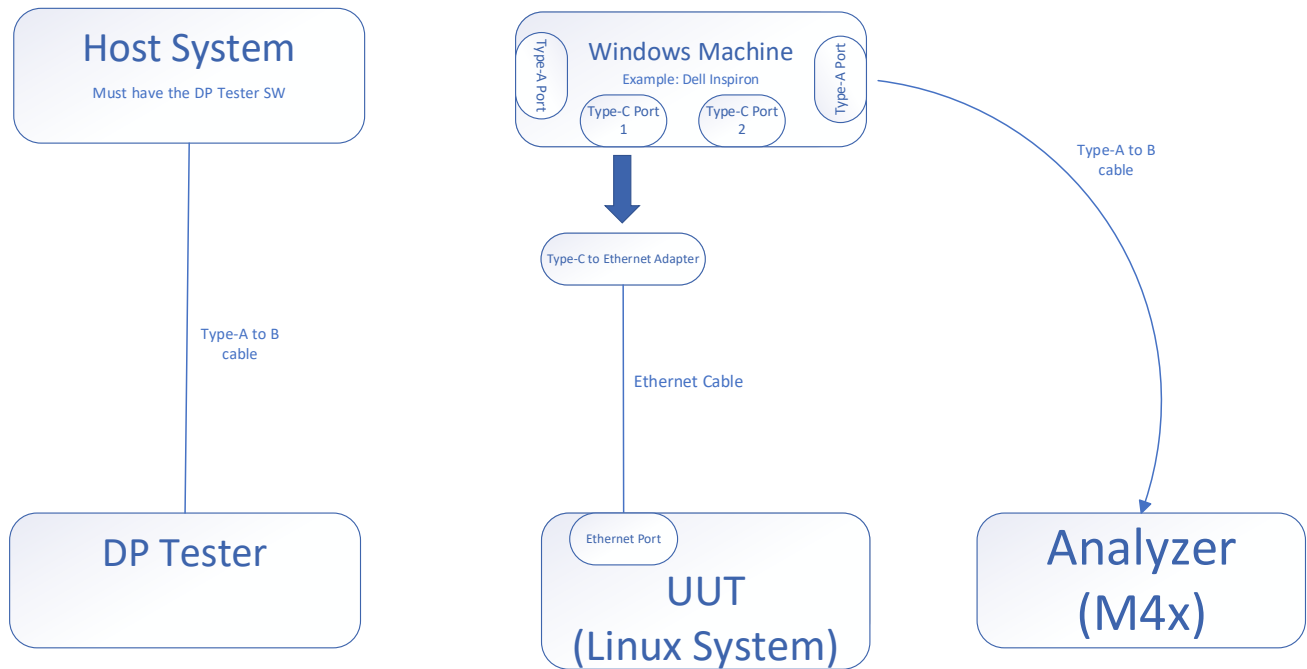
```
./cv-gui -app USB4CV
```

CmdLineCV command:

```
./cv-cli -app USB4CV -testsuite <name of the test>
```

Setup

Users must follow the CTS to connect the compliance device and M4x to the PUT



7.7 Chapter 11

sudo -E must be used when running chapter 11 test

cv-gui command to launch USB4CV:

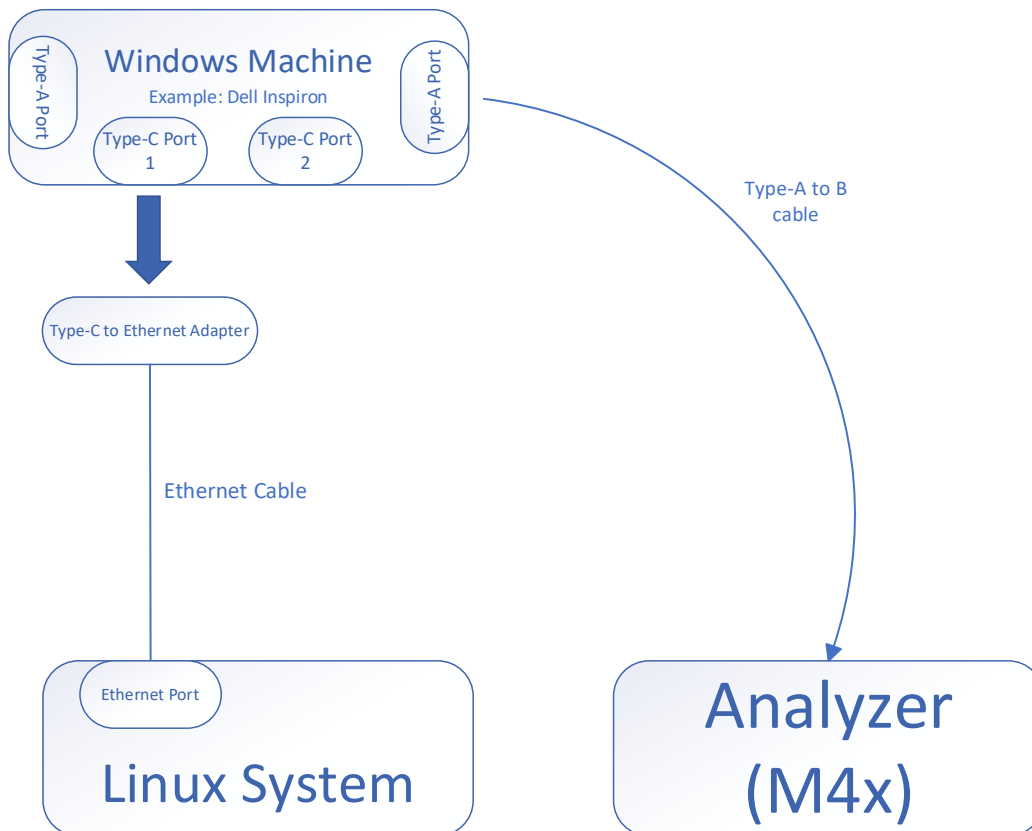
```
sudo -E ./cv-gui -app USB4CV
```

cv-cli command:

```
sudo -E ./cv-cli -app USB4CV -testsuite <name of the test> -vif <VIF file name>
```

Setup

Users must follow the CTS to connect the compliance device and M4x to the PUT (Port Under Test)



7.8 Chapter 12

cv-gui command to launch USB4CV:

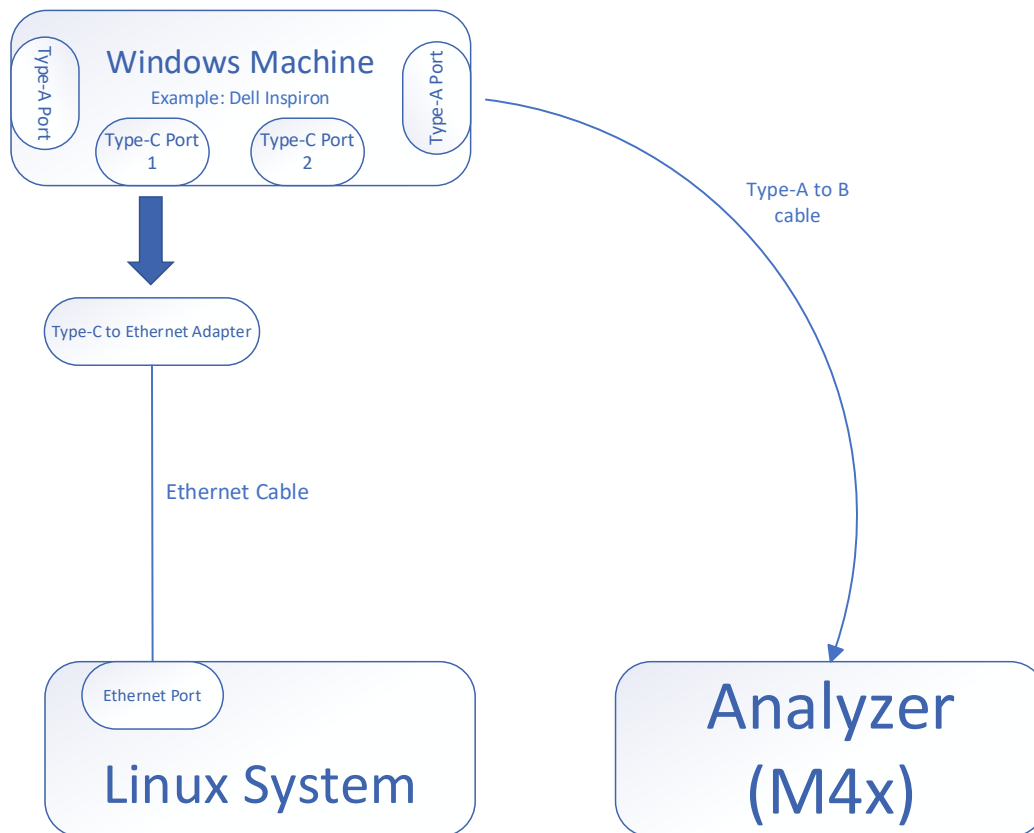
```
./cv-gui -app USB4CV
```

cv-cli command:

```
./cv-cli -app USB4CV -testsuite <name of the test>
```

setup

The setup below is only for windows/M4x/Linux machines connections. **Users must follow the CTS to connect the compliance device and M4x to the PUT (Port Under Test)**



7.9 Chapter 13

cv-gui command to launch USB4CV:

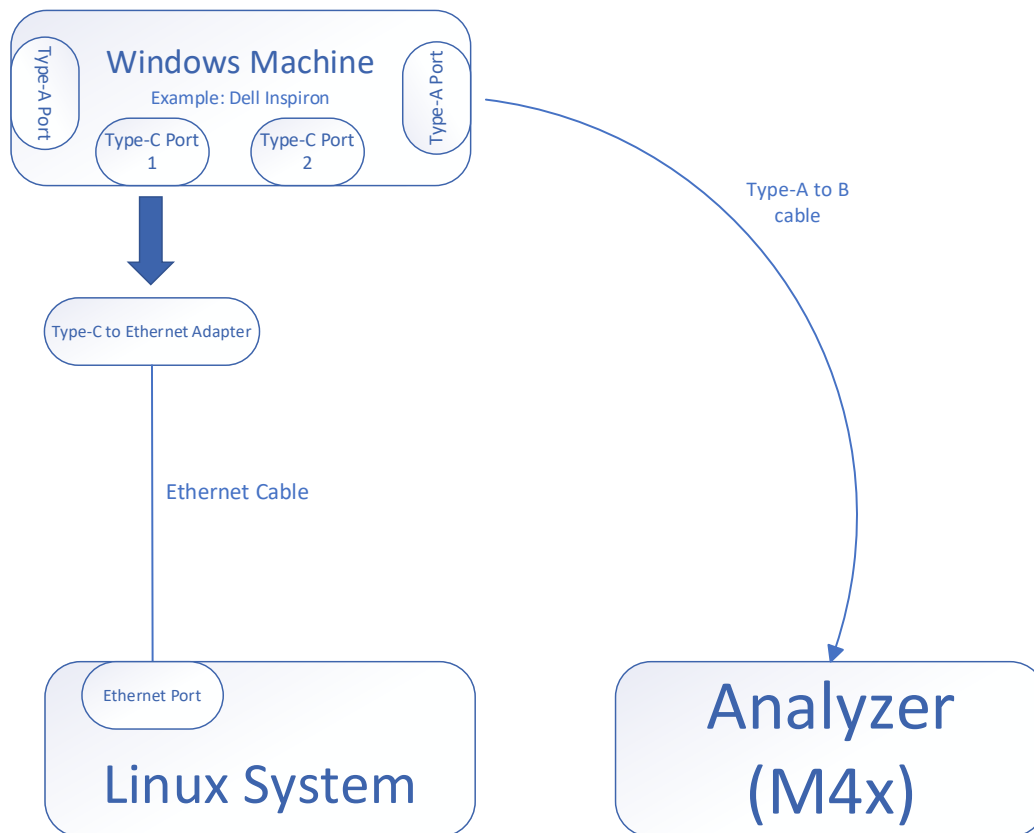
```
./cv-gui -app USB4CV
```

cv-cli command:

```
./cv-cli -app USB4CV -testsuite <name of the test>
```

setup

The setup below is only for windows/M4x/Linux machines connections. **Users must follow the CTS to connect the compliance device and M4x to the PUT (Port Under Test)**



8. Appendix

8.1 Time Measurement Equipment – Setup Procedure

Copyright © 2021 USB Implementers Forum, Inc.

All rights reserved

THIS PROCEDURE IS PROVIDED TO YOU “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE. THE AUTHORS OF THIS SPECIFICATION DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO USE OR IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. THE PROVISION OF THIS SPECIFICATION TO YOU DOES NOT PROVIDE YOU WITH ANY LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS.

USB Type-C™, USB-C™, USB 2.0 Type-C™, and USB4™ are trademarks of the Universal Serial Bus Implementers Forum (USB-IF). Thunderbolt™ is a trademark of Intel Corporation. All product names are trademarks, registered trademarks, or service marks of their respective owners.

8.1.1 Introduction

Time Measurement Equipment is an essential part of Time Synchronization Compliance testing. It is based on the Serial Time Link Protocol (STLP) that transmits current Grand Master time every 16 μ sec. STLP is parsed and analyzed by a *Time Measurement Equipment*, which is FPGA based hardware.

There are two usages for such equipment:

8.1.2 Standalone Measurement

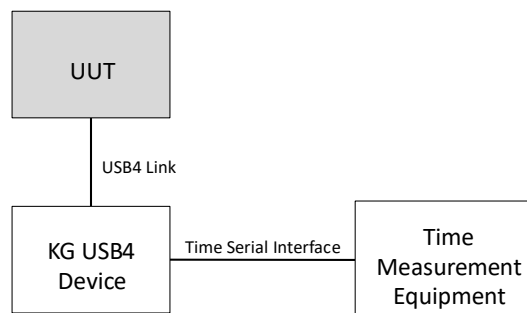


Figure 8.1: Standalone setup

8.1.3 Comparative Time Measurement

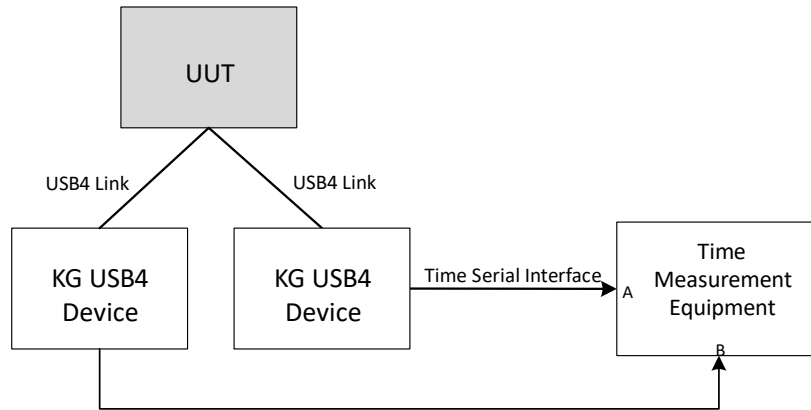


Figure 8.2: Pair Setup

8.1.4. Required Hardware and Software

Hardware	Vendor	Link
Quad Motherboard	proFPGA	proFPGA quad
Virtex 7 Module XC7V2000T	proFPGA	proFPGA XC7V2000T
Case (optional)	proFPGA	proFPGA duo quad Casing
Goodway Gatkex Board with TMU CLK Out Board (DBD1210L1)	Good Way Technology Co., Ltd.	https://www.goodway.com.tw/

Software	Vendor	Link
proFPGA-2020C-install.exe	proFPGA	<i>*Contact proFPGA for this software</i>
Files to use with USB4CV	USB-IF	https://usb.org/compliancetools#anchor_usb4tools

Please contact proFPGA about getting the necessary items listed above.



<https://www.proFPGA.com/>
email: profpga@prodesign-

europe.com

8.1.4.1. Quad Motherboard



Figure 8.3: Quad Motherboard

8.1.4.2. Duo & Quad Casing (optional)



Figure 8.4: Optional case for FPGA

8.1.4.3. Virtex® 7 module XC7V2000T



Figure 8.5: Virtex® 7 module

8.1.4.4. Goodway Gatkex Board with TMU CLK Out as USB4 KG Device:

Contact Info: raymond_chan@goodway.com.tw & vivian_liao@goodway.com.tw

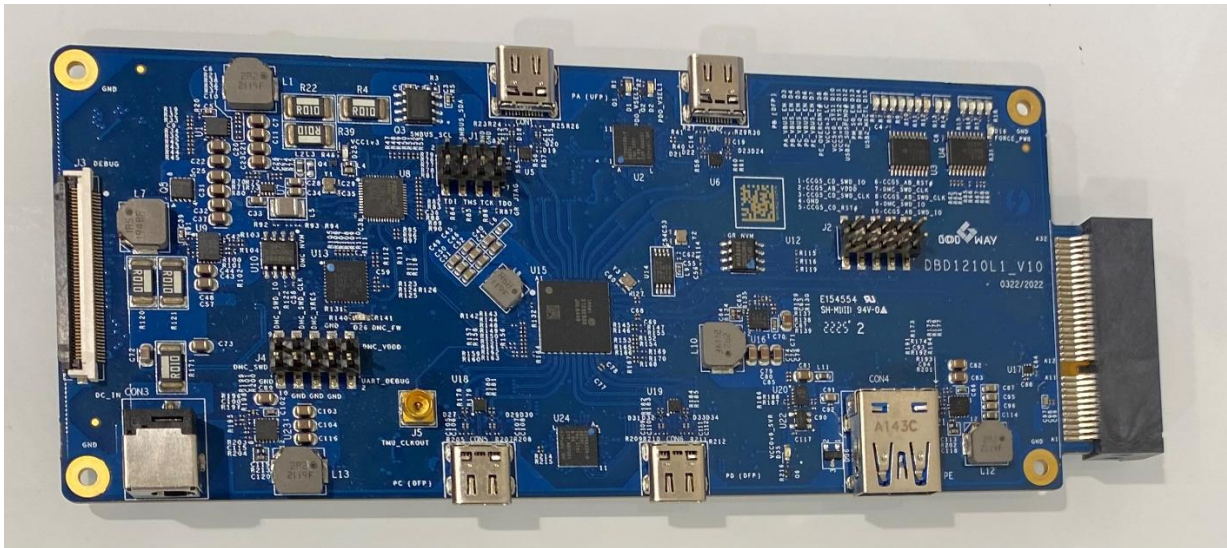


Figure 8.6: Goodway Gatkex Board with TMU CLK Out

8.1.4.5. STLP Cable

STLP cable connects Goodway Gatkex Board TMU_CLK_OUT output and carries STLP protocol to FPGA.

The cable has 3 ends as shown on the following figure.

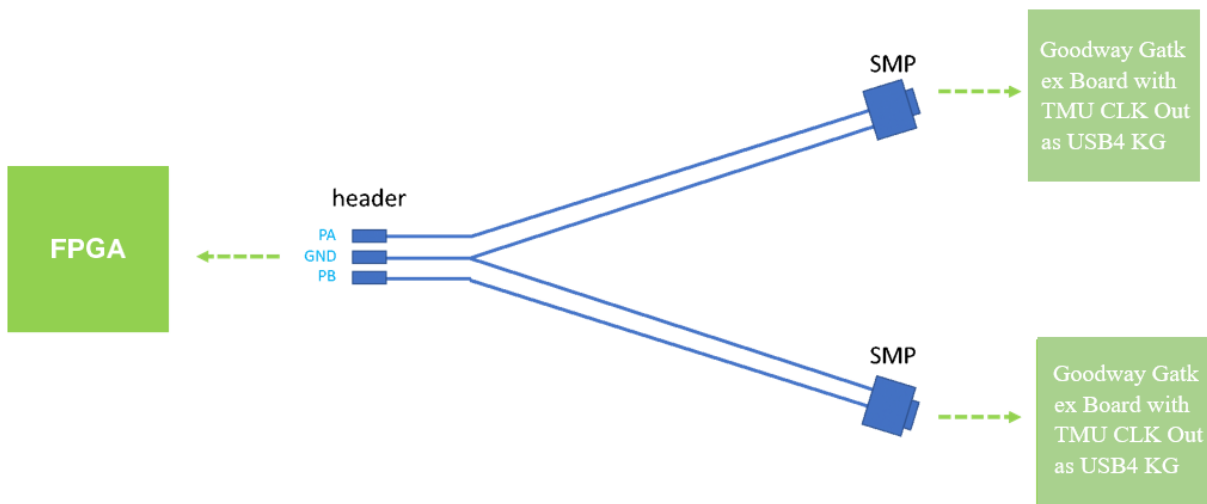


Figure 8.7: Example

8.1.4.5.1. *STLP Cable construction procedure for the Goodway Gatkex Board with TMU CLK Out*
These instructions can be used to make custom cables for testing products that do not use SMP connectors for

TMU_CLK_OUT.

The following materials will be necessary:

- 2x RF cables each at least 48inches long with push-on SMP connectors.
Note: Do not use semi-rigid type. Choose braided shielding (eg: PE3C3584/PH180-48)
- 1x 0.1inch (2.54mm) Crimp connector housing 1x6-pin
- Wire stripper
- Heat shrink tubes
- 3x Heat shrink tubes with solder ring
- Heat gun
- 3x 2inch pre-crimped wires
- Hot glue gun

1. Strip at least 1" of the outer insulation.



2. Carefully undo the braided shielding. It is recommended to use a small and fine tool to poke into the braids and pull towards the open end of the cable. Start from the end of the cable and slowly work down to slowly undo the braiding.



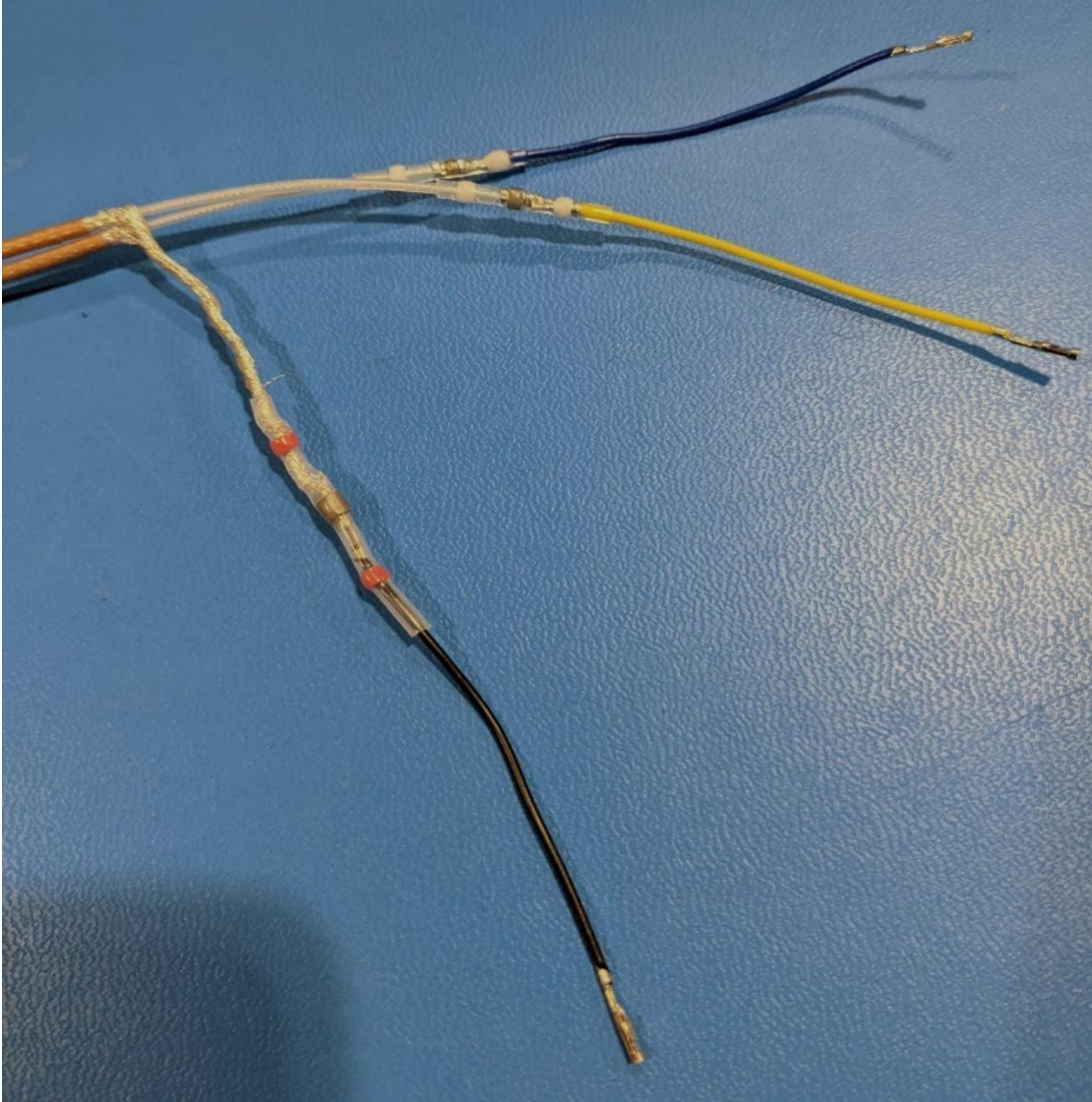
3. Strip about $\frac{1}{4}$ inch of the dielectric insulation to expose the center conductor.



4. Take a pre-crimped wire and feed into a heat shrink tube with solder ring. Take the exposed center conductor and insert into one of the crimps. Then pull the heat shrink tube over this connection and line the solder ring up with the metal-to-metal contact portion.



5. Use heat gun to shrink the tube and pay extra attention to melt the solder ring. Solder ring should visibly deform or fill in gaps of the crimp contact.
6. Repeat steps 1-5 for the second cable.
7. Combine both Ground braids by twisting them together. Repeat steps 4 and 5 on this combined braid.

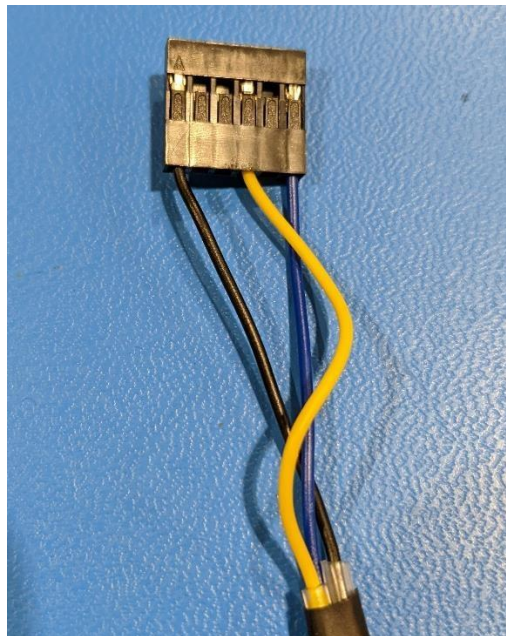


8. Use heat shrink (no solder ring) to cover excess exposed braiding.

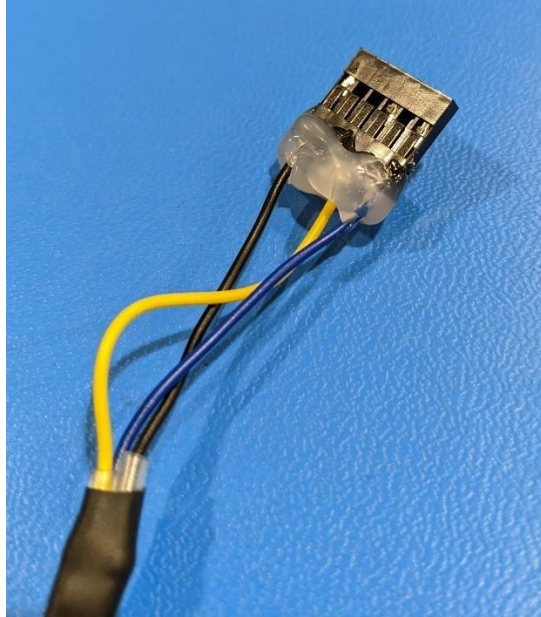


9. Insert pins into the housing and give it a light tug test to ensure that the pin is secured in place:

- GND braids to pin 1 (GND)
- Choice of cable to pin 4 (PB)
- Remaining cable to pin 6 (PA)



10. Apply hot glue to the bottom of housing. Make sure to fill in spaces between the three cables and overlap onto the plastic housing to prevent the cables from bending when attaching this cable to FPGA.



8.1.5. Connecting the system

8.1.5.1. Installing the module

Virtex® 7 module shall be installed on connector TA1 of Quad Motherboard

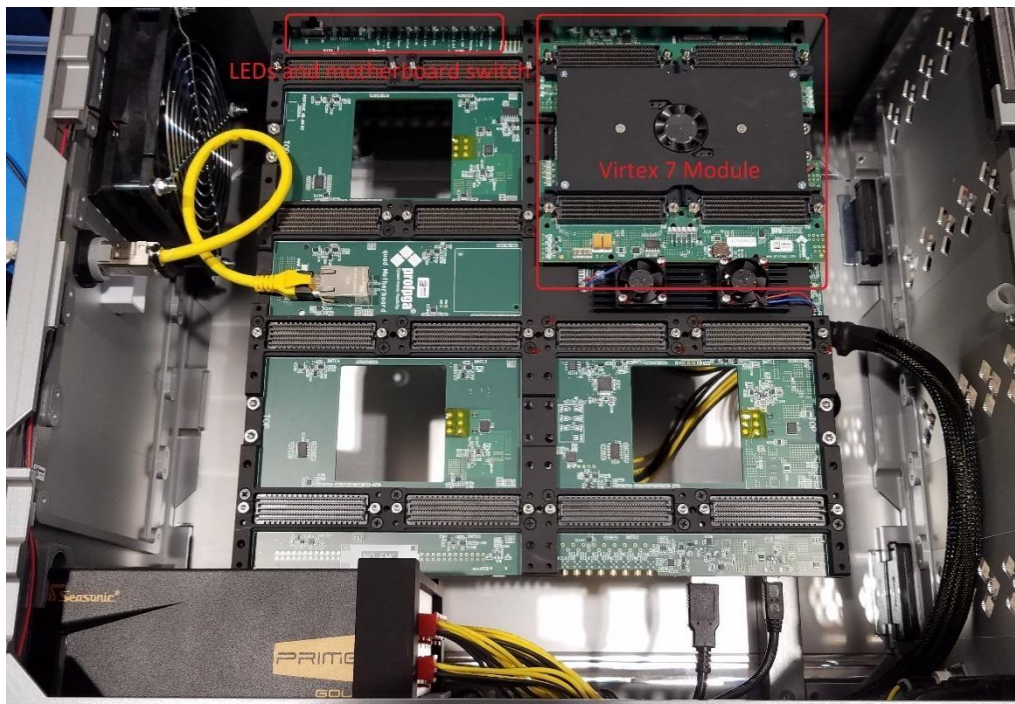


Figure 8.10: Virtex® 7 Module position on Quad Motherboard

8.1.5.2. Connecting STLP cable

STLP cable will need to be connected to the following pins located on the Virtex module:

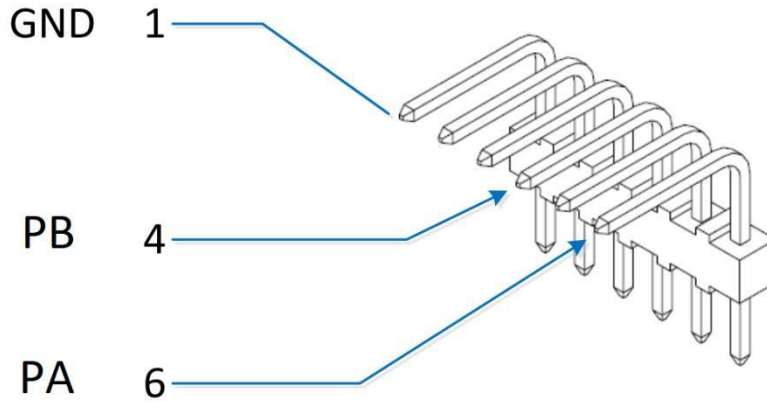


Figure 8.11: Virtex[®] 7 Module Pins

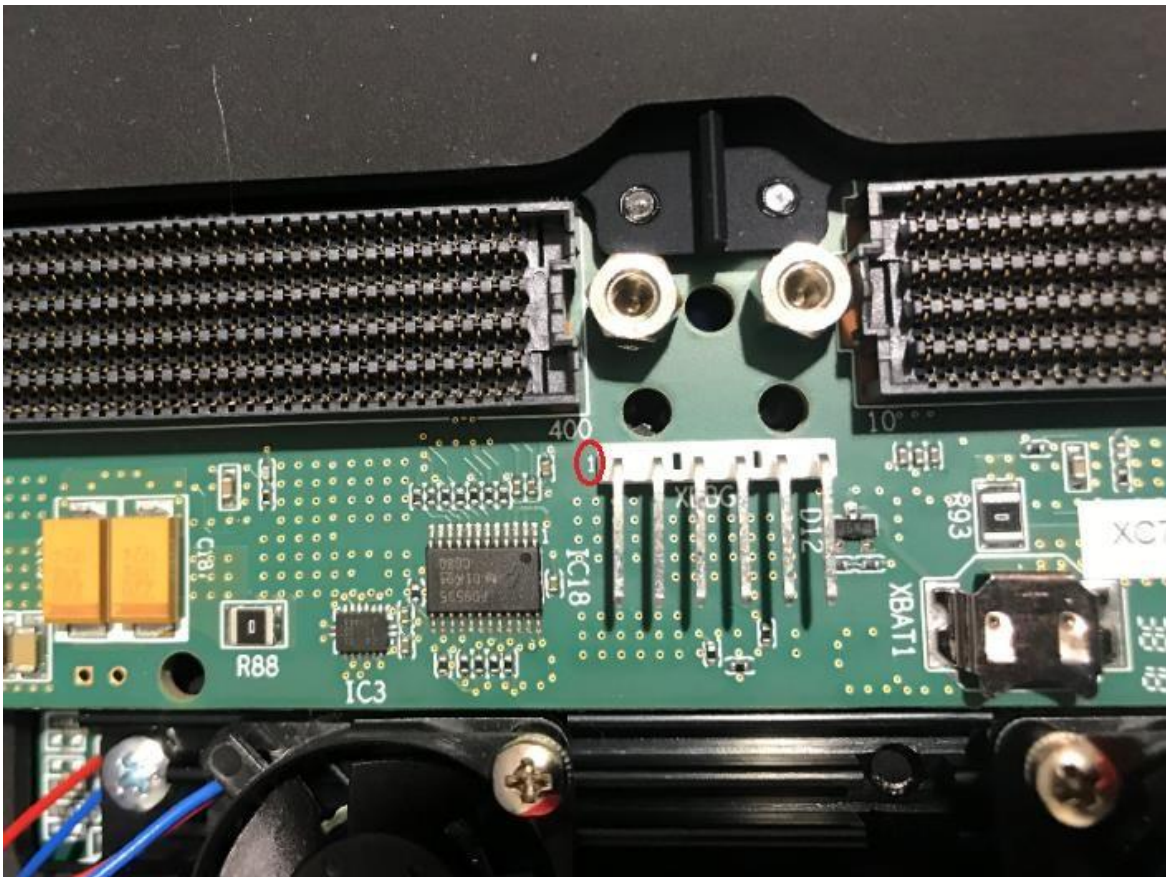


Figure 8.12: Pin 1 on ProFPGA

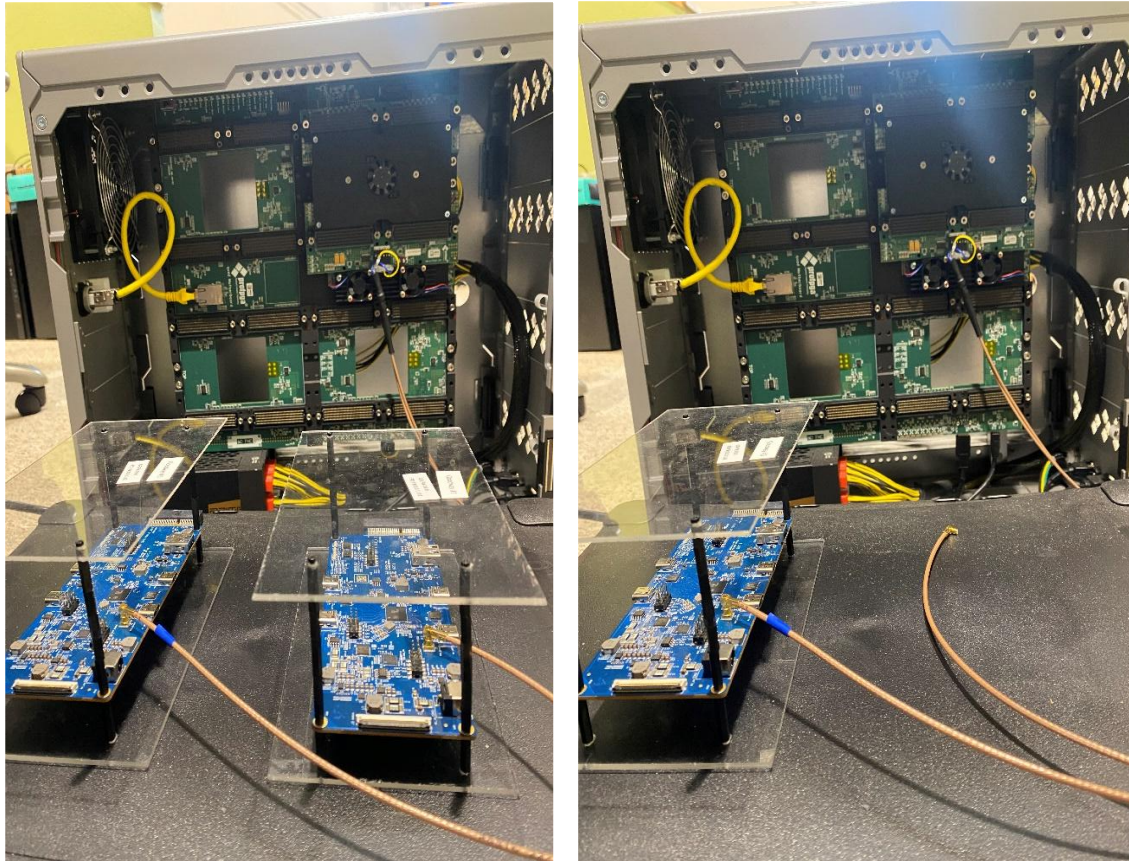


Figure 8.13: Connecting proFPGA and Goodway Gatkex Board using STLP cable. (Left) Both PA and PB connected. (Right) Only PA connected.

8.1.6. proFPGA Initial Setup on USB4CV System

This setup procedure only needs to be done once on a system and tester will only need to rely on the provided batch files for startup and shut down of the FPGA.

You will need to contact proFPGA for software install executable. Their contact can be found in the Required Hardware and Software section of this document.

The following files are provided by the USB-IF at https://usb.org/compliancetools#anchor_usb4tools to use proFPGA with USB4CV:

- TMU_FPGA.bat
- tmu_fpga.exe
- proj.cfg
- fmx7v2000tr2.bit

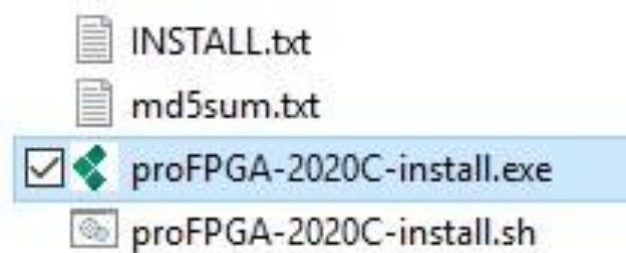
- fpga_startup.bat
- fpga_shutdown.bat

All files should be placed into the following path to use with USB4CV:

C:\Users\Public\Documents\TMU_FPGA\USB4_Compliance

8.1.6.1. Install proFPGA software

Run proFPGA software installation executable and set install path to **C:\ProDesign\proFPGA-2020C**

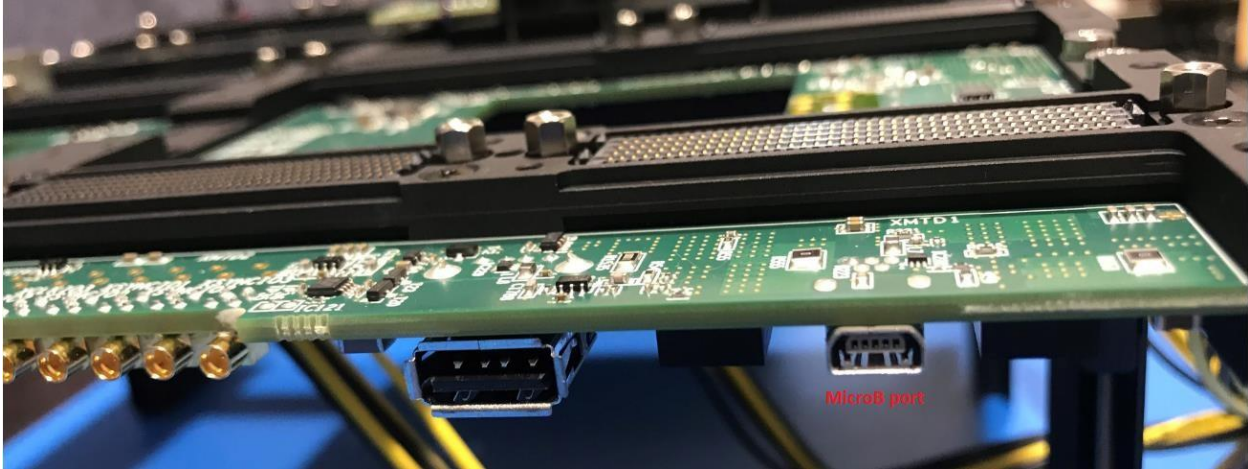


8.1.6.2. Connect proFPGA to USB4CV System

1. Turn on the power supply.
2. Change the switch on mother board from OFF to ON
3. Connect proFPGA to USB4CV System:
 - i. If you purchase the optional case for the FPGA, use a USB2.0 STD B to A cable to **connect STD B port on front of proFPGA to USB-A port on USB4CV System.** (Do not use the ethernet port on the FPGA)



- ii. If you did not get the optional case for the FPGA, use a USB2.0 microB to A cable to **connect from microB port on proFPGA to Type A port on USB4CV System.**



8.1.6.3. Install Driver for USB Ethernet/RNDIS Gadget

This driver is needed to interface with the FPGA through the USB-A port.

8.1.6.3.1. Identify Device Driver for FPGA

Open Device Manager. The device is most likely listed as a Serial Device COM under 'Ports (COM & LPT)'.

Verify this device is the FPGA by unplugging FPGA from the USB4CV System and seeing if the device goes away.

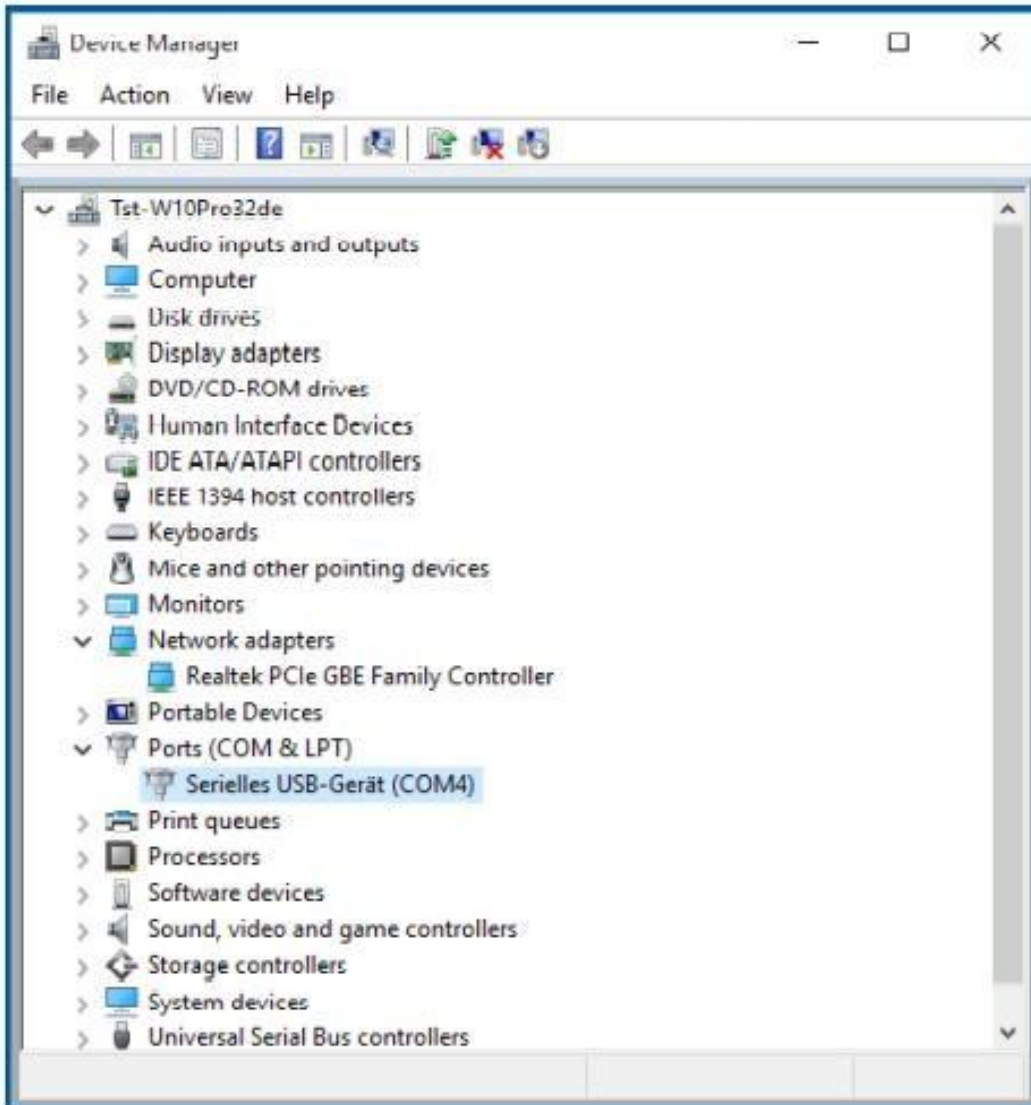


Figure 8.14: Update driver in Device Manager

8.1.6.3.2 Update Driver Software

Right-click the Serial USB Device and select “Update Driver Software”. Select “Browse my computer for software”.

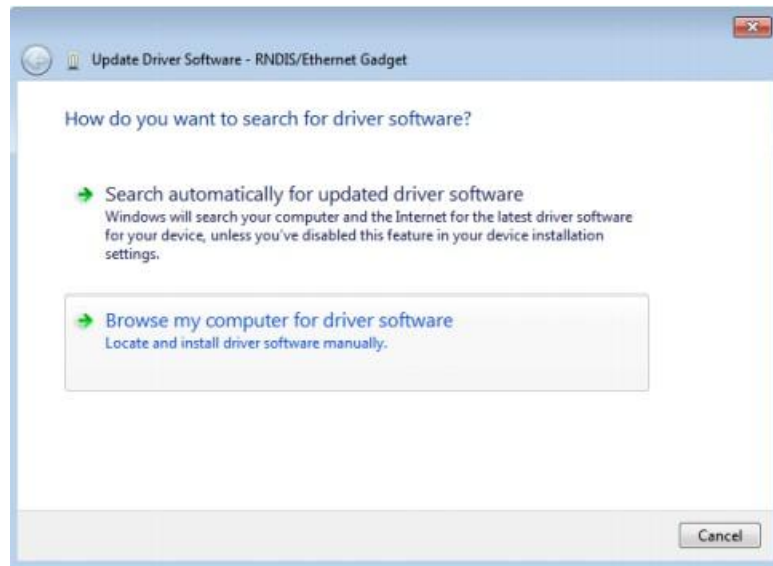


Figure 8.15: Select “Browse my computer for driver software”

Specify path to the proFPGA Windows drivers which came with the proFPGA software release and press ‘Next’:

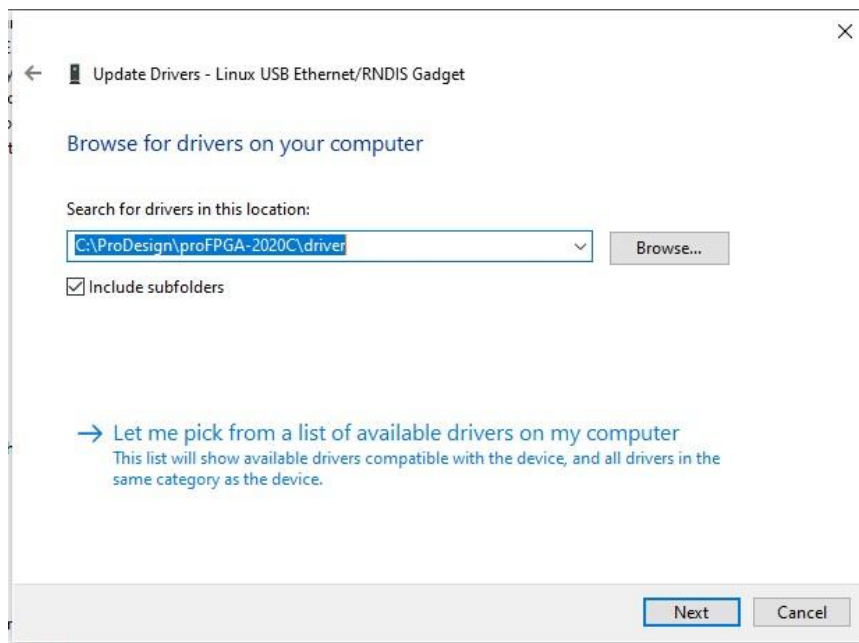


Figure 8.16: Specify path to the proFPGA

Allow installation of the driver. Press 'Install' to continue.



Figure 8.17: Allow installation

8.1.7. Set Static IP Address for FPGA

To use a Static IP address, first set up your USB4CV System's IP address to Static IP Address.

1. Open **Control Panel** and select **Network and Sharing Center**.



Figure 8.18: Network and Sharing Center

2. Click on Change Adapter Settings

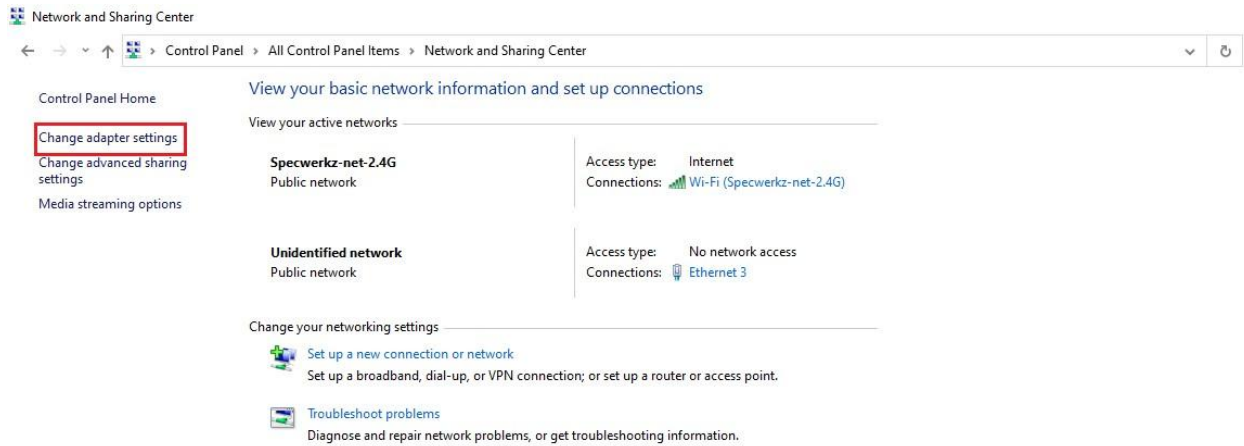


Figure 8.19: Change adapter setting

3. Click on “USB Ethernet/RNDIS Gadget”



Figure 8.20: USB Ethernet/RNDIS Gadget

4. Click on “Properties”.

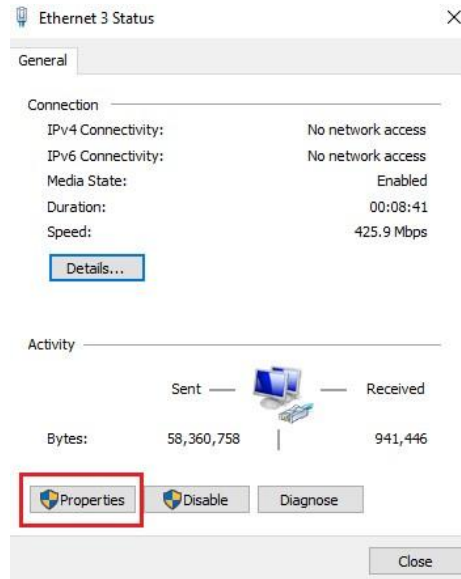


Figure 8.21: Choose “Properties”

5. Click on “Internet Protocol Version 4 (TCP/IPv4)” and open Properties.

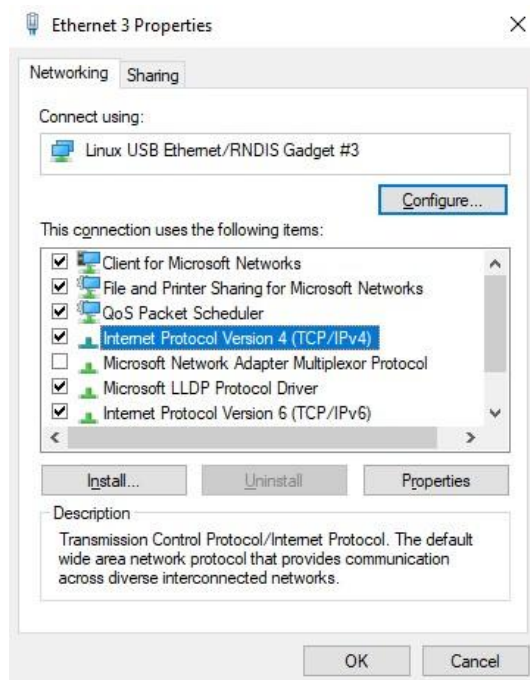


Figure 8.22: Internet Protocol Version 4 (TCP/IPv4)

6. Click on **“Use the following IP address”**

In the IP address section type **169.254.0.1** (this is USB4CV System IP address)

In the subnet mask type **255.255.255.0**

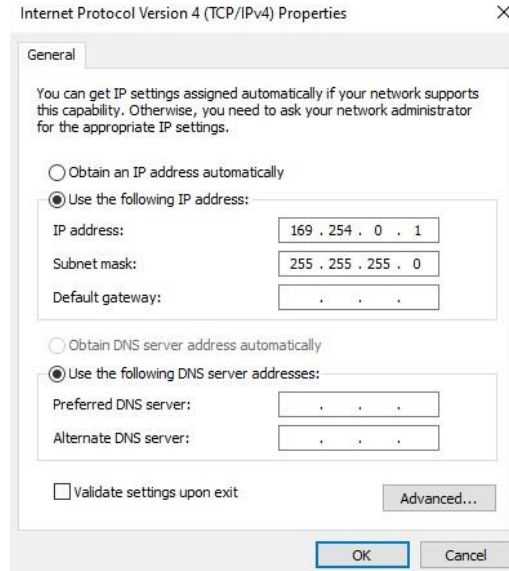


Figure 8.23: IP address and the subnet mask

7. Click **“OK”**

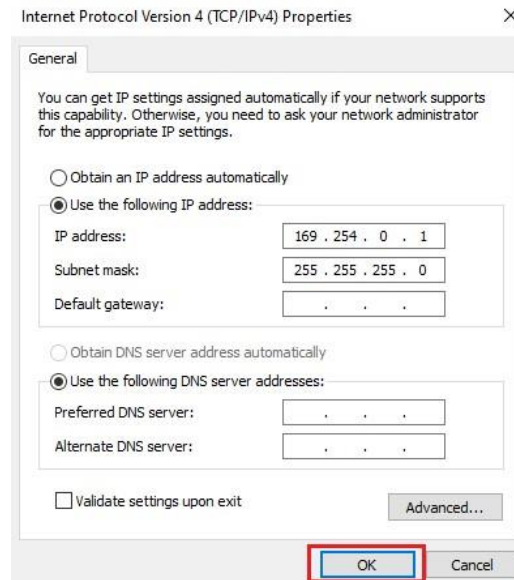


Figure 8.24: Click OK

Verify that FPGA connection to USB4CV System works

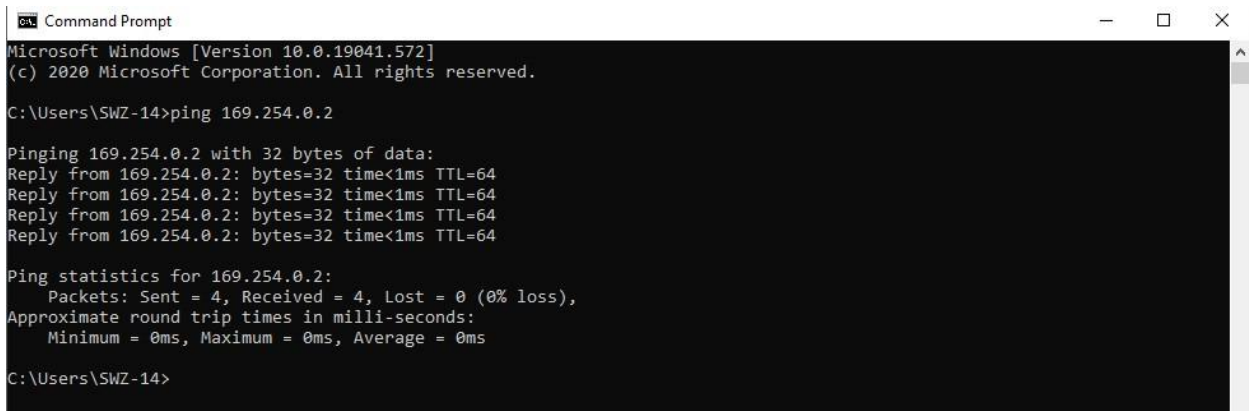
1. Open Command Prompt and enter **ping 169.254.0.2**



```
Select Command Prompt
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\SWZ-14>ping 169.254.0.2
```

Figure 8.25: Ping proFPGA IP address

2. Make sure you received all packets



```
Command Prompt
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\SWZ-14>ping 169.254.0.2

Pinging 169.254.0.2 with 32 bytes of data:
Reply from 169.254.0.2: bytes=32 time<1ms TTL=64
Reply from 169.254.0.2: bytes=32 time<1ms TTL=64
Reply from 169.254.0.2: bytes=32 time<1ms TTL=64
Reply from 169.254.0.2: bytes=32 time<1ms TTL=64

Ping statistics for 169.254.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\SWZ-14>
```

Figure 8.26: All packets received

5.1.8. Using proFPGA with USB4CV

With all files in the correct locations, all you need to do now is use the provided Windows Batch files to start up and shut down the FPGA. Every time the FPGA is powered on, you need to burn the bit file again using the provided startup batch file. Once you start up the FPGA, follow USB4 TMU CTS for connections to test setups.

A good indication for when the FPGA is ready to use is by looking for a line of LEDs next to the power switch on the Quad motherboard. When initially powering on the FPGA, only a few LEDs will be lit:

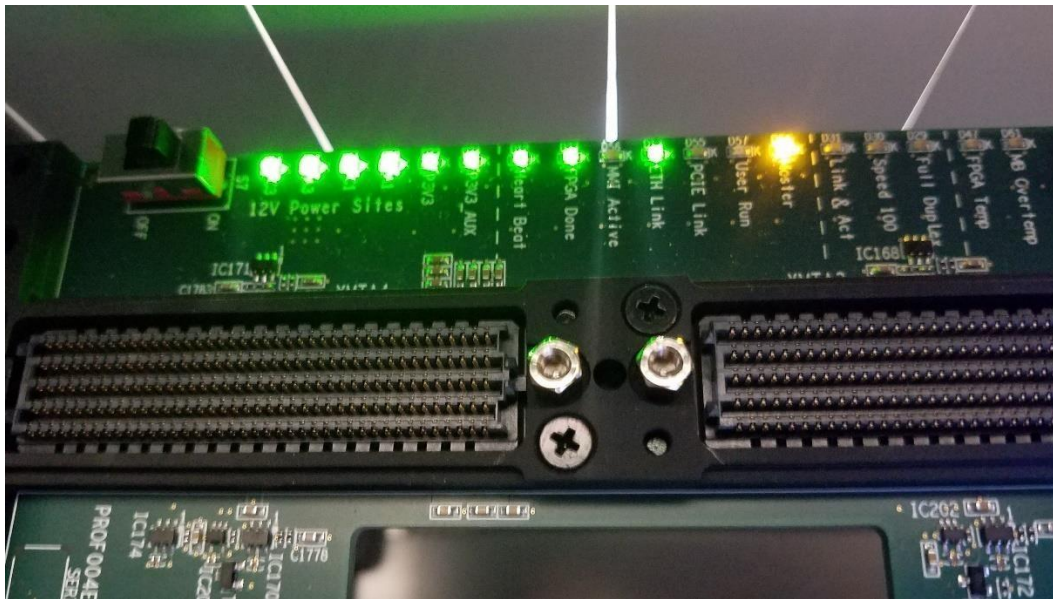


Figure 8.29: LEDs indicating bit file burned

8.1.8.1. Start up proFPGA (burning the bit file)

Every time the FPGA is powered up, it is necessary to burn the FPGA with the bit file. *fpga_start-up.bat* will do this by running the following commands:

- a. **cd C:\Users\Public\Documents\TMU_FPGA\USB4_Compliance**

```
Command Prompt
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\SWZ-14>cd C:\Users\Public\Documents\TMU_FPGA\USB4_Compliance
C:\Users\Public\Documents\TMU_FPGA\USB4_Compliance>
```

- b. **profpga_run.exe proj.cfg -u**

```
Command Prompt
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\SWZ-14>cd C:\Users\Public\Documents\TMU_FPGA\USB4_Compliance
C:\Users\Public\Documents\TMU_FPGA\USB4_Compliance>profpga_run.exe proj.cfg -u
```

- c. Wait until loading .bit file is done

```

INFO : power up FPGA modules on motherboard_1
INFO : power up FPGA module <MB1 TA1 0>
INFO : setting over-temperature shutdown to 84 deg. C
INFO : enable PSUs (except IO voltage)
INFO : enable VIO PSUs
INFO : checking power-goods of FPGA modules on motherboard_1
INFO : checking power-goods on FPGA module <MB1 TA1 0>
INFO : Release resets from FPGA modules on motherboard_1
INFO : configure FPGAs on motherboard_1
INFO : configure FPGAs 1
INFO : loading bitstream "C:/Users/Public/Documents/TMU_FPGA/USB4_Compliance/fmxc7v2000tr2.bit" into FPGA <MB1 TA1 0> ... on MMI64 address 05 with id 0x
30000005 please wait
00.0 [=====]
INFO : done
INFO : reset on sync_0 on motherboard_1 released
INFO : Rescanning for MMI64 modules inside User FPGAs. This operation may
INFO : deadlock if MMI64 inside the user design does not respond correctly.
INFO : Success.
INFO : MB 1.FPGA_TA1: Found profpga_ctrl module inside user design.
INFO : Done.

```

8.1.8.2. Shut Down proFPGA

Please run the shutdown script and wait for proFPGA to fully shutdown before powering off the FPGA.

fpga_shutdown.bat will shut down the FPGA by running the following command:

profpga_selftest C:\Users\Public\Documents\TMU_FPGA\USB4_Compliance\proj.cfg -d

Wait until proFPGA finishes shutdown.

```

INFO : Power down motherboard_1
INFO : deactivating P12V_A1...
INFO : deactivating P12V_C1...
INFO : deactivating P12V_A3...
INFO : deactivating P12V_C3...
INFO : deactivating P3V3...
INFO : deactivating P3V3_AUX...

```

Troubleshooting:

If you are not able to turn on the FPGA with the batch files, please make sure that you followed the initial setup procedure correctly. Files need to be in a specific folder path and uses the specific IP address indicated in this procedure.

8.1.8.3. LEDs for PA and PB

When connecting to test setups defined in TMU CTS, FPGA module has dedicated LEDs that will light up when it is getting TMU_CLK_OUT from a device that is connected to a host and has established a stable link.

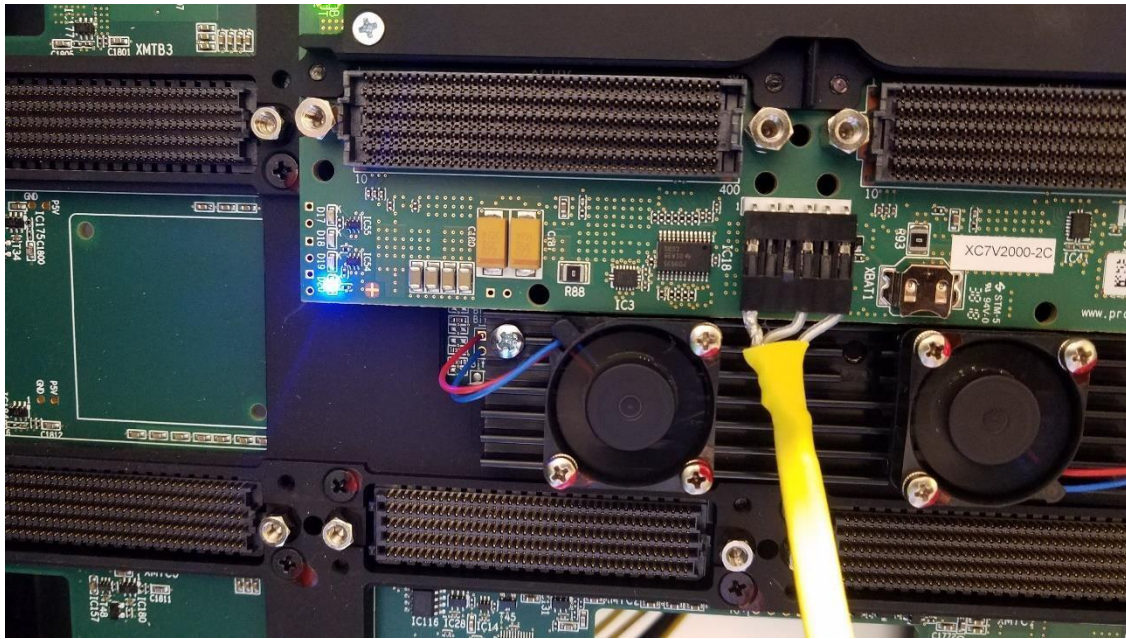


Figure 8.30: Blue LED lit for PA connection

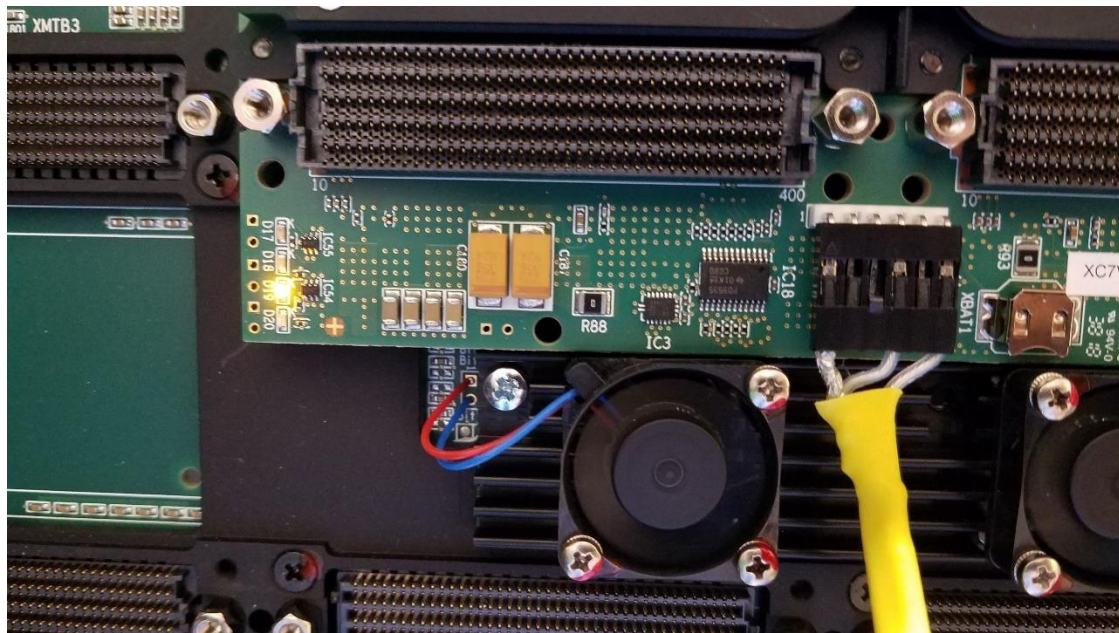


Figure 8.31: Orange LED lit for PB connection

8.2 DP Testers Setup

8.2.1 Required Hardware and Software

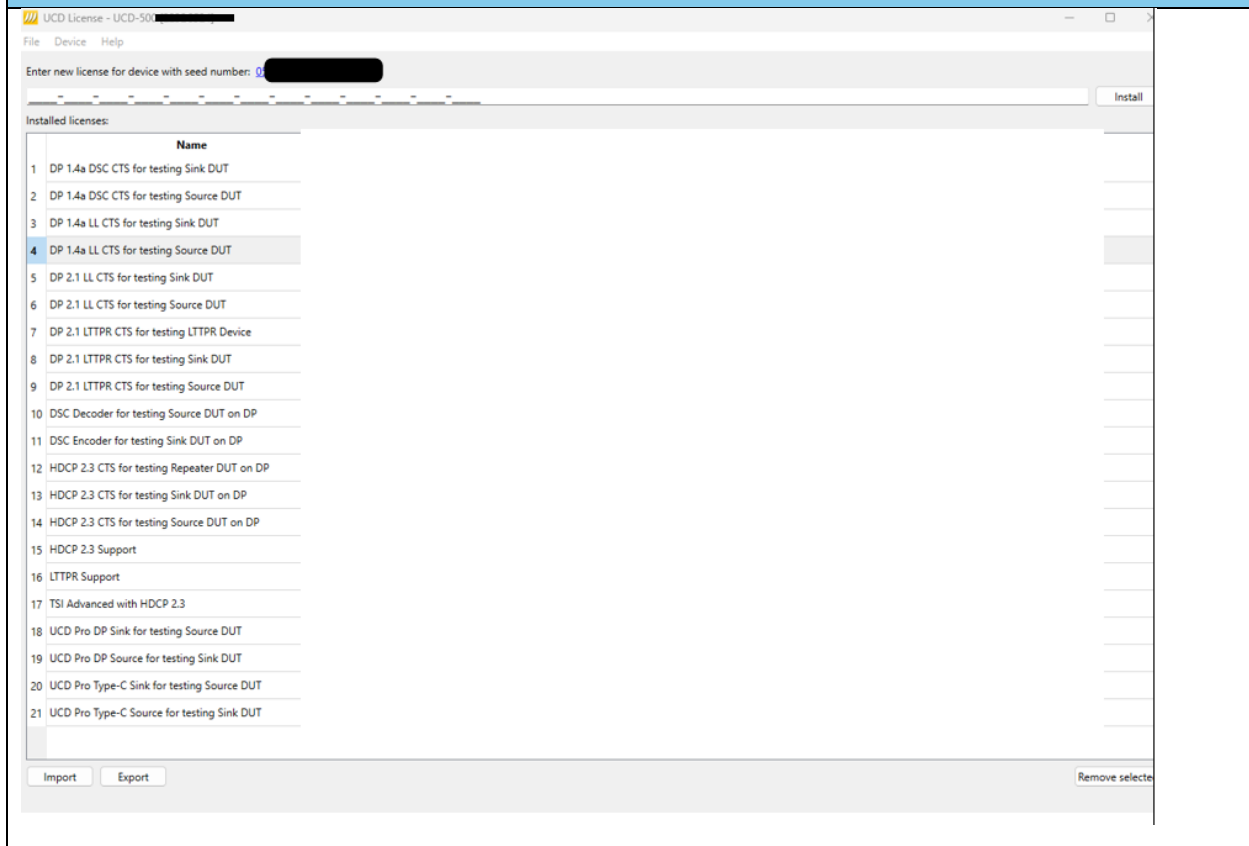
8.2.1.1 Unigraf Testing Solution – UCD-500 Gen2

Contact Information
https://www.unigraf.fi/support/

Hardware	
UCD-500 Gen2 DisplayPort 2.1 Generator & Analyzer *Should include cables necessary for connecting to DUTs: <ul style="list-style-type: none">• DP-to-C Bidirectional cable• DP cable	https://www.unigraf.fi/product/displayport-2-video-generator-analyzer-ucd-500/

Software	
UCD-500 Gen2 SW Bundle (Download latest version on website)	https://www.unigraf.fi/downloads/

Necessary Licenses



8.2.1.2 [WIP] Teledyne LeCroy Testing Solution – M42DE

Contact Information

Email: psgsupport@teledynelecroy.com

Please provide the following information in your email:

- Device Model number
- Device Serial number
- Firmware version installed.
- Software version (if applicable)
- Your location

Phone: 800-909-7211 or 408-727-6600

Fax: 408-727-622

Telephone support is available 8:00 am – 4:45 pm CST

Hardware

M42DE

80Gbps Video
Analyzer/Generator for
DisplayPort 2.0 Testing

<https://www.teledynelecroy.com/protocolanalyzer/quantumdata-m42de.aspx>

Software

Video Protocol Suite

ATP Manager Graphical User Interface

<https://www.quantumdata.com/downloads.html>

Necessary Licenses

Description

DP Auxiliary Channel Analyzer
DisplayPort Capture/Analysis
M42d DP 2.0 HDCP 2.3 Function
DP HDCP 2.3 Compliance Test - Source
DP HDCP 2.3 Compliance Test - Sink
DisplayPort CT Package 2 LLCT 1.2 Core - Sink
DisplayPort CT Package 3 LLCT HBR3 Source Tests
M42d DP 2.0 Video Generator Function
M42d DP 2.0 Basic Video Analyzer Software
DP 1.4 Display Stream Compression (DSC) Rx Analysis
DP 1.4 FEC/DSC Tx
DisplayPort CT Package 4 LLCT 1.4 - Sink Tests
M42d DP 2.0 USB-C Alt Mode Enable
M42d DP 2.0 DP Port Enable
M42d eDP Enable
DP DSC Compliance Test of Source
DP DSC Compliance Test of Sink
DP EDID/DisplayID Sink Compliance
DP EDID/DisplayID Source Compliance
M42d DP Source Link Layer Compliance 1.2 + 1.4
M42d DP Sink Link Layer Compliance 1.2 + 1.4
M42d DP 1.4 Sink Test Enhanced Functional test
M42d DP 1.4 SourceTest Enhanced Functional test
M42d DP 2.0 Sink Test Enhanced Functional test
M42d DP 2.0 Source Test Enhanced Functional test
M42d DP 1.4 to DP 2.0 Upgrade
M42d Passive Probing
DP 2.0 Sink Test Panel Replay
DP 2.0 Source Test Panel Replay
DP 2.0 Sink Test LTPR
DP 2.0 Source Test LTPR
DP 2.0 Capture/Analysis
DP 2.0 Basic Analysis
DP 2.0 Video Generation
LTPR 1.4 Sink Device Testing
LTPR 1.4 Source Testing
M42d DP 2.0 Link Layer Source Compliance Test
M42d DP 2.0 Link Layer Sink Compliance Test
M42d DP Adaptive Sync Source Compliance Test
M42d DP Adaptive Sync Sink Compliance Test
M42d DP 2.0 DSC Source Compliance Test
M42d DP 2.0 DSC Sink Compliance Test
M42d LTPR Source Compliance Test
M42d LTPR Sink Compliance Test
M42d LTPR Device Compliance Test

8.2.1.3 Recommended External Host, Cables and Adapters

Hardware	Vendor	Link
External Host	NA	*At minimum should be able to support software for testing solutions above
DP-to-DP cable	Club 3D	Club 3D DisplayPort 1.4 to DisplayPort 1.4 M/M 1m/3.28ft
C-to-DP Bidirectional cable	Club 3D	Club 3D Club CAC-1557 3D USB Adapter Cable 1.8m Type C to DisplayPort 8K60Hz HDR Black
USB-C Digital AV Multiport Adapter	Apple	https://www.apple.com/shop/product/MUF82AM/A/usb-c-digital-av-multiport-adapter
8K HDMI cable	Zeskit	https://www.zeskit.com/maya21
Moshi Cable	Moshi	Amazon.com: Moshi USB C to DisplayPort Cable 1.5m/5ft, Support 5K@60 Hz, 4K HDR, VESA Certified, Aluminum Housing, Thunderbolt 3 Compatible (Ensure Monitors Accept DisplayPort Signal) : Electronics

8.2.2 Setting up DisplayPort Testing Solutions

It is advised to run DisplayPort Testing solution software on a separate host from what is part of the USB4 test setup. This can be on an external host or an embedded host on the testing solution hardware.

8.2.2.1 Unigraf Solution – UCD-500 Gen2

8.2.2.1.1 Install UCD-500 Gen2 SW Bundle (latest version)

Download the latest version of UCD-500 Gen2 SW Bundle from Unigraf’s website and install it on an external host.

Important note: When choosing components, please also check **DSC Content library** for installation

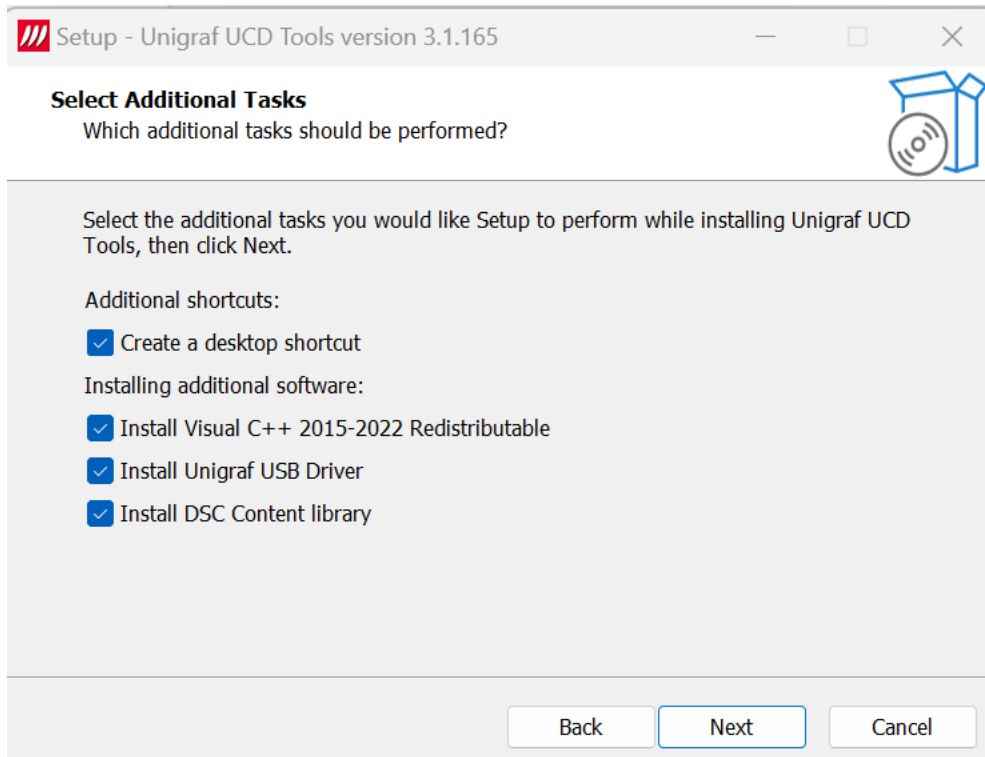


Figure 8.32 - Installing UCD Software Bundle: include DSC Content library.

8.2.2.1.2 Activate licenses on UCD-500 Gen2

Connect UCD-500 Gen2 standard-B port to a computer with SW bundle installed and open License Manager (installed as part of SW Bundle) and choose **UCD Devices**.

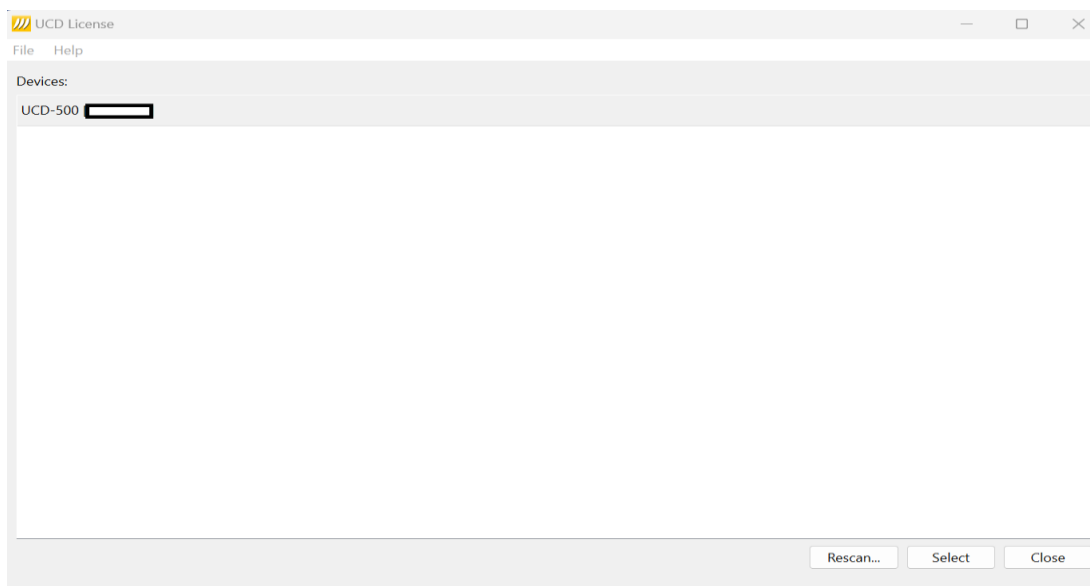


Figure 8.33 - License Manager: select UCD device.

Select your UCD-500 Gen2 unit and install your licenses. It is possible to copy and paste keys or import a .INI file with your licenses in it.

You should only have to install licenses once, but it is suggested to export your licenses as an .INI file for quick activation if you need to reactivate in the future.

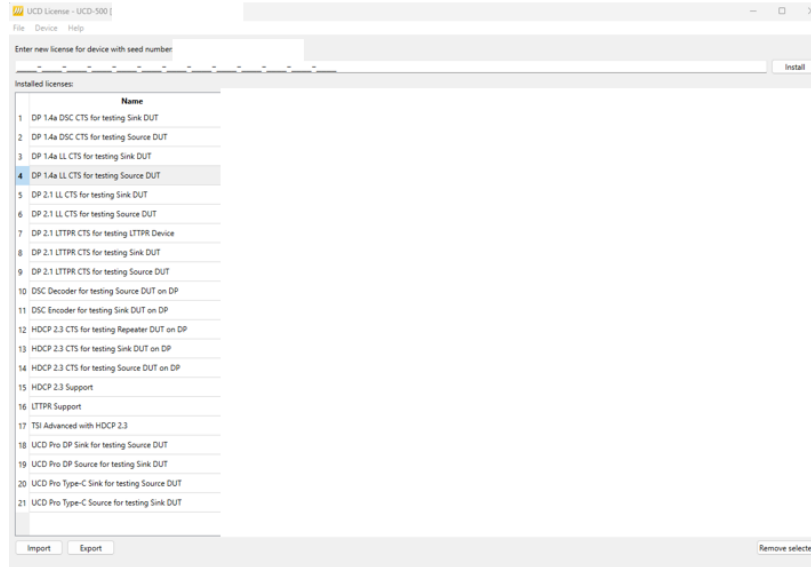


Figure 8.35 - License Manager: select UCD-500 and add your licenses.

8.2.2.1.3 Update firmware with SW Bundle

Here is how to check firmware and update it if it does not match what that software version provides.

Open UCD Config and a list of update files will appear. Press **next**.

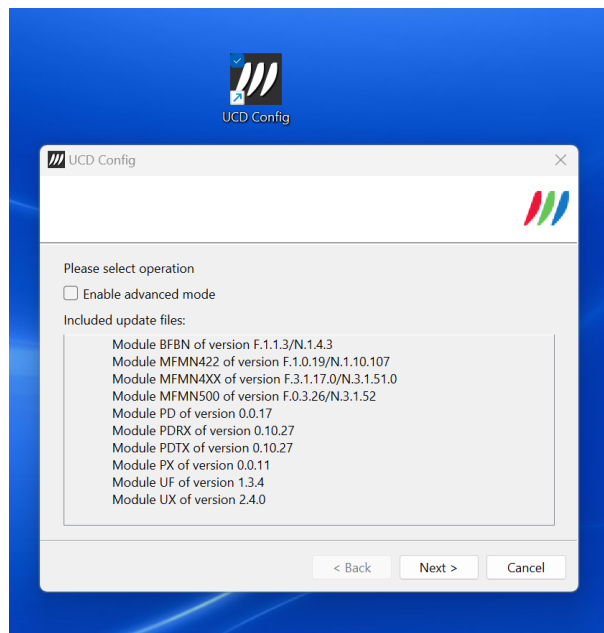


Figure 8.36 - UCD Config: firmware update

This will show you the UCD500 that you want to update the firmware version for. **Click Next.**

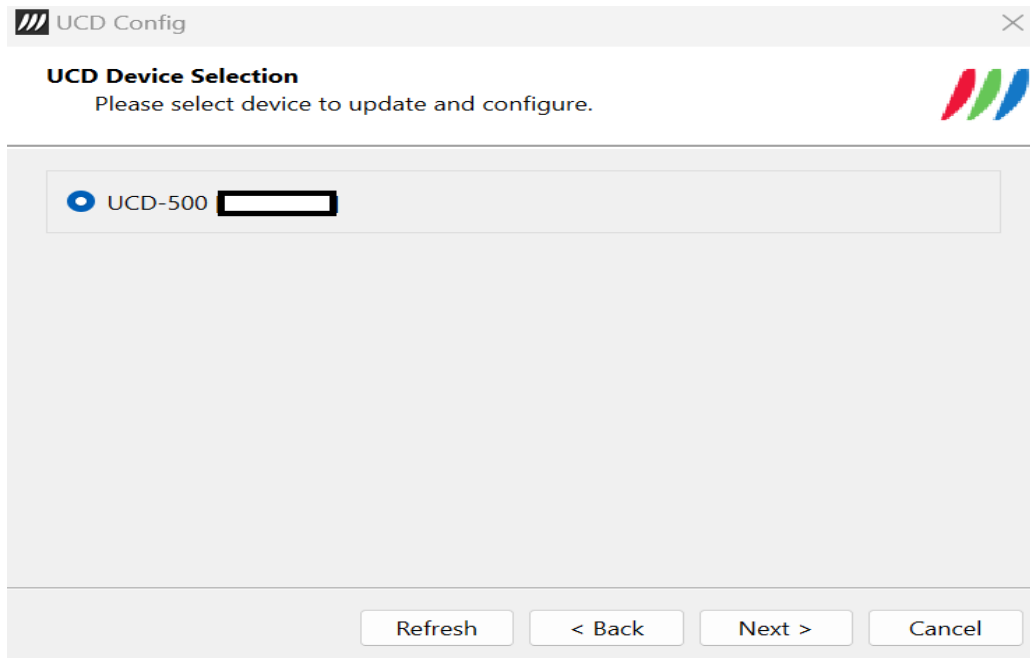


Figure 8.37 - UCD Config: select device and update.

If New and Existing do not match, you do not need to update. Otherwise, make sure the device is selected and click Next to proceed with updating the firmware.

Follow Firmware Update Utility instructions and UCD-500 Gen2 will be ready to use.

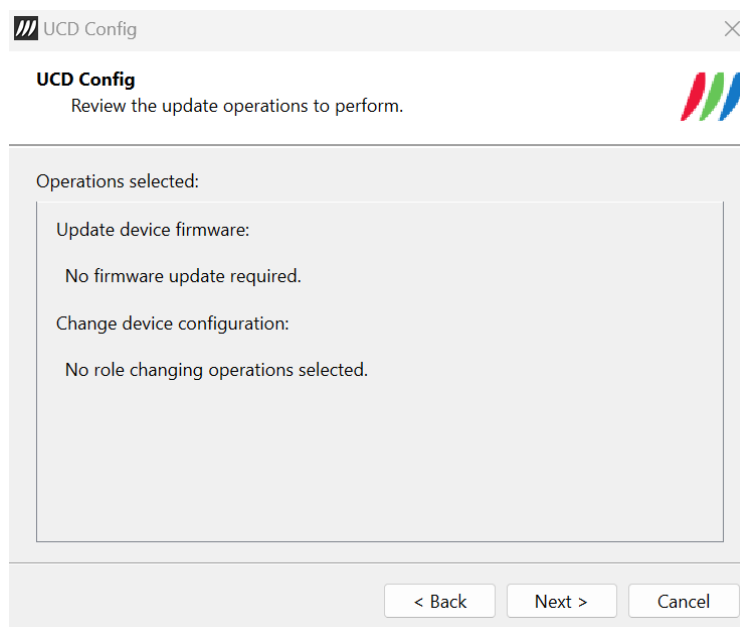


Figure 8.38 - UCD Config: Update Operations to perform.

8.2.2.2 [WIP] Teledyne LeCroy Solution – M42DE

This test solution comes with an embedded host. Test operators can connect this unit to a monitor, mouse, and keyboard and run tests without an external host.

8.2.2.2.1 Setting up IP address for external host

If the test operator wants to use an external host, this requires an ethernet connection between the M42de and the external host. On the external host, open the Command console and use command **ipconfig/all**.

Look for an Ethernet adapter associated with the Ethernet port that M42de is connected to.

```
Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : 
Description . . . . . : Intel(R) Ethernet Controller (3) I225-V
Physical Address. . . . . : FC-34-97-9E-67-36
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::d14a:adb8:b623:5e00%14(Preferred)
Autoconfiguration IPv4 Address. . : 169.254.94.0(Preferred)
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . : 
DHCPv6 IAID . . . . . : 251409559
DHCPv6 Client DUID. . . . . : 00-01-00-01-29-24-62-01-FC-34-97-9E-67-36
DNS Servers . . . . . : fec0:0:0:ffff::1%1
                       : fec0:0:0:ffff::2%1
                       : fec0:0:0:ffff::3%1
NetBIOS over Tcpip. . . . . : Enabled
```

Figure 8.39: M42de External Host - find local IP address and subnet mask.

On the external host, Go to Control Panel Control Panel → Network and Internet → Network and Sharing Center. Then Press on the Ethernet

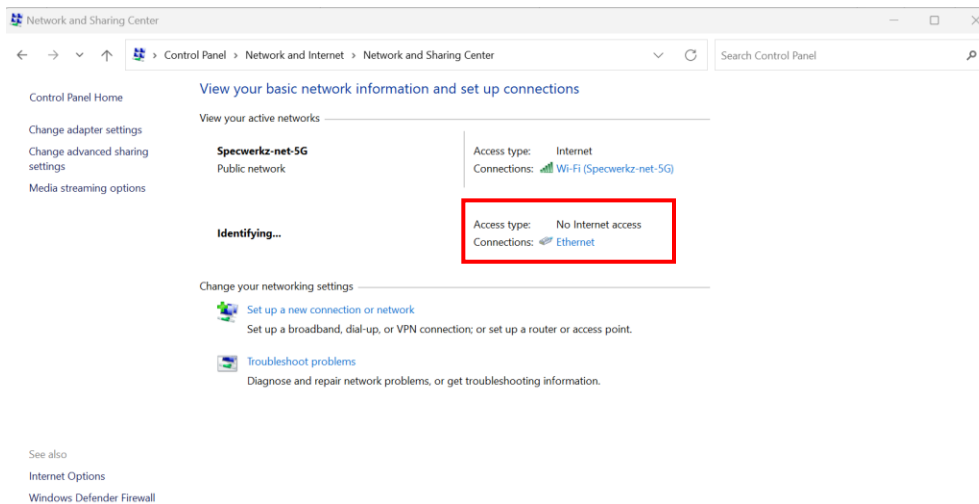


Figure 8.40: External Host Control Panel

Make sure that you see both receive and sent not zero.

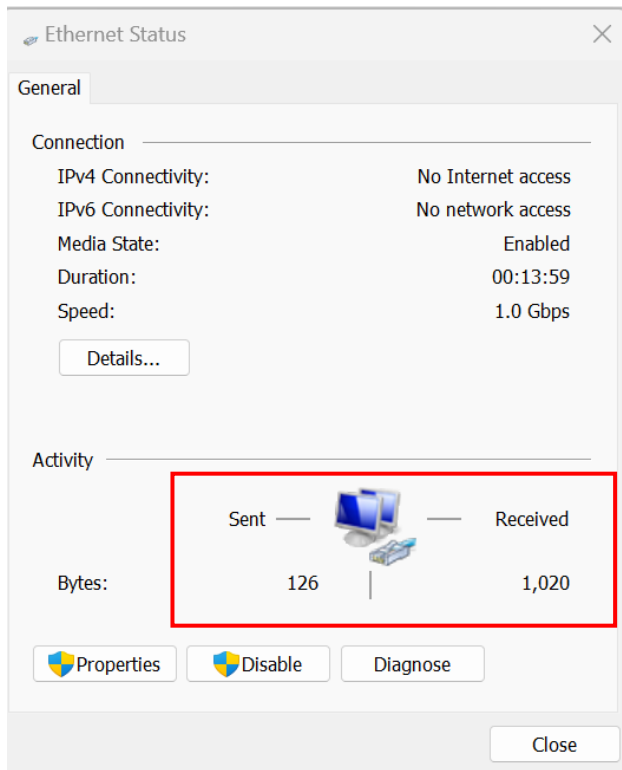


Figure 8.41: Ethernet Connection

If the received is zero, then press on Properties → Internet Protocol Version 4 (TCP/IPv4) → Properties. Make sure you set up the IP address as follows. Note, to find the IP address and Subnet for the M42DE, on the M42DE switch to ATP Manager then go to Other → Instrument Network Settings

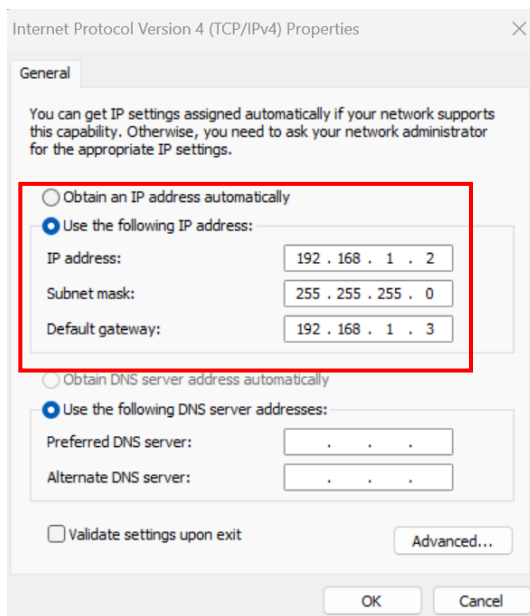


Figure 8.42: IP connection on External Host

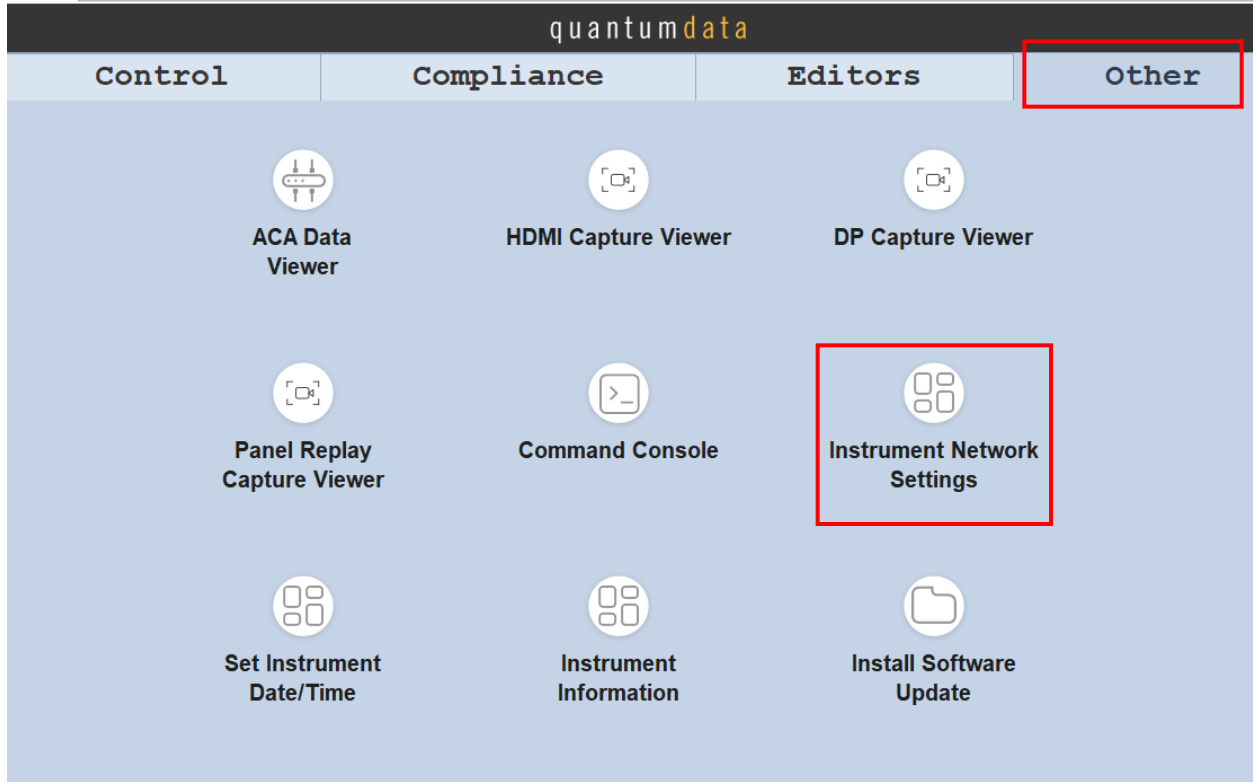


Figure 8.43: Checking Instrument Network IP address.

Once you get the connections set up, install ATP Manager Graphical User Interface. and click on “Instruments” Then in Instruments, click on “Add” and set the IP address to 192.168.1.3 as the address for the instrument. Then press on “Connect.”

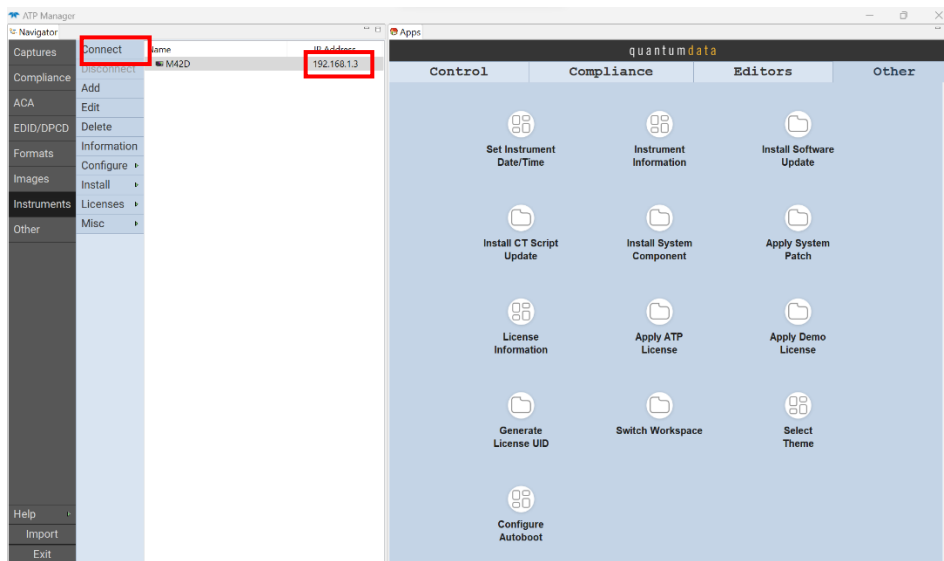


Figure 8.44: M42de External Host - add instrument.

8.2.3 UCD-500 Testing Solution

8.2.3.1 Settings for UCD-500 as Reference Source

The following settings should always be enabled while using UCD-500 as a Reference Source in the test setup.

Prefer FEC Enabled

DP Tx >> FEC >> Prefer FEC Enabled

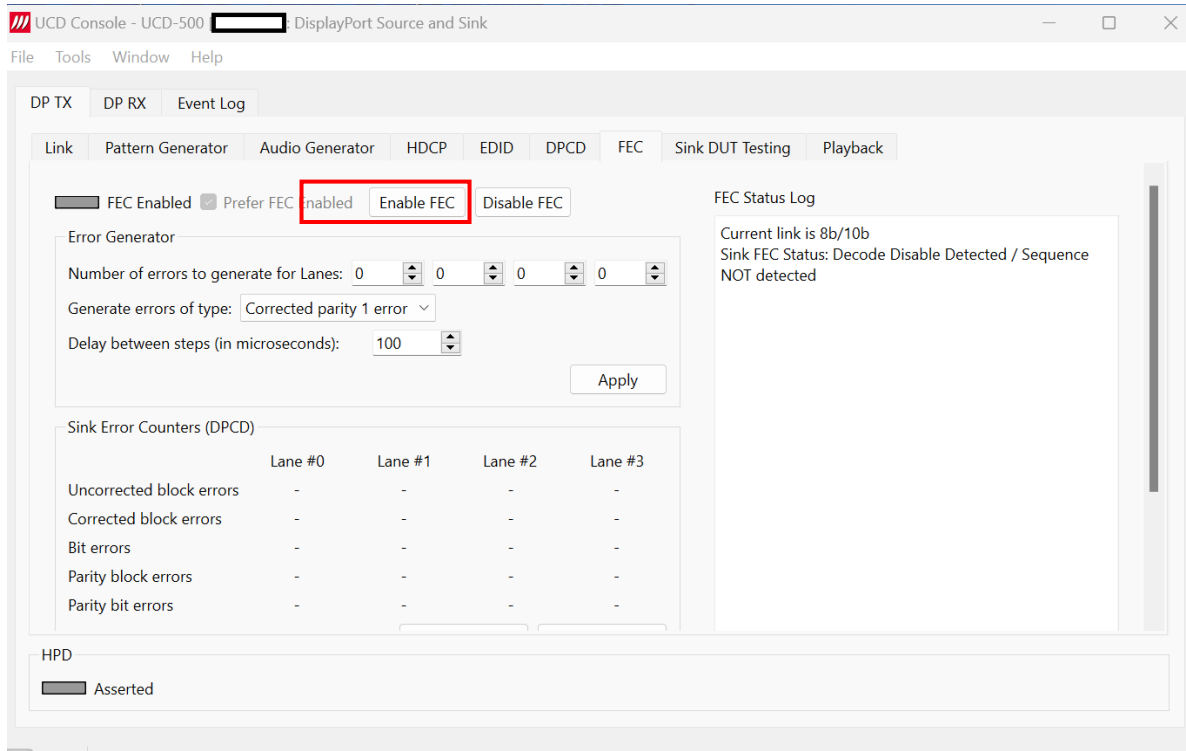


Figure 8.45 - UCD-500 Gen2: Prefer FEC Enabled on DP Tx

HDCP 2.3 Authentication Enabled

This is only needed for HDCP testing. This setting does not stay enabled unless there is an active link. User will need to remember to enable this after starting DP Tunnel using USB4CV.

DP Tx >>> HDCP >>> enable Authenticate under HDCP 1.3 and HDCP 2.3

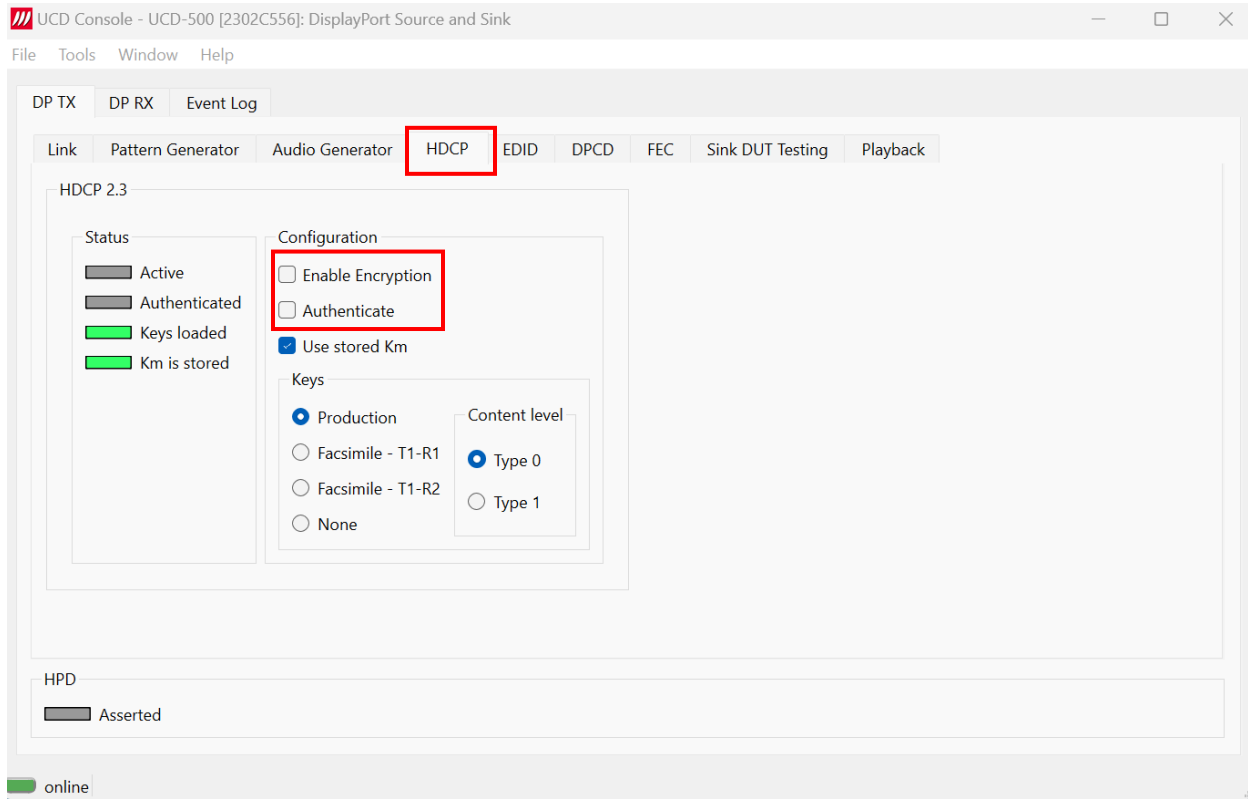


Figure 8.46: UCD-500 Gen2: Enable HDCP 2.3 Authentication on DP Tx

8.2.3.2 Settings for UCD-500 as Reference Sink

The following settings should always be enabled while using UCD-500 as a Reference Sink in the test setup.

FEC Capable

DP Rx >>> FEC >>> enable FEC Capable

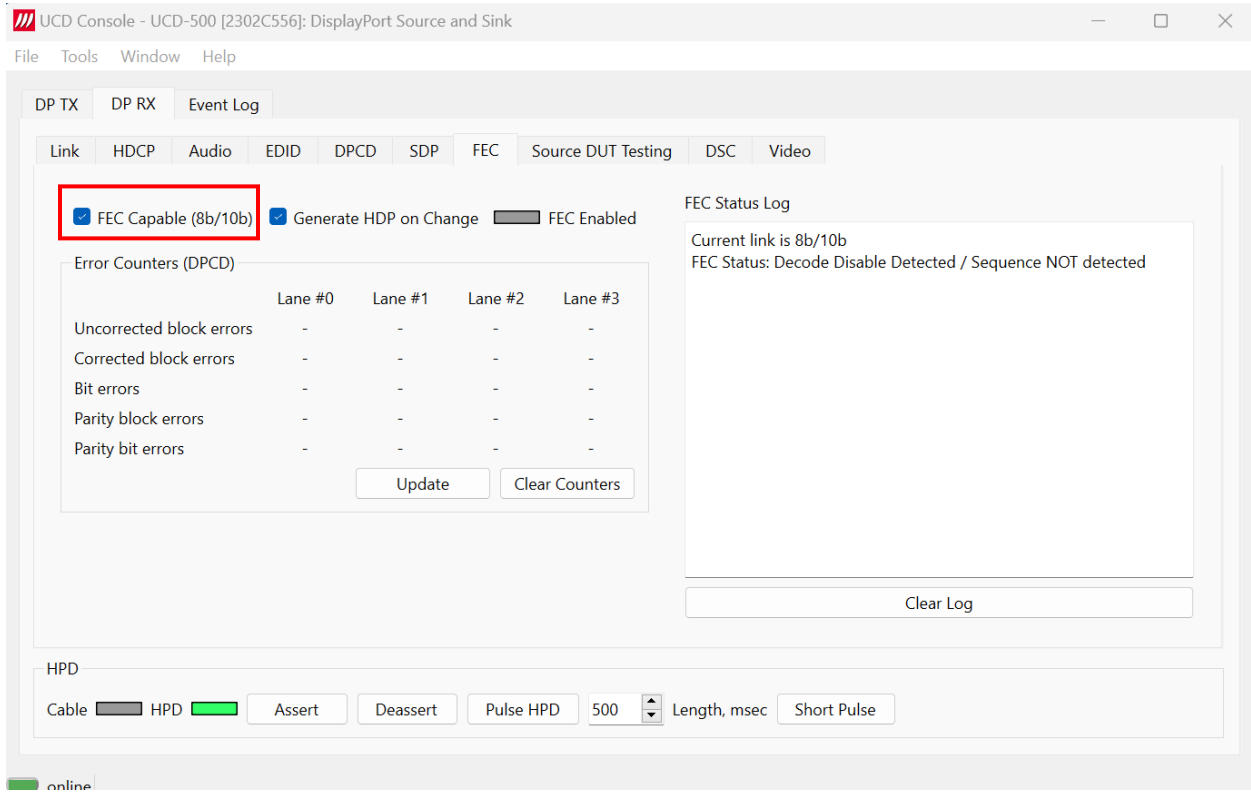


Figure 8.47: UCD-500 Gen2: FEC Capable set on DP Rx

HDCP Capable

DP Rx >>> HDCP >>> Enable HDCP Capable

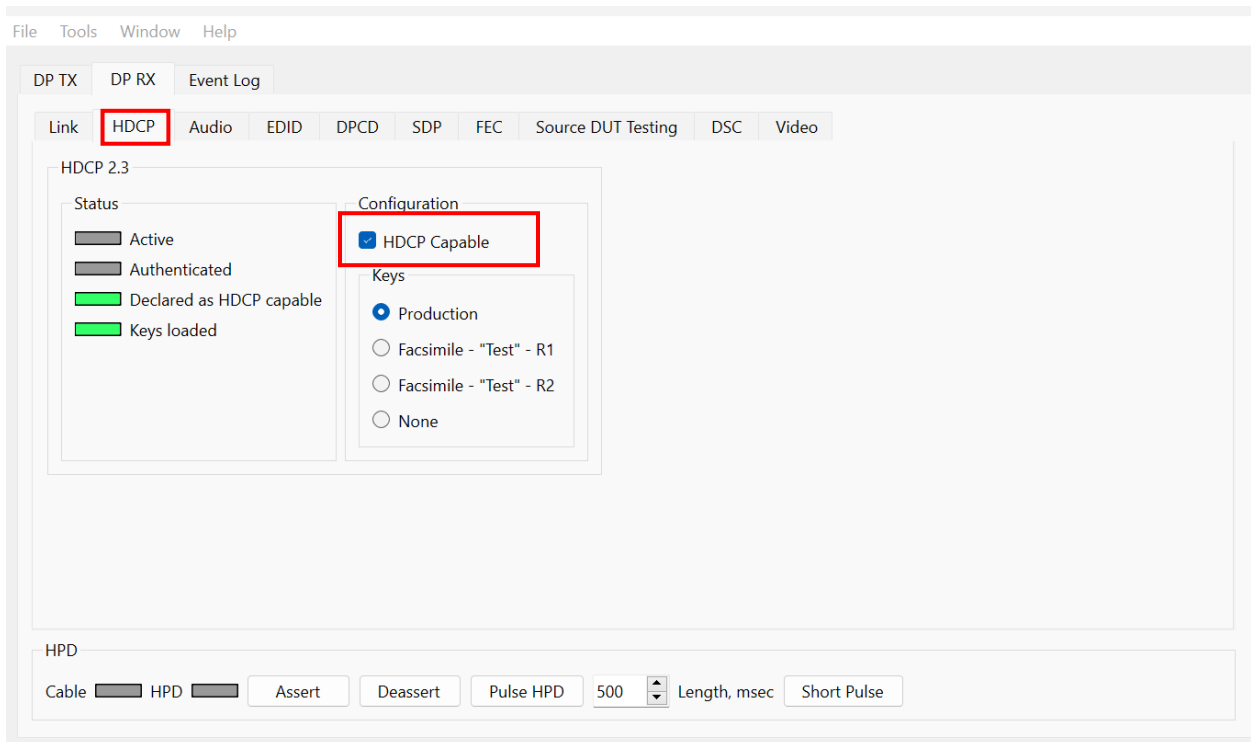


Figure 8.48: UCD-500 Gen2: HDCP Capable set on DP Rx

Load Source CDF Preset

For accurate testing results for DisplayPort CTS Assertions, it is important to input the capabilities of the source being tested. Vendors should provide a Capabilities Declaration Form (CDF) file to test with UCD-500.

All preset files should be located at the following directory for accessibility through UCD-Console:

C:\ProgramData\Unigraf\UCD_Console\Presets\DP_RX_LL_CTS

You can load this preset file in UCD Console by going to DP Rx >> Source DUT Testing >> DP 1.4 LL CTS >>Configure >> Preset (see figures below).

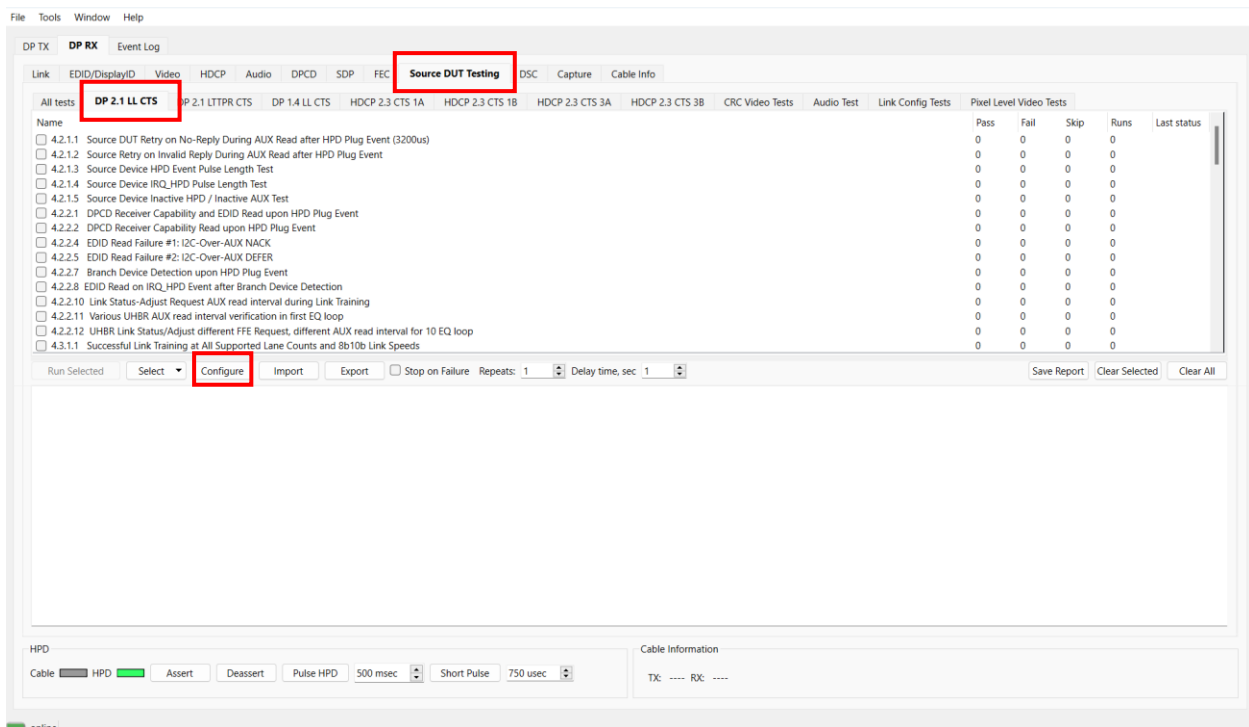


Figure 8.49 - DP Rx: Source DUT Testing.

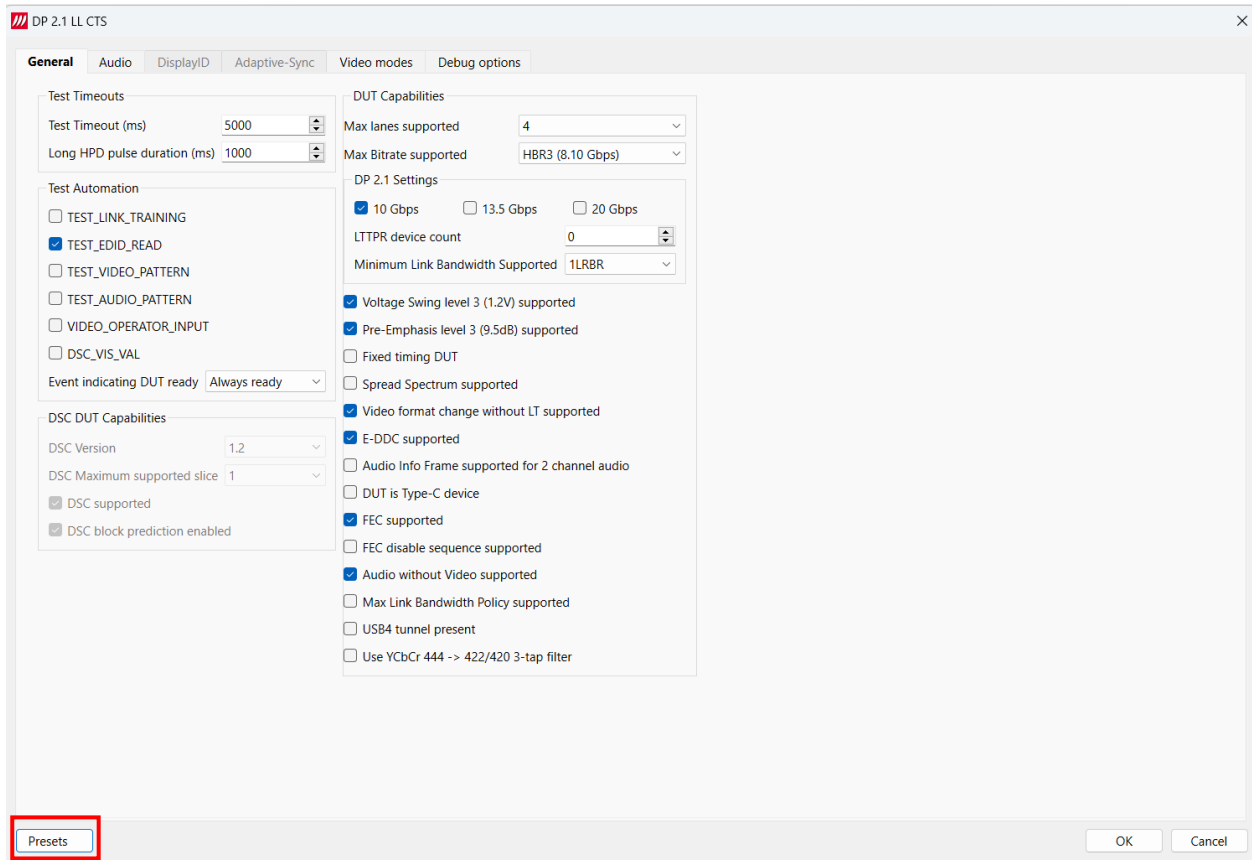


Figure 8.50 - Source DUT Capabilities, Presets button location in red

DP CTS Assertion Testing - Add delay for Source DUT Testing

Modify the delay to 5 seconds in UCD Console to run automated tests.

Although it is possible to select multiple tests and have UCD Console step through each item automatically, it tends to be too fast for the USB4 Link to establish a stable link before it moves on to the next test. This delay is necessary to allow the link to be established before tests execute.

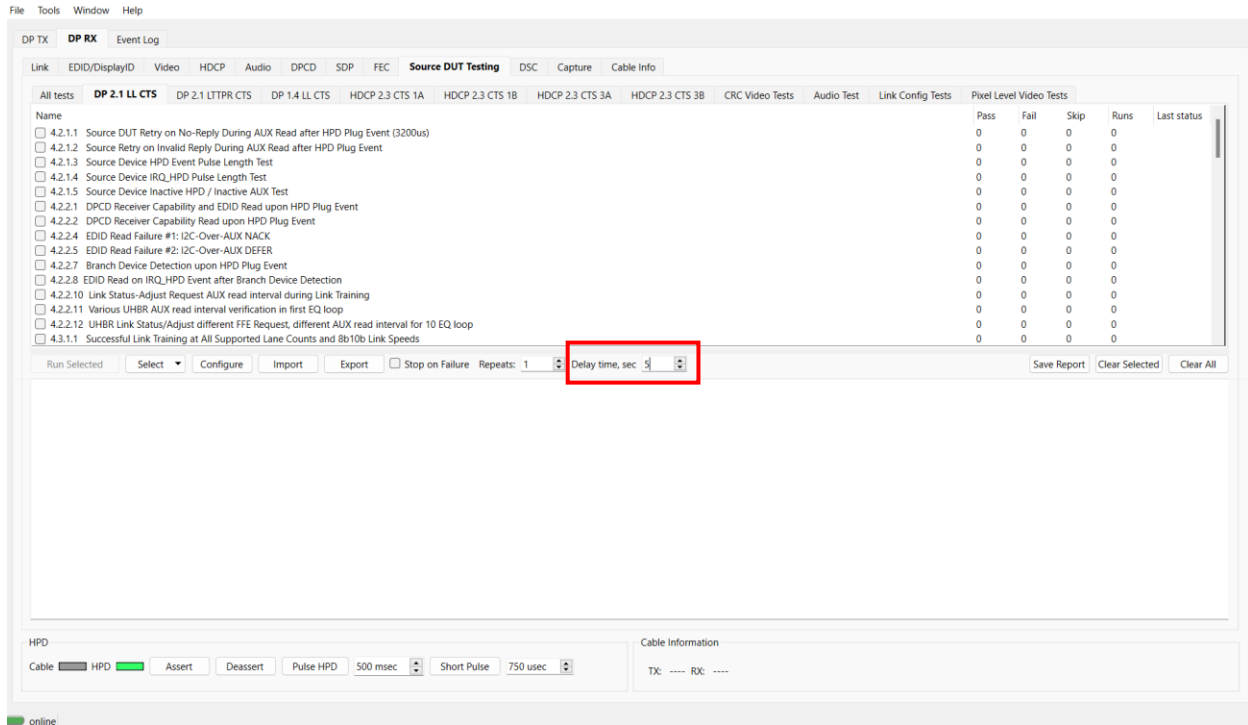


Figure 8.51 - Source DUT Testing: change delay to 5 seconds

Generate DSC Content for UCD-500 as Reference Source

DSC content will need to be generated for Source DUT testing by using a script provided in the Unigraf SW Bundle.

This can be found in **Windows Start >>> Unigraf >>> Generate DSC Content**.

Users can edit this script in Notepad and change the location that they want the content to be located. By default the location is:

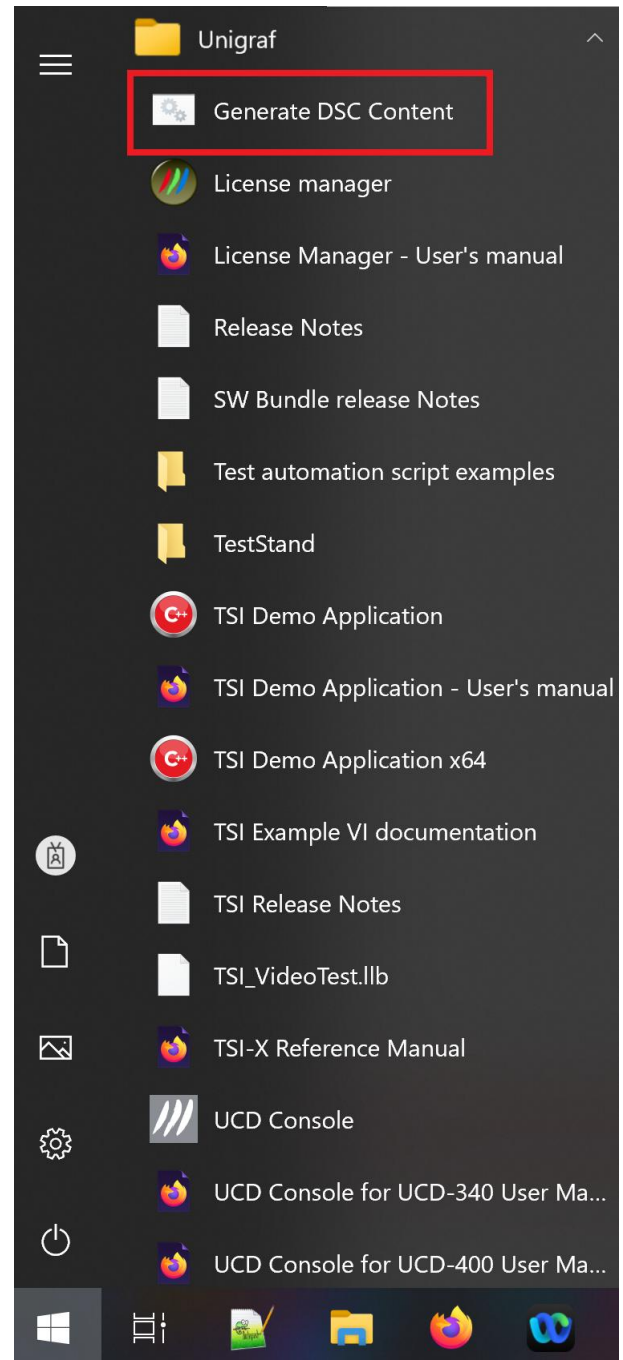
C:\ProgramData\Unigraf\DSC_content_library

***Note:**

There is a setting to **Automatically create missing contents** but this only works for Sink DUT testing.

****Note:**

This process may take over 24 hours to finish generating DSC content for all timings.



8.2.3.3 Prompts for Video and Audio configuration

Modify Pattern and Audio Generator to match prompt's video timing request as close as possible.

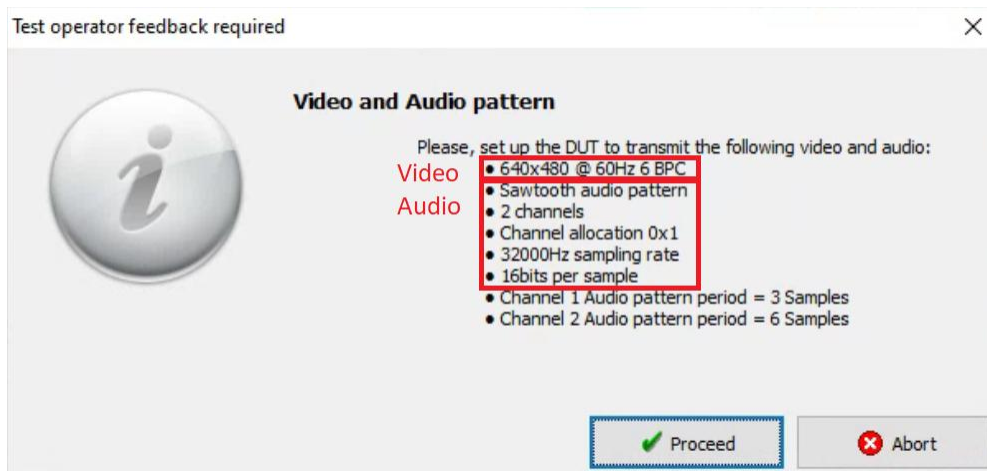


Figure 8.52: Video and Audio Timing Request, circled are details to match in Pattern and Audio Generator

8.2.3.4 Using UCD-500 as Reference Source for DSC

When testing with UCD-500 as Source, DSC must be selected (*Figure 18c*) in the Pattern Generator tab and video output needs to match the prompt.

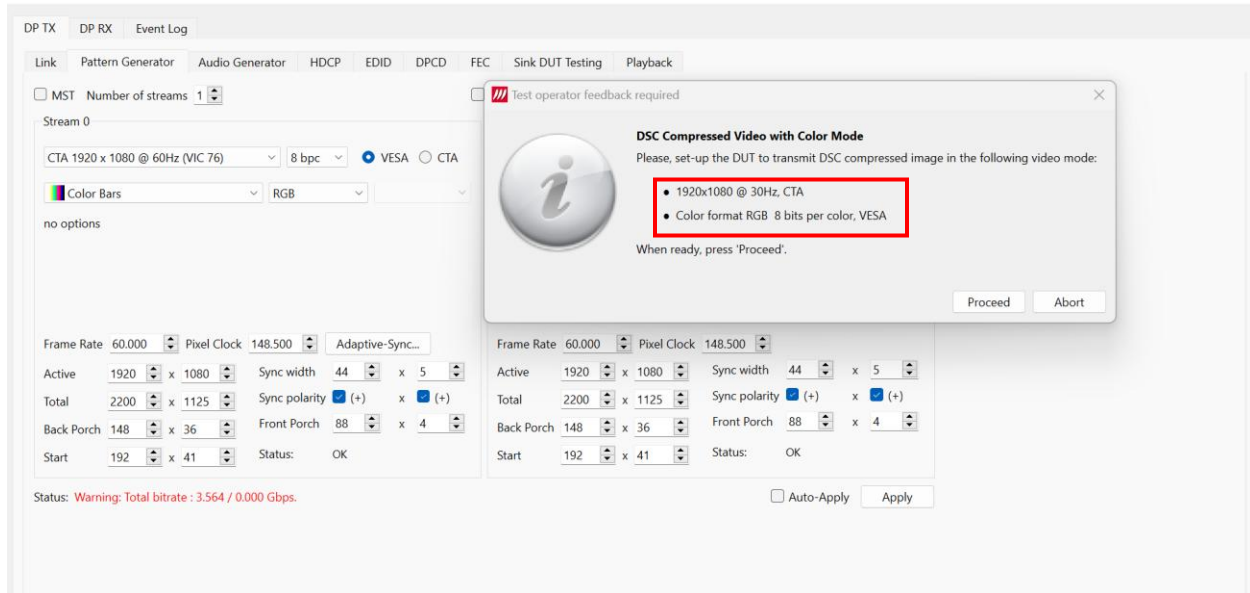


Figure 8.53 – Being asked for DSC image.

Testing with UCD-500 as Source will also require user to load a custom image using the generated DSC content from [step D5 of this document](#). Choose **DSC Image** then **select** the picture then **Apply**.

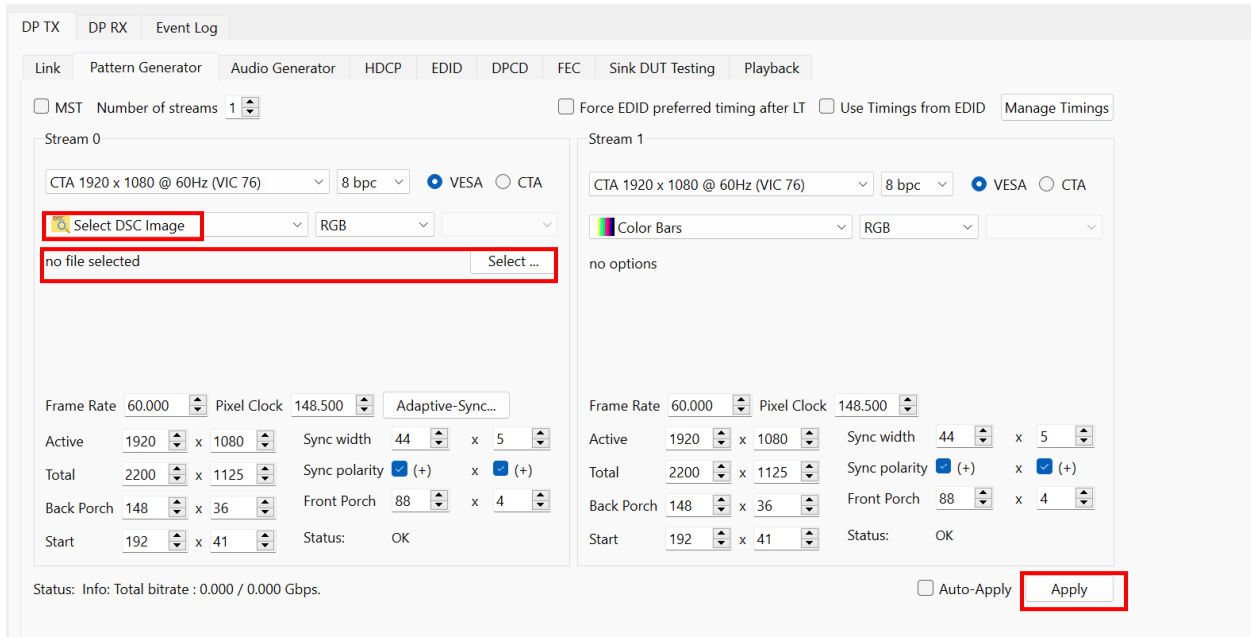


Figure 8.54: Uploading up the DSC Image

Choose the DSC file that matches the resolution, coloring, and bits-per-color.

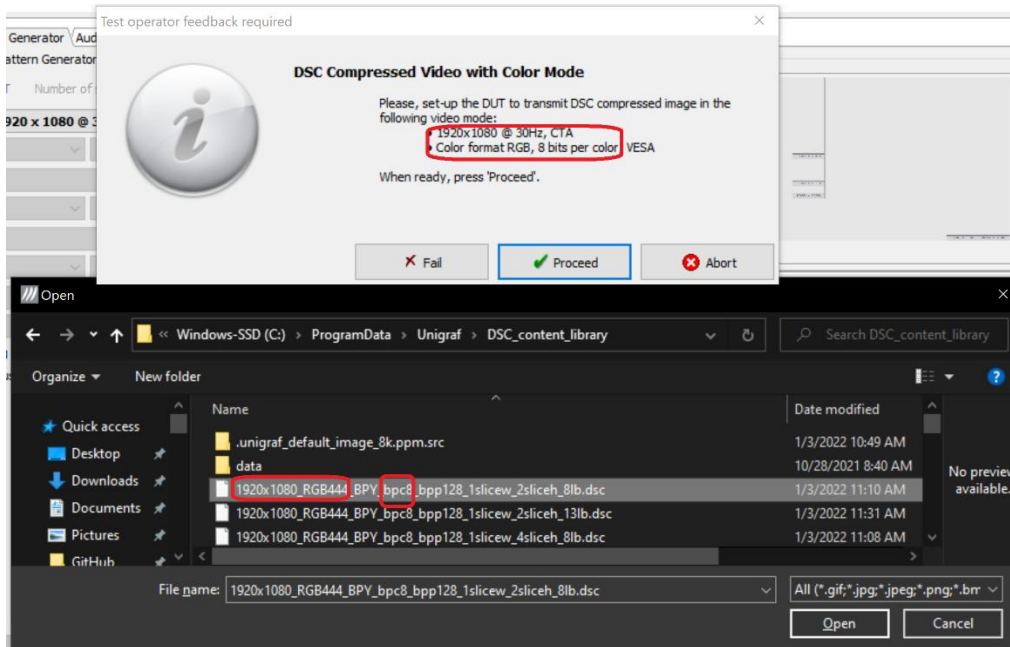


Figure 8.55: How to Check Which Image to Choose From the DSC Images list.

8.2.3.5 Using UCD-500 as Reference Sink for DSC

When using UCD-500 as Reference Sink for DSC testing, DSC capable needs to be enabled:

DP Rx >>> DSC >>> Enable DSC Capable

Click **Apply** after checking off this option to apply default capabilities.

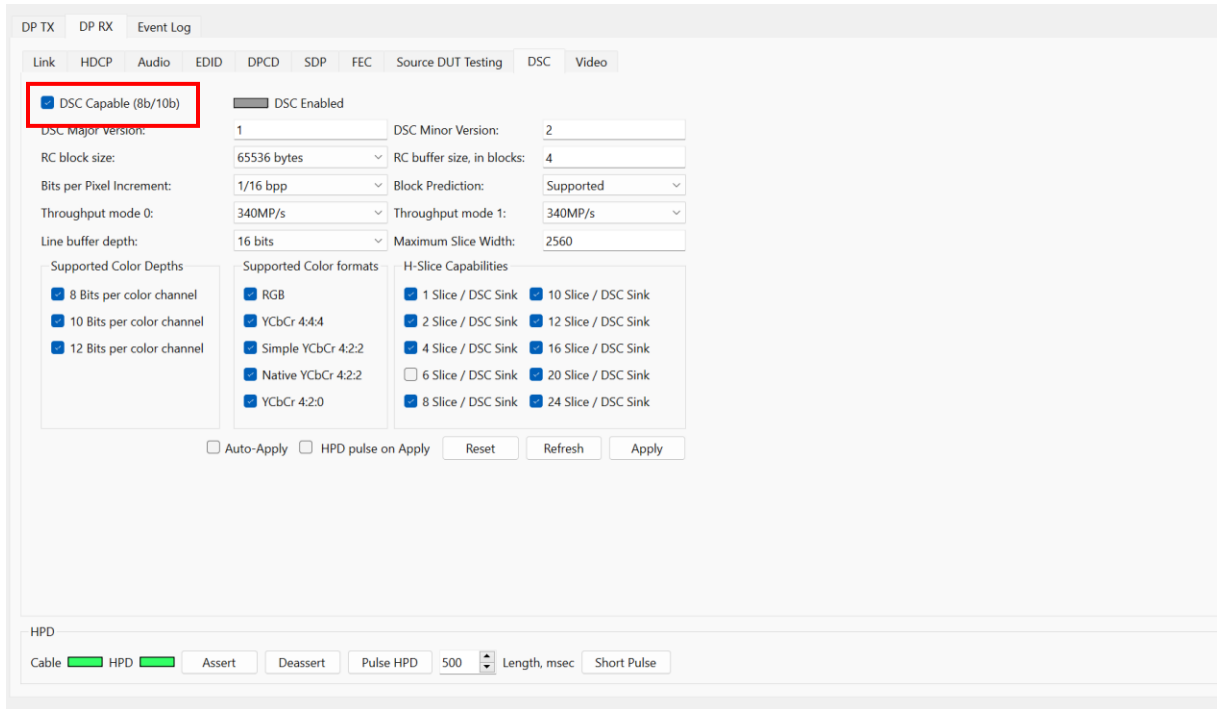


Figure 8.56: UCD-500, DSC Capable set on DP Rx

Visual Inspection

When UCD Console asks for video test pattern result, user must take a snapshot of the video stream and inspect the captured frame. If the snapshot looks as expected with no distortions, then it passes.

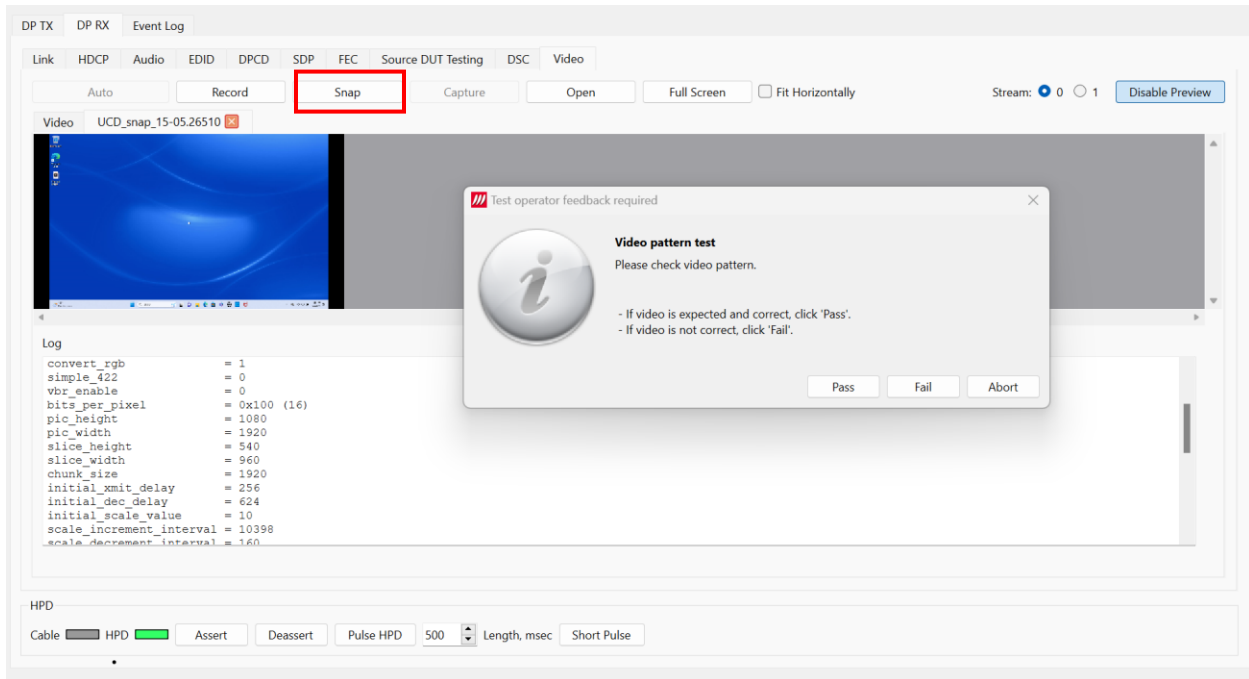


Figure 8.57: UCD-500 DP Rx - Visual inspection for DSC

8.2.4 M42DE Testing Solution

8.2.4.1 Settings for M42DE as Reference Source

HDCP 2.3 Authentication Enabled - Source

This is only needed for HDCP testing. HDCP needs to be enabled on both the sink and source. To enable it for the M42DE when it's the source, go to Generator → Tools → HDCP Test, then choose the HDCP mode. Note, you must check the Authenticated and make sure it's a "YES." If it's a no that means HDCP is not enabled on the sink.

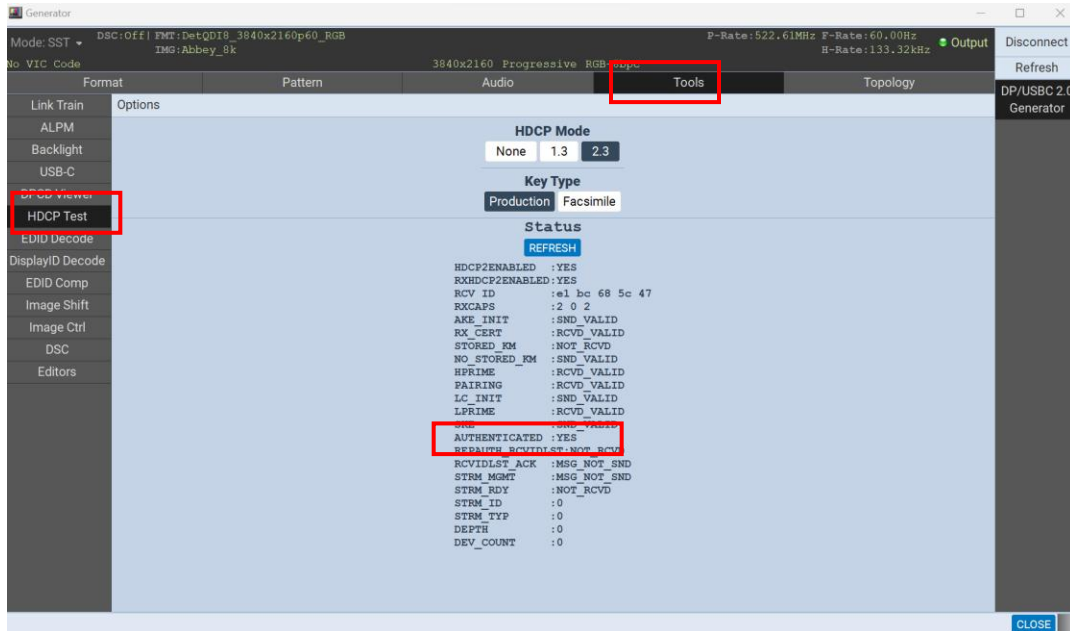


Figure 8.58: M42d Source – HDCP enabled.

8.2.4.2 Settings for M42DE as Reference Sink

HDCP 2.3 Authentication Enabled - Sink

This is only needed for HDCP testing. HDCP needs to be enabled on both the sink and source. To enable it for the M42DE when it's the sink, go to Receiver → HDCP Test, then choose the HDCP mode. Note, you must check the Authenticated and make sure it's a "YES." If it's a no that means HDCP is not enabled on the source.

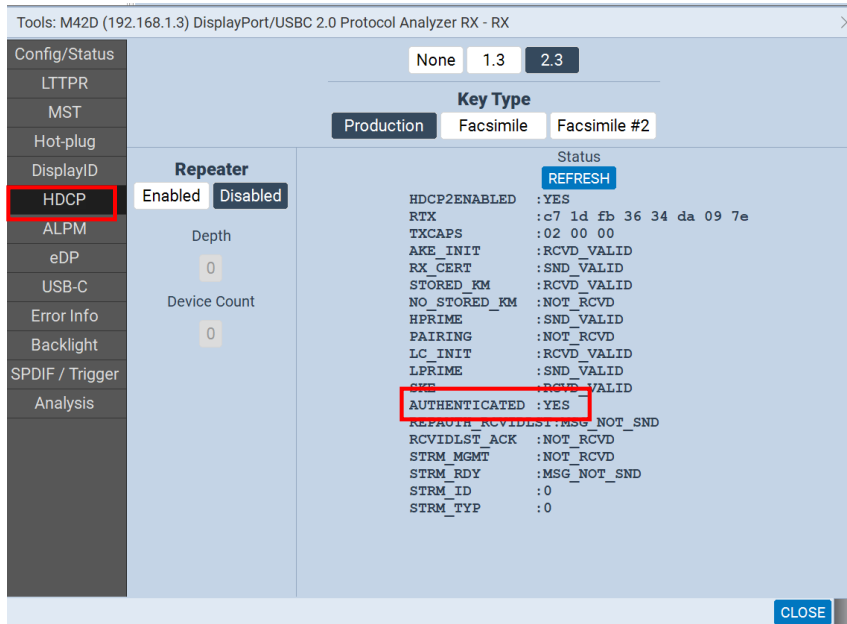


Figure 8.59: M42de Sink – HDCP enabled.

8.2.4.3 Load Source CDF

For accurate testing results for DisplayPort CTS Assertions, it is important to input the capabilities of the source being tested. Vendors should provide a Capabilities Declaration Form (CDF) file to test with M42DE.

All CDF files should be located at the following directory for accessibility through M42DE:

For Source Tests: C:\QuantumData\workspace\dp14srcct\cdf

For Sink Test: C:\QuantumData\workspace\dp14sinkct\cdf

You can load this file in M42DE by going to "CDF Entry tab." Press on the Open and choose which CDF you want to load.

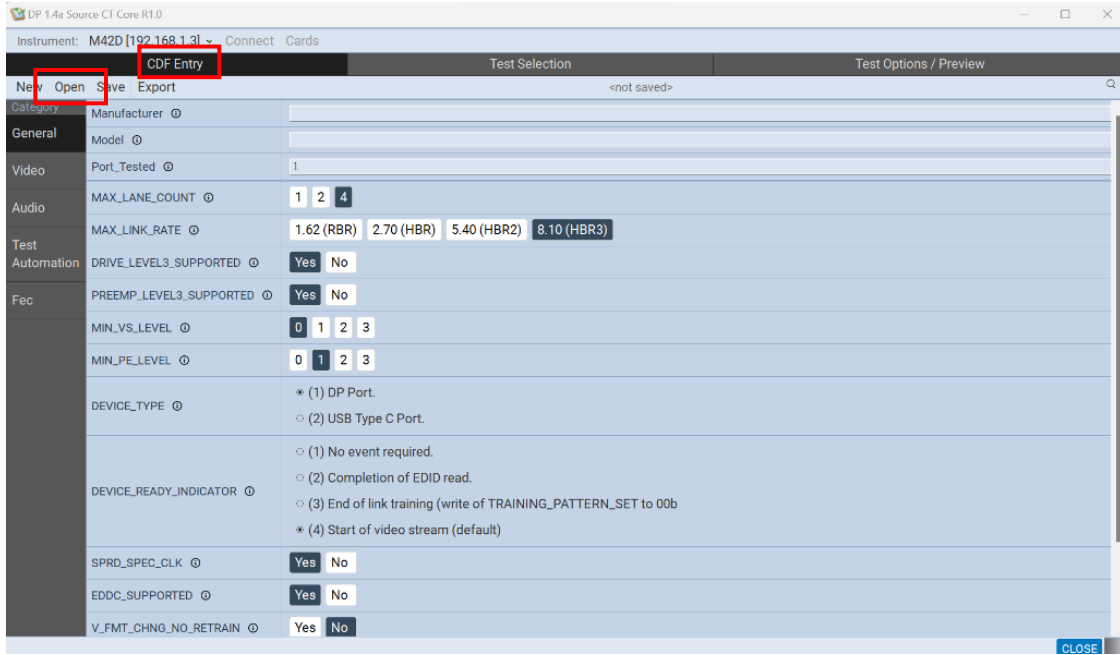


Figure 8.60: M42DE Upload CDF

8.3 Troubleshooting CV System (MR system)

8.3.1 BIOS Configuration for USB4 CV System

CV System has a special BIOS configuration for testing with USB4 Command Verifier. If there is any concern that BIOS has been modified, users can check that their BIOS settings match the following table:

Advanced\Platform Misc Configuration	
PCI Express Native Power Management	Enabled
Native ASPM	Auto
DMI Link ASPM Control	Auto
ASPM	L1
L1 Substates	L1.1 & L1.2
DMI Gen3 ASPM Control	ASPM L1
DMI Gen3 ASPM	ASPM L1

Advanced\System Agent Configuration\Graphics Configuration	
iGPU Multi-Monitor	Enabled

Advanced\Thunderbolt(TM) Configuration	
Discrete Thunderbolt(TM) Support	Enabled

Advanced\Thunderbolt(TM) Configuration\Discrete Thunderbolt(TM) Configuration	
Thunderbolt Boot Support	Enabled
GPIO filter	Disabled

Advanced\Thunderbolt(TM) Configuration\Discrete Thunderbolt(TM) Configuration\DTBT Controller 0 Configuration	
DTBT Controller 0	Enabled

Advanced\System Agent (SA) Configuration	
Control Iommu Pre-boot Behavior	Enable IOMMU during boot

Can't manually locate the following:	
PCIEX16_3 Bandwidth	X4 Mode

8.3.2 Check Power Connections to USB4 Host Card

CAUTION!

Power off Motherboard before making changes to hardware connections

USB4 CV System is setup to use Port A on USB4 Host Card as PUT.

In reference to the figures below, please make sure the following connections are set up:

USB4 CV System – Motherboard	USB4 Host Card
Onboard DisplayPort	DPSNK1
Onboard USB2.0 Type A Port	USB2 PA
USB2.0 Header – USB+5V Pin	JB1 – IN_5v0
USB2.0 Header – GND	JB1 – GND

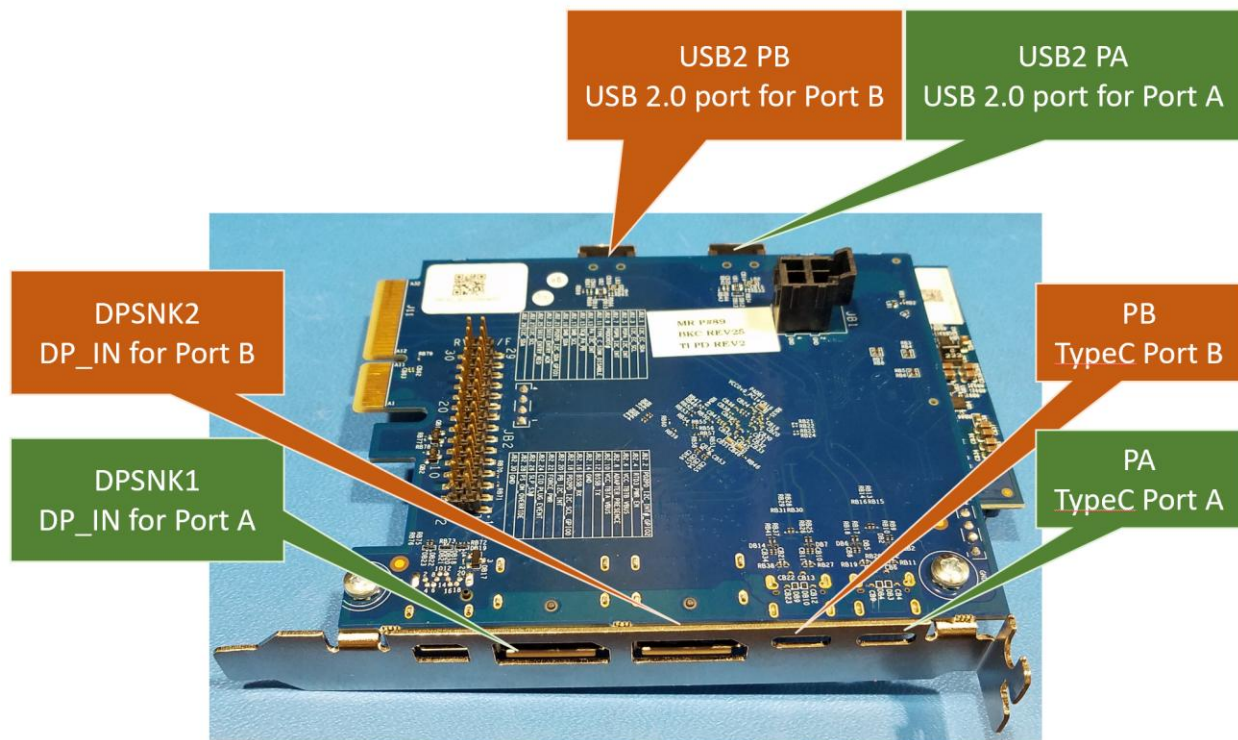


Figure 5.61: USB4 Add-in Host Card - Ports

USB4 Host Card goes
in bottom PCIe Slot

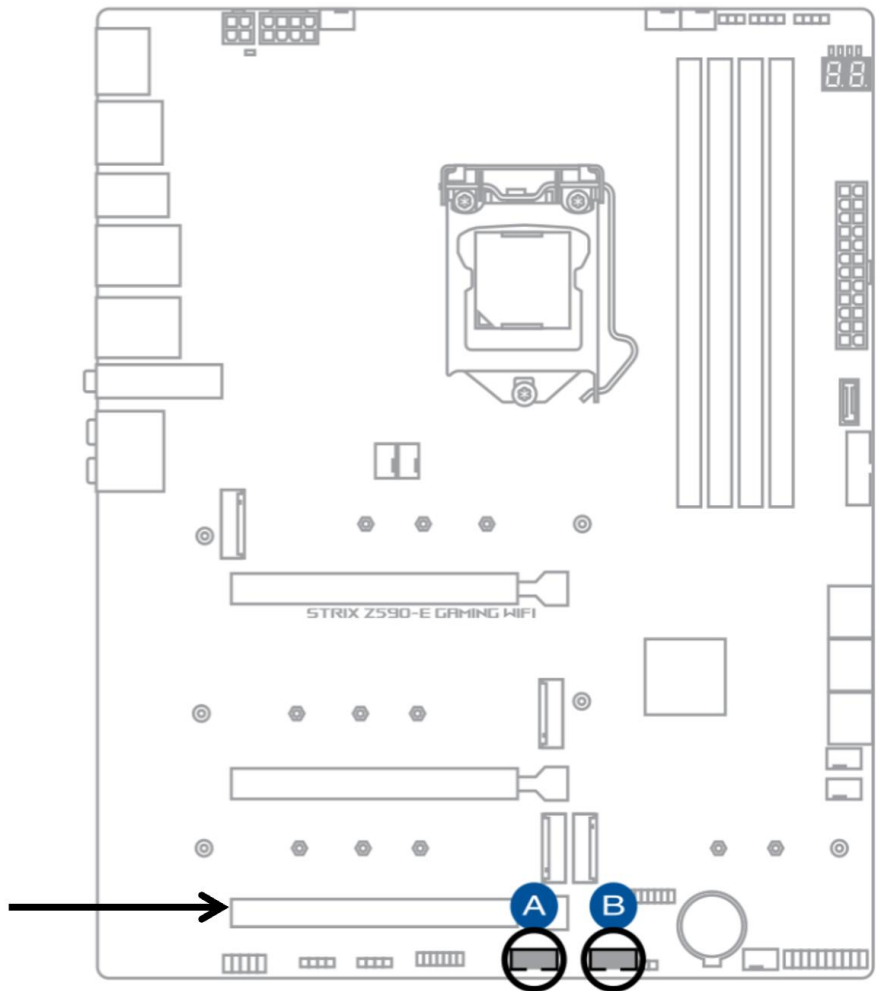


Figure5.62: USB4 CV System Motherboard Diagram

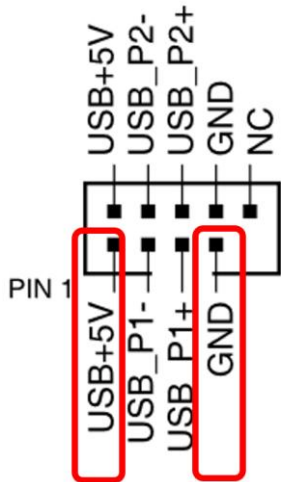


Figure 5.63: USB4 CV System Motherboard - USB2.0 Headers

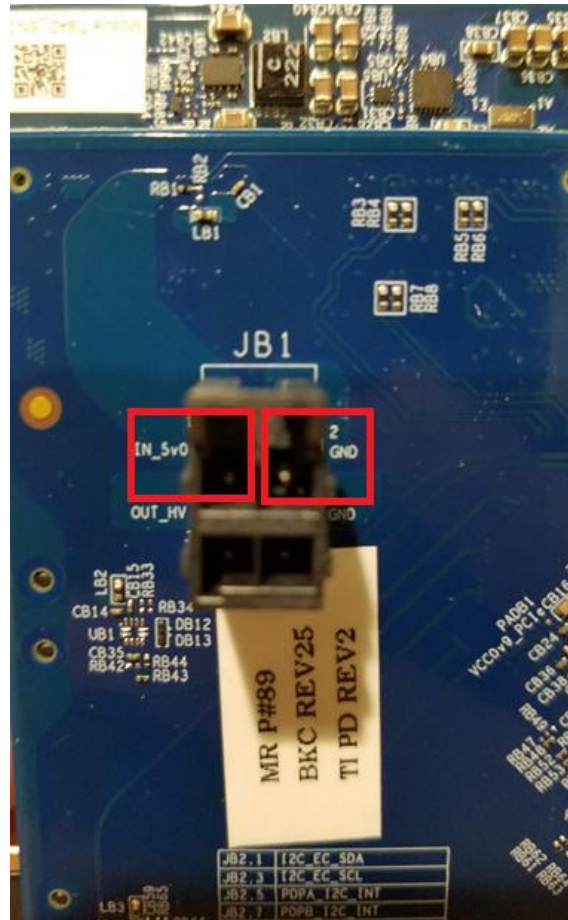


Figure 1.64: USB4 Add-in Host Card - Power Pinout JB1

8.4 Chapter 7 Linux Host

To run chapter 7 test on the Linux system, two computers are needed. One is the UUT (Linux), and the second computer is the remote Window machine (USB4 Win-x64 system). Make sure both systems have the same USB4 CV version, and that the Windows machine has the LeCroy USB Protocol Suite.

8.4.1 Set Up Remote Windows Machine

8.4.1.1 Set Up the proFPGA.

Refer to [Time Measurement Equipment - Setup Procedure](#) above to set up the proFPGA on remote Window machine (USB4 Win-x64 system).

8.4.1.2 Set up IP Address on remote Windows machine

Refer to [Set up IP Address on Window](#) above.

8.4.1.3 Install USB4CV and Protocol Suite

Make sure that USB4CV version on the remote Windows machine is the same version as USB4CV on the Linux system

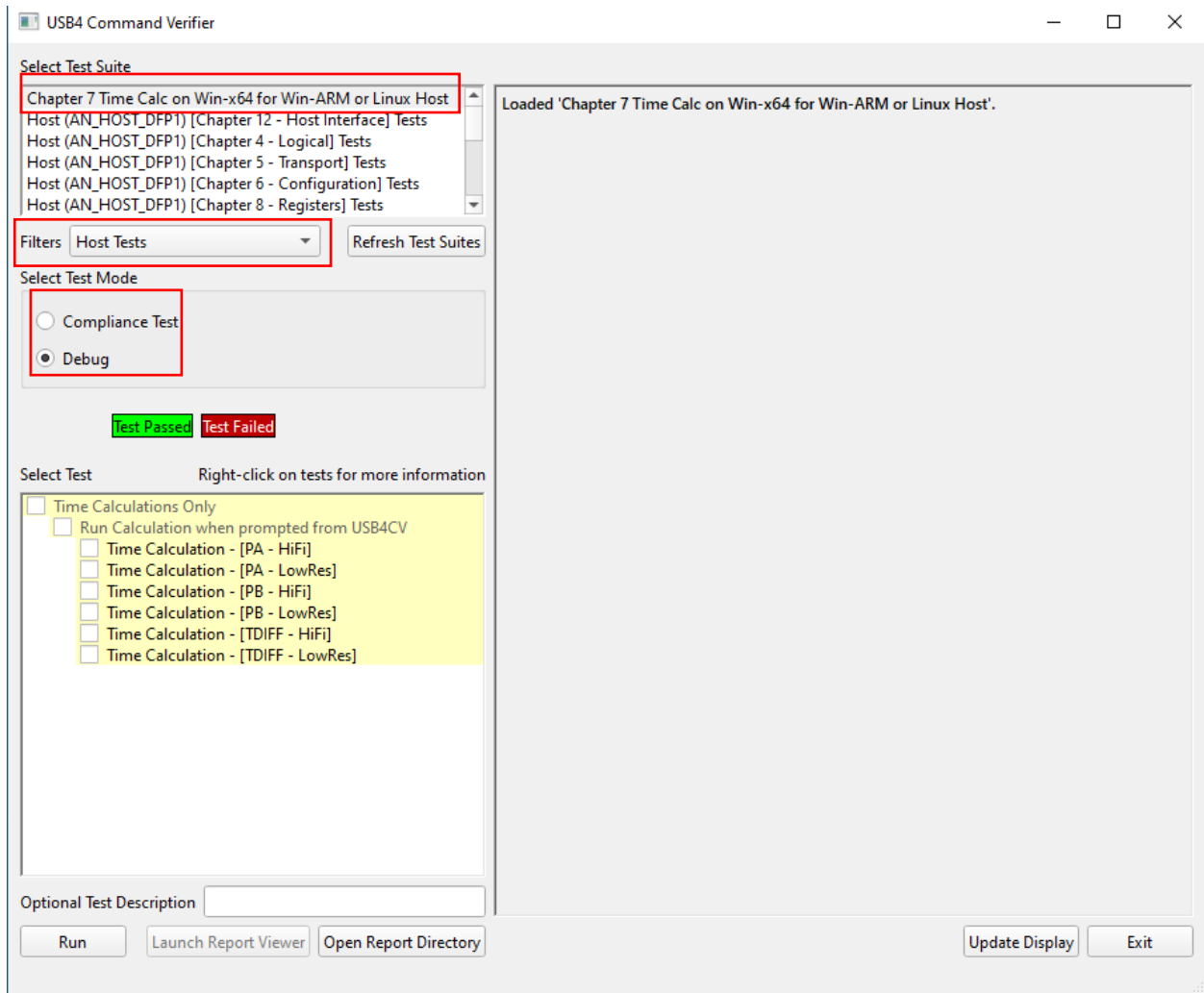
Also, install the appropriate USB Protocol Suite version for the USB4CV version. See [ReleaseNotes_USB4CV.txt](#) for the correct LeCroy Protocol Suite version.

8.4.1.4 Start USB Protocol Suite

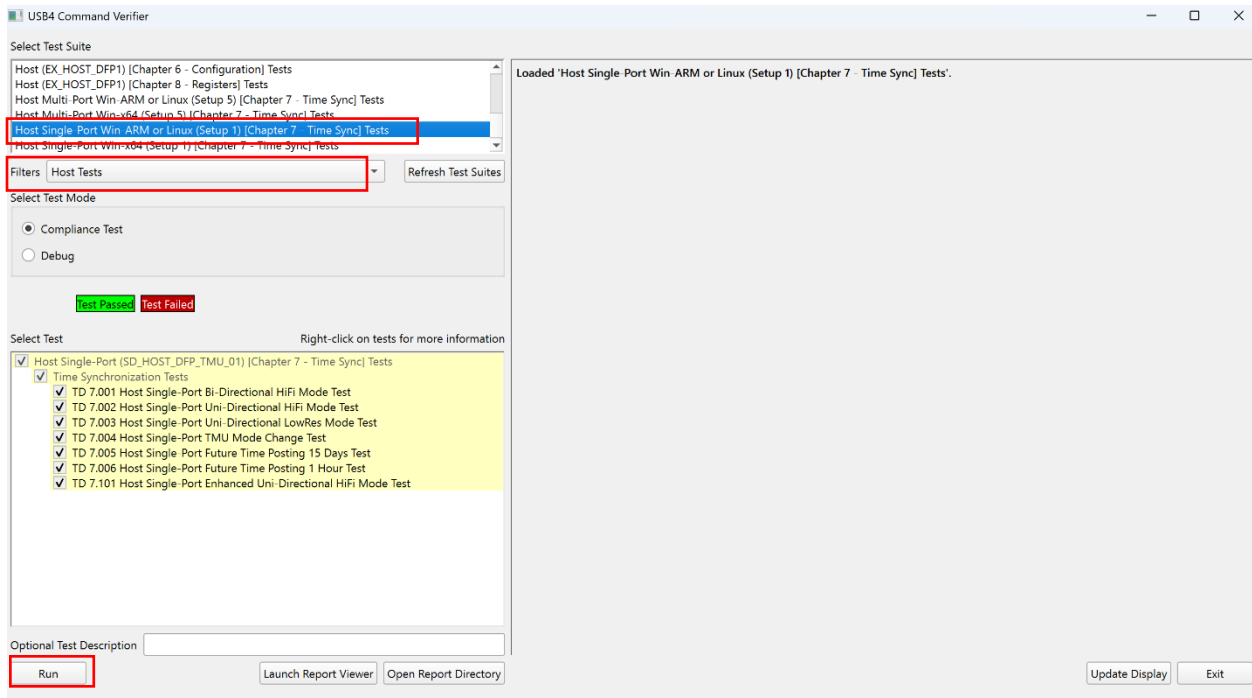
Refer to [Starting LeCroy on Windows Machine](#) above to start the USB Protocol Suite on the remote Windows machine.

8.4.2 Running The Test

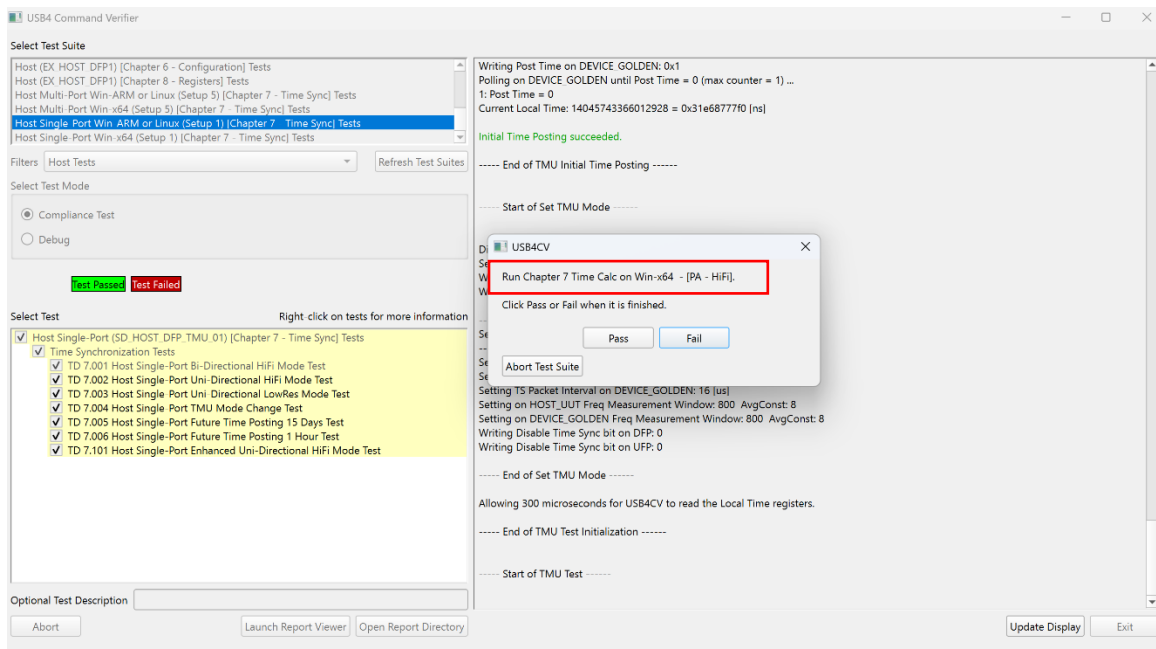
Now on the remote Windows machine (Win-x64 machine), open USB4 CV and filter the tests by Host tests and make sure Debug is selected in the test mode.



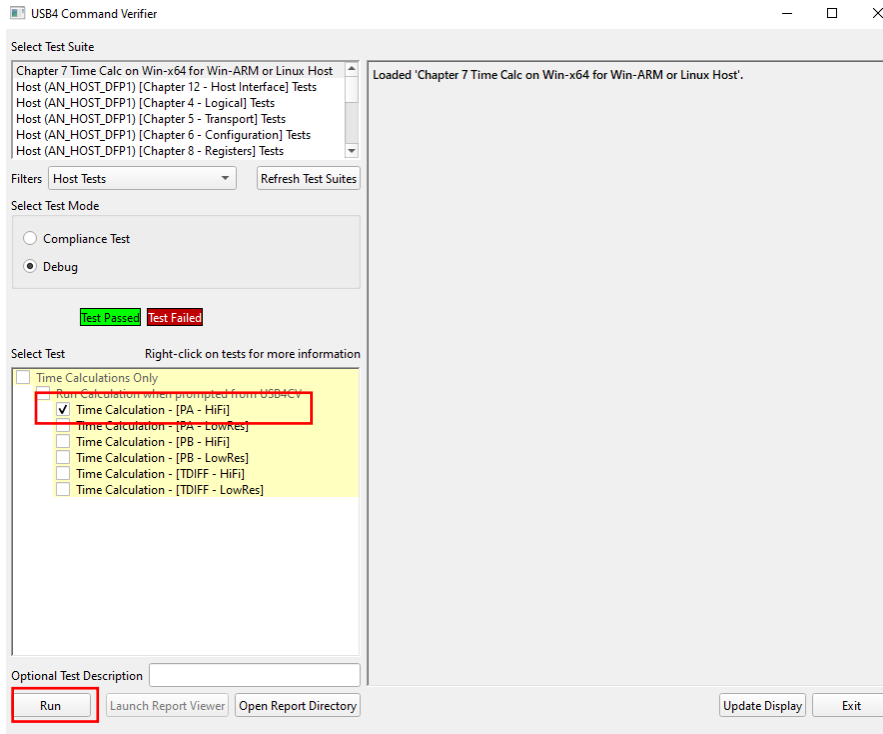
On the UUT system, make sure the setup for chapter 7 is connected to the UUT. Open USB4 CV and filter the tests by Host. Select *Host Single-Port Win ARM or Linux (setup 1)* if the UUT has one router for one port. Press Run



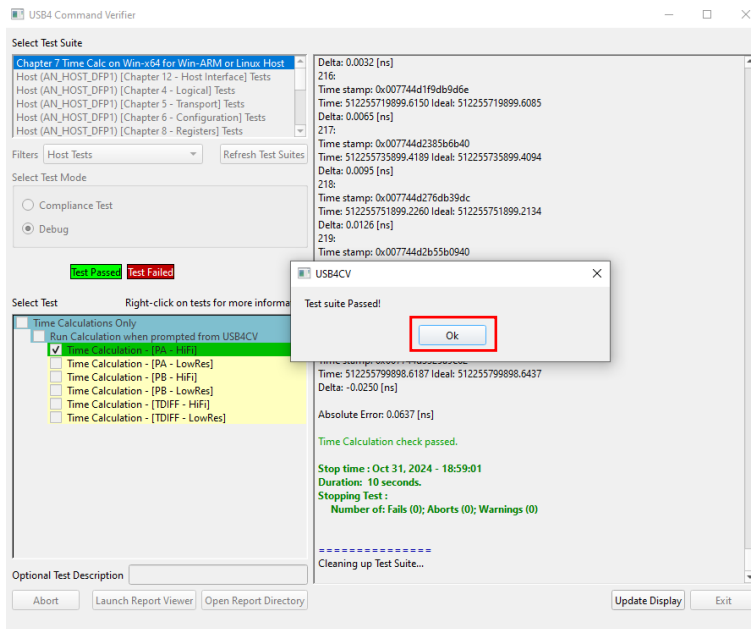
Once you press run in the UUT system, a pop-up window will appear while the test is running. The pop-up window will ask the user to run a test in the remote Windows machine (USB4 Win-x64 system)



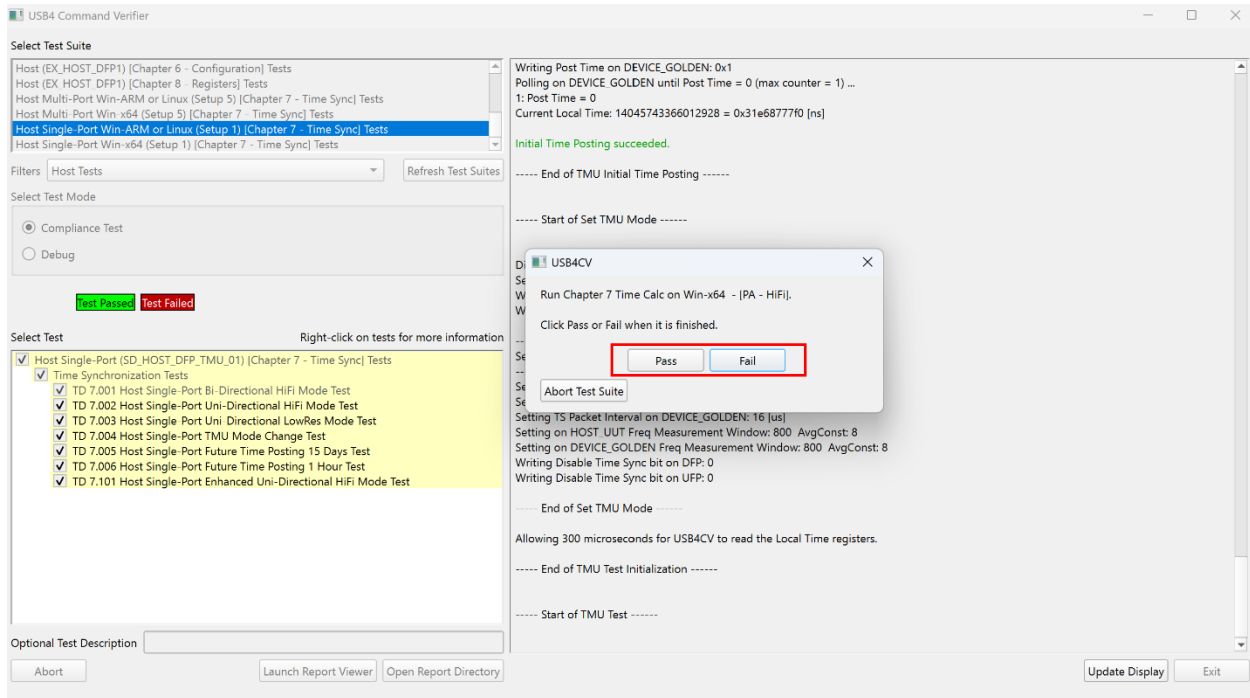
Now go back to the remote Windows machine (USB4 Win-x64 system) and look for the test in the UUT system pop-up window. Press Run



Wait for the test to finish in the remote Windows machine (USB4 Win-x64 system) first. Then press OK



Go back to the UUT system, and press pass if the test passes in the remote Windows machine (USB4 Win-x64 system). Otherwise, press fail



In some test cases, the user must run the Time calculation test in the remote Windows machine (USB4 Win-x64 system) multiple times.

Once all tests are finished in the UUT system, save all logs and traces in both the UUT system and the remote Windows machine (USB4 x64 system). The user will need to submit both results.

Any other USB connections between hosts/devices/hubs setup that are not going to the analyzer should use a 0.8m Accell USB4 passive cable.

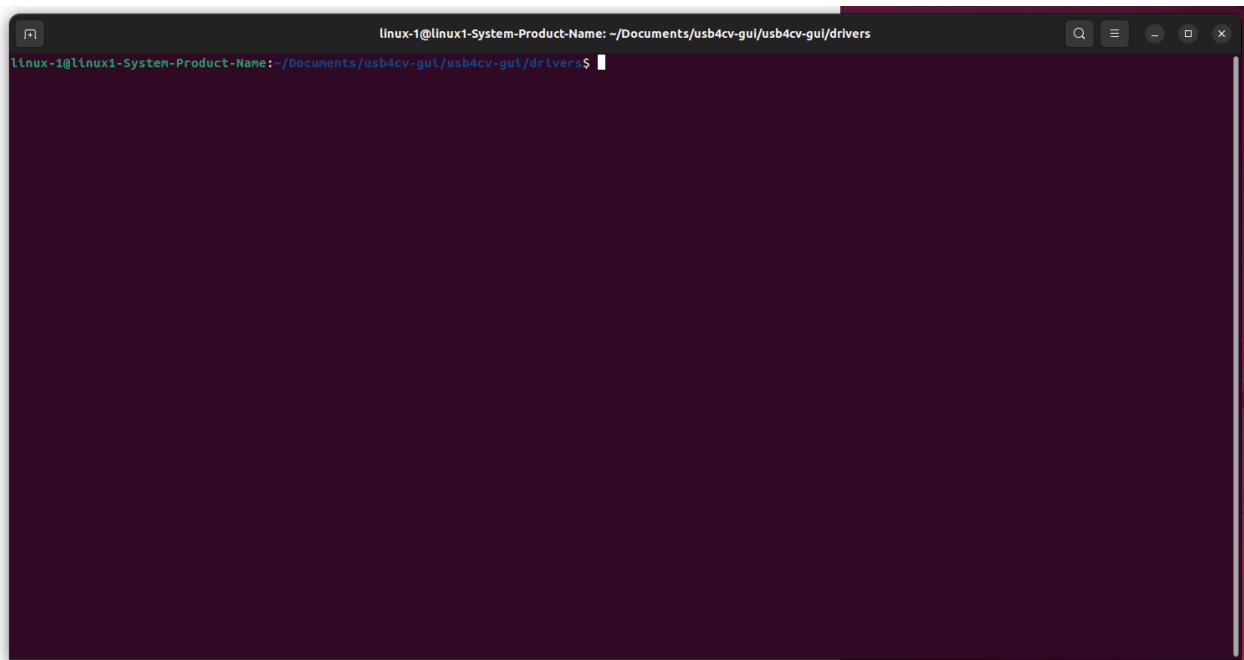
8.5 Linux usb4_hi Driver

8.5.1 Load usb4_hi Driver Manually

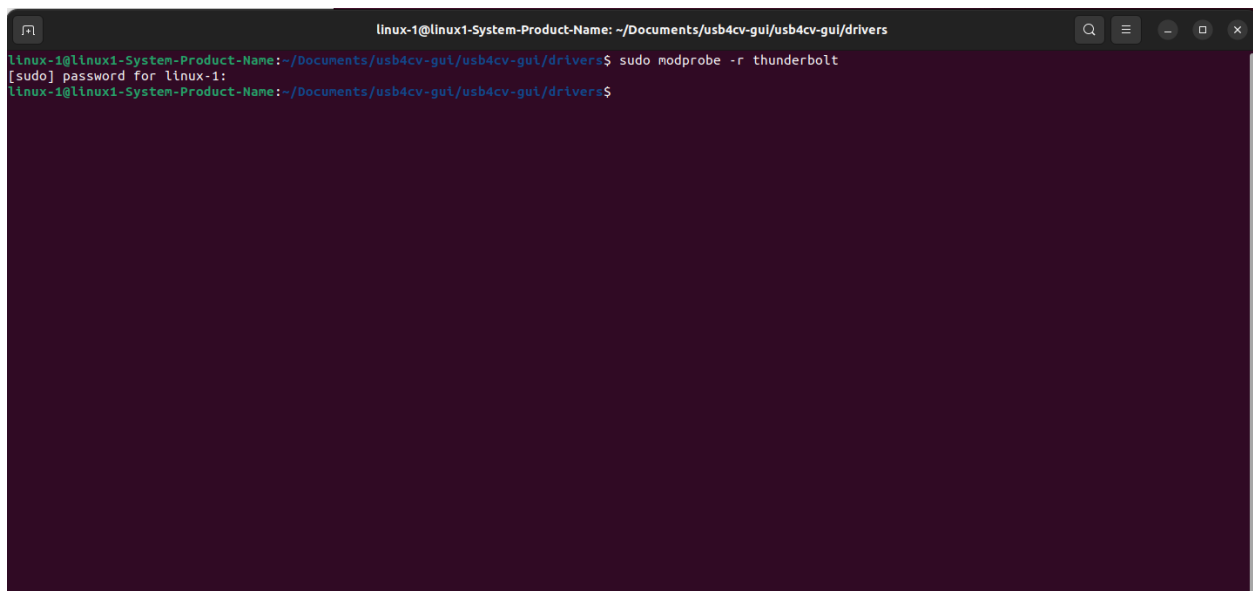
In the usb4cv-gui directory locate the driver file

Or

In the usb4cv-cli directory locate the driver file



`sudo modprobe -r thunderbolt`



sudo modprobe usb4_hi

```
linux-1@linux1-System-Product-Name: ~/Documents/usb4cv-gui/usb4cv-gui/drivers
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-gui/usb4cv-gui/drivers$ sudo modprobe -r thunderbolt
[sudo] password for linux-1:
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-gui/usb4cv-gui/drivers$ sudo modprobe usb4_hi
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-gui/usb4cv-gui/drivers$
```

sudo dmesg | tail

```
linux-1@linux1-System-Product-Name: ~/Documents/usb4cv-gui/usb4cv-gui/drivers
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-gui/usb4cv-gui/drivers$ sudo modprobe -r thunderbolt
[sudo] password for linux-1:
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-gui/usb4cv-gui/drivers$ sudo modprobe usb4_hi
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-gui/usb4cv-gui/drivers$ sudo dmesg | tail
[ 27.363481] audit: type=1400 audit(1730453280.280:63): apparmor="DENIED" operation="capable" class="cap" profile="/usr/lib/snapd/snap-confine" pid=2060 com
m="snap-confine" capability=12 capname="net_admin"
[ 27.363487] audit: type=1400 audit(1730453280.280:64): apparmor="DENIED" operation="capable" class="cap" profile="/usr/lib/snapd/snap-confine" pid=2060 com
m="snap-confine" capability=38 capname="perfmon"
[ 263.346369] xhci_hcd 0000:37:00.0: xHC error in resume, USBSTS 0x401, Reinit
[ 263.346372] usb usb3: root hub lost power or was reset
[ 263.346373] usb usb4: root hub lost power or was reset
[ 263.353685] ACPI: bus type thunderbolt unregistered
[ 303.634921] usb4_hi: loading out-of-tree module taints kernel.
[ 303.634924] usb4_hi: module verification failed: signature and/or required key missing - tainting kernel
[ 303.635398] usb4_hi 0000:03:00.0: 12 paths supported
[ 303.635499] usb4_hi 0000:03:00.0: driver initialized
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-gui/usb4cv-gui/drivers$
```

Note, the number of paths supported is based on the host.

8.5.2 Re-build usb4_hi Driver

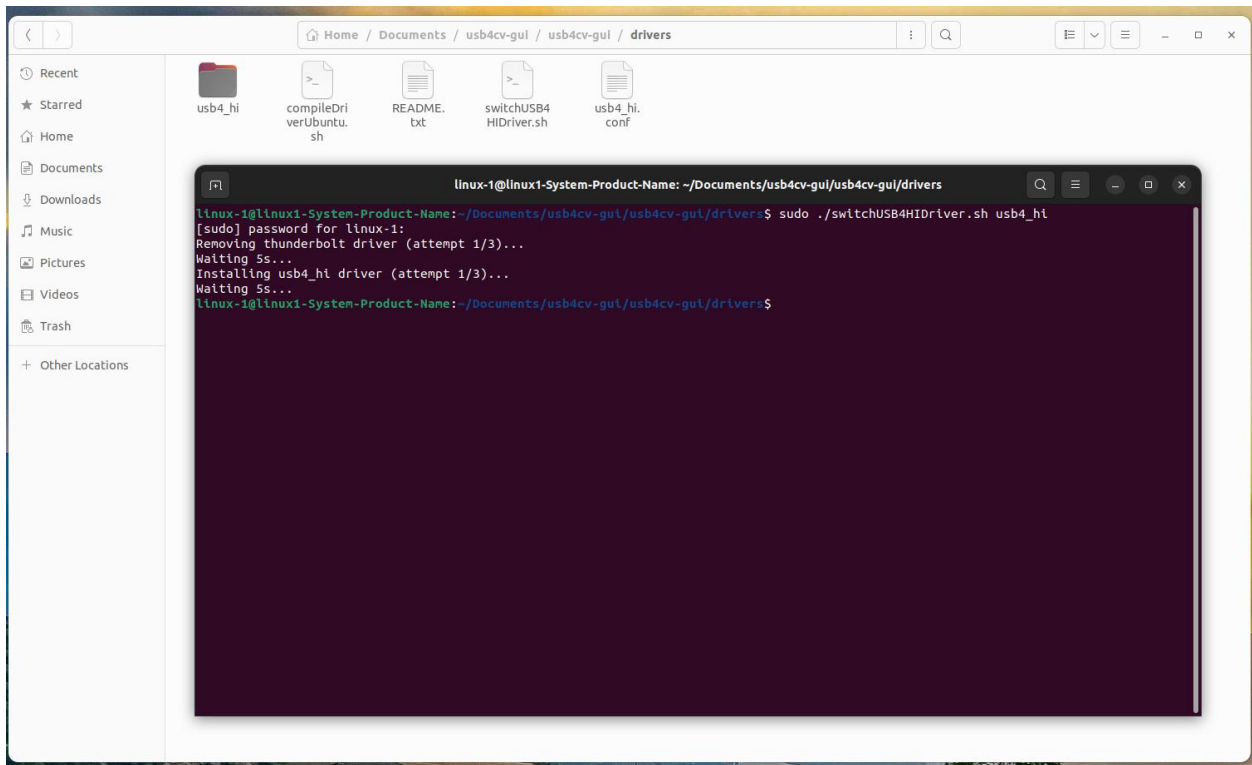
If the kernel is updated, then the user must re-build and reinstall the driver.

For cv-gui users: Go to `~/Documents/usb4cv-gui/usb4cv-gui/driver`

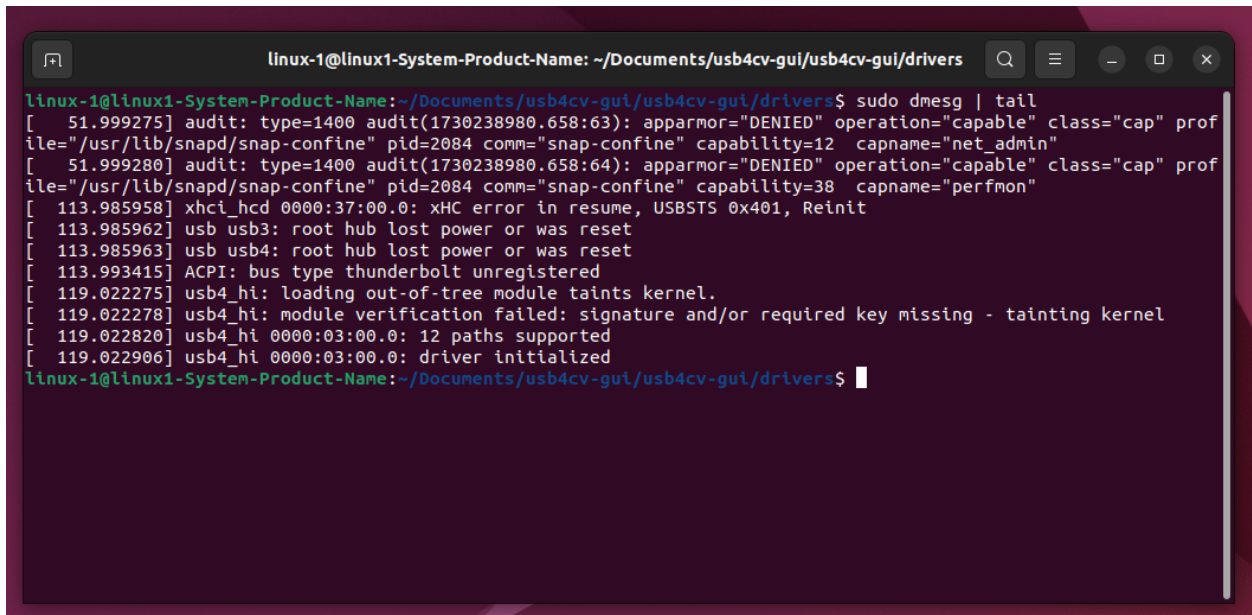
Run `sudo ./compileDriverUbuntu.sh`

```
linux-1@linux1-System-Product-Name: ~/Documents/usb4cv-gui/usb4cv-gui/drivers
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-gui/usb4cv-gui/driver$ sudo ./compileDriverUbuntu.sh
make -C /lib/modules/6.8.0-45-generic/build M=/home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi clean
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-45-generic'
CLEAN /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-45-generic'
make -C /lib/modules/6.8.0-45-generic/build M=/home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi modules
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-45-generic'
warning: the compiler differs from the one used to build the kernel
The kernel was built by: x86_64-linux-gnu-gcc-12 (Ubuntu 12.3.0-1ubuntu1-22.04) 12.3.0
You are using: gcc-12 (Ubuntu 12.3.0-1ubuntu1-22.04) 12.3.0
CC [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi-core.o
CC [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi-chardev.o
CC [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi-ioctl.o
CC [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi-pci.o
LD [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi.o
MODPOST /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/Module.symvers
CC [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi.mod.o
LD /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi.ko
BTF [M] /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi.ko
Skipping BTF generation for /home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi/usb4_hi.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-45-generic'
make -C /lib/modules/6.8.0-45-generic/build M=/home/linux-1/Documents/usb4cv-gui/usb4cv-gui/drivers/usb4_hi modules_install
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-45-generic'
INSTALL /lib/modules/6.8.0-45-generic/updates/usb4_hi.ko
SIGN /lib/modules/6.8.0-45-generic/updates/usb4_hi.ko
DEPMOD /lib/modules/6.8.0-45-generic
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-45-generic'
Driver installed.
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-gui/usb4cv-gui/drivers$
```

Load the usb4_hi driver



Make sure usb4_hi is loaded



For cv-cli users: Go to ~/Documents/usb4cv-cli/usb4cv-cli/drivers

Run sudo ./compileDriverUbuntu.sh

```
linux-1@linux1-System-Product-Name: ~/Documents/usb4cv-cli/usb4cv-cli/drivers
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/drivers$ sudo ./compileDriverUbuntu.sh
[sudo] password for linux-1:
make -C /lib/modules/6.8.0-53-generic/build M=/home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi clean
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-53-generic'
  CLEAN   /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-53-generic'
make -C /lib/modules/6.8.0-53-generic/build M=/home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi modules
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-53-generic'
warning: the compiler differs from the one used to build the kernel
The kernel was built by: x86_64-linux-gnu-gcc-13 (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0
You are using:          gcc-13 (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0
CC [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4-hi-core.o
CC [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4-hi-chardev.o
CC [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4-hi-ioctl.o
CC [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4-hi-pci.o
LD [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4_hi.o
MODPOST /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/Module.symvers
CC [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4_hi.mod.o
LD [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4_hi.ko
BTF [M] /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4_hi.ko
Skipping BTF generation for /home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi/usb4_hi.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-53-generic'
make -C /lib/modules/6.8.0-53-generic/build M=/home/linux-1/Documents/usb4cv-cli/usb4cv-cli/drivers/usb4_hi modules_install
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-53-generic'
  INSTALL /lib/modules/6.8.0-53-generic/updates/usb4_hi.ko
  SIGN    /lib/modules/6.8.0-53-generic/updates/usb4_hi.ko
  DEPMOD  /lib/modules/6.8.0-53-generic
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-53-generic'
Driver installed.
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/drivers$
```

Load the usb4_hi driver

```
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/drivers$ sudo ./switchUSB4HIDriver.sh usb4_hi
[sudo] password for linux-1:
Removing thunderbolt driver (attempt 1/3)...
Waiting 5s...
Installing usb4_hi driver (attempt 1/3)...
Waiting 5s...
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/drivers$
```

Make sure the usb4_hi driver will load

```
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/drivers$
linux-1@linux1-System-Product-Name:~/Documents/usb4cv-cli/usb4cv-cli/drivers$ sudo dmesg | tail
[ 1200.631728] usb usb3: root hub lost power or was reset
[ 1200.631730] usb usb4: root hub lost power or was reset
[ 1200.649814] ACPI: bus type thunderbolt unregistered
[ 1205.683010] usb4_hi: loading out-of-tree module taints kernel.
[ 1205.683014] usb4_hi: module verification failed: signature and/or required key missing - tainting kernel
[ 1205.683573] usb4_hi 0000:03:00.0: 12 paths supported
[ 1205.683662] usb4_hi 0000:03:00.0: driver initialized
```

8.5.3 Driver Signature for usb4_hi Driver

USB4CV communicates directly with the usb4_hi driver. This might require the user to do one of the following:

- Disable secure boot in the BIOS firmware setup screen. We suggest restoring this setting when you are done running tests.
- Sign the usb4_hi driver, and then install the signed driver. Details on this are beyond the scope of this document.