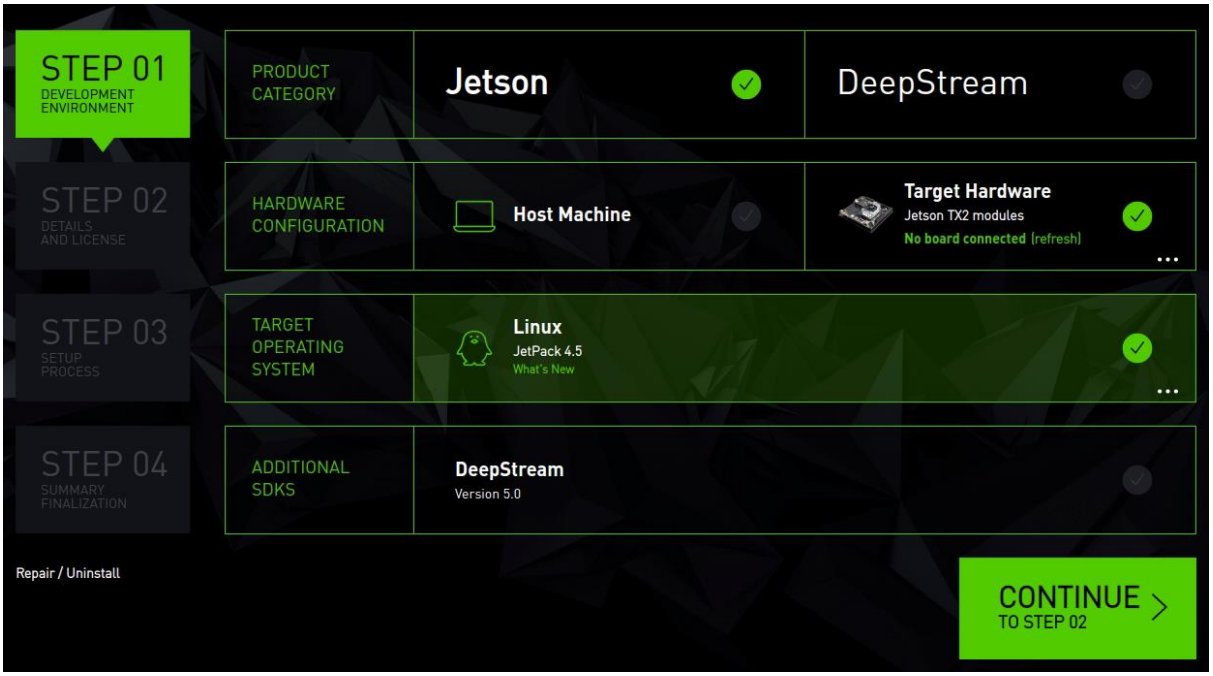
Procedure to Create Custom Image for Testing

Setup Base Resource via NVidia SDK Manager

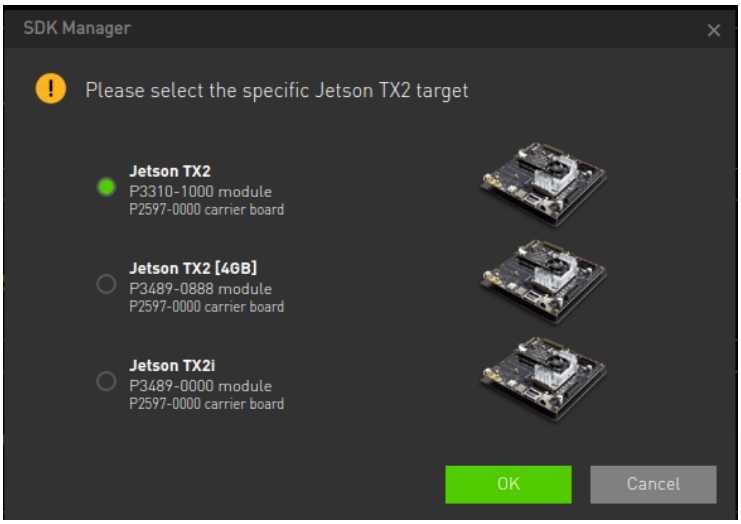
Install the NVidia SDK Manager (reference link).

This is based on CTI's article. For PMG2, we use the following settings:

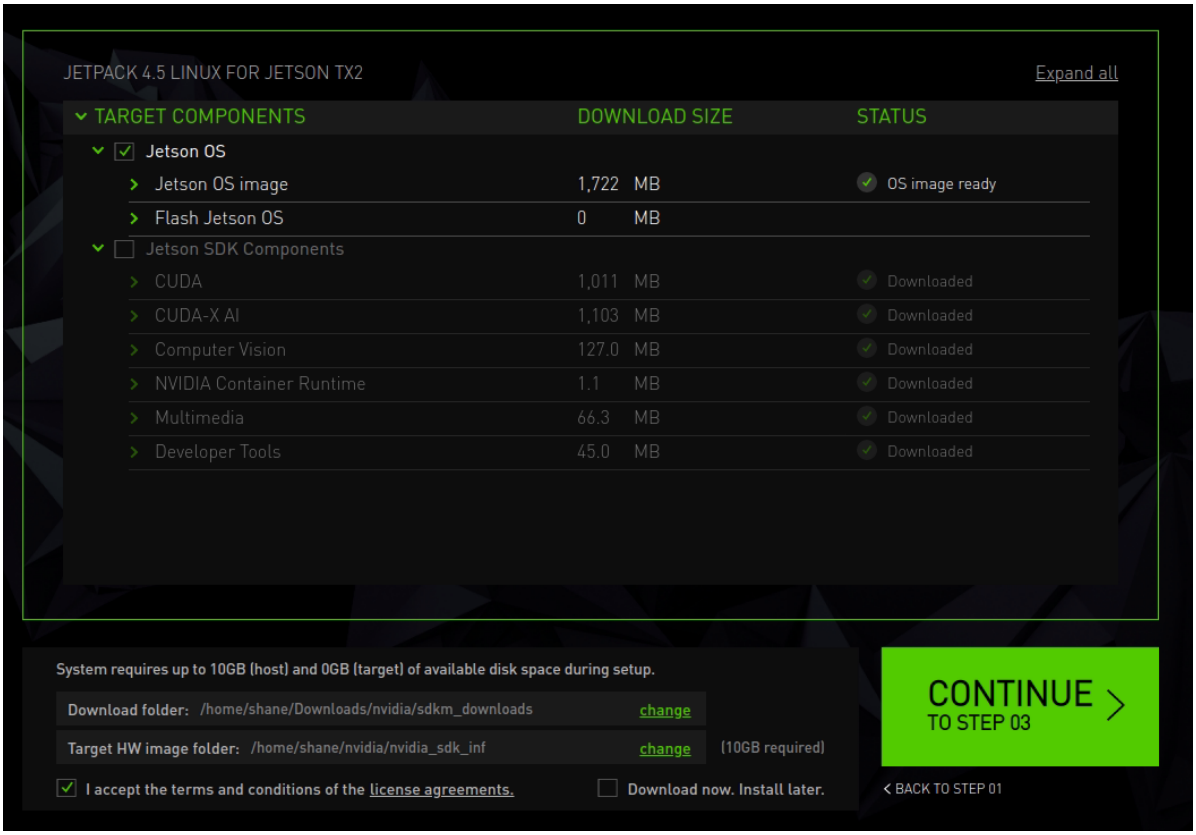
- No host machine config
- TX2 Modules
- Click on Continue



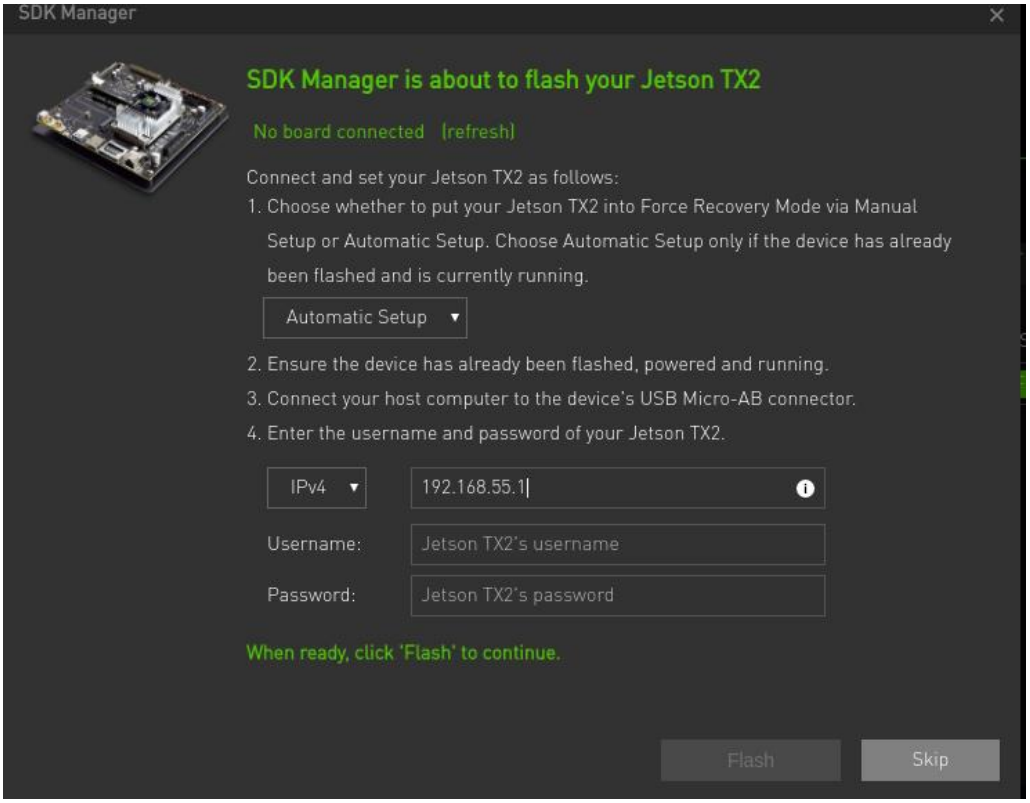
Select the top TX2 version



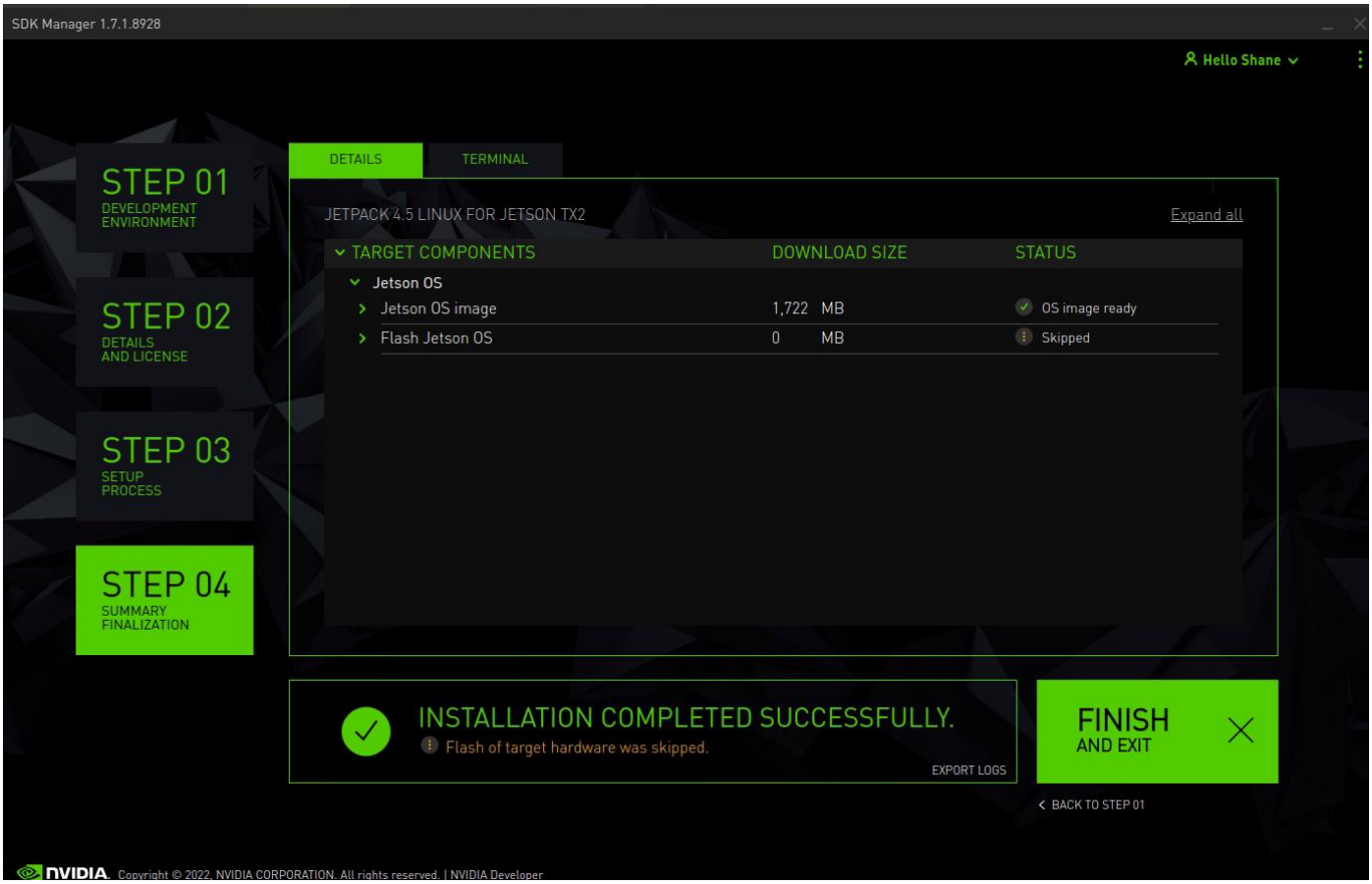
Uncheck “Jetson SDK Component” and only leave the “Jetson OS” category checked. You need to create a folder (target hw image folder) to host the resource to be generated. The example directory shown in screenshot is “/home/shane/nvidia/nvidia_sdk_inf2”. Check the checkbox to accept terms and conditions, and click “Continue” to install.



Once installation is done, SDK manager will request to flash TX2. Skip this step.



Click “Finish and Exit”. Check the “/home/shane/nvidia/nvidia_sdk_inf2” folder to see if it is populated with new contents.



Apply ConnectTech Board Support Package (BSP)

For current PortaMetrics, we uses the Sony IMX 577 camera, and CTI created a BSP with customer driver for us. This section is based on the readme.txt inside the BSP package. Download the package to the specified directory using the following command.

```
cd /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra
wget https://connecttech.com/ftp/dropbox/cti-l4t-src-tx2-32.5-v001-li-imx577.tgz
tar -xzf ./cti-l4t-src-tx2-32.5-v001-li-imx577.tgz
```

Move the contents inside sources/confs up to your Linux_for_Tegra directory with the following command.

```
cd /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra
mv sources/confs/* .
```

Replace the nvidia-l4t-camera and nvidia-l4t-oem-config debian packages inside nv_tegra/l4t_deb_packages/ with the ones in sources/extra. Use the following commands.

```
cd /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra
cp ./sources/extra/nvidia-l4t-camera_32.5.0-20210115151051_arm64.deb ./nv_tegra/l4t_deb_packages/
cp ./sources/extra/nvidia-l4t-oem-config_32.5.0-20210115151051_arm64.deb ./nv_tegra/l4t_deb_packages/
```

Run the following command to update the binaries.

```
sudo ./apply_binaries.sh
```

The ConnectTech BSP's readme requires a few more steps such as building the kernel, but we will not do that until we apply more configuration related to the Wi-Fi driver in sections below.

Apply Wi-Fi driver patch

This is based on reference link to install CYW54591 driver on Jetson Xavier NX. That IC is using PCIe bus and can be detected with lspci command. TX2 is using brcm4354, which cannot be detected by lspci command. The steps below follow the article but has some modifications.

Download the patch from the article using the following commands.

```
cd /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2
wget https://community.infineon.com/gfawx74859/attachments/gfawx74859/KnowledgeBaseArticles/6773/1/patches.zip
```

Extract the patch, which shows modification to tegra_defconfig. This is actually similar the README.txt in the driver package released on 2022-05-11.

```
diff --git a/arch/arm64/configs/tegra_defconfig b/arch/arm64/configs/tegra_defconfig
index 3d34446..ea6746c 100644
--- a/arch/arm64/configs/tegra_defconfig
+++ b/arch/arm64/configs/tegra_defconfig
@@ -544,7 +544,7 @@ CONFIG_WL18XX=m
 CONFIG_WLCORE_SDIO=m
 CONFIG_USB_ZD1201=m
 CONFIG_ZD1211RW=m
-CONFIG_BCMDHD=m
+CONFIG_BCMDHD=n
 CONFIG_BCMDHD_SDIO=y
 CONFIG_BCMDHD_PCIE=y
 CONFIG_BCM4354=y
@@ -1207,3 +1207,7 @@ CONFIG_ARCH_TEGRA_18x_SOC=y
 CONFIG_ARCH_TEGRA_19x_SOC=y
 CONFIG_ARCH_TEGRA_23x_SOC=y
 CONFIG_TEGRA_SAFETY=y
+CONFIG_ASYMMETRIC_KEY_TYPE=y
+CONFIG_ASYMMETRIC_PUBLIC_KEY_SUBTYPE=y
+CONFIG_X509_CERTIFICATE_PARSER=y
+CONFIG_PKCS7_MESSAGE_PARSER=y
```

Since we are using ConnectTech's BSP, we actually will not be using tegra_defconfig, but cti_tegra_defconfig. We will manually change cti_tegra_defconfig, located in "/home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/sources/kernel/kernel-4.9/arch/arm64/configs/cti_tegra_defconfig"

Line 548 (replace)	**CONFIG_BCMDHD=n** CONFIG_BCMDHD_SDIO=y CONFIG_BCMDHD_PCIE=y CONFIG_BCM4354=y CONFIG_BCMDHD_FW_PATH="/lib/firmware/brcm/fw_bcmdhd.bin" CONFIG_BCMDHD_NVRAM_PATH="/lib/firmware/brcm/nvram.txt" CONFIG_BCMDHD_HW_OOB=y CONFIG_DHD_USE_SCHED_SCAN=y

	CONFIG_BCMDHD_DISABLE_MCC=y CONFIG_BCMDHD_PCIE_FW_PATH="/lib/firmware/brcm/fw_bcmdhd_4356.bin" CONFIG_BCMDHD_PCIE_NVRAM_PATH="/lib/firmware/brcm/nvram_4356.txt" CONFIG_BCMDHD_PCIE_ES4_NVRAM_PATH="/lib/firmware/brcm/nvram_murata_4359_b1_es4.txt"
Line 1228-1231 (insert)	**CONFIG_ASYMMETRIC_KEY_TYPE=y** **CONFIG_ASYMMETRIC_PUBLIC_KEY_SUBTYPE=y** **CONFIG_X509_CERTIFICATE_PARSER=y** **CONFIG_PKCS7_MESSAGE_PARSER=y**

Set up the Jetson Linux Toolchain

This is based on reference link. Run the following command:

cd ~
mkdir l4t-gcc && cd l4t-gcc
wget http://releases.linaro.org/components/toolchain/binaries/7.3-2018.05/aarch64-linux-gnu/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz
tar xf gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz

Build the NVidia kernel

This is based on reference link. Run the following command:

cd /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/sources/kernel/kernel-4.9
TEGRA_KERNEL_OUT=outputKernel
export CROSS_COMPILE=$HOME/l4t-gcc/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu/bin/aarch64-linux-gnu-
export LOCALVERSION=-tegra
mkdir -p $TEGRA_KERNEL_OUT
make ARCH=arm64 O=$TEGRA_KERNEL_OUT cti_tegra_defconfig
make ARCH=arm64 O=$TEGRA_KERNEL_OUT -j2

Copy Resources

This is based on the Infineon article. Run the following commands.

cd /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/sources/kernel/kernel-4.9
cp $TEGRA_KERNEL_OUT/arch/arm64/boot/Image /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/kernel/Image
mv /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/kernel/dtb /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/dtb_orig
mkdir /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/kernel/dtb
cp -R $TEGRA_KERNEL_OUT/arch/arm64/boot/dts/* /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/kernel/dtb
sudo make ARCH=arm64 O=$TEGRA_KERNEL_OUT modules_install INSTALL_MOD_PATH=~/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/rootfs/

Apply Binaries Again

This is based on the Infineon article, which indicates running this right after compiling kernel and before copying resources files. Run the following commands. Note: running this will override .config, so the .config replacement needs to be before this command.

```
cd /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra
sudo ./apply_binaries.sh
```

Copy Other Resources

This has to be done after applying the binaries again. Reasons are given in Appendix A.

```
sudo cp $TEGRA_KERNEL_OUT/.config
/home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/rootfs/usr/src/linux-headers-4.9.201-tegra-ubuntu18.04_aarch64/kernel-4.9/.config
```

Add the following line to disable the NetworkManager service controlling the wlan0 interface. Run the following command.

```
sudo gedit
/home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/rootfs/etc/NetworkManager/NetworkManager.conf
```

Append the following two line and a new line at the end. Save the file.

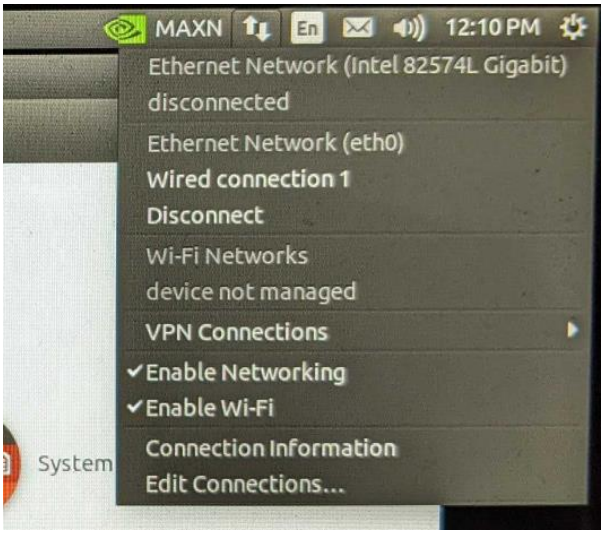
```
[keyfile]
unmanaged-devices=interface-name:wlan0

```

Flash the Image

Put the TX2 into recovery mode and open a terminal in the Linux_for_Tegra directory and run the following command.

```
cd /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra
sudo ./flash.sh cti/tx2/spacely-li-imx577-2cam mmcblk0p1
```

Observation after first bootup. LAN is working. Wi-Fi is enabled but it is not able to see any access points.



If I edit the "/etc/NetworkManager/NetworkManager.conf" file and comment out the last two lines. I will be able to see and also connect to Wi-Fi access points.

Compile FMAC driver

This is based on the reference link. Use the following commands on the TX2 platform (NOT the flash machine)

cd ~
mkdir dev && cd dev
wget https://community.infineon.com/gfawx74859/attachments/gfawx74859/WiFiBluetoothLinux/2383/1/cypress-fmac-v5.10.9-2022_0511.zip
unzip cypress-fmac-v5.10.9-2022_0511.zip
cd cypress-fmac-v5.10.9-2022_0511
tar -xvf cypress-backports-v5.10.9-2022_0511-module-src.tar.gz
cd v5.10.9-backports
export MY_KERNEL=/usr/src/linux-headers-4.9.201-tegra-ubuntu18.04_aarch64/kernel-4.9/
make KLIB=$MY_KERNEL KLIB_BUILD=$MY_KERNEL defconfig-brcmfmac
make KLIB=$MY_KERNEL KLIB_BUILD=$MY_KERNEL modules
mkdir ~/dev/kernel_modules
cp ./compat/compat.ko ~/dev/kernel_modules/
cp ./net/wireless/cfg80211.ko ~/dev/kernel_modules/
cp ./drivers/net/wireless/broadcom/brcm80211/brcmutil/brcmutil.ko ~/dev/kernel_modules/
cp ./drivers/net/wireless/broadcom/brcm80211/brcmfmac/brcmfmac.ko ~/dev/kernel_modules/
sudo systemctl reboot -i

The results will be the following:

```
-rw-rw-r-- 1 mmi mmi 10453808 Jun 30 11:56 brcmfmac.ko
-rw-rw-r-- 1 mmi mmi   675680 Jun 30 11:55 brcmutil.ko
-rw-rw-r-- 1 mmi mmi 14287568 Jun 30 11:55 cfg80211.ko
-rw-rw-r-- 1 mmi mmi  3683632 Jun 30 11:54 compat.ko
```

Add Firmware

This is based on the reference link. Use the following commands on the TX2 platform (NOT the flash machine)

Note that the "/lib/firmware/brcm/nvram.txt" was copied and renamed to "cyfmac4354-sdio.txt".

cd ~/dev/cypress-fmac-v5.10.9-2022_0511/
tar -xvf cypress-firmware-v5.10.9-2022_0511.tar.gz
cd firmware
sudo mkdir -p /lib/firmware/cypress
sudo cp /home/mmi/dev/cypress-fmac-v5.10.9-2022_0511/firmware/cyfmac4354-sdio.bin /lib/firmware/cypress
sudo cp /home/mmi/dev/cypress-fmac-v5.10.9-2022_0511/firmware/cyfmac4354-sdio.clm_blob /lib/firmware/cypress
sudo cp /lib/firmware/brcm/nvram.txt /lib/firmware/cypress
sudo mv /lib/firmware/cypress/nvram.txt /lib/firmware/cypress/cyfmac4354-sdio.txt

Load Kernel Modules

This is based on the reference link. Use the following commands on the TX2 platform (NOT the flash machine).

It was noted that the bcmdhd kernel module needs to be removed before removing cfg80211. This was not mentioned in the Infineon article.

The insmod for brcmfmac.ko was slightly modified (with bolded text) based on Infineon_Wi-Fi_Software_UserGuide.pdf. The reason is to provide the value to the newly created firmware location.

```
cd ~/dev/kernel_modules
sudo rmmod bcmdhd
sudo rmmod cfg80211.ko
sudo insmod compat.ko
sudo insmod cfg80211.ko
sudo insmod brcmutil.ko
sudo insmod brcmfmac.ko alternative_fw_path=/etc/firmware/cypress
```

After this, the Wi-Fi icon in user interface could not show any detected access point. Using ifconfig, we also cannot see the Wi-Fi connection.

```
mmi@pm-1000:~/dev/kernel_modules$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.1.21  netmask 255.255.255.0  broadcast 192.168.1.255
      inet6 fe80::2124:3e2d:8322:9985  prefixlen 64  scopeid 0x20<link>
      ether 48:b0:2d:18:30:92  txqueuelen 1000  (Ethernet)
      RX packets 1442  bytes 265430 (265.4 KB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 1458  bytes 365637 (365.6 KB)
      TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
      device interrupt 41

eth1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
      ether 00:0c:8b:92:09:65  txqueuelen 1000  (Ethernet)
      RX packets 0  bytes 0 (0.0 B)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 0  bytes 0 (0.0 B)
      TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
      device interrupt 125  memory 0x40100000-40120000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1  netmask 255.0.0.0
      inet6 ::1  prefixlen 128  scopeid 0x10<host>
      loop  txqueuelen 1  (Local Loopback)
      RX packets 472  bytes 35794 (35.7 KB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 472  bytes 35794 (35.7 KB)
      TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

The following dmesg was retrieved.

```
[  888.364122] compat: loading out-of-tree module taints kernel.
[  888.371908] compat: no symbol version for hash_algo_name
[  888.381153] Loading modules backported from Linux version v5.10.9-2022_0511-0-ge060a4740b3d
[  888.381159] Backport generated by backports.git v5.10.42-1-0-gbee5c545
[  897.302676] cfg80211: Loading compiled-in X.509 certificates for regulatory database
[  897.319423] PKCS#7 signature not signed with a trusted key
```

```
[  897.324958] cfg80211: loaded regulatory.db is malformed or signature is missing/invalid
[ 1018.702513] mmc1: Timeout waiting for hardware cmd interrupt.
[ 1018.708328] sdhci: =========== REGISTER DUMP (mmc1)==========
[ 1018.714201] sdhci: Sys addr: 0x00000000 | Version:  0x00000404
[ 1018.720063] sdhci: Blk size: 0x00000000 | Blk cnt:  0x00000000
[ 1018.725927] sdhci: Argument: 0x80022040 | Trn mode: 0x00000000
[ 1018.731787] sdhci: Present:  0x01fb00f1 | Host ctl: 0x00000001
[ 1018.737644] sdhci: Power:    0x00000000 | Blk gap:  0x00000000
[ 1018.743508] sdhci: Wake-up:  0x00000000 | Clock:    0x00000403
[ 1018.749366] sdhci: Timeout:  0x00000000 | Int stat: 0x00000000
[ 1018.755229] sdhci: Int enab: 0x00ff1003 | Sig enab: 0x00fc1003
[ 1018.761088] sdhci: AC12 err: 0x00000000 | Slot int: 0x00000000
[ 1018.766946] sdhci: Caps:     0x3f6cd08c | Caps_1:   0x18006f73
[ 1018.772808] sdhci: Cmd:      0x0000341a | Max curr: 0x00000000
[ 1018.778664] sdhci: Host ctl2: 0x00003000
[ 1018.782638] sdhci: ADMA Err: 0x00000000 | ADMA Ptr: 0x00000000fc100410
[ 1018.789238] sdhci: ============================================
[ 1018.795509] brcmfmac: probe of mmc1:0001:1 failed with error -110
[ 1018.812031] mmc1: Controller never released inhibit bit(s).
[ 1018.817616] sdhci: =========== REGISTER DUMP (mmc1)==========
[ 1018.823464] sdhci: Sys addr: 0x00000000 | Version:  0x00000404
[ 1018.829319] sdhci: Blk size: 0x00000000 | Blk cnt:  0x00000000
[ 1018.835178] sdhci: Argument: 0x80022040 | Trn mode: 0x00000000
[ 1018.841035] sdhci: Present:  0x01fb00f1 | Host ctl: 0x00000001
[ 1018.846895] sdhci: Power:    0x00000000 | Blk gap:  0x00000000
[ 1018.852758] sdhci: Wake-up:  0x00000000 | Clock:    0x00000403
[ 1018.858622] sdhci: Timeout:  0x00000000 | Int stat: 0x00000000
[ 1018.864484] sdhci: Int enab: 0x00ff1003 | Sig enab: 0x00fc1003
[ 1018.870341] sdhci: AC12 err: 0x00000000 | Slot int: 0x00000000
[ 1018.876201] sdhci: Caps:     0x3f6cd08c | Caps_1:   0x18006f73
[ 1018.882061] sdhci: Cmd:      0x0000341a | Max curr: 0x00000000
[ 1018.887915] sdhci: Host ctl2: 0x00003000
[ 1018.891872] sdhci: ADMA Err: 0x00000000 | ADMA Ptr: 0x00000000fc100410
[ 1018.898474] sdhci: ============================================
[ 1018.905191] brcmfmac: probe of mmc1:0001:2 failed with error -5
[ 1018.921292] mmc1: Controller never released inhibit bit(s).
[ 1018.926902] sdhci: =========== REGISTER DUMP (mmc1)==========
[ 1018.932766] sdhci: Sys addr: 0x00000000 | Version:  0x00000404
[ 1018.938628] sdhci: Blk size: 0x00000000 | Blk cnt:  0x00000000
[ 1018.944490] sdhci: Argument: 0x80022040 | Trn mode: 0x00000000
[ 1018.950354] sdhci: Present:  0x01fb00f1 | Host ctl: 0x00000001
[ 1018.956200] sdhci: Power:    0x00000000 | Blk gap:  0x00000000
[ 1018.962035] sdhci: Wake-up:  0x00000000 | Clock:    0x00000403
[ 1018.967872] sdhci: Timeout:  0x00000000 | Int stat: 0x00000000
[ 1018.973720] sdhci: Int enab: 0x00ff1003 | Sig enab: 0x00fc1003
[ 1018.979566] sdhci: AC12 err: 0x00000000 | Slot int: 0x00000000
[ 1018.985421] sdhci: Caps:     0x3f6cd08c | Caps_1:   0x18006f73
[ 1018.991281] sdhci: Cmd:      0x0000341a | Max curr: 0x00000000
[ 1018.997136] sdhci: Host ctl2: 0x00003000
[ 1019.001094] sdhci: ADMA Err: 0x00000000 | ADMA Ptr: 0x00000000fc100410
[ 1019.007694] sdhci: ============================================
[ 1019.014343] brcmfmac: probe of mmc1:0001:3 failed with error -5
[ 1019.023764] usbcore: registered new interface driver brcmfmac
```

Replace BCM Resource File with Latest Firmware

An alternative way to load the driver was attempted. This is based on suggestions from our previous attempt and also from this article, where one moderator showed the method of replacing files in the /lib/firmware/brcm/ folder.

It should be noted that both wl and wlarm64 has been tried on the image before and after inserting kernel modules.

```
mmi@pm-1000:~/dev$ sudo ./wl64 ver
[sudo] password for mmi:
./wl64: wl driver adapter not found
mmi@pm-1000:~/dev$ sudo ./wlarm64_dhd ver
./wlarm64_dhd: wl driver adapter not found
```

We comment out the last two lines in the "/etc/NetworkManager/NetworkManager.conf" file and reboot the system.

Command:

```
sudo vim /etc/NetworkManager/NetworkManager.conf
```

```
#[keyfile]
#unmanaged-devices=interface-name:wlan0
```

After a reboot, running wl and wlarm64 shows the following result:

```
mmi@pm-1000:~/dev$ sudo ./wl64 ver
1.28 RC0.0
wl0: Jul 21 2020 16:35:45 version 7.35.349.94 (r726700 CY) FWID 01-aa356e32
mmi@pm-1000:~/dev$ sudo ./wlarm64_dhd ver
1.28 RC0.0
wl0: Jul 21 2020 16:35:45 version 7.35.349.94 (r726700 CY) FWID 01-aa356e32
```

We then proceed to replace the resource for the bcmdhd with the following commands

```
mkdir ~/dev/bcmdhd_backup
cp /lib/firmware/brcm/fw_bcmdhd.bin ~/dev/bcmdhd_backup/
cp /lib/firmware/brcm/brcmfmac4354-sdio.bin ~/dev/bcmdhd_backup/
sudo cp /lib/firmware/cypress/cyfmac4354-sdio.bin /lib/firmware/brcm/
sudo mv /lib/firmware/brcm/cyfmac4354-sdio.bin /lib/firmware/brcm/fw_bcmdhd.bin
sudo cp /lib/firmware/cypress/cyfmac4354-sdio.bin /lib/firmware/brcm/
sudo mv /lib/firmware/brcm/cyfmac4354-sdio.bin /lib/firmware/brcm/brcmfmac4354-sdio.bin
sudo cp /lib/firmware/cypress/cyfmac4354-sdio.clm_blob /lib/firmware/brcm/
sudo mv /lib/firmware/brcm/cyfmac4354-sdio.clm_blob /lib/firmware/brcm/brcmfmac4354-sdio.clm_blob
sudo cp /lib/firmware/cypress/cyfmac4354-sdio.clm_blob /lib/firmware/brcm/
sudo cp /lib/firmware/brcm/cyfmac4354-sdio.clm_blob /lib/firmware/brcm/bcmdhd.clm_blob
sudo systemctl reboot -i
```

After the reboot, we can verify the firmware has changed.

```
mmi@pm-1000:~/dev$ sudo ./wl64 ver
```

```
1.28 RC0.0
wl0: Mar 28 2022 22:00:00 version 7.35.349.117 (af057cf CY) FWID 01-ea8f72c7
mmi@pm-1000:~/dev$ sudo ./wlarm64_dhd ver
1.28 RC0.0
wl0: Mar 28 2022 22:00:00 version 7.35.349.117 (af057cf CY) FWID 01-ea8f72c7
```

We then proceed to test turning Wi-Fi down and up. We have the same result as before from our post.

```
mmi@pm-1000:~/dev$ ip -br address show wlan0
wlan0            UP             192.168.27.253/24 fe80::306c:4c71:597a:a246/64
mmi@pm-1000:~/dev$ sudo ./wlarm64_dhd down
mmi@pm-1000:~/dev$ ip -br address show wlan0
wlan0            DORMANT        192.168.27.253/24 fe80::306c:4c71:597a:a246/64
mmi@pm-1000:~/dev$ ip -br address show wlan0
wlan0            DORMANT
mmi@pm-1000:~/dev$ sudo ./wlarm64_dhd up
mmi@pm-1000:~/dev$ ip -br address show wlan0
wlan0            DORMANT
```

Replace BCM Resource File with the mfg_firmware Package

We have previously received a mfg_firmware package. We use the same method to replace the bcmdhd related files with the commands below. Assume the mfg_firmware package is already extracted and copied to the ~/dev folder.

```
sudo cp /home/mmi/dev/mfg_firmware/cyfmac4354-sdio.bin /lib/firmware/brcm/fw_bcmdhd.bin
sudo cp /home/mmi/dev/mfg_firmware/cyfmac4354-sdio.bin /lib/firmware/brcm/brcmfmac4354-sdio.bin
sudo cp /home/mmi/dev/mfg_firmware/cyfmac4354-sdio.clm_blob /lib/firmware/brcm/brcmfmac4354-sdio.clm_blob
sudo cp /home/mmi/dev/mfg_firmware/cyfmac4354-sdio.clm_blob /lib/firmware/brcm/bcmdhd.clm_blob
```

After reboot, we can verify the firmware version has changed.

```
mmi@pm-1000:~/dev$ sudo ./wl64 ver

1.28 RC0.0

wl0: Sep  1 2021 02:23:00 version 7.35.349.112 (ae8cc29 CY WLTEST) FWID 01-6e1af42a

mmi@pm-1000:~/dev$ sudo ./wlarm64_dhd ver

1.28 RC0.0

wl0: Sep  1 2021 02:23:00 version 7.35.349.112 (ae8cc29 CY WLTEST) FWID 01-6e1af42a
```

We then proceed to test turning Wi-Fi down and up. We have the same result as before from our post.

```
mmi@pm-1000:~/dev$ ip -br address show wlan0
```

```
wlan0            UP             192.168.1.25/24 fe80::91e3:4876:ec40:a0e1/64
mmi@pm-1000:~/dev$ sudo ./wlarm64_dhd down
mmi@pm-1000:~/dev$ ip -br address show wlan0
wlan0            DORMANT        192.168.1.25/24 fe80::91e3:4876:ec40:a0e1/64
mmi@pm-1000:~/dev$ ip -br address show wlan0
wlan0            DORMANT
mmi@pm-1000:~/dev$ sudo ./wlarm64_dhd up
mmi@pm-1000:~/dev$ ip -br address show wlan0
wlan0            DORMANT
```

Appendix A: Cannot flash to TX2 if I use the .config from the generated kernel output

One thing I found is that after kernel is compiled, the .config created in the kernel output folder would no longer have the BCMDHD relatd configs. This should be the effect of setting tegra_defconfig (we are using cti_tegra_defconfig) with:

CONFIG_BCMDHD=n

Before:

CONFIG_BCMDHD=m

CONFIG_BCMDHD_SDIO=y

CONFIG_BCMDHD_PCIE=y

CONFIG_BCMDYNAMIC is not set

CONFIG_BCM43241 is not set

CONFIG_BCM4354=y

CONFIG_BCMDHD_FW_PATH="/lib/firmware/brcm/fw_bcmdhd.bin"

CONFIG_BCMDHD_NVRAM_PATH="/lib/firmware/brcm/nvram.txt"

CONFIG_BCMDHD_HW_OOB=y

CONFIG_DHD_USE_STATIC_BUF is not set

CONFIG_DHD_USE_SCHED_SCAN=y

CONFIG_BCMDHD_DISABLE_MCC=y

CONFIG_BCMDHD_CUSTOM_SYSFS_TEGRA is not set

CONFIG_BCMDHD_CUSTOM_NET_PERF_TEGRA is not set

CONFIG_BCMDHD_QMONITOR is not set

CONFIG_BCMDHD_CUSTOM_NET_BW_EST_TEGRA is not set

CONFIG_BCMDHD_CUSTOM_NET_DIAG_TEGRA is not set

CONFIG_BCM4359=y

CONFIG_BCMDHD_PCIE_FW_PATH="/lib/firmware/brcm/fw_bcmdhd_4356.bin"

CONFIG_BCMDHD_PCIE_NVRAM_PATH="/lib/firmware/brcm/nvram_4356.txt"

CONFIG_BCMDHD_CLM_PATH="/lib/firmware/brcm/bcmdhd.clm_blob"

CONFIG_BCMDHD_PCIE_ES4_NVRAM_PATH="/lib/firmware/brcm/nvram_murata_4359_b1_es4.txt"

CONFIG_DHD_SET_RANDOM_MAC_VAL=0x001A11

After:

CONFIG_BCMDHD is not set

Based on the article, I did run "sudo ./apply_binaries.sh", and I noticed that bcmdhd.ko was placed in /rootfs/lib/modules/4.9.201-tegra/kernel/drivers/net/wireless/bcmdhd folder. The article also said I need to run module_install command. My command is: "sudo make ARCH=arm64 O=$TEGRA_KERNEL_OUT modules_install INSTALL_MOD_PATH=~/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/rootfs/" After running it, I noticed that the bcmdhd.ko was removed.

If I proceed to flash the image, the flash process will abort with the following error:

Making recovery ramdisk for recovery image...

Re-generating recovery ramdisk for recovery image...

~/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/bootloader/ramdisk_tmp
~/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/bootloader
~/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra

30402 blocks

_BASE_KERNEL_VERSION=4.9.201-tegra

cp: cannot stat '/home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/rootfs/lib/modules/4.9.201-tegra/kernel/drivers/net/wireless/bcmdhd/bcmdhd.ko': No such file or directory

failed command: cp -fv /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/rootfs/lib/modules/4.9.201-tegra/kernel/drivers/net/wireless/bcmdhd/bcmdhd.ko /home/shane/nvidia/nvidia_sdk_inf2/JetPack_4.5_Linux_JETSON_TX2/Linux_for_Tegra/bootloader/ramdisk_tmp//lib/modules/4.9.201-tegra/kernel/drivers/net/wireless/bcmdhd/bcmdhd.ko

To resolve this issue, I ran the "sudo ./apply_binaries.sh" and bring back the bcmdhd.ko. But the .config will be modified. I then copy the .config again. This allows me to flash the image successfully, but the change of sequence is a bit concerning. It also imply some dependency still on bcmdhd.