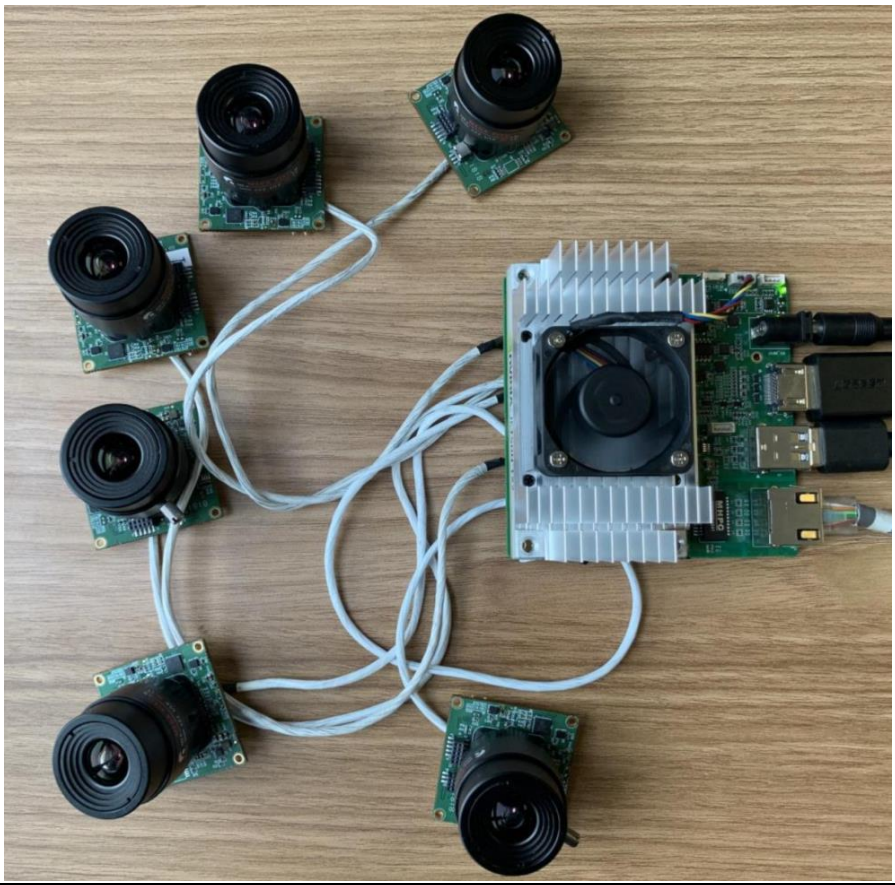
Overview

This driver is for LI-TX1-CB-6CAM + IMX265 camera kit with Nvidia Jetson TX2 CPU module (SOM).
This driver supports up to six IMX265 cameras.
This driver supports 2048 x 1554 @ 38fps and 1936x1106@30fps.
This driver doesn't support camera sync function.
This driver is based on R32.3.1 (Jetpack 4.3).

Download link

https://www.dropbox.com/sh/teiv92ywz5mz325/AAC-KDhHJW0hCQ3h1aEEb-Lba?dl=0

Platform	Camera
Nvidia Jetson TX2 SOM with R32.3.1	1 ~ 6 x LI-IMX265-MIPI
Cable	Adapter/Carrier Board
1 ~ 6 x FAW-1233	1 x LI-TX1-CB-6CAM Rev1.0



Revision	SVN version	Release Date	Author	Tested By
2020_02_22	Rev199	02/22/2020	Lincheng Yang	Simon Zhu

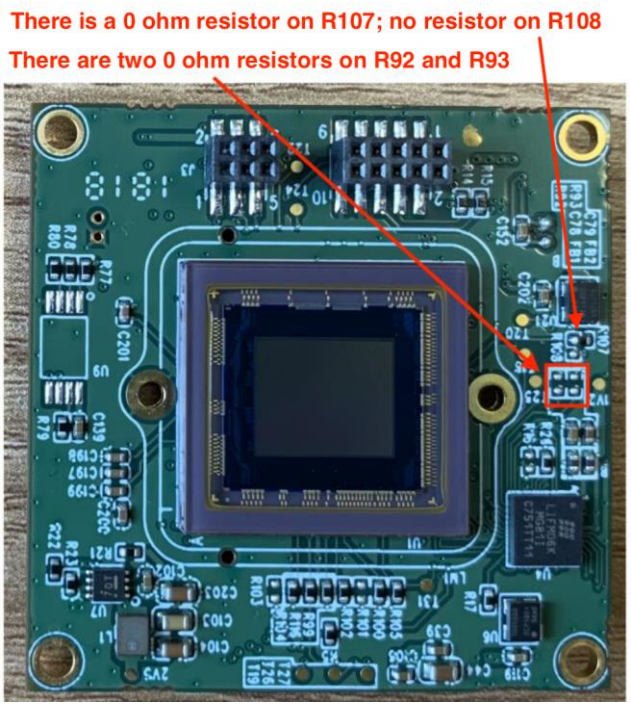
Updates

Revision	Description	Release Date
2020_02_22	First release based on R32.3.1	02/22/2020

Known bugs

Rework board

LI-IMX264-MIPI V1.0



LI-IMX264-MIPI V1.1

Setup Procedure 1/2

Hardware:

1. LI-TX1-CB-6CAM + TX2 SOM x 1
2. LI-IMX265-MIPI x 1 ~ 6
3. FAW-1233 cable x 1 ~ 6
4. USB 2.0 micro-B cable x 1 (for flashing OS image and dtb file)
5. 12VDC power supply x1
6. Monitor with HDMI cable x 1
7. Keyboard and Mouse (with USB hub) x 1

Note: Please make sure the LI-TX1-CB-6CAM board has been programmed FPGA IMX264_6cam_v134.jed and IMX265 board has been programmed FPGA lvds2mipi_imx264_2lane_v151.rbt .

Driver installation:

1. Download the L4T R32.3.1 for TX2 from link below to your Ubuntu OS on Intel x64 host PC (we are using Ubuntu 18.04, virtual machine is fine) and follow the quick start guide (l4t_quick_start_guide) to flash the R32.3.1 image to TX2.

R32.3.1 OS Image: https://www.dropbox.com/sh/usqu5j0p4b3hh3q/AADnGf98IkHqFU-Tnm1g4NWza?dl=0

Note: The USB port won't work after flashing the R32.3.1 OS image. USB will work after doing step4 below.

2. Reboot TX2 and Put your system into "reset recovery mode" again.
3. Copy the tegra186-quill-p3310-1000-c03-00-base.dtb (which was downloaded from the Driver link in page 1) and replace the same file under Linux_for_Tegra/kernel/dtb on your Ubuntu host PC.

```
simon@ubuntu:~/R32.3.1-TX2/Linux_for_Tegra$ sudo cp ../tegra186-quill-p3310-1000
-c03-00-base.dtb kernel/dtb/
```

4. Under Linux_for_Tegra/ do
   sudo ./flash.sh -k kernel-dtb jetson-tx2 mmcblk0p1

```
simon@ubuntu:~/R32.3.1-TX2/Linux_for_Tegra$ sudo ./flash.sh -k kernel-dtb jetson
-tx2 mmcblk0p1
```

If flash the dtb file successfully, the log should be like below.

```
[   10.2377 ] [.............................................] 100%
[   10.2708 ]
[   10.2729 ] Coldbooting the device
[   10.2749 ] tegradevflash_v2 --reboot coldboot
[   10.2765 ] Bootloader version 01.00.0000
[   10.3054 ]
*** The [kernel-dtb] has been updated successfully. ***
simon@ubuntu:~/R32.3.1-TX2/Linux_for_Tegra$
```

Setup Procedure 2/2

5. After boot up TX2, copy “Image” to /boot on TX2.

```
nvidia@nvidia-desktop:~/Downloads$ sudo cp Image /boot/
```

6. Reboot TX2. After boot up, open a terminal and do "nvgstcapture-1.0”. You will get live video output.

Note: Please make sure there is a camera on port **J1** (A) of LI-TX1-CB-6CAM board.

7. Use Ctrl+C to close the video and copy camera_overrides.isp to /var/nvidia/nvcam/settings and do below two command (if there is an isp file)
   sudo chmod 664 /var/nvidia/nvcam/settings/camera_overrides.isp
   sudo chown root:root /var/nvidia/nvcam/settings/camera_overrides.isp

```
nvidia@nvidia-desktop:~/Downloads$ sudo cp camera_overrides.isp /var/nvidia/nvca
m/settings/
nvidia@nvidia-desktop:~/Downloads$ sudo chmod 664 /var/nvidia/nvcam/settings/cam
era_overrides.isp
nvidia@nvidia-desktop:~/Downloads$ sudo chown root:root /var/nvidia/nvcam/settin
gs/camera_overrides.isp
nvidia@nvidia-desktop:~/Downloads$
```

8. Try "nvgstcapture-1.0” again. You should be able to see the image with better image quality.

Run Camera

1. Argus software

Download the Multimedia package from link below and copy it to TX2.

https://www.dropbox.com/s/4qo4ca52se0lptq/Multimedia_JXAV_R32.3.1.tgz?dl=0

Open a terminal, do
sudo apt-get update
sudo apt-get install cmake libgtk-3-dev libjpeg-dev libgles2-mesa-dev libgstreamer1.0-dev

Uncompress the tgz package,
tar zxvf Multimedia_JXAV_R32.3.1.tgz

Under jetson_multimedia_api/argus/cmake, do
cmake ..
make
sudo make install

Do "argus_camera --device=0" to get the video.

2. Gstreamer

gst-launch-1.0 nvarguscamerasrc sensor-id=0 ! 'video/x-raw(memory:NVMM), width=(int)2048, height=(int)1554, framerate=38/1' ! nvvidconv flip-method=0 ! 'video/x-raw, format=(string)I420' ! xvimagesink –e

3. v4l2-ctl capture raw

v4l2-ctl --set-fmt-video=width=2048,height=1554,pixelformat=RG12 --set-ctrl bypass_mode=0 --stream-mmap --stream-count=3 --stream-to=imx265.raw -d /dev/video0

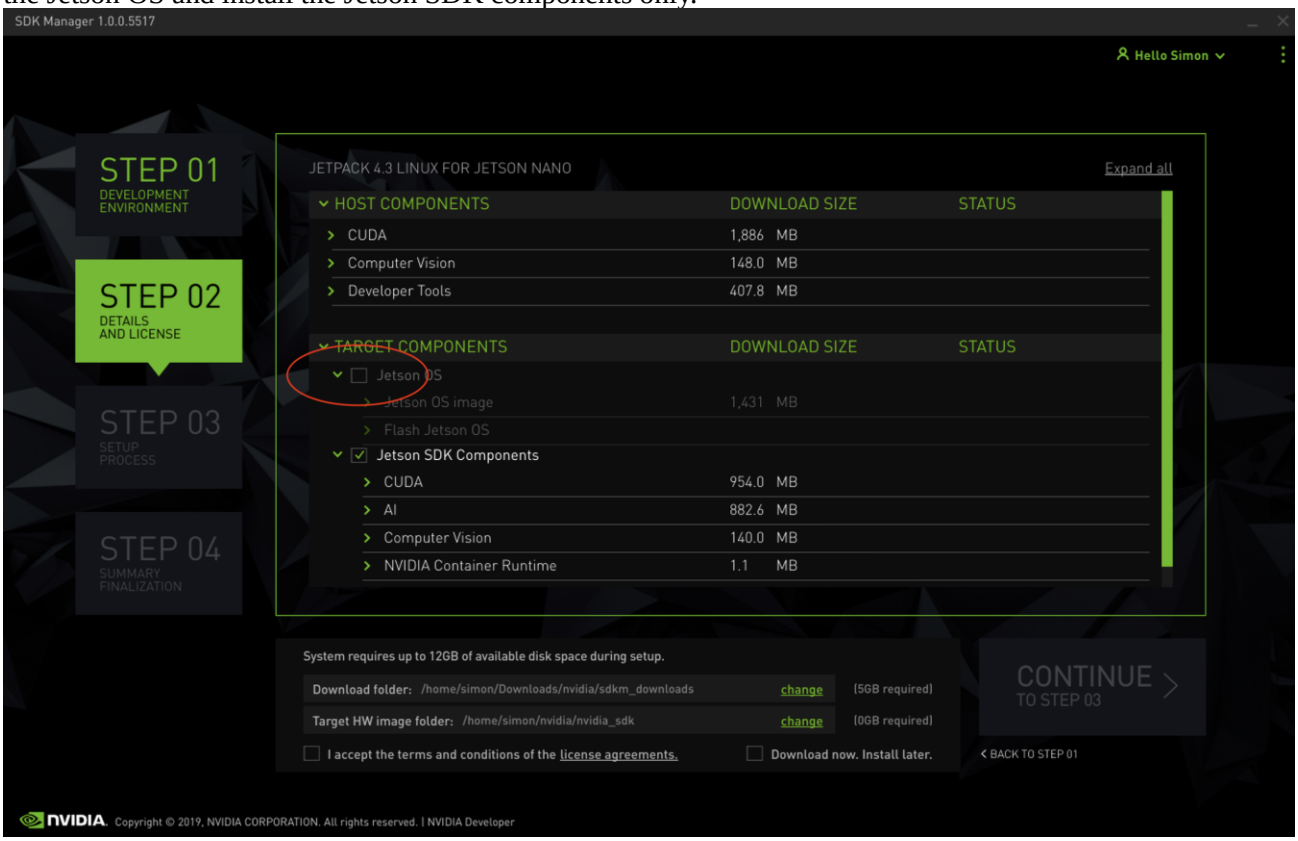
Note:
1) The 0 can be changed to 1 ~ 5 if there are other cameras connected.

Connector J1 (A) ---- video0
Connector J3 (B) ---- video1
Connector J6 (C) ---- video2
Connector J7 (D) ---- video3
Connector J8 (E) ---- video4
Connector J9 (F) ---- video5

2) Please use below commands to install v4l2.
sudo apt-get update
sudo apt-get install v4l-utils

Note 1/6

1. If you would like to install the Jetpack 4.3 but don't want to re-flash the whole OS image, you can uncheck the Jetson OS and install the Jetson SDK components only.



2. If you have cooling system (fan), you can use below commands to turn on/off the fan.
1) switch to the root user.
   sudo su
2) echo 255 > /sys/devices/pwm-fan/target_pwm //turn on
   echo 0 > /sys/devices/pwm-fan/target_pwm //turn off

3. If there are any new drivers, we will add them into link below.
https://www.dropbox.com/sh/td09lt3hakvc2ex/AAC3hp-4t6aTBfc_FuJP-yfqa?dl=0

Note 2/6

4. Compile the driver
If you would like to re-compile the driver, please follow below steps.
Download the driver code and Tool chain from links below.

Kernel code: https://www.dropbox.com/s/xpetnme80qg5wci/kernel_src_JXAV_R32.3.1.tbz2?dl=0
GCC ToolChain: https://www.dropbox.com/sh/f21qck6f29h3n20/AABP8B1b4DgmUgO2MYO32Nyza?dl=0

Compile the kernel under 64 bit Ubuntu OS on Intel x64 PC. (Virtual machine is fine. We are using Ubuntu 16.04 64 bit OS)

1) Copy compile tool gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz to /opt, and unzip it

```
sudo tar xpf gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz
```

2) Copy kernel_src_JXAV_R32.3.1.tbz2 and two patch files to /usr/src

```
sudo tar xpf kernel_src_JXAV_R32.3.1.tbz2
sudo chown -R <user_name> kernel
sudo chown -R <user_name> hardware
patch -p0 < 6_streaming_imx265_base32.3.1_CB_tx2_no-sync_dtbs_20200222.patch
patch -p0 < 6_streaming_imx265_base32.3.1_CB_tx2_no-sync_kernel_20200222.patch
```

Note: <user_name> is the user name of your Ubuntu OS. For example: sudo chown -R leopard kernel

3) Copy tx2.sh to /usr/src/kernel.

```
under /usr/src/kernel,do
source tx2.sh
```

4) Create a work folder under /home:

```
sudo mkdir /home/work
sudo chown -R <user_name> /home/work
```

5) In "kernel/kernel-4.9" folder, run:

```
make O=$TEGRA_KERNEL_OUT tegra_defconfig
make O=$TEGRA_KERNEL_OUT zImage
make O=$TEGRA_KERNEL_OUT dtbs
```

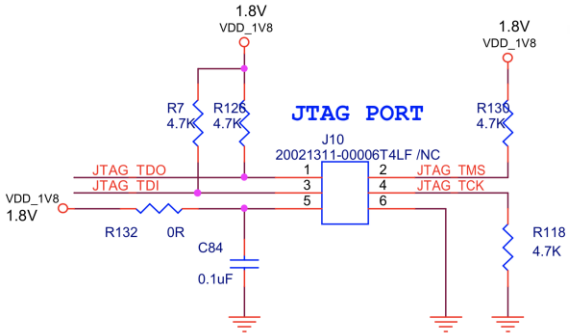
You will get Image under /home/work/TX2/kernel/kernel_out/arch/arm64/boot and tegra186-quill-p3310-1000-c03-00-base.dtb under /home/work/TX2/kernel/kernel_out/arch/arm64/boot/dts.

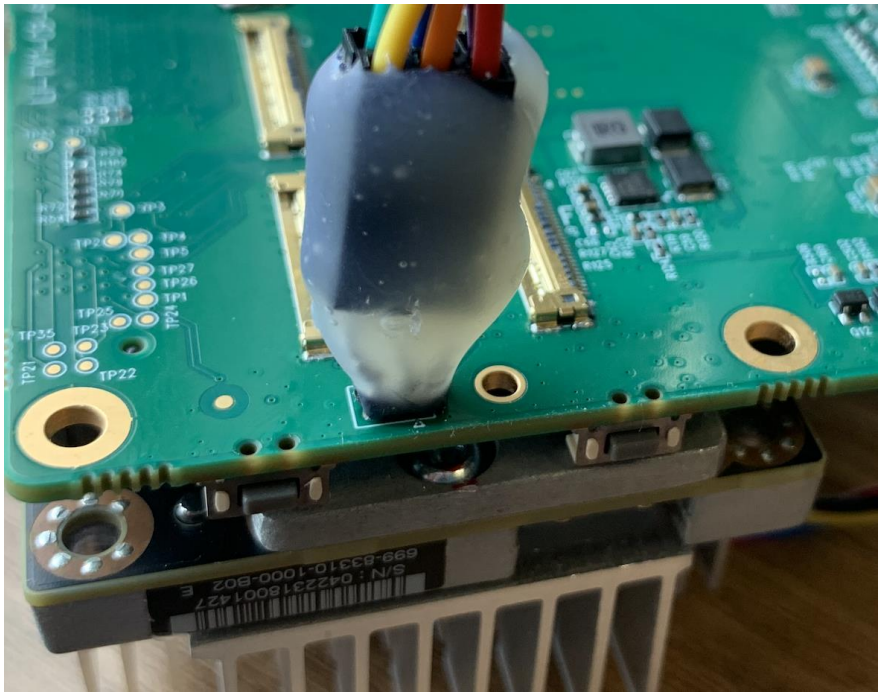
Note 3/6

5. Programming FPGA file to LI-TX1-CB-6CAM board
FPGA file name: IMX264_6cam_v134.jed

1) Download the Lattice programmer tool software **Programmer Standalone 3.* 64-bit for Windows** from link below and install it to your PC.
http://www.latticesemi.com/programmer
2) Connect the HW-USBN-2A or HW-USBN-2B to J10 of LI-TX1-CB-6CAM board. Below is the pinout. You may need to build an adapter cable for the programmer tool and J10. J10 is a 1.27mm pitch interface. Below pins in red need to be connected.

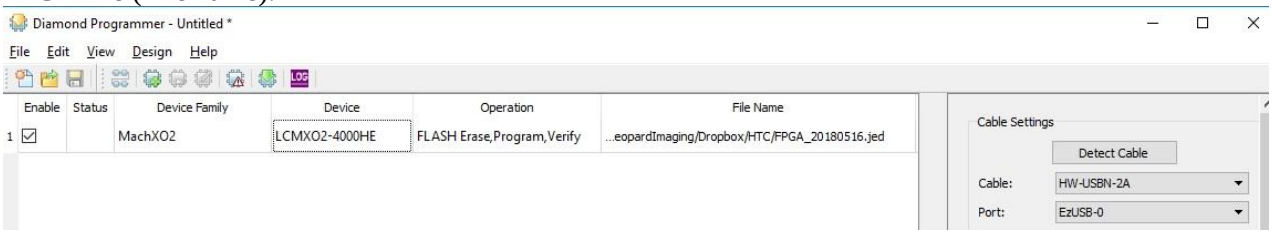
LI-TX1-CB-6CAM <—> HW-USBN-2B	LI-TX1-CB-6CAM <—> HW-USBN-2B
Pin1: JTAG_TDO <—> TDO/SO	Pin2: JTAG_TMS <—> TMS
Pin3: JTAG_TDI <—> TDI/SI	Pin4: JTAG_TCK <—> TCK/SCLK
Pin5: 1V8 <—> VCC	Pin6: GND <—> GND



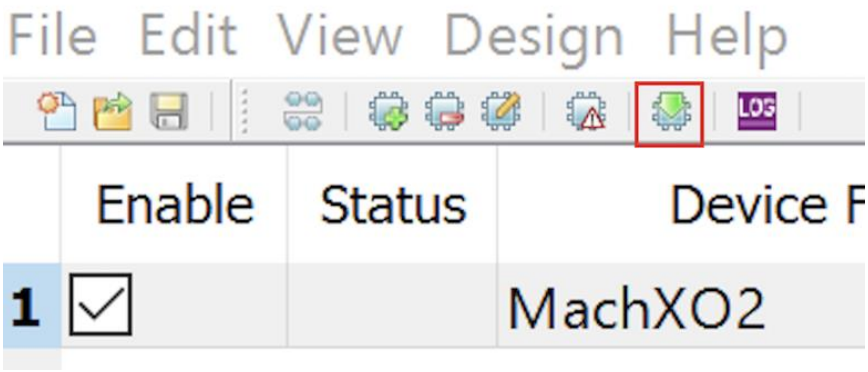


Note 4/6

3) Power on TX2.
4) Open Lattice software (which you installed in 1). Select the Device name “LCMXO2-4000HE” and the FPGA file (File name).



5) Click program icon.



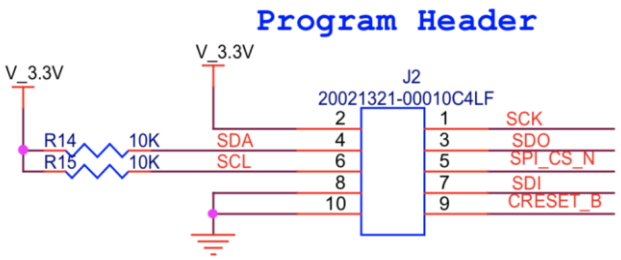
6) It may take 30s to program the FPGA.

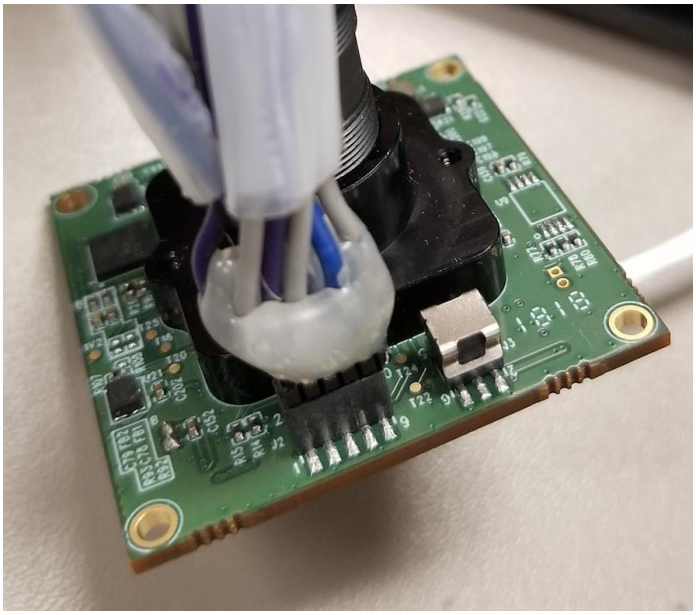
```
INFO - JTAG Chain Verification. No Errors.

INFO - Check configuration setup: Successful.

INFO - Device1 LCMXO2-4000HE: FLASH Erase,Program,Verify

INFO - Operation Done. No errors.

INFO - Elapsed time: 00 min : 30 sec

INFO - Operation: successful.
```

Note 5/6

6. Programming FPGA file to IMX264 board
   FPGA file name: lvds2mipi_imx264_2lane_v151.rbt
1) Download the Lattice programmer tool software **Programmer Standalone 3.10 64-bit for Windows** from link below and install it to your PC.
http://www.latticesemi.com/programmer
2) Connect the HW-USBN-2A or HW-USBN-2B to J2 of LI-IMX265-MIPI board. Below is the pinout. You may need to build an adapter cable for the programmer tool and J2. J2 is a 1.27mm pitch interface.
Below pins in red need to be connected.

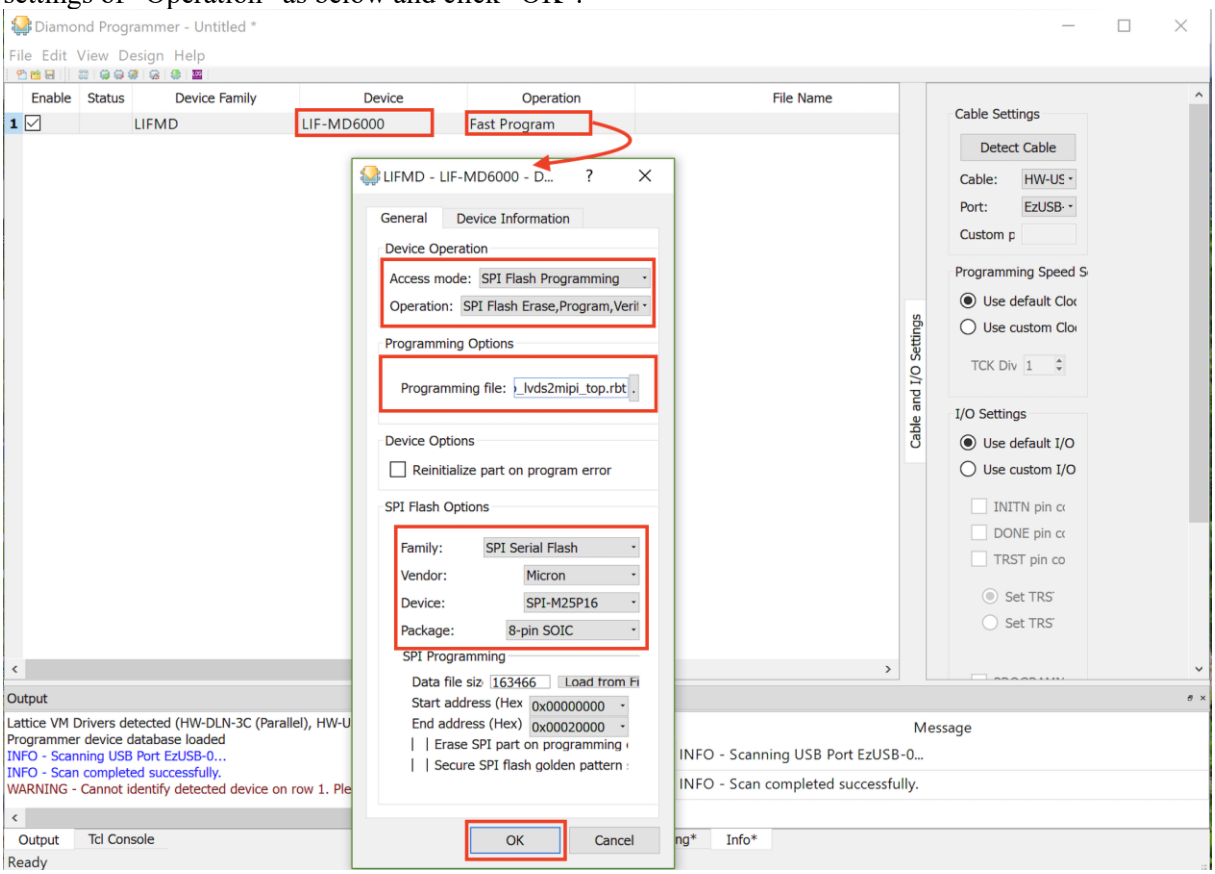
IMX265-MIPI Board <—> HW-USBN-2B	IMX265-MIPI Board <—> HW-USBN-2B
Pin2: V_3.3V <—> VCC	Pin1: SCK <—> TCK/SCLK
Pin4: N/A	Pin3: SDO <—> TDO/SO
Pin6: N/A	Pin5: SPI_CS_N <—> ISPEN/PROG
Pin8: GND <—> GND	Pin7: SDI <—> TDI/SI
Pin10: N/A	Pin9: CRESET_B <—> TRST





Note 6/6

3) Connect the IMX265 camera to TX2 kit and power on the LI-TX1-CB-6CAM board.
4) Open Lattice software (which you installed in 1). The Device name should be “LIF-MD6000” . Set the settings of “Operation” as below and click “OK”.



5) Click program icon.

6) It may take 9s to program the FPGA.