

Downloading and Loading Kria-Prophesee-Event-VitisAI Application Firmware in KV260 Board

Getting Started Guide

Overview of Kria-App

This Kria-App consists of setup steps for running the event camera based Machine learning inference in Kria KV260 Board. Here we are taking event camera from Prophesee and also performing recorded event file based ML inference.

This kria-app has Kria-Ubuntu and Kria-Petalinux both OS support.



Prerequisites

Following are the prerequisites of this Kria-App:

- KV260 board and power supply
- Ethernet cable
- USB cable (for serial connection to host machine)
- HDMI monitor and cable
- Prophesee CCAM5 (optional if using event file based ML)
- Mouse and Keyboard
- SD card (32 GB)

Setup Kria Ubuntu SD card for KV260:

Get the Kria Ubuntu 22.04 from - [Link](#)

Create SD card using balenaethcer tool and prepare the KV260 for SD card boot. Along with this getting started guide you can also perform test on SmartCam application for verification of your hardware and setup.

Getting the Kria Ubuntu Firmwares

Get the Kria Ubuntu firmware and necessary installation scripts from following repository :

<https://github.com/LogicTronixInc/Kria-Prophesee-Event-VitisAI>

After cloning the repository, run following scripts to update Ubuntu Kernel followed by Prophesee driver installation.

Update Kria Ubuntu kernel:

Run scripts to update the Ubuntu kernel:

Kria-Prophesee-Event-VitisAI/Kria-Ubuntu/scripts/install_updated_linux_kernel.sh

Setup Prophesee camera drivers :

Run following script to install Prophesee camera drivers:

Kria-Prophesee-Event-VitisAI/Kria-Ubuntu/scripts/install_psee_drivers.sh

Get the application:

```
'''
```

```
Git clone
```

```
'''
```

Docker-Based Application Preparation

Pull the docker image:

```
'''
```

```
docker pull logictronixinc/prophesee-ml-kria:v0.1
```

```
'''
```

View the available images using docker command:

```
'''
```

```
docker images
```

```
'''
```

Load the hardware overlay using xmutil :

```
'''
```

```
Sudo xmutil unloadapp
```

```
Sudo xmutil loadapp <hardware overlay>
```

```
'''
```

```

base login: Mon Jun 10 11:41:05 2024 from 192.168.0.197
ubuntu@kria:~$ sudo xmutil listapps
[sudo] password for ubuntu:
      Accelerator      Accel_type      Base      Base_type      #slots(PL+AIE)      Active_slot
      psee-vitis-mipi-dpu      XRT_FLAT      psee-vitis-mipi-dpu      XRT_FLAT      (0+0)      -1
      prophesee-kv260-imx636      XRT_FLAT      prophesee-kv260-imx636      XRT_FLAT      (0+0)      -1
      kv260-smartcam      XRT_FLAT      kv260-smartcam      XRT_FLAT      (0+0)      -1
      kv260-benchmark-b4096      XRT_FLAT      kv260-benchmark-b4096      XRT_FLAT      (0+0)      -1
      k26-starter-klts      XRT_FLAT      k26-starter-klts      XRT_FLAT      (0+0)      0,
      kv260-psee-dpu      XRT_FLAT      kv260-psee-dpu      XRT_FLAT      (0+0)      -1
ubuntu@kria:~$ sudo xmutil unloadapp
remove from slot 0 returns: 0 (Ok)
ubuntu@kria:~$ sudo xmutil loadapp
-load expects a package name. Try again.
ubuntu@kria:~$ sudo xmutil loadapp psee-vitis-mipi-dpu
psee-vitis-mipi-dpu: loaded to slot 0
ubuntu@kria:~$

```

After loading the hardware overlay one can view Prophesee sensor media graph running `media-ctl -p` command:

```
ubuntu@kria:~$ media-ctl -p
Media controller API version 5.15.136

Media device information
-----
driver      psee-video
model       Prophesee Video Pipeline
serial
bus info
hw revision  0x0
driver version 5.15.136

Device topology
- entity 1: ps_host_if output 0 (1 pad, 1 link)
    type Node subtype V4L flags 0
    device node name /dev/video0
    pad0: SINK
        <- "a0050000.event_stream_smart_tra":1 [ENABLED]

- entity 5: a0010000.mipi_csi2_rx_subsystem (2 pads, 2 links, 0 routes)
    type V4L2 subdev subtype Unknown flags 0
    device node name /dev/v4l-subdev0
    pad0: SINK
        [stream:0 fmt:SRGB8_1X8/1920x1080 field:none colorspace:srgb]
        <- "imx636 6-003c":0 [ENABLED]
    pad1: SOURCE
        [stream:0 fmt:SRGB8_1X8/1920x1080 field:none colorspace:srgb]
        -> "a0040000.axis_tkeep_handler":0 [ENABLED]

- entity 8: a0050000.event_stream_smart_tra (2 pads, 2 links, 0 routes)
    type V4L2 subdev subtype Unknown flags 0
    device node name /dev/v4l-subdev1
    pad0: SINK
        [stream:0 fmt:unknown/0x0]
        <- "a0040000.axis_tkeep_handler":1 [ENABLED]
    pad1: SOURCE
        [stream:0 fmt:unknown/0x0]
        -> "ps_host_if output 0":0 [ENABLED]

- entity 11: a0040000.axis_tkeep_handler (2 pads, 2 links, 0 routes)
    type V4L2 subdev subtype Unknown flags 0
    device node name /dev/v4l-subdev2
    pad0: SINK
        [stream:0 fmt:unknown/0x0]
        <- "a0010000.mipi_csi2_rx_subsystem":1 [ENABLED]
    pad1: SOURCE
        [stream:0 fmt:unknown/0x0]
        -> "a0050000.event_stream_smart_tra":0 [ENABLED]

- entity 14: imx636 6-003c (1 pad, 1 link, 0 routes)
    type V4L2 subdev subtype Sensor flags 0
    device node name /dev/v4l-subdev3
```

Next update the media format to `PSEE_EVT21` running `load.sh` script.

And enable the camera sensor running following command:

```
...
```

```
echo on > /sys/class/video4linux/v4l-subdev3/device/power/control
```

```
...
```

Note: Here v4l2 subdev number may change depending upon loading of device driver, so update the above command with respect to imx636 node in media graph as obtained using media-ctl command.

```
ubuntu@kria:~$ cat load.sh
#!/bin/bash
: ${FORMAT:="PSEE_EVT21"}

# Force load of tkeep handler driver, so that we don't get the pass-through driver
# probed on tkeep handler
#modprobe psee-tkeep-handler

# Load the FPGA
#xmutil unloadapp
#xmutil loadapp prophesee-kv260-imx636

# Wait long enough for the drivers to be probed
sleep 1

# Set the pipeline to run in $FORMAT (eg PSEE_EVT3)
media-ctl -V "'imx636 6-003c':0[fmt:$FORMAT/1280x720]"
media-ctl -V "'a0010000.mipi_csi2_rx_subsystem':1[fmt:$FORMAT/1280x720]"
media-ctl -V "'a0040000.axis_tkeep_handler':1[fmt:$FORMAT/1280x720]"
media-ctl -V "'a0050000.event_stream_smart_tra':1[fmt:$FORMAT/1280x720]"
# The config of the pipeline can be seen with 'media-ctl -p'

# Force the sensor to be ON, so that Metavision can do register accesses on it
# Write "auto" to let the sensor be powered down when not streaming
#echo on > /sys/class/video4linux/v4l-subdev3/device/power/control

# run with
# MV_LOG_LEVEL=TRACE V4L2_HEAP=reserved V4L2_SENSOR_PATH=/dev/v4l-subdev3 metavision_viewer
ubuntu@kria:~$ sudo su
root@kria:/home/ubuntu# ./load.sh
root@kria:/home/ubuntu# echo on > /sys/class/video4linux/v4l-subdev3/device/power/control
root@kria:/home/ubuntu#
```

For GUI to get the output from application running in docker container, run following command:

```
...
```

```
xhost local:docker
```

```
...
```

Then run the docker run command:

```
...
```

```
docker run \
-e DISPLAY=unix$DISPLAY \
-h "xlnx-docker" \
--env="XDG_SESSION_TYPE" \
```

For any Queries, please contact us at : info@logictronix.com

```
--net=host \
--privileged \
--volume="/home/ubuntu/.Xauthority:/root/.Xauthority:rw" \
-v /tmp:/tmp \
-v /dev:/dev \
-v /sys:/sys \
-v /etc/vart.conf:/etc/vart.conf \
-v /lib/firmware/xilinx:/lib/firmware/xilinx \
-v /run:/run \
-v /home/ubuntu:/ubuntu \
-it <psee-openeb-ml-vitis:latest> bash
```

...

Things to check before running application:

- DPU check
 - Run `show_dpu` command to get information about available DPU
 - Run `xdputil query` to get detail information

```
root@xlnx-docker:/ubuntu# show_dpu
WARNING: Logging before InitGoogleLogging() is written to STDERR
F20240612 11:20:53.917568 1711 xrt_device_handle_tmp.cpp:327] Check failed: r == 0 cannot set read range! cu_index 0 cu_base_addr 2952790016 fingerprint 0x101000056010407 : Invalid argument [22]
device_core_id=0 device=0 core=0 fingerprint = 0x101000056010407 batch = 1 full_cu_name=DPU CZDX8G:DPU CZDX8G_1

root@xlnx-docker:/ubuntu# xdputil query
WARNING: Logging before InitGoogleLogging() is written to STDERR
F20240612 11:22:50.015642 1713 xrt_device_handle_tmp.cpp:327] Check failed: r == 0 cannot set read range! cu_index 0 cu_base_addr 2952790016 fingerprint 0x101000056010407 : Invalid argument [22]
Not support this dpu arch !
{
  "DPU IP Spec": {
    "DPU Core Count": 1,
    "IP version": "v4.1.0",
    "generation timestamp": "2023-02-21 21:30:00",
    "git commit id": "7d32c41",
    "git commit time": "2023022121",
    "regmap": "itot version"
  },
  "VAI Version": {
    "libvart-runner.so": "Xilinx vart-runner Version: 3.5.0-966e125cfe19b51a6f938d6071f25dfd501fcd7e 6 2023-06-21-11:18:16 ",
    "libvitis_at_library-dpu_task.so": "libopencytngcodecs.so.406: cannot open shared object file: No such file or directory",
    "libxrt.so": "Xilinx xrt Versions: xrt-d5929771399d0ea13f483fb091f602d1bb42f21a0 2023-06-21-11:16:50",
    "target_factory": "target-factory.3.5.0 e618c965234bcd7aea337e5fb3d910cc6ba5faea"
  },
  "kernels": [
    {
      "AIE Frequency (MHz)": 0,
      "DPU Arch": "DPU CZDX8G 1sA1_B4096",
      "DPU Frequency (MHz)": 200,
      "IP Type": "DPU",
      "Load Parallel": 2,
      "Load augmentation": "enable",
      "Load minus mean": "disable",
      "Save Parallel": 2,
      "XRT Frequency (MHz)": 200,
      "cu_addr": "0xb0000000",
      "cu_handle": "0xaaaaad45632d0",
      "cu_idx": 0,
      "cu_mask": 1,
      "cu_name": "DPU CZDX8G:DPU CZDX8G_1",
      "device_id": 0,
      "fingerprint": "0x101000056010407",
      "name": "DPU Core 0"
    }
  ]
}
```

- Camera Check
 - `media-ctl -p` to get the v4l2 media graph and sensor device number


```
-----
driver      psee-video
model      Prophesee Video Pipeline
serial
bus info
hw revision 0x0
driver version 5.15.136

Device topology
- entity 1: ps_host_if output 0 (1 pad, 1 link)
    type Node subtype V4L flags 0
    device node name /dev/video0
    pad0: Sink
        <- "a0050000.event_stream_smart_tra":1 [ENABLED]

- entity 5: a0010000.mipi_csi2_rx_subsystem (2 pads, 2 links)
    type V4L2 subdev subtype Unknown flags 0
    device node name /dev/v4l-subdev0
    pad0: Sink
        [fmt:unknown/1280x720 field:none colorspace:raw xfer:none]
        <- "imx636 6-003c":0 [ENABLED]
    pad1: Source
        [fmt:unknown/1280x720 field:none colorspace:raw xfer:none]
        -> "a0040000.axis_tkeep_handler":0 [ENABLED]

- entity 8: imx636 6-003c (1 pad, 1 link)
    type V4L2 subdev subtype Sensor flags 0
    device node name /dev/v4l-subdev1
    pad0: Source
        [fmt:unknown/1280x720 field:none colorspace:raw xfer:none]
        -> "a0010000.mipi_csi2_rx_subsystem":0 [ENABLED]

- entity 10: a0040000.axis_tkeep_handler (2 pads, 2 links)
    type V4L2 subdev subtype Unknown flags 0
    device node name /dev/v4l-subdev2
    pad0: Sink
        [fmt:unknown/1280x720 field:none colorspace:raw xfer:none]
        <- "a0010000.mipi_csi2_rx_subsystem":1 [ENABLED]
    pad1: Source
        [fmt:unknown/1280x720 field:none colorspace:raw xfer:none]
        -> "a0050000.event_stream_smart_tra":0 [ENABLED]

- entity 13: a0050000.event_stream_smart_tra (2 pads, 2 links)
    type V4L2 subdev subtype Unknown flags 0
    device node name /dev/v4l-subdev3
    pad0: Sink
        [fmt:unknown/1280x720 field:none colorspace:raw xfer:none]
        <- "a0040000.axis_tkeep_handler":1 [ENABLED]
    pad1: Source
        [fmt:unknown/1280x720 field:none colorspace:raw xfer:none]
        -> "ps_host_if output 0":0 [ENABLED]

root@xlnx-docker:/ubuntu#
```

In above media graph imx636 device node is /dev/v4l-subdev1, which is required to enable and running the application.

Run Event File based YoloV7

Run Event camera based YoloV7