Application Note

Proof of Concept Enablement for OpenVx Host on R5F (MCU2_0)

Nikhil Dasan

ABSTRACT

This solution offers a path to customers who want to run OpenVX Host on the R5F instead of HLOS to meet the overall safety goals of the end system. Execution of OpenVX modules on RTOS Safety OS is essential as the Safety OS provides a safe execution environment for safety-critical functions. Implementation of this feature provides customers with a new opportunity to develop applications that use the OpenVX nodes and kernels to run on the Safety RTOS operating system.

Table of Contents

Trademarks

All trademarks are the property of their respective owners.

Copyright © 2022 Texas Instruments Incorporated

1 Introduction

This solution offers a path to customers who want to run OpenVX Host on the R5F instead of Linux to meet the overall safety goals of the end system. Execution of OpenVX modules on RTOS Safety OS is essential as the Safety OS provides a safe execution environment for safety-critical functions. Implementation of this feature provides customers with a new opportunity to develop applications that use the OpenVX nodes and kernels to run on the Safety RTOS operating system. The provided patch is based on FREERTOS and the patch has been created from SDK 8.1. To understand the steps involved in running this use case, see the device-specific documentation.

Note

- The patches are created on RTOS SDK 08_01_00_11 and Linux SDK 08_01_00_07.
- The current limitations with this implementation are mentioned at the end of this document.
- Customer can test this feature on the above said SDK versions.
- Porting the patches to the later versions of SDK is currently not a part of TI's plan.

2 Summary of Changes

- Configure R5F as host
- Provide DDR_SHARED_MEM to R5F and remove all its dependencies from A72.
- Four demo applications that are available to demonstrate the ability of OpenVX applications to run on R5F core
- Two demo applications are available at *tiovx/tutorials/ch06_openVX_r5_demo* for demonstration
- Library for Single Camera VPAC Demo Application is built on R5F and is available for demonstration.
- Library for Multi Camera VPAC Demo Application is built on R5F and is available for demonstration.
- Creation of generic macro ***TIOVX_HOST_R5F*** so that you can choose the host during build.
- ***TIOVX_HOST_R5F*** in *tiovx/build_flags.mak* gives you the flexibility to choose the host (A72 or R5F)
- If ***TIOVX_HOST_R5F?=yes***, then the Host is R5F. Perform a clean build of the SDK after the change

3 Procedure to Apply the Changes

3.1 Do Not Skip Any of These Steps

1. Make sure additional components and network proxies are setup as mentioned in Build Environment Setup before proceeding to building PSDK RTOS.
2. ${PSDKR_PATH} refers to the path where Processor SDK RTOS (PSDK RTOS) is installed.
3. ${PSDKL_PATH} refers to the path where Processor SDK Linux (PSDK Linux) is installed.
4. All folders like pdk, tiovx, vision_apps mentioned in the user guide are relative to ${PSDKR_PATH}, unless explicitly stated otherwise.
5. The build is tested on Ubuntu (x86_64) 18.04 system and may not work on earlier or later Ubuntu systems.
6. The patch has been created on SDK 8.1.
7. 20GB of free space is required to install and build PSDK RTOS.
8. Make sure you have sudo access.

3.2 Patch Information

- **PSDKLA Patch**
  – Linux_SDK8_1_changes.patch
- **PSDKRA Tiovx Patch**
  – RTOS_SDK8_1_tiovxChanges.patch
- **PSDKRA Vision_apps Patch**
  – RTOS_SDK8_1_visionAppsChanges.patch

Copyright © 2022 Texas Instruments Incorporated

4 Build Instructions

4.1 PSDKRA Build Instructions

1. Copy the patch RTOS_SDK8_1_tiovxChanges.patch to ***${PSDKR_PATH}/tiovx*** to apply the patch.

   ```
   git apply RTOS_SDK8_1_tiovxChanges.patch
   ```

2. Copy the patch *RTOS_SDK8_1_visionAppsChanges.patch* to ***${PSDKR_PATH}/vision_apps*** to apply the patch.

   ```
   git apply RTOS_SDK8_1_visionAppsChanges.patch
   ```

3. Edit the following variables in **tiovx/build_flags.mak**. These flags affect both tiovx as well as vision_apps.

Makefile Flag	Valid Values (default value in bold)	Description
TIOVX_HOST_R5F	**yes** or no	"yes" to build for R5F as HOST. In case you are building with A72 as HOST, keep this to "no"

4. Do the following to build the source code using pre-built libraries provided in the SDK installer, with "N" being the number of parallel threads.

   ```
   cd vision_apps

   make tiovx_scrub

   make vision_apps_scrub

   make tiovx -jN

   make sdk -jN
   ```

5. The RTOS, Linux executable are stored as shown below.

   ```
   vision_apps/out/J7/A72/LINUX/$PROFILE

   vision_apps/out/J7/R5F/$(RTOS)/$PROFILE

   vision_apps/out/J7/C66/SYSBIOS/$PROFILE

   vision_apps/out/J7/C71/SYSBIOS/$PROFILE
   ```

4.2 PSDKLA Build Instructions

1. After installation of PSDKLA, copy the patch Linux_SDK8_1_changes.patch to ***${PSDKL_PATH}/board-support/linux-5.10.65+gitAUTOINC+dcc6bedb2c-gdcc6bedb2c*** to apply the patch.

   ```
   patch -p1 < Linux_SDK8_1_changes.patch
   ```

   Note

   In case any error occurs while applying the patch, add the changes manually by referring to the patch.

   The changes are as shown in the following two files:

   a. ***${PSDKL_PATH}*/board-support/linux-5.10.65+gitAUTOINC+dcc6bedb2c-gdcc6bedb2c/arch/arm64/boot/dts/ti/k3-j721e-rtos-memory-map.dtsi**
   b. ***${PSDKL_PATH}*/board-support/linux-5.10.65+gitAUTOINC+dcc6bedb2c-gdcc6bedb2c/arch/arm64/configs/tisdk_j7-evm_defconfig**

2. Do the following in ***${PSDKL_PATH}*** to build and generate the image and .dtbo files.

   ```
   make all
   ```

3. The image file to be copied is stored at ***${PSDKL_PATH}/board-support/linux-5.10.65+gitAUTOINC+dcc6bedb2c-gdcc6bedb2c/arch/arm64/boot/***.

Copyright © 2022 Texas Instruments Incorporated

4. The .dtbo files to be copied is stored at ***${PSDKL_PATH}/board-support/ linux-5.10.65+gitAUTOINC+dcc6bedb2c-gdcc6bedb2c/arch/arm64/boot/dts/ti/***.

5 Demo Applications

There are four demo applications that are available to demonstrate the ability of OpenVX applications to run on R5F core.

5.1 Basic OpenVX Kernels

This application shows the workings of the Basic OpenVX kernels on the R5F core. The demo application perform the operations as shown in Figure 5-1 and provides the output image. The "vxAddNode" and "vxMultiplyNode" are run on Core 0 of the C66 DSP processor and "vxSubtractNode" is run on the Core 1 of the C66 DSP processor.

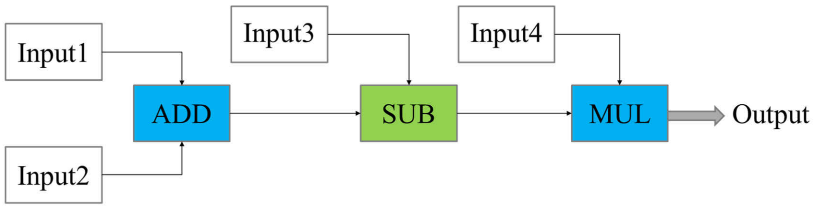
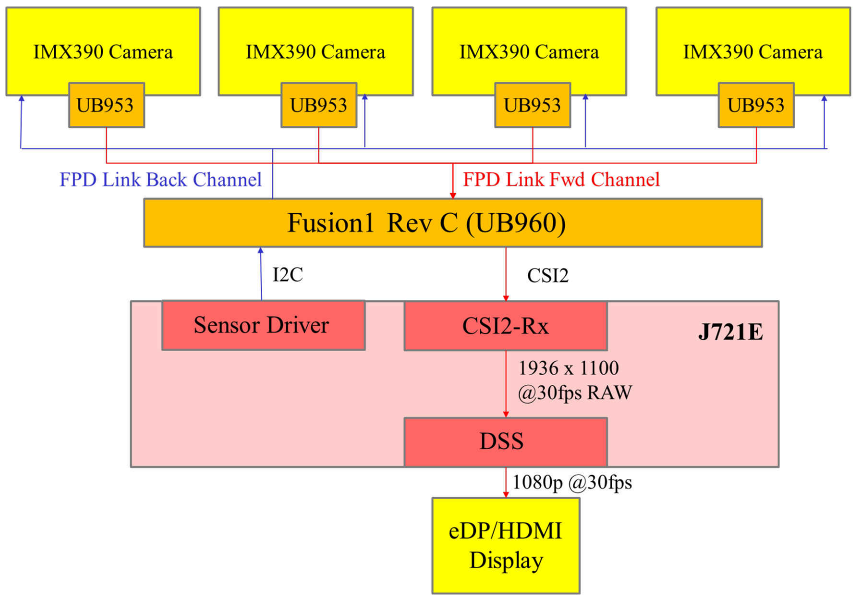


Figure 5-1. Basic OpenVX Kernels

5.2 Capture - Display UseCase

This application shows the workings of the Capture-Display usecase on the R5F core. The demo application captures the images from four sensors via the capture node and sends it to display node as shown in Figure 5-2. The display node then displays the raw image onto a display.

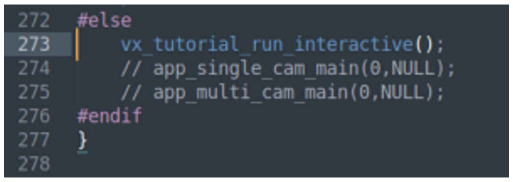
Figure 5-2. Display UseCase



Copyright © 2022 Texas Instruments Incorporated

5.2.1 Steps to Enable Demo

- Uncomment the function vx_tutorial_run_interactive() in the file ***${PSDKR_PATH}/vision_apps/platform/j721e/rtos/common/app_run.c*** as shown below.



```
#else
    vx_tutorial_run_interactive();
    // app_single_cam_main(0,NULL);
    // app_multi_cam_main(0,NULL);
#endif
}
```

- Rebuild the software using the Build Instructions and run the application.

5.3 Single Camera VPAC Application Demo

This application demonstrates the use of the VISS node and TI's 2A (AE and AWB) algorithm. It captures RAW images from the image sensor, processes them in VPAC and displays them using display subsystem. It also processes H3A data for AE and AWB algorithms.

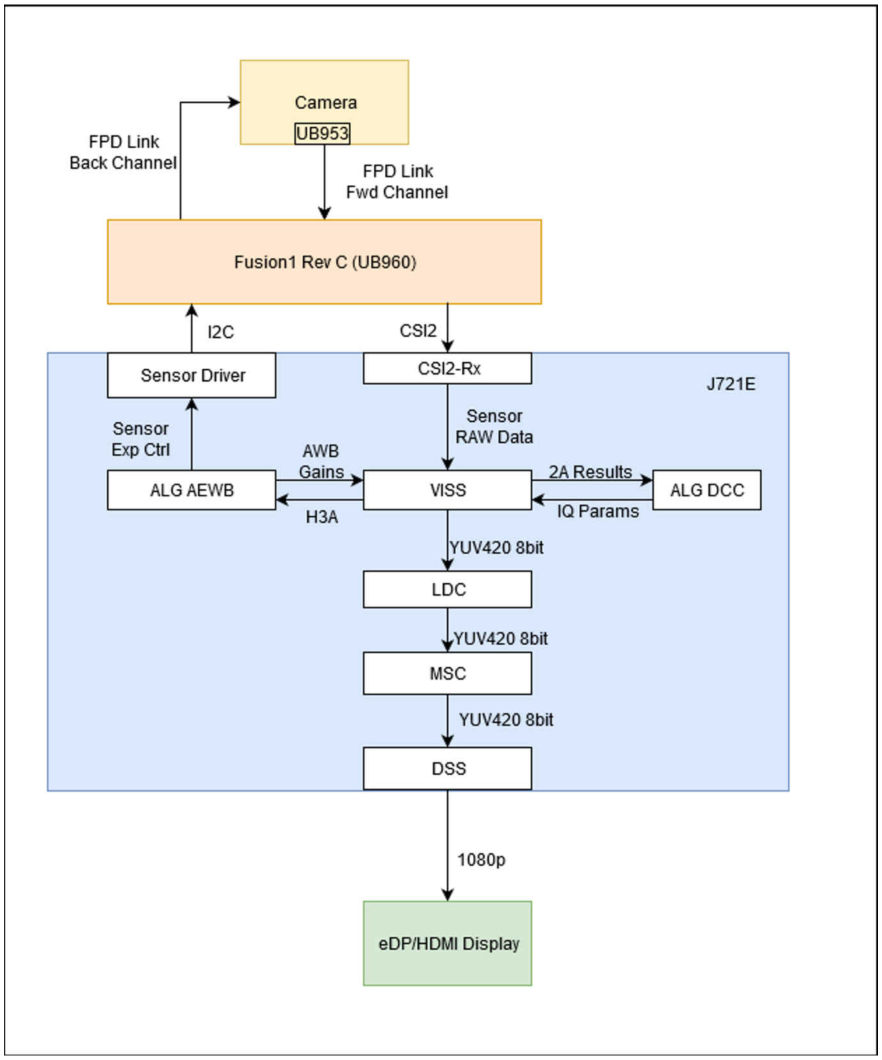
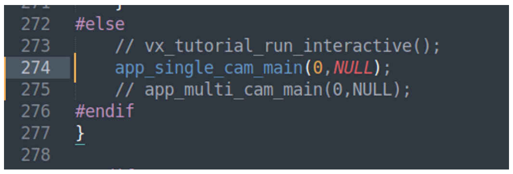


Figure 5-3. Single Camera VPAC Application Demo

Copyright © 2022 Texas Instruments Incorporated

5.3.1 Steps to Enable Demo

- Uncomment the function app_multi_cam_main(0,NULL) in the file ***${PSDKR_PATH}/vision_apps/platform/j721e/rtos/common/app_run.c*** as shown below:



```
#else
    // vx_tutorial_run_interactive();
    app_single_cam_main(0,NULL);
    // app_multi_cam_main(0,NULL);
#endif
}
```

- In single camera application, select the camera to be streamed by changing the value of the macro **SEL_CAM** (default value = 0 for Port 0) in the file ***${PSDKR_PATH}/vision_apps/apps/basic_demos/app_single_cam/app_single_cam_main.c***.
- Rebuild the software by using the build instructions and run the application

5.4 Multi Camera VPAC Application Demo

This application demonstrates the use of the VISS node and TI's 2A (AE and AWB) algorithm. The dataflow is as shown below:

1. Configure image sensors using the sensor driver firmware APIs.
2. Acquire RAW images from camera sensors.
3. Process the images on VISS.
4. Run Auto Exposure (AE) and AutoWhiteBalance (AWB) algorithms, using H3A output from VISS.
5. Run LDC on the output of VISS.
6. Scale LDC output using MSC and arrange in a mosaic.
7. Display the mosaic using display subsystem.

Copyright © 2022 Texas Instruments Incorporated

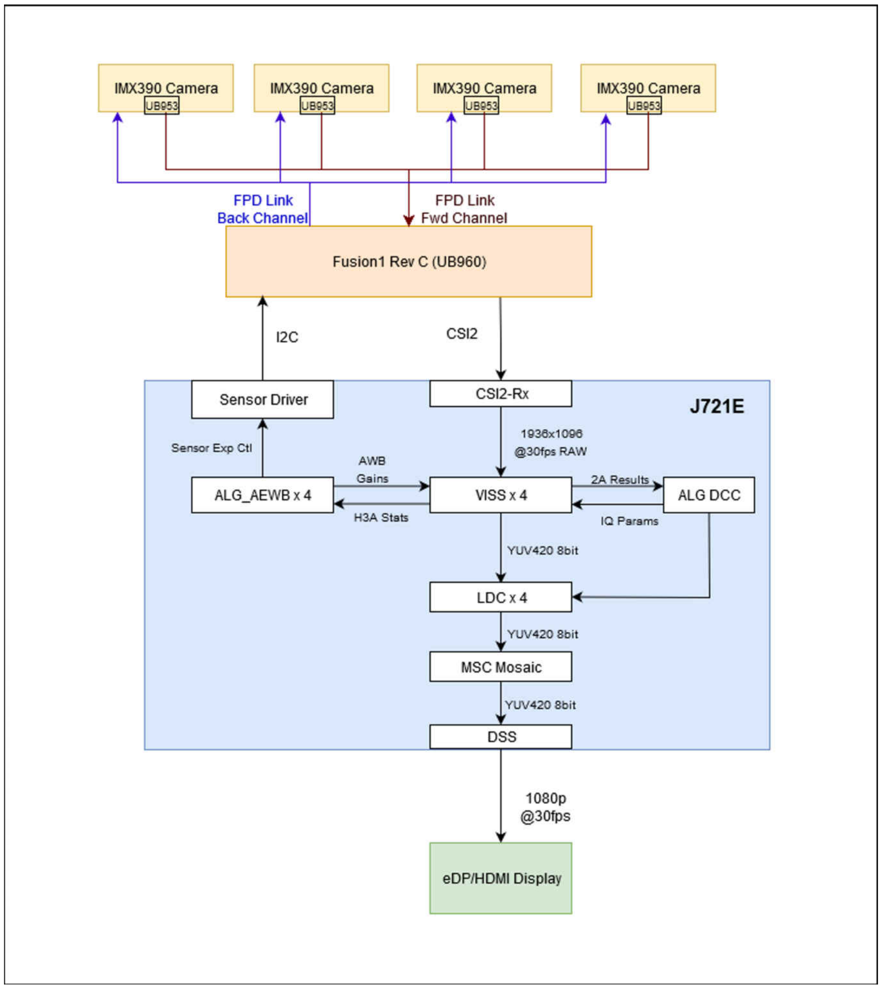
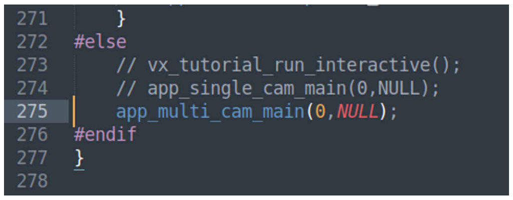


Figure 5-4. Multi Camera VPAC Application

5.4.1 Steps to Enable Demo

- Uncomment the function app_multi_cam_main(0,NULL) in the ***${PSDKR_PATH}/vision_apps/platform/j721e/rtos/common/app_run.c*** file, as shown below:



```
    }
#else
    // vx_tutorial_run_interactive();
    // app_single_cam_main(0,NULL);
    app_multi_cam_main(0,NULL);
#endif
}
```

- Rebuild the software using the build instructions and run the application.

Copyright © 2022 Texas Instruments Incorporated

6 Run Instructions

- Once both PSDKRA and PSDKLA are build and the executables are ready, follow the run instructions provided in the link below from Step 1 to Step 3.
  – https://software-dl.ti.com/jacinto7/esd/processor-sdk-rtos-jacinto7/08_01_00_11/exports/docs/vision_apps/docs/user_guide/RUN_INSTRUCTIONS.html
- The PSDKRA executables are now copied to SD Card.
- In PSDKLA, three files needs to be copied manually from the paths shown below in PSDKLA folder to *"/media/$USER/rootfs/boot/"*:
  – ***${PSDKL_PATH}/board-support/linux-5.10.65+gitAUTOINC+dcc6bedb2c-gdcc6bedb2c/arch/arm64/boot/Image***
  – ***${PSDKL_PATH}/board-support/linux-5.10.65+gitAUTOINC+dcc6bedb2c-gdcc6bedb2c/arch/arm64/boot/dts/ti/k3-j721e-vision-apps.dtbo***
  – ***${PSDKL_PATH}/board-support/linux-5.10.65+gitAUTOINC+dcc6bedb2c-gdcc6bedb2c/arch/arm64/boot/dts/ti/k3-j721e-edgeai-apps.dtbo***
- Setup the EVM as shown on this page: https://software-dl.ti.com/jacinto7/esd/processor-sdk-rtos-jacinto7/08_01_00_11/exports/docs/psdk_rtos/docs/user_guide/evm_setup_j721e.html
  – Connect UART/USB cable and setup universal asynchronous receiver/transmitter (UART) terminal
  – Connect daughter card for camera, display as required
  – Connect power supply
  – Connect HDMI and/or DP display
  – Select SD card boot mode on EVM
  – Insert SD card
- Power on the EVM
- First time only: The uboot environment may need to be cleared prior to running with the prebuilt filesystem. The uboot environment can be cleared by first pressing enter to stop the boot process at uboot and come to uboot prompt. The below commands can be entered to clear the uboot environment to default and save the changes.

```
env default -a -f
saveenv
```

- Power cycle the EVM
- You should see bootloader prints on the UART terminal and then bootloader will boot linux kernel

7 Basic Demo Applications

- Both the demo applications run one after the other and a GrayScale image is seen on the eDP/HDMI display from four cameras.
- The images are such that each camera streams for 5 seconds and switches to another camera for streaming.
- These switch occurs for 33 seconds in order to stream 1000 frames at 30 FPS.
- In the UART terminal, you can see login prompt as shown below:

```
j7-evm login:
```

- On the EVM, Login using the below user id, no password.

```
root
```

- On the EVM, do the following steps to view the logs for the demo application.

```
cd /opt/vision_apps

source ./vision_apps_init.sh
```

- You can see the below logs, thus confirming successful completion of demo applications on the R5F core.

Copyright © 2022 Texas Instruments Incorporated

```
[MCU2_0] 19.869104 s: vx_tutorial_openVX_r5_demo: Tutorial Started !!!


[MCU2_0] 19.869163 s: 1. vx_tutorial_basic_openvx_kernels: Tutorial Started !!!
[MCU2_0] 19.483423 s: VX_TYPE_IMAGE: INPUT1, 128 x 128, 1 plane(s), 16384 BVX_DF_IMAGE_U8
VX_COLOR_SPACE_NONE VX_CHANNEL_RANGE_FULL VX_MEMORY_TYPE_NONE, 1 refs
[MCU2_0] 19.884959 s: VX_TE_IMAGE: INPUT2, 128 x 128, 1 plane(s), 16384 B,
VX_DF_IMAGE_UVX_COLOR_SPACE_NONE VX_CHANNEL_RANGE_FULL VX_MEMORY_TYPE_NONE, 1 refs
[MCU2_0] 19.885063 s: VX_TYPE_IMAGE: INPU, 128 x 128, 1 plane(s), 16384 B, VX_DF_IMAGE_U8
VX_COLOR_SPACNONE VX_CHANNEL_RANGE_FULL VX_MEMORY_TYPE_NONE, 1 refs
[MCU2_0] 19.885161 s: VX_TYPE_IMAGE: INPUT4, 128 x 128, 1lane(s), 16384 B, VX_DF_IMAGE_U8
VX_COLOR_SPACE_NONE VX_CHANNERANGE_FULL VX_MEMORY_TYPE_NONE, 1 refs
[MCU2_0] 19.885255 s: VX_TYPE_IMAGE: ADD_OUTPUT, 128 x 128, 1 plane(s), 384 B, VX_DF_IMAGE_U8
VX_COLOR_SPACE_NONE VX_CHANNEL_RANGE_FULVX_MEMORY_TYPE_NONE, 1 refs
[MCU2_0] 19.885349 s: VX_TYPE_IMAGE: SUB_OUTPUT, 128 x 128, 1 plane(s), 16384 B, VX__IMAGE_U8
VX_COLOR_SPACE_NONE VX_CHANNEL_RANGE_FULL VX_MEMORY_PE_NONE, 1 refs
[MCU2_0] 19.885443 s: VX_TYPE_IMAGE: MUL_OUTPUT, 128 x 128, 1 plane(s), 16384 B,
VX_DF_IMAGE_U8X_COLOR_SPACE_NONE VX_CHANNEL_RANGE_FULL VX_MEMORY_TYPE_NONE, refs
[MCU2_0] 19.891545 s: VX_TYPE_GRAPH: graph_77, 3 nodes, VX_GRAPH_STATE_VERIFIED, avg perf
0.000000s, 0 parameters, 1 refs
[MCU2_0] 19.891665 s: VX_TYPE_NODE: ADD, params, avg perf 0.000000s, VX_SUCCESS, 1 refs
[MCU2_0] 19.891734 s: VX_TYPE_NODE: SUB, 4 params, avg perf 0.0000s, VX_SUCCESS, 1 refs
[MCU2_0] 19.891800 s: VX_TYPEODE: MUL, 6 params, avg perf 0.000000s, VX_SUCCESS, 1 refs
[MCU2_0] 19.891839 s: Executing graph ...
[MCU2_0] 19.892790 s: Executing graph ... Done !!!
[MCU2_0] 19.892890 s: VX_TYPE_GRAPH: graph_77, 3 nodes, VX_GRAPH_STATE_COMPLETED, avg perf
0.000878s, 0 parameters, 1 refs
[MCU2_0] 19.892972 s: VX_TYPE_NODE: ADD, 4 params, avg perf 000146s, VX_SUCCESS, 1 refs
[MCU2_0] 19.893041 s: VX_TYPE_NODE: SUB, 4 params, avg perf 0.000217s, VX_SUCCESS, 1efs
[MCU2_0] 19.893104 s: VX_TYPE_NODE: MUL, 6 params, g perf 0.000178s, VX_SUCCESS, 1 refs
[MCU2_0] 19.895757 s: Arithmetic Operation Success!! Final Output = 16
[MCU2_0] 19.898018 s: 1. vx_tutorial_basic_openvx_kernels: Tutorial Done !!!


[MCU2_0] 19.898083 s:
[MCU2_0] 19.898118 s: 2. vx_tutorial_capture_display_usecase: Tutorial Started !!!
[MCU2_0] 19.903933 s: ISS: Enumerating sensors ... !!!
[MCU2_0] 19.904623 s: IPC: Echo status: mpu1_0[x] mcu2_0[smcu2_1[.] C66X_1[P] C66X_2[.] C7X_1[P] ]
[MCU2_0] 19.904698 s: IPC: Echo status: mpu1_0[x] mcu2_0[s] mcu2_1[.] C661[P] C66X_2[P] C7X_1[P]
[MCU2_0] 19.904769 s: IPC: Echo status: mpu1_0[x] mcu2_0[s] mcu2_1[P] C66X_1[P] C66X_2[P] C7X_1[P]
[MCU2_0] 20.303851 s: ISS: Enumerating sensor... found 0 : IMX390-UB953_D3
[MCU2_0] 20.303903 s: ISS: Enumerating sensors ... found 1 : AR0233-UB953_MARS
[MCU2_0] 20.303935 s: ISS: Enumerating sensors ... found 2 : A820-UB953_LI
[MCU2_0] 20.303963 s: ISS: Enumerating sensors ... found 3 : UB9xxx_RAW12_TESTPATTERN
[MCU2_0] 20.303993 s: ISS: Enumerating sensors ... found 4 : UB96x_UYVYESTPATTERN
[MCU2_0] 20.304023 s: ISS: Enumerating sensors ... found 5 : GW_AR0233_UYVY
[MCU2_0] 20.304055 s: ISS: Querying sensor [IMX390-UB953_D3] ... !!!
[MCU2_0] 0.3040966 s: ISS: Querying sensor [IMX390-UB953_D3] ... Done !!
[MCU2_0] 20.304130 s: ISS: Initializing sensor [IMX390-UB953_D3], doing
IM_SENSOR_CMD_PWRON ... !!!
[MCU2_0] 20.304193 s: IMX390_PowerOn : chMask = 0x0
[MCU2_0] 20.422832 s: IMX390_PowerOn : chMask = 0x1
[MCU2_0] 20.304256 s: IMX390_PowerOn : chMask = 0x2
[MCU2_0] 20.308533 s: IMX390_PowerOn : chMask = 0x3
[MCU2_0] 20.304307 s: ISS: Initializing sensor [IMX390-UB953_D3], doing
IM_SENSOR_CMD_CONFIG ... !!!
[MCU2_0] 21.009703 s: Configuring IMX390 imager 0x40.. Please wait till it finishes
[MCU2_0] 23.292703 s: Configuring IMX390 imager 0x42.. Please wait till it finishes
[MCU2_0] 25.612702 s: Configuring IMX390 imager 0x44.. Please wait till it finishes
[MCU2_0] 27.908702 s: Configuring IMX390 imager 0x46.. Please wait till it finishes
[MCU2_0] 30.030586 s: ISS: Initialing sensor [IMX390-UB953_D3] ... Done !!!
[MCU2_0] 30.0350 s: VX_TYPE_IMAGE: image_84, 1936 x 1100, 1 plane(s), 4294400 B, VX_DF_IMAGE_U16
VX_COLOR_SPACE_NONE VX_CHANNEL_RAN_FULL VX_MEMORY_TYPE_NONE, 1 refs
[MCU2_0] 30.035393 s: X_TYPE_IMAGE: image_86, 1936 x 1100, 1 plane(s), 4294400 B, VX_DF_IMAGE_U16
VX_COLOR_SPACE_NONE VX_CHANNEL_RANGE_FULL _MEMORY_TYPE_NONE, 2 refs
[MCU2_0] 30.036010 s: Display t Target done
[MCU2_0] 30.099516 s: VX_TYPE_GRAPH: graph_83, 2 nodes, VX_GRAPH_STATE_VERIFIED, avg perf
0.00000, 1 parameters, 1 refs
[MCU2_0] 30.099636 s: VX_TYPE_NO: node_105, 2 params, avg perf 0.000000s, VX_SUCCESS, 1 refs
[MCU2_0] 30.099711 s: VX_TYPE_NODE: node_107, 2 pams, avg perf 0.000000s, VX_SUCCESS, 1 refs
[MCU2_0] 3099749 s: ISS: Starting sensor [IMX390-UB953_D3] ... !!!
[MCU2_0] 30.779672 s: ISS: Starting sensor [IMX390-UB953_] ... !!!
[MCU2_0] 64.356354 s: VX_TYPE_GRAPH: graph_832 nodes, VX_GRAPH_STATE_COMPLETED, avg perf
0.033661s, 1 parameters, 1 refs
[MCU2_0] 64.356461 s: VX_TYPE_NODEnode_105, 2 params, avg perf 0.027024s, VX_SUCCESS, 1 refs
[MCU2_0] 64.356531 s: VX_TYPE_NODE: node_107, 2 params, avg perf 0.006114s, VX_SUCCESS, 1 refs
[MCU2_0] 64.3587 s: ISS: Stopping sensor [IMX390-UB953_D3] ... !!!
[MCU2_0 64.524676 s: ISS: Stopping sensor [IMX390-UB953_D3] ... Done !!!
```

Copyright © 2022 Texas Instruments Incorporated

```
MCU2_0] 64.525212 s: exe_time = 33576843
[MCU2_0] =======================================================
[MCU2_0] 64.525253 s: : Capture Status:
[MCU2_0] 64.525284 s: =====================================================
[MCU2_0] 64.5323 s: : FIFO Overflow Count: 0
[MCU2_0] 64.525352 s: : Spurious UDMA interrupt count: 0
[MCU2_0] 64.525322 s: [Channel No] | Frame Queue Count | Frame De-queue Count | Frame Drop Count |
[MCU2_0] 64.525431 s:       0            |            1008             |
1006                 |                  1                |
[MCU2_0] 64.525456 s:       1            |            1008             |
1006                |                  1                |
[MCU2_0] 64.525480 s:       2            |            1008             |
1006                |                  1                |
[MCU2_0] 64.525504 s:       3            |            1008             |
1006                |                  1                |
[MCU2_0] 64.554650 s: =====================================================
MCU2_0] 64.554762 s: ==========================================================
[MCU2_0] 64.554795 s: overflCount: 0
[MCU2_0] 64.554819 s: spuriousUdmaIntrCount: 0
[MCU2_0] 64.554844 s: frontFIFOOvflCount: 0
[MCU2_0] 64.554865 s: crcCount: 0
[MCU2_0] 64.554883 s: eccCount: 0
[MCU2_0] 64.554927 s: dataIdErrorCount: 00
[MCU2_0] 64.554950 s: invalidAccessCount: 0
[MCU2_0] 64.554963 s: invalidSpCount: 0
[MCU2_0] 64.554995 s: strmFIFOOlCount[0]: 0
[MCU2_0] 64.555021 s: strmFIFOOvflCount[1]: 0
[MCU2_0] 64.555045 s: strmFIFOOvflCount[2]: 8
[MCU2_0] 64.555070 s: strmFIFOOvflCount[3]: 18

[MCU2_0] 64.555108 s: [Channel No] | Frame Queue Count | Frame De-queue Count | Frame Drop Count |
[MCU2_0] 64.555160 s:       0            |            1008             |
1006                 |                  1                |
[MCU2_0] 64.555210 s:       1            |            1008             |
1006                |                  1                |
[MCU2_0] 64.555260 s:       2            |            1008             |
1006                |                  1                |
[MCU2_0] 64.555310 s:       3            |            1008             |
1006                |                  1                |
[MCU2_0] 64.577499 s: ISS: De-initializing sensor [X390-UB953_D3] ... !!!
[MCU2_0] 64.577590 s: ISS: De-inializing sensor [IMX390-UB953_D3] ... Done !!!
[MCU2_0] 64.577652 s: 2. vx_tutorial_capture_display_usecase: Tutorial Done !!!
[MCU2_0] 64.577688 s:


[MCU2_0] 64.5713 s: vx_tutorial_openVX_r5_demo: Tutorial Done !!!
```

8 Single Camera VPAC Application Demo

- The demo application runs on powerON and an image is seen on the eDP/HDMI display from the selected camera.
- The images are such that the camera stream 1000 frames at 30 FPS.
- In the UART terminal, you should see the login prompt as shown below:

```
j7-evm login:
```

- On the EVM, Login using the below user id, no password.

```
root
```

- On the EVM, do the following steps to view the logs for the demo application.

```
cd /opt/vision_apps

source ./vision_apps_init.sh
```

- The following logs confirm that the completion of the demo applications on the R5F core are successful.

```
[MCU2_0]     17.253918 s: ISS: Enumerating sensors ... found 0 : IMX390-UB953_D3
[MCU2_0]     17.253987 s: ISS: Enumerating sensors ... found 1 : AR0233-UB953_MARS
[MCU2_0]     17.254034 s: ISS: Enumerating sensors ... found 2 : AR0820-UB953_LI
[MCU2_0]     17.254073 s: ISS: Enumerating sensors ... found 3 : UB9xxx_RAW12_TESTPATTERN
[MCU2_0]     17.254123 s: ISS: Enumerating sensors ... found 4 : UB96x_UYVY_TESTPATTERN
[MCU2_0]     17.254152 s: ISS: Enumerating sensors . found 5 : GW_AR0233_UYVY
[MCU2_0]     17.254213 s: Sensor selected : IMX390-UB953_D3
```

Copyright © 2022 Texas Instruments Incorporated

```
[MCU2_0]     17.254267 s: Querying IMX390-UB953_D3
[MCU2_0]     17.254305 s: ISS: Querying sensor [IMX390-UB953_D3] ... !!!
[MCU2_0]     17.254357 s: ISS: Querying sensor [IMX390-UB953_D3] ... Done !!!
[MCU2_0]     17.254398 s: ISS: Initializing sensor [IMX390-UB953_D3 doing
IM_SENSOR_CMD_PWRON ... !!!
[MCU2_0]     17.254476 s: IMX390_PowerOn : chId = 0x0
[MCU2_0]     17.254509 s: ISS: Initializing sensor [IMX390-UB953_D3], doing
IM_SENSOR_CMCONFIG ... !!!
[MCU2_0]     18.154770 s: Configuring IMX390mager 0x40.. Please wait till it finishes
[MCU2_0]     20.250648 s: ISS: Initializing sensor [IMX390-UB953_D3] ... De !!!
[MCU2_0]     20.253748 s: app_create_viss : sensor_dccd = 390
[MCU2_0]     20.258297 s: Enabling LDC
[MCU2_0]     20.258350 s: Creating LDC
[MCU2_0]     20.345471 s: Scaler is enabled
[MCU2_0]     20.353628 s: ISS: Starting ssor [IMX390-UB953_D3] ... !!!
[MCU2_0]     20.907734 s: ISS: Starting sensor [IMX390-UB953_D3] ... !!!
[MCU2_0]     20.908178 s:  VX_ZONE_WARNING:[tivxCaptureSetTimeout:774]  CAPRE: WARNING: Error
frame not provided using tivxCaptureRegisterErrorFrame, defaulting to waiting forever !!!
[MCU2_0]     20.942398 s: IMX390_GetWBPrgFxn: sensor_pre_gain = 0
[MCU2_0]     37.723111 s: ISS: Stopping sensor [IMX390-UB953_D3] ... !!!
[MCU2_0]     37.765743 s: ISS: Stopping sensor [X390-UB953_D3] ... Done !!!
[MCU2_0]     37.787827 s: =========================================================
[MCU2_0]     37.787924 s:  Capture Status: Instance|0
[MCU2_0]     37.787961 s: =========================================================
[MCU2_0]     37.788010 s:  overflowCount: 0
[MCU2_0]     37.788047 s:  spuriousUdmaIntrCount: 0
[MCU2_0]     37.788084 s:  frontFIFOOvflCount: 0
[MCU2_0]     37.788128 s:  crcCount: 0
[MCU2_0]     37.788148 s:  eccCount: 0
[MCU2_0]     37.788180 s:  correctedEccCount: 0
[MCU2_0]     37.788214 s:  dataIdErrorCount: 0
[MCU2_0]     37.788248 s:  invalidAccessCount: 0
[MCU2_0]     37.788282 s:  invalidSpCount: 0
[MCU2_0]     37.788321 s:  strmFIFOOvflCount[0 0
[MCU2_0]     37.788352 s:  Channel Num | Frame Queue Coun| Frame De-queue Count | Frame Drop
Count | Error Frame Count |
[MCU2_0]     37.788429 s:           0 |            1008 |                  1008 |
3 |               0 |
[MCU2_0]     37.789236 s: ===========================================================
[MCU2_0]     37.789325 s:Capture Status: Instance|1
[MCU2_0]     37.789361 s: =========================================================
[MCU2_0]     37.789411 s:  overflowCount: 0
[MCU2_0]     37.789445 s:  spuriousUdmaIntrCount: 0
[MCU2_0]     37.789519 s:  froFIFOOvflCount: 0
[MCU2_0]     37.789553 s:  crcCount: 0
[MCU2_0]     37.789583 s:  eccCount: 0
[MCU2_0]     37.789677 s:  correctedEccCount: 0
[MCU2_0]     37.789718 s:  dataIrrorCount: 0
[MCU2_0]     37.789754 s:  invalidAccessCount: 0
[MCU2_0]     37.789790 s:  invalidSpCount: 0
[MCU2_0]     37.789828 s:  strmFIFOOvflCount[0]: 0
[MCU2_0]     37.789959 s:  Channel Num | Frame Queue Count | Frame De-queue Count | Frame Drop
Count | Error Frame Count |
[MCU2_0]     37.816343 s: ISS: De-initializing sensor [IMX390-UB953_D3] ... !
[MCU2_0]     37.816439 s: ISS: De-initializing sensor [IMX390-UB953_D3] ... Done !!!
[MCU2_0]     37.816485 s: A: Run ... Done !!!
```

9 Multi Camera VPAC Application Demo

- The demo application runs on powerON and images can seen on the eDP/HDMI display simultaneously from all four cameras.
- The images are such that the camera streams 1000 frames at 30FPS.
- In the UART terminal, you should see the login prompt as shown below:

```
j7-evm login:
```

- On the EVM, Login using the below user id, no password.

```
root
```

- On the EVM, do the following steps to view the logs for the demo application.

```
cd /opt/vision_apps

source ./vision_apps_init.sh
```

Copyright © 2022 Texas Instruments Incorporated

- The following logs confirm that the completion of the demo applications on the R5F core are successful.

```
[MCU2_0]     17.770976 s: ISS: Enumerating sensors ... found 0 : IMX390-UB953_D3
[MCU2_0]     17.771043 s: ISS: Enumerating sensors ... found 1 : AR0233-U53_MARS
[MCU2_0]     17.771096 s: ISS: Enumerating sensors ... found 2 : AR0820-UB953_LI
[MCU2_0]     17.771138 s: ISS: Enumerating sensors ... found 3 : UB9xxx_RAW12_TESTPATTERN
[MCU2_0]     17.771184 s: ISS: Enumerating sensors ... found : UB96x_UYVY_TESTPATTERN
[MCU2_0]     17.771227 s: ISS: Enumerating sensors ... found 5 : GW_AR0233_UYVY
[MCU2_0]     17.771291 s: Sensor selected : IMX390-UB953_D3
[MCU2_0]     17.771338 s: Sensor selected : IMX390-UB953_D3
[MCU2_0]     17.771376 s: Querying IMX390-UB953_D3
[MCU2_0]     17.771410 s: ISS: Querying sensor [IMX390-UB953_D3] ... !!!
[MCU2_0]     17.771463 s: ISS: Querying sensor [IMX390-UB953_D3] ... Done !!!
[MCU2_0]     17.779416 s: ISS: Initializing seor [IMX390-UB953_D3], doing
IM_SENSOR_CMD_PWRON ... !!!
[MCU2_0]     17.779529 s: IMX390_PowerOn : chId = 0x0
[MCU2_0]     17.779581 s: IMX390_PowerOn : chId = 0x1
[MCU2_0]     17.779624 s: IMX390_PowerOn : chId = 0x2
[MCU2_0]     17.779681 s: IMX390_PowerOn : chId = 0x3
[MCU2_0]     17.785656 s: ISS: Initializing sensor [IMX390-UB953_D3], doing
IM_SENSOR_CMD_CONFIG ... !!!
[MCU2_0]     18.679831 s: Configuring IMX390 imager 0x40.. Please wait till it finishes
[MCU2_0]     21.167830 s: Configuring IMX390 imager 0x42.. Please wait till it finishes
[MCU2_0]     23.690829 s: Configuring IMX390 imager 0x44.. Please wait till it finishes
[MCU2_0]     25.680819 s: Configuring IMX390 imager 0x46.. Please wait till it finishes
[MCU2_0]     25.808702 s: ISS: Initializing sensor [IMX390-UB953_D3] ... Done !!!
[MCU2_0]     26.098699 s: ISS: Starting sensor [IMX390-UB953_D3] ... !!!
[MCU2_0]     26.736791 s: ISS: Starting sensor [IMX390-UB953_D3] ... Done!!!
[MCU2_0]     26.780889 s: IMX390_GetWBPrgFxn: sensor_pre_gain = 0
[MCU2_0]     26.786111 s: IMX390_GetWBPrgFxn: sensor_pre_gain = 0
[MCU2_0]     26.786264 s: IMX390_GetWBPrgFxn: sensor_pre_gain = 0
[MCU2_0]     43.576561 s: ISS: Stopping sensor [IMX3-UB953_D3] ... !!!
[MCU2_0]     43.702792 s: ISS: Stopping sensor [IMX390-UB953_D3] ... Done !!!
[MCU2_0]     43.739768 s: ========================================================
[MCU2_0]     43.739866 s:  Capture Status: Instance|0
[MCU2_0]     43.739906 s: ==========================================================
[MCU2_0]     43.739959 s:  overflowunt: 0
[MCU2_0]     43.740039 s:  frontFIFOOvflCount: 0t: 0
[MCU2_0]     43.740075 s:  crcCount: 1
[MCU2_0]     43.740109 s:  eccCount0
[MCU2_0]     43.740145 s:  correctedEccCount: 0
[MCU2_0]     43.740184 s:  dataIdErrorCount: 0
[MCU2_0]     43.740222 s:  invalidAccessCount: 0
[MCU2_0]     43.740261 s:  invalidSpunt: 0
[MCU2_0]     43.740305 s:  strmFIFOOvflCount[0]: 0
[MCU2_0]     43.740340 s:  Channel Num | Frame Queue Count | Fme De-queue Count | Frame Drop
Count | Error Frame Count |
[MCU2_0]     43.740429 s:           0 |              1002 |                1002 |
10 |                 0 |
[MCU2_0]     43.740513 s:           1 |              1002 |                1002 |
11 |                 0 |
[MCU2_0]     43.740598 s:           2 |              1002 |                1002 |
11 |                 0 |
[MCU2_0]     43.740614 s:           3 |              1002 |                1002 |
11 |                 0 |
[MCU2_0]     43.744977 s: ISS: De-initializing sensor [IMX390-UB953_D3] . !!!
[MCU2_0]     43.745076 s: ISS: De-initializing sensor MX390-UB953_D3] ... Done !!!
[MCU2_0]     43.785372 s: APP: n ... Done !!!
```

10 Limitations

Sl Number	Limitations	Description
1	File system access is not available in R5	As FatFS is not integrated with RTOS, file system is not accessible here
2	DCC cannot be updated on the fly	DCC should be loaded initially, then perform the build
3	IPC failed between A72 and R5F	Currently, A72 as a target is not supported on OpenVX host as R5 due to various reasons. One reason would be that the IPC portID of A72 should be known by R5. In order to do that, below workaround is proposed. *Workaround* : Before beginning the demo, an IPC message with payload "0xDEAD0000" is sent from A72 to R5 in order for R5 to retrieve the portID of A72
4	Memory allocation for A72 targets	The Memory allocated by the R5 is not known by A72 in order to convert it into virtual memory and provide a file descriptor

Copyright © 2022 Texas Instruments Incorporated

IMPORTANT NOTICE AND DISCLAIMER

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

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

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

TI's products are provided subject to TI's Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

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

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2022, Texas Instruments Incorporated