SRV Optimizations SDK 8.2

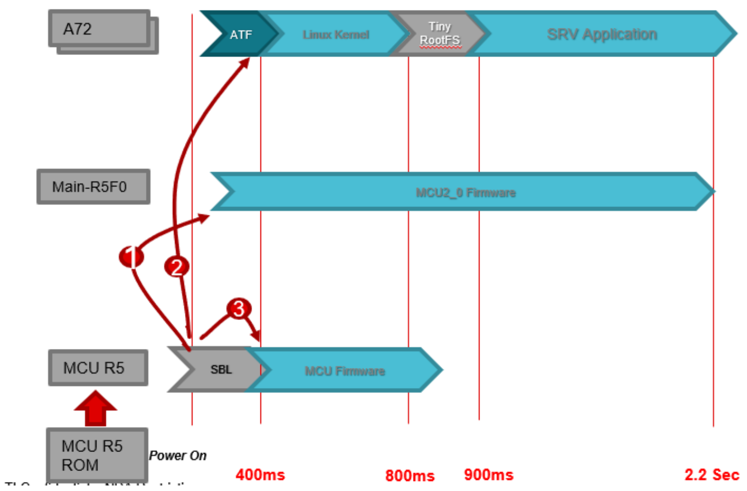
Jacinto Software Apps

Exported on 06/20/2022

Table of Contents

Abstract:

Surround view is application involves multiple cores running in parallel. MCU R5F starts with booting the device, A72 to triggering the vision apps application, R5F starting the camera & display. Using the default Linux SDK with vision_apps the time taken to achieve surround view is of the order of 25S as depicted in the below picture. The application report focuses on optimizing the time to get out the first frame of surround view to display in ~2.2 Seconds.



Optimizations:

The application report addresses the optimizations methodically at every stage of boot.

- Bootloader switch to SBL from OSPI boot media.
- Linux device tree optimizations
- File system switch to tiny rootfs
- Vision_apps Application re-design

/*<![CDATA[*/ div.rbtoc1655723597713 {padding: 0px;} div.rbtoc1655723597713 ul {list-style: disc;margin-left: 0px;} div.rbtoc1655723597713 li {margin-left: 0px;padding-left: 0px;} /*]]>*/ Abstract: Optimizations in Linux(PSDKLA): Optimizations

- Camera driver optimizations.
- ***Eventually leading to Surround view demo in ~2Seconds from power-on.***

1.1 Optimizations in Linux(PSDKLA):

8.2-srv-linux.zip

Download the attached tar ball to: $PSDK_Linux/board-support/linux-5.10.100+gitAUTOINC+7a7a3af903-g7a7a3af903

```
cd $PSDKLA_PATH/board-support/linux-5.10.100+gitAUTOINC+7a7a3af903-g7a7a3af903
unzip 8.2-srv-linux.zip
git am 8.2-srv-linux/0001-vision_apps-Integrated.patch
git am 8.2-srv-linux/0002-arch-arm64-boot-dts-ti-k3-j721e-common-proc-board.dt.patch
cd ..
make linux
```

Note: Apply 8.0-srv/0003-arch-arm64-boot-dts-ti-k3-j721e-common-proc-board.dt.patch Only after you shift the file system to eMMC as a last step.

1.2 Optimizations in Imaging(PSDK RTOS):

```
cd $PSDKRA_PATH/imaging
git init
git add .
git commit -m "Initial commit"
```

/*<![CDATA[*/ div.rbtoc1655723597713 {padding: 0px;} div.rbtoc1655723597713 ul {list-style: disc;margin-left: 0px;} div.rbtoc1655723597713 li {margin-left: 0px;padding-left: 0px;} /*]]>*/ Abstract: Optimizations in Linux(PSDKLA): Optimizations

Download the attached tar ball to: $PSDKRA_PATH/imaging

8.2-srv-imaging.zip

```
unzip xvf 8.2-srv-imaging.zip
git am 8.2-srv-imaging/*
cd ../vision_apps
make imaging
```

1.3 Optimizations in vision_apps(PSDKRA):

```
cd $PSDKRA_PATH/vision_apps
git init
git add .
git commit -m "Initial commit"
```

Download the attached patch to: $PSDKRA_PATH/vision_apps

0001-apps-basic...-Optimiza.patch

```
git am 0001-apps-basic_demos-app_rtos-common-app_init.c-Optimiza.patch
make vision_apps
make sdk
```

1.4 Optimizations in PDK(PSDK RTOS):

```
cd $PSDK_RTOS_PATH/pdk_jacinto_08_02_00_21
git init
git add .
git commit -m "Initial commit"
```

1.4.1 *Build R5 SBL for OSPI boot mode*

```
cd $PSDK_RTOS_PATH/pdk_jacinto_08_02_00_21/packages/ti/build
make ipc_echo_testb_freertos
make sbl_ospi_img_hlos
```

Image will be generated here: $PSDK_RTOS_PATH/pdk_jacinto_08_02_00_21/packages/ti/boot/sbl/binary/j721e_evm/ospi/bin/sbl_ospi_img_hlos_mcu1_0_release.tiimage

1.4.2 *Building combined_appImage*

Download the attached patch to: $PSDK_RTOS_PATH/pdk_jacinto_08_02_00_21

0001-packages-t...e-config..patch

Download & open the above patch & edit the paths.

HLOS_BIN_PATH
GCC_LINUX_ARM_PATH
DTB_IMG
DTBO_IMG
SPL_IMG
VISION_APPS_FW_PATH
KERNEL_IMG

To reflect your Paths.

Build the combined appImage using the below commands:

```
$PSDK_RTOS_PATH/pdk_jacinto_08_02_00_21
git am 0001-packages-ti-boot-sbl-tools-combined_appimage-config..patch
cd $PSDK_RTOS_PATH/pdk_jacinto_08_02_00_21/packages/ti/boot/sbl/tools/combined_appimage
make clean
make BOARD=j721e_evm HLOS_BOOT=optimized
```

You will have the binary generated under:

$PSDK_RTOS_PATH/pdk_jacinto_08_02_00_21/packages/ti/boot/sbl/tools/combined_appimage/bin/j721e_evm

1.4.3 Pick the tifs.bin prebuilt from $PSDKRA_PATH/pdkpdk_jacinto_07_03_00_29/packages/ti/drv/sciclient/soc/V1/tifs.bin

1.4.4 Pick the attached nor_spi_patterns.bin:

nor_spi_patterns.bin

1.4.5 Flashing the binaries to OSPI

Now copy 4 Images to SD card boot partition:

1. combined.appimage
2. tifs.bin
3. sbl
4. sbl_ospi_img_hlos_mcu1_0_release.tiimage

Flash to OSPI using the below commands:

```
sf probe
sf erase 0x0 0x4000000

fatload mmc 1 ${loadaddr} sbl_ospi_img_hlos_mcu1_0_release.tiimage;
sf update $loadaddr 0x0 $filesize;

fatload mmc 1 ${loadaddr} combined.appimage;
sf update $loadaddr 0x100000 $filesize;

fatload mmc 1 ${loadaddr} tifs.bin;
sf update $loadaddr 0x80000 $filesize;

fatload mmc 1 ${loadaddr} nor_spi_patterns.bin;
sf update $loadaddr 0x3fe0000 $filesize;
```

/*<![CDATA[*/ div.rbtoc1655723597713 {padding: 0px;} div.rbtoc1655723597713 ul {list-style: disc;margin-left: 0px;} div.rbtoc1655723597713 li {margin-left: 0px;padding-left: 0px;} /*]]>*/ Abstract: Optimizations in Linux(PSDKLA): Optimizations

Switch to OSPI Boot mode.
All the boot binaries are now in OSPI.

1.5 Steps to install vision_apps with tiny-rootfs on SD card

Insert the SD card and check if /dev/sdb exists(already formatted card) if yes do below

```
cp $PSDK_LINUX_PATH/filesystem/tisdk-tiny-image-j7-evm.tar.xz $PSDK_RTOS_PATH
umount /dev/sdb1
umount /dev/sdb2
cd ${PSDK_RTOS_PATH}
sudo psdk_rtos/scripts/mk-linux-card.sh /dev/sdb
```

install_to_sd_card_tiny.sh

download the above to $PSDKRA_PATH/scripts/ folder and give executable permissions.
Use the above script:

```
cd $PSDKRA_PATH
chmod +x psdk_rtos/scripts/install_to_sd_card_tiny.sh
cp $PSDKLAfilesystem/tisdk-tiny-image-j7-evm.tar.xz .
psdk_rtos/scripts/install_to_sd_card_tiny.sh
```

1.6 Copy test data to SD card (one time only)

```
cd /media/$USER/rootfs/
```

```
mkdir -p opt/vision_apps
cd opt/vision_apps
tar --strip-components=1 -xf ${path/to/file}/psdk_rtos_ti_data_set_xx_xx_xx.tar.gz
sync

cp -r $PSDK_LINUX_PATH/targetNFS/lib/* /media/$USER/rootfs/lib
cp -r $PSDK_LINUX_PATH/targetNFS/usr/lib/* /media/$USER/rootfs/usr/lib/
cp -r $PSDK_LINUX_PATH/targetNFS/etc/*  /media/$USER/rootfs/etc/

cd ${PSDK_RTOS_PATH}/vision_apps
make linux_fs_install_sd
```

1.7 Init script

To avoid losing time on filesystem mounting. We bypass using an init script.

- After flashing the minimal filesystem to the card, boot up the first time. Login as root.
- In the /home/root directory, create a file called init.sh with the following contents:

```
#!/bin/sh

export PATH=/usr/bin:/sbin:/bin
export LD_LIBRARY_PATH=/lib:/usr/lib:$LD_LIBRARY_PATH

mount -t proc proc /proc
mount -n -t sysfs none /sys
mount -n -t tmpfs none /run
insmod /lib/modules/5.10.100-g7a7a3af903/kernel/drivers/rpmsg/virtio_rpmsg_bus.ko
insmod /lib/modules/5.10.100-g7a7a3af903/kernel/drivers/remoteproc/ti_k3_r5_remoteproc.ko
insmod /lib/modules/5.10.100-g7a7a3af903/kernel/drivers/remoteproc/ti_k3_dsp_remoteproc.ko
insmod /lib/modules/5.10.100-g7a7a3af903/extra/pvrsrvkm.ko
insmod /lib/modules/5.10.100-g7a7a3af903/kernel/drivers/rpmsg/rpmsg_char.ko
insmod /lib/modules/5.10.100-g7a7a3af903/kernel/drivers/rpmsg-kdrv/rpmsg_kdrv_switch.ko


export VX_TEST_DATA_PATH=/opt/vision_apps/test_data

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib:/usr/lib/python3.8/site-packages/dlr
export APP_STEREO_DATA_PATH=$VX_TEST_DATA_PATH/psdkra/stereo_test_data
export APP_CONFIG_BASE_PATH=/opt/vision_apps/ptk_app_cfg
/opt/vision_apps/vx_app_arm_remote_log.out &
sleep .7s
/opt/vision_apps/vx_app_srv_camera.out --cfg /opt/vision_apps/app_srv.cfg
/bin/sh
```

From the Linux command prompt on the evm:

```
chmod +x init.sh
cd /sbin/
```

```
rm init
ln -s /home/root/init.sh init
```

1.8 Moving File system from SD to eMMC:

Boot to Linux command prompt on the TDA4-EVM. Execute the following commands to copy filesystem from MMC-SD to eMMC.

```
mkdir /mnt/emmc
mkdir /mnt/sd
mount /dev/mmcblk0p1 /mnt/emmc
mount /dev/mmcblk1p2 /mnt/sd

cp -r /mnt/sd/* /mnt/emmc
sync
```

After the above step on can optimize the MMC-SD aka sdhci1 node from DT by applying the patch 0003-arch-arm64-boot-dts-ti-k3-j721e-common-proc-board.dt.patch from 8.2-srv-linux.zip of step1

Insert the SD card in ubuntu host machine. Build the dtb & update the combined_appImage with the latest DTB.

```
cd $PSDK_LINUX_PATH
git am 8.2-srv-linux/0003-arch-arm64-boot-dts-ti-k3-j721e-common-proc-board.dt.patch
make linux-dtbs
cd $PSDK_RTOS_PATH/pdk_jacinto_08_02_00_21/packages/ti/boot/sbl/tools/combined_appimage
make clean
make BOARD=j721e_evm HLOS_BOOT=optimized
cp $PSDK_RTOS_PATH/pdk_jacinto_08_02_00_21/packages/ti/boot/sbl/tools/combined_appimage/bin/j721e_evm/
combined.appimage /media/$USER/boot
sync
```

`Now insert the SD card to TDA4-EVM & burn the new combined.appimage to OSPI`

```
sf probe
fatload mmc 1 ${loadaddr} combined.appimage;
sf update $loadaddr 0x100000 $filesize;
```

`Power off the board. Switch to OSPI Dip switch settings:`

`SW8: 00000000`
`SW9: 01000000`

`Boot the board. SRV demo should be up in ~2.2 Seconds.`

Logs:

```
[2022-06-16 14:08:08.175] SBL Revision: 01.00.10.01 (Jun 14 2022 - 11:01:15)
[2022-06-16 14:08:08.255] TIFS ver: 22.1.1--v2022.01 (Terrific Llam
[2022-06-16 14:08:08.591] NOTICE: BL31Built : 21:03:57, Mar 23 2022
[2022-06-16 14:08:08.591] ERROR: GTC_CNTFID0 is 0! Assung 200000000 Hz. Fix Bootloader
[2022-06-16 14:08:08.623] [ 0.000000] Booting Linux on physical CPU 0x0000000000 [0x411fd0]
[2022-06-16 14:08:08.639] [ 0.000000] Linux version 5.10.100-g7a7a3af903 (keerthy@erthy) (aarch64-none-linux-gnu-gcc (GNU Toolchain for the A-profile Architecture 9.2-2019.12 (arm-9.10)) 9.2.1 201025,2
[2022-06-16 14:08:08.671] [ 0.000000] Machine model: TexaInstruments K3 J721E SoC
[2022-06-16 14:08:08.671] [ 0.000000] earlycon: ns16550a0 aMMIO32 0x0000000002800000 (options '')
[2022-06-16 14:08:08.675] [ 0.000000] printk: bootconsole [ns16550a0] enabled
[2022-06-16 14:08:08.703] ERROR: GTC_CNTFID0 is 0! Assuming 200000000 Hz. Fix Bootloader
[2022-06-16 14:08:09.182] Start of init.sh script
[2022-06-16 14:08:09.263] before vision_apps_init
[2022-06-16 14:08:09.998] APP: Init ... !!!
[2022-06-16 14:08:09.998] MEM: Init ... !!!
[2022-06-16 14:08:09.998] MEM: Initialized DMA HEA(fd=4) !!!
[2022-06-16 14:08:09.998] MEM: Init ... Done !!!
[2022-06-16 14:08:09.998] IPC: Init ... !!!
[2022-06-16 14:08:09.998] IPC: In ... Done !!!
[2022-06-16 14:08:10.002] REMOTE_SERVICE: Init ... !!!
[2022-06-16 14:08:10.002] REMOTE_SERVICE: Init ... Done !!!
[2022-06-16 14:08:10.014] 0.000000 s: GTC Frequency = 0 MH
[2022-06-16 14:08:10.014] APP: Init ... Done !!!
[2022-06-16 14:08:10.014] 0.000000 s: VX_ZONE_INIT:Enable
[2022-06-16 14:08:10.014] 0.000000 s: VX_ZONE_ERROR:Enabled
[2022-06-16 14:08:10.017] 0.000000 s: VX_ZONE_WARNING:Enabled
[2022-06-16 14:08:10.031] 0.000000 s: VX_ZONE_INIT:[txInitLocal:130] Initialization Done !!!
[2022-06-16 14:08:10.031] 0.000000 s: VX_NE_INIT:[tivxHostInitLocal:86] Initialization Done for HOST !!!
[2022-06-16 14:08:10.062] 0.000000 s: ISS: Enumerating sensors ... !!!
[2022-06-16 14:08:10.063] 0.0000 s: ISS: Enumerating sensors ... found 0 : IMX390-UB953_D3
[2022-06-16 14:08:10.081] 0.000000 s: ISS: Enumeting sensors ... found 1 : AR0233-UB953_MARS
[2022-06-16 14:08:10.081] 0.000000 s:SS: Enumerating sensors ... found 2 : AR0820-UB953_LI
[2022-06-16 14:08:10.086] 0.000000 s: ISS: Enumerating sensors ... found 3 : UB9xxx_W12_TESTPATTERN
[2022-06-16 14:08:10.095] 0.000000 s: ISS: Enumerating sensors ...ound 4 : UB96x_UYVY_TESTPATTERN
[2022-06-16 14:08:10.104] 0.000000 s: ISS: Enumerating sensors ... found 5 : GW_AR0233_UYVY
[2022-06-16 14:08:10.111] Sensor selecte: IMX390-UB953_D3
[2022-06-16 14:08:10.111] 0.000000 s: ISS: Querying sensor [IMX3-UB953_D3] ... !!!
[2022-06-16 14:08:10.117] 0.000000 s: ISS: Querying sensor [IMX390-UB953_D3] ... Done !!!
[2022-06-16 14:08:10.127] Reading calmat file
[2022-06-16 14:08:10.127] file rd completed
[2022-06-16 14:08:10.206] EGL: version 1.4
[2022-06-16 14:08:10.351]
[2022-06-16 14:08:10.351]
[2022-06-16 14:08:10.352] =========================
[2022-06-16 14:08:10.352] Demo : Integrated SRV
[2022-06-16 14:08:10.352] =======================
[2022-06-16 14:08:10.352]
[2022-06-16 14:08:10.352] p: Print performance statistics
```

```
[2022-06-16 14:08:10.352]
[2022-06-16 14:08:10.352] eExport performance statistics
[2022-06-16 14:08:10.373]
[2022-06-16 14:08:10.373] x: Exit
[2022-06-16 14:08:10.373]
[2022-06-16 14:08:10.373] Enter Choice: 0.0000 s: ISS: Starting sensor [IMX390-UB953_D3] ... !!!
[2022-06-16 14:08:10.398] 0.000000 s: ISS: Starting sensor [IMX390-UB953_D3] ... !!!
```

Time to first frame to display: 10.398 - 08.175 = ~2.2 Seconds