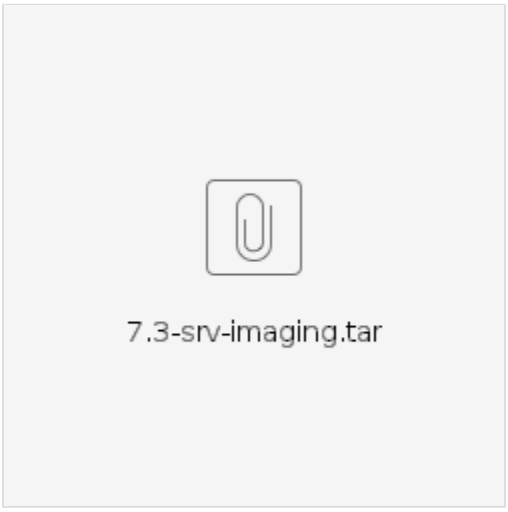
SRV in 2 Seconds (SBL based boot flow)

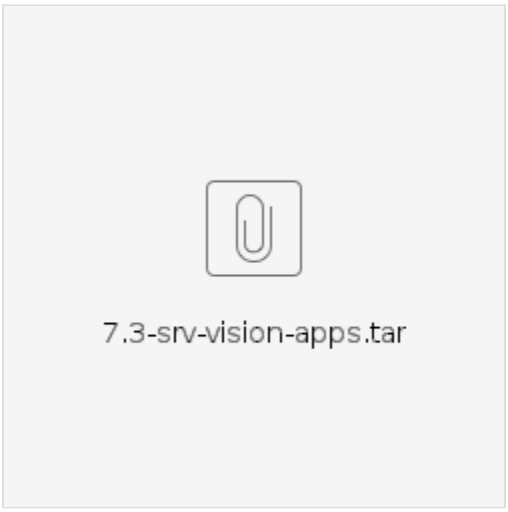
Bootflow SBL ATF/OPTEE Linux TinyRootfs SRV

OPtimizations in imaging folder:



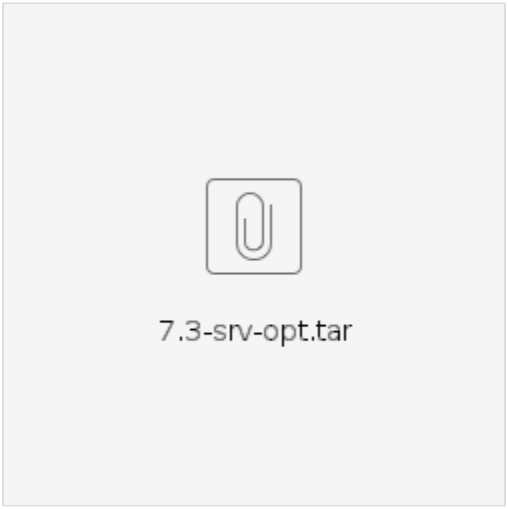
cd $PSDKRA_PATH/imaging
tar xvf 7.3-srv-imaging.tar
git am 7.3-srv-imaging/*
cd ../vision_apps
make imaging

Optimizations in vision_apps:



cd $PSDKRA_PATH/vision_apps
tar xvf 7.3-srv-vision-apps.tar
git am 7.3-srv-vision-apps/*
cd ../vision_apps
make vision_apps

Optimizations in Linux:



cd $PSDKLA_PATH/board-support/linux-5.4.106+gitAUTOINC+023faefa70-g023faefa70
tar xvf 7.3-srv-opt.tar
git am 7.3-srv-opt/0001-SBL-DT-Inteegrate-vision_apps-DTBO-in-CP-board-DTB.patch
git am 7.3-srv-opt/0002-arm64-boot-dts-ti-k3-j721e-Linux-DTS-optimizations-f.patch
cd ../..
make linux

Note: the patch 0003-arch-arm64-boot-dts-ti-k3-j721e-common-proc-board.dt.patch needs to be applied only when the file system is shifted to eMMC.

Note: Change: bootargs = "console=ttyS2,115200n8 earlycon=ns16550a,mmio32,0x02800000 root=/dev/mmcblk0p1 rw rootfstype=ext4 rootwait loglevel=0";
in arch/arm64/boot/dts/ti/k3-j721e-common-proc-board.dts

root parameter to SD. The above points to eMMC file system. Once the file system is shifted to eMMC then that can be changed back.

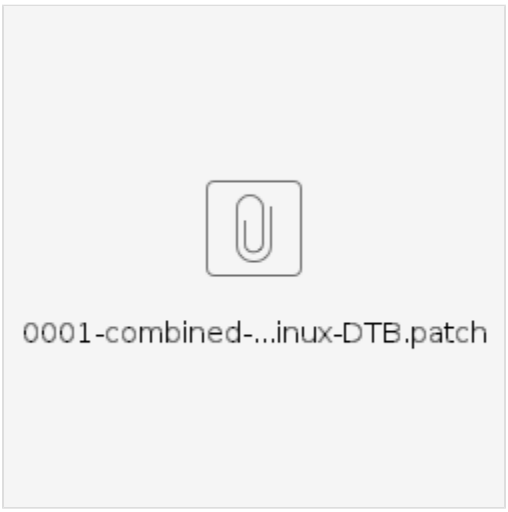
Changed needed in SBL:

Step 1: Build R5 SBL for OSPI boot mode

cd $PSDKRA/packages/ti/build
make sbl_ospi_img_hlos

Image will be generated here: $PSDKRA_PATH/pdk_jacinto_07_03_00_29/packages/ti/boot/sbl/binary/j721e_evm/ospi/bin
/sbl_ospi_img_hlos_mcu1_0_release.tiimage

Step 2: Building combined_appImage



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:

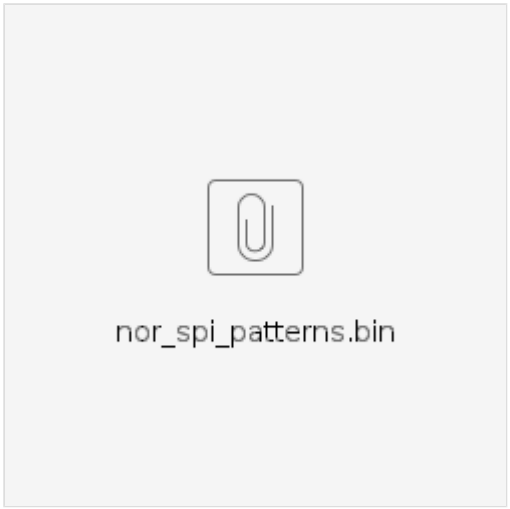
cd $PSDKRA_PATH/pdk_jacinto_07_03_00_29/packages/ti/boot/sbl/tools/combined_appimage
make clean
make BOARD=j721e_evm HLOS_BOOT=optimized

You will have the binary generated under:

$PSDKRA_PATH/pdk_jacinto_07_03_00_29/packages/ti/boot/sbl/tools/combined_appimage/bin/j721e_evm
ls -l
total 22244
-rw-rw-r-- 1 keerthy keerthy 22776632 Aug 20 19:49 **combined.appimage**

Step 3: Pick the tifs.bin prebuilt from $PSDKRA_PATH/pdkpdk_jacinto_07_03_00_29/packages/ti/drv/sciclient/soc/V1/tifs.bin

Step 4: Pick the attached nor_spi_patterns.bin:



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

You have already done the above steps. Only difference is **combined.appimage**.

Flash to OSPI using the below commands:

```
sf probe
sf erase 0x100000 0x3000000
fatload mmc 1 ${loadaddr} combined.appimage;
sf update $loadaddr 0x100000 $filesize;

/opt/vision_apps/vx_app_arm_remote_log.out &

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;
```

Switch to OSPI Boot mode.

All the boot binaries are now in OSPI.

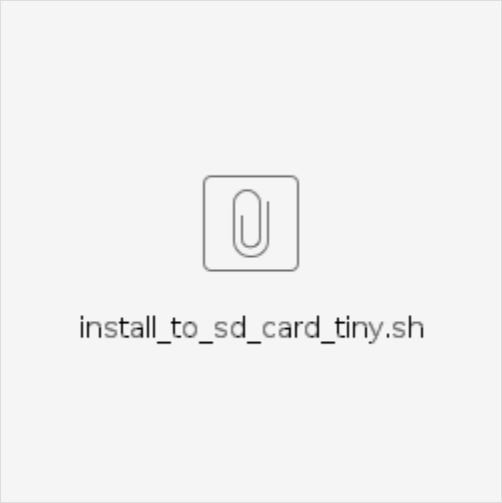
Steps to install vision_apps with tiny-rootfs on SD card

```
cp $PSDKLA_PATH/filesystem/tisdk-tiny-image-j7-evm.tar.xz $PSDKRA_PATH
# Insert the SD card and check if /dev/sdb exists(already formatted card) if yes do below

umount /dev/sdb1
umount /dev/sdb2

cd ${PSDKRA_PATH}
sudo psdk_rtos/scripts/mk-linux-card.sh /dev/sdb

mount /dev/sdb1
mount /dev/sdb2
```



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
psdk_rtos/scripts/install_to_sd_card_tiny.sh
```

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


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

Additional libraries that need to be copied from full SDK to tiny-rootfs for vision_apps to run:

Source: Full fledged PSDKRA rootfs
Destination: tiny rootfs

```
cp /lib/firmware/* from tisdk-default to tiny-rootfs
cp /usr/lib/* from tisdk-default to tiny-rootfs
cp /lib/* from tisdk-default to tiny-rootfs
cp /etc/* from tisdk-default to tiny-rootfs
```

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

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


#echo "entering init script"
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
#mount -n -t devpts none /dev/pts
#mount -t tmpfs -o size=51200K /var/volatile
#mount -n -t devtmpfs none /dev
#mount -n -t tmpfs none /tmp


insmod /lib/modules/5.4.106-g023faefa70/kernel/drivers/rpmsg/virtio_rpmsg_bus.ko
insmod /lib/modules/5.4.106-g023faefa70/kernel/drivers/remoteproc/ti_k3_r5_remoteproc.ko
insmod /lib/modules/5.4.106-g023faefa70/extra/pvrsrvkm.ko
insmod /lib/modules/5.4.106-g023faefa70/kernel/drivers/rpmsg/rpmsg_char.ko
insmod /lib/modules/5.4.106-g023faefa70/kernel/drivers/rpmsg-kdrv/rpmsg_kdrv_switch.ko



#cd /opt/vision_apps


export VX_TEST_DATA_PATH=/opt/vision_apps/test_data


# Location of DLR library
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib:/usr/lib/python3.8/site-packages/dlr


# Location of the input data for PTK demos
#export TIAP_DATABASE_PATH=/opt/vision_apps/test_data_ptk
# Location of the input data for Stereo demo
export APP_STEREO_DATA_PATH=$VX_TEST_DATA_PATH/psdkra/stereo_test_data


# APP config path for PTK demos
export APP_CONFIG_BASE_PATH=/opt/vision_apps/ptk_app_cfg


/opt/vision_apps/vx_app_arm_remote_log.out &


sleep .5
/opt/vision_apps/vx_app_srv_camera.out --cfg /opt/vision_apps/app_srv.cfg
/bin/sh
```

```
chmod +x init.sh
cd /sbin/
rm init
ln -s /home/root/init.sh init
```

Moving File system from SD to eMMC: https://e2e.ti.com/support/processors-group/processors/f/processors-forum/886615/tda4vmxevm-emmc-booting-issue/3330218#3330218

Presentation describing the feature optimization: