INTEGRATE MEMTOOL INTO I.MX8MM ANDROID

CAS MAGGIE JIANG

2023 MAY

EXTERNAL USE

SECURE CONNECTIONS
FOR A SMARTER WORLD

Topics

Memtool are default supported in i.MX Yocto, while not supported in Android. Some customer need to use memtool in Android, such as use memtool to do USB eye diagram to ouput test pattern.

This document describe on how to integrate memtool into i.MX8 Android.

Platform:

SW: Android-12.0.0_2.0.0

HW: i.MX8MM EVK

Build memtool in Android environment (way 1)

There are two ways to compile memtool into Android, one is build memtool in Android environment:

- git clone https://github.com/nxp-imx/imx-test
- cd imx-test/
- git tag
- git checkout lf-5.15.32-2.0.0
- cp -r imx-test/test/memtool ${MY_ANDROID}/external
- cd ${MY_ANDROID}
- source build/envsetup.sh
- lunch evk_8mm-userdebug
- mmm external/memtool

Use static link to build out memtool on Linux(way 2)

If you have no Android environment, it can also build memtool using static link toolchain.

1.Install Linaro toolchain

- Go to following link https://releases.linaro.org/components/toolchain/binaries/latest-7/aarch64-linux-gnu/
- download gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu.tar.xz
- tar –xvf gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu.tar.xz
- export PATH=$PATH:~/tools/toolchain/gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu/bin/
- aarch64-linux-gnu-gcc -v

2.Build memtool using this toolchain:

```
aarch64-linux-gnu-gcc -static -o memtool *.c
```

Memtool can run but doesn’t make effect

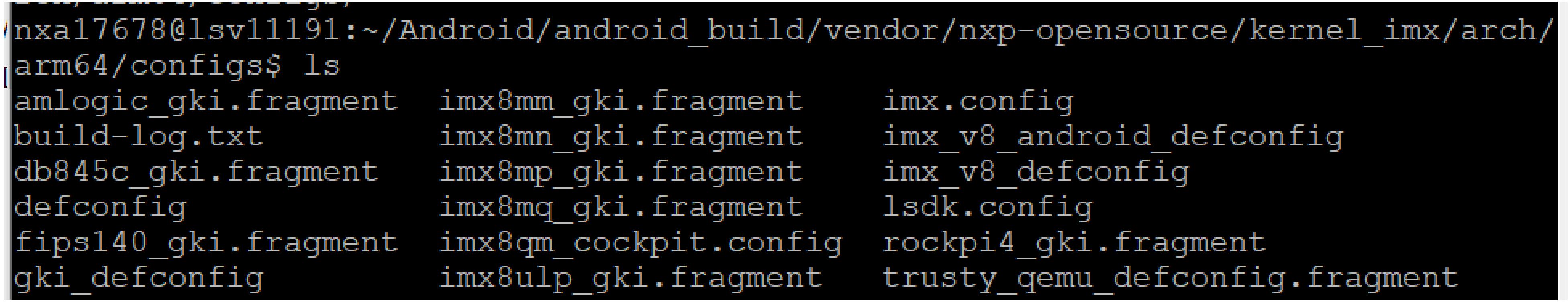
- Use either of way to build out memtool and pun in Android board. When run memtool on Android board, it is abnormal. Actually it doesn’t make effect although system is still working and no error report.

```
130|evk_8mm:/data # /data/memtool_static 0x32E40184 1
E
Reading 0x1 count starting at address 0x32E40184
// When reading, it doesn’t feedback register value
130|evk_8mm:/data # /data/memtool_static 0x32E40184=0x10050000
Writing 32-bit value 0x10050000 to address 0x32E40184
// when writing, it seems can write to register successfully. But actually it doesn’t make effect.
```

- The root cause: There is no /dev/mem on Android. The /dev/mem device is used to access areas of physical memory. We need to add DEVMEM into kernel config.

Default kernel config file on DEVMEM config

- It combine two config files: gki_defconfig and imx8mm_gki.fragment



```
nxa17678@lsv11191:~/Android/android_build/vendor/nxp-opensource/kernel_imx/arch/
arm64/configs$ ls
amlogic_gki.fragment  imx8mm_gki.fragment    imx.config
build-log.txt         imx8mn_gki.fragment    imx_v8_android_defconfig
db845c_gki.fragment   imx8mp_gki.fragment    imx_v8_defconfig
defconfig             imx8mq_gki.fragment    lsdk.config
fips140_gki.fragment  imx8qm_cockpit.config  rockpi4_gki.fragment
gki_defconfig         imx8ulp_gki.fragment   trusty_qemu_defconfig.fragment
```

- In gki_defconfig, it is setting following in default code

  # CONFIG_DEVMEM is not set

Tried several way to enable DEVMEM while failed

We tried either of following way to enable DEVMEM while all failed. (similar way is correct for other config such as CONFIG_I2C_CHARDEV=Y or M)

1. Enable DEVMEM in GKI config file:

Modify gki_defconfig, change to CONFIG_DEVMEM=y

2. Enable DEVMEM in i.MX8MM config file

Modify imx8mm_gki.fragment and add CONFIG_DEVMEM=y

3. Build DEVMEM into module (Consider Android GKI structure, build to module)

- Modify imx8mm_gki.fragment ,add CONFIG_DEVMEM=m,
- Modify android_build/device/nxp/imx8m/evk_8mm/SharedBoardConfig.mk to add into automatic loading module when bootup:

$(KERNEL_OUT)/drivers/char/mem.ko

Why failed when build DEVMEM into module

There is no mem.ko module output after building :
~/android_build/out/target/product/evk_8mm/obj/KERNEL_OBJ/drivers/char/

Root cause is:

1. drivers/char/Kconfig:

```
 config DEVMEM
          bool "/dev/mem virtual device support“
```

//it describe DEVMEM as bool, so only “Y” or “N” is supported, it doesn’t support build as module.

tristate : set as “Y” or “N” or “M”, which can also build as module.

2. drivers/char/Makefile

```
#obj-y     += mem.o random.o
```

//It means can only build into kernel

(obj-$(CONFIG_TEST)   += test.o   ------ It means depend on CONFIG_TEST setting, Y or M)

So please check corresponding Kconfig/Makefile to check whether the config an be set as module or not.

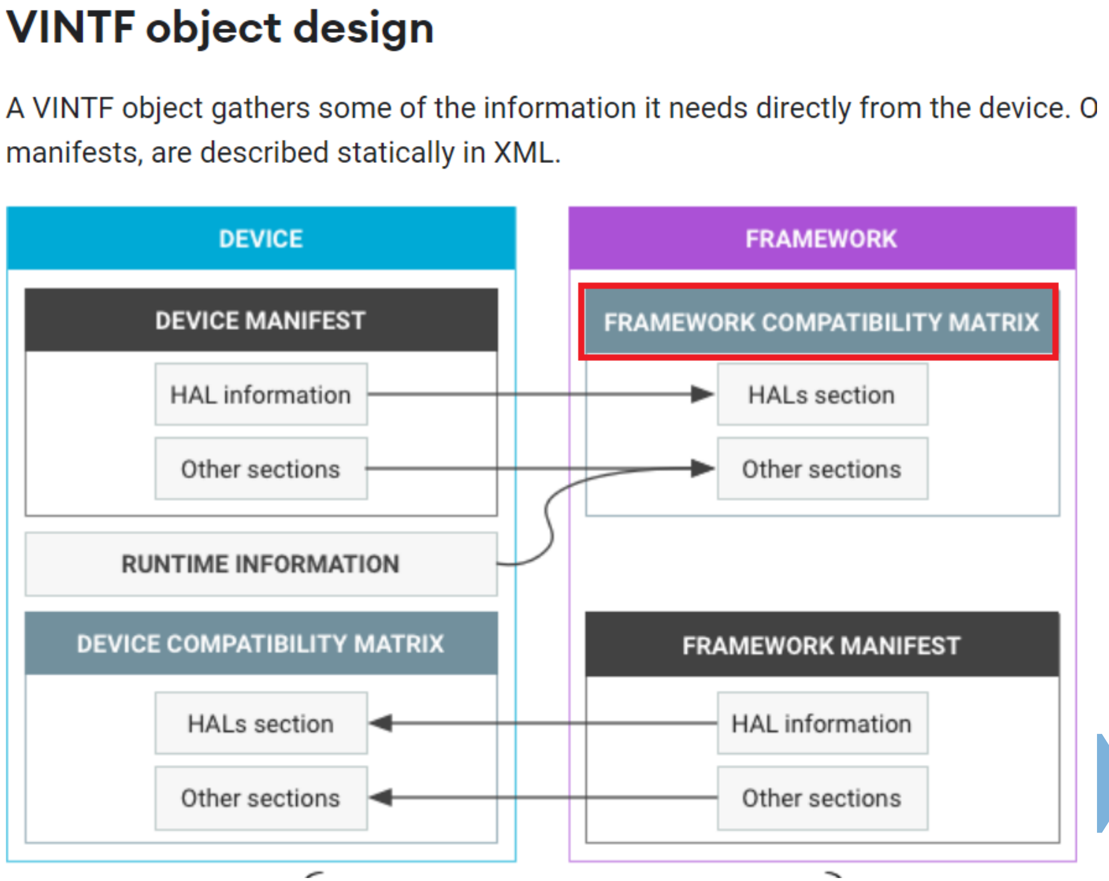
Why failed when build DEVMEM into kernel

- When build into kernel, it have following error log:

```
checkvintf E check_vintf.cpp:620] files are incompatible: Runtime info and framework compatibility matrix are incompatible: No compatible kernel requirement found (kernel FCM version = 6).
checkvintf E check_vintf.cpp:620] For kernel requirements at matrix level 6, For config CONFIG_DEVMEM, value = y but required n
```

- VINTF： Vendor Interface Object
- FCM：framework compatibility matrix

  框架兼容性矩阵



VINTF object design

A VINTF object gathers some of the information it needs directly from the device. O manifests, are described statically in XML.

Android FCM rules

https://source.android.google.cn/docs/core/architecture/vintf/comp-matrices?authuser=0

- Matching Rules:
- Framework compatibility matrix version matches
- HAL matches
- Kernel matches

- The VINTF object checks kernel compatibility against requirements on 4.19-r kernel branch, which is specified in FCM version 5. These requirements are built from kernel/configs/r/android-4.19 in the Android source tree.
- If the <kernel> section does match, the process continues by attempting to match config elements against /proc/config.gz. When a config item is set to n in the compatibility matrix for the matching <kernel> section, it must be absent from /proc/config.gz.

Fix FCM error issue when build DEVMEM into kernel

We need to remove Android FCM limitation on kernel config on DEVMEM setting:

~/Android/android_build/kernel/configs/android-5.15

1.android-base-conditional.xml

Mark following code in two locations:

```
  -    <config>
 +   <!--    <config>
               <key>CONFIG_DEVKMEM</key>
               <value type="bool">n</value>
  +      </config>  -->
  -    </config>
```

2. android-base.config

Remove following code

```
 -   CONFIG_DEVMEM is not set
```

Recompile NXP kernel instead of GKI kernel

Use following build command line to recompile whole Android:

- Build android all images:

```
./imx-make.sh -j4 2>&1 | tee build-log.txt
```

- After it, boot.img is GKI kernel. So it didn’t make effect, still need to add following command to reproduce NXP kernel and produce boot.image

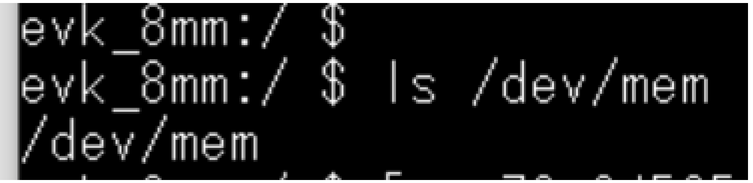
```
TARGET_IMX_KERNEL=true make bootimage
```

Then check .config file after building to confirm it is “y”

```
android_build/out/target/product/evk_8mm/obj/KERNEL_OBJ$ vi .config
CONFIG_DEVMEM=y
```

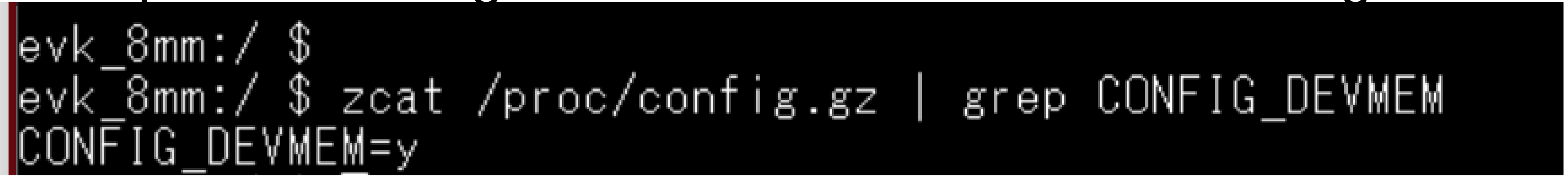
Test memtool in i.MX8MM EVK

- Confirm there is /dev/mem node



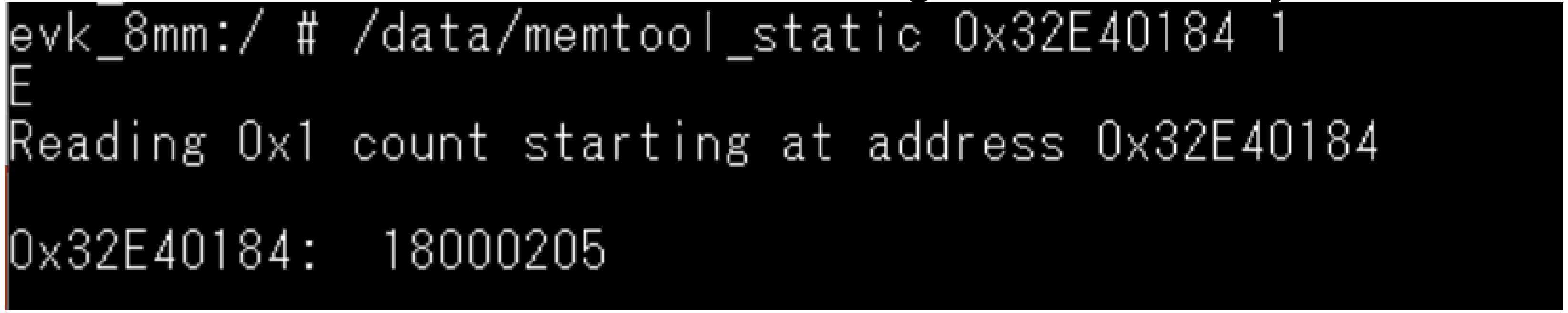
```
evk_8mm:/ $
evk_8mm:/ $ ls /dev/mem
/dev/mem
```

- Dump board config file to confirm DEVMEM is setting to Y:



```
evk_8mm:/ $
evk_8mm:/ $ zcat /proc/config.gz | grep CONFIG_DEVMEM
CONFIG_DEVMEM=y
```

- Run memtool, it can read back register correctly:



```
evk_8mm:/ # /data/memtool_static 0x32E40184 1
E
Reading 0x1 count starting at address 0x32E40184

0x32E40184:  18000205
```

Conclusion

1. Build memtool sourcecode into android environment, or build memtool with static link toolchain.

2. DEVMEM is set as bool by default, so it can’t build as module with “M”. It need o set config file to “Y”.

3. Modify Android code to remove FCM limitation on DEVMEM config setting.

4. Above modification is based on NXP kernel instead of GKI kernel, so it need to add additional building command to recompile NXP kernel image.

SECURE CONNECTIONS
FOR A SMARTER WORLD