OpenSTLinux Boot Customization

life.augmented

Lab *Boot Customization*

Lab Objective

- Run standard U-Boot command
- Example of different U-Boot configurations
- Start GDB debugger in TF-A
- Example of TF-A modification for secure peripheral assignments

Linux command on the host

```
root@stm32mp1:/#
```

Linux command on the target

theory

practice

Lab *Boot Customization*

Enter U-Boot

wiki user guide

https://wiki.st.com/stm32mpu/index.php/U-Boot_overview

- Open Terminal on Linux host and type;

```
minicom -D /dev/ttyACM0
```

- Enter U-Boot by pressing any key at boot time of board

```
NAND:   0 MiB
MMC:    STM32 SDMMC2: 0, STM32 SDMMC2: 1
In:     serial
Out:    serial
Err:    serial
Net:    eth0: ethernet@5800a000
Boot over mmc0!
Hit any key to stop autoboot:  0
STM32MP>
STM32MP>
STM32MP>
STM32MP>
STM32MP>
STM32MP>
```

Lab *Boot Customization*

U-Boot standard commands

Show U-Boot standard commands:

```
help
```

Dump memory mapping

```
mmc part
```

```
STM32MP> mmc part

Partition Map for MMC device 0  --   Partition Type: EFI

Part    Start LBA       End LBA         Name
        Attributes
        Type GUID
        Partition GUID
  1     0x00000022      0x00000221      "fsbl1"
        attrs:  0x0000000000000000
        type:   ebd0a0a2-b9e5-4433-87c0-68b6b72699c7
        type:   data
        guid:   8ef917d1-2c6f-4bd0-a5b2-331a19f91cb2
  2     0x00000222      0x00000421      "fsbl2"
        attrs:  0x0000000000000000
        type:   ebd0a0a2-b9e5-4433-87c0-68b6b72699c7
        type:   data
        guid:   77877125-add0-4374-9e60-02cb591c9737
  3     0x00000422      0x00001421      "ssbl"
        attrs:  0x0000000000000000
```

Partition number

Read content of bootfs partition (partition 4)

```
ext2ls mmc 0:4
```

```
STM32MP> ext2ls mmc 0:4
<DIR>       1024 .
<DIR>       1024 ..
<DIR>      12288 lost+found
           6569432 uImage
<DIR>       1024 mmc0_stm32mp157c-ed1_extlinux
          69558 stm32mp157c-dk2-a7-examples.dtb
          46180 splash.bmp
<DIR>       1024 mmc0_stm32mp157c-dk2_extlinux
<DIR>       1024 mmc0_stm32mp157c-ev1_extlinux
          69510 stm32mp157c-dk2.dtb
          74489 stm32mp157c-ev1-a7-examples.dtb
<DIR>       1024 mmc1_stm32mp157c-ed1-optee_extlinux
          1553 boot.scr.uimg
<DIR>       1024 mmc0_stm32mp157c-ev1-optee_extlinux
<DIR>       1024 mmc1_stm32mp157c-ed1_extlinux
<DIR>       1024 mmc1_stm32mp157c-ev1-optee_extlinux
<DIR>       1024 mmc0_stm32mp157c-ed1-optee_extlinux
```

Lab *Boot Customization*

U-Boot - view board as usb mass storage of linux host

Set board as USB mass storage (use USB OTG type cable to Linux host)

```
ums 0 mmc 0
```

```
STM32MP>
STM32MP> ums 0 mmc 0
UMS: LUN 0, dev 0, hwpart 0, sector 0x0, count 0x1dacc00
/
```

(Board is halted in mass storage mode)

Then on linux host you can view the SD card partitions

```
lsblk
```

```
osboxes@osboxes: ~
osboxes@osboxes:~$ lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sdb      8:16   1  14.9G  0 disk
└─sdb4   8:20   1    64M  0 part /media/osboxes/bootfs
└─sdb2   8:18   1   256K  0 part
└─sdb7   8:23   1    14G  0 part /media/osboxes/userfs
└─sdb5   8:21   1    16M  0 part /media/osboxes/vendorfs
└─sdb3   8:19   1     2M  0 part
└─sdb1   8:17   1   256K  0 part
└─sdb6   8:22   1 750.4M  0 part /media/osboxes/rootfs
sr0     11:0    1  1024M  0 rom
sda      8:0    0   500G  0 disk
└─sda4   8:4    0 302.7G  0 part /home
└─sda2   8:2    0   4.7G  0 part /boot
└─sda3   8:3    0   5.6G  0 part [SWAP]
└─sda1   8:1    0 187.1G  0 part /
osboxes@osboxes:~$
```

Lab *Boot Customization*

U-Boot - view board as USB mass storage of linux host

Then on linux host can observe the SD card contents

mount

mount | grep bootfs

ls /dev/disk/by-partlabel/

ls /media/$USER

ls /media/$USER/bootfs

```
osboxes@osboxes:~$ mount | grep bootfs
/dev/sdb4 on /media/osboxes/bootfs type ext4 (rw,nosuid,nodev,relatime,data=orde
red,uhelper=udisks2)
osboxes@osboxes:~$ ls -l /dev/disk/by-partlabel/
total 0
lrwxrwxrwx 1 root root 10 May 14 04:57 bootfs -> ../../sdb4
lrwxrwxrwx 1 root root 10 May 14 04:57 fsbl1 -> ../../sdb1
lrwxrwxrwx 1 root root 10 May 14 04:57 fsbl2 -> ../../sdb2
lrwxrwxrwx 1 root root 10 May 14 04:57 rootfs -> ../../sdb6
lrwxrwxrwx 1 root root 10 May 14 04:57 ssbl -> ../../sdb3
lrwxrwxrwx 1 root root 10 May 14 04:57 userfs -> ../../sdb7
lrwxrwxrwx 1 root root 10 May 14 04:57 vendorfs -> ../../sdb5
osboxes@osboxes:~$ ls /media/$USER
bootfs  rootfs  userfs  vendorfs
osboxes@osboxes:~$ ls /media/$USER/bootfs
boot.scr.uimg                          nand0_stm32mp157c-ev1_extlinux
lost+found                             nor0-mmc1_stm32mp157c-ev1_extlinux
mmc0_stm32mp157a-dk1_extlinux          nor0_stm32mp157c-ev1_extlinux
mmc0_stm32mp157a-dk1-optee_extlinux    splash.bmp
mmc0_stm32mp157c-dk2_extlinux          stm32mp157a-dk1.dtb
mmc0_stm32mp157c-dk2-optee_extlinux    stm32mp157c-dk2-a7-examples.dtb
mmc0_stm32mp157c-ed1_extlinux          stm32mp157c-dk2.dtb
```

Lab *Boot Customization*

U-Boot - update new splash screen

Set the board as usb mass storage (USB OTG type cable is connected to Linux host)

```
ums 0 mmc 0
```

Note: this should already be done from previous stage

Go to lab directory

```
cd $HOME/Desktop/InputLabMaterial/Lab-BspCustomization/
```

Copy new splash screen file and synchronize storage

```
sudo cp splash.bmp /media/$USER/bootfs/splash.bmp
sync
```

Reboot the board ("Ctrl+C" closes the "ums 0 mmc 0" command)

```
reset
```

Lab *Boot Customization*

U-Boot - list Uboot environment variable

To see U-Boot all environment variables

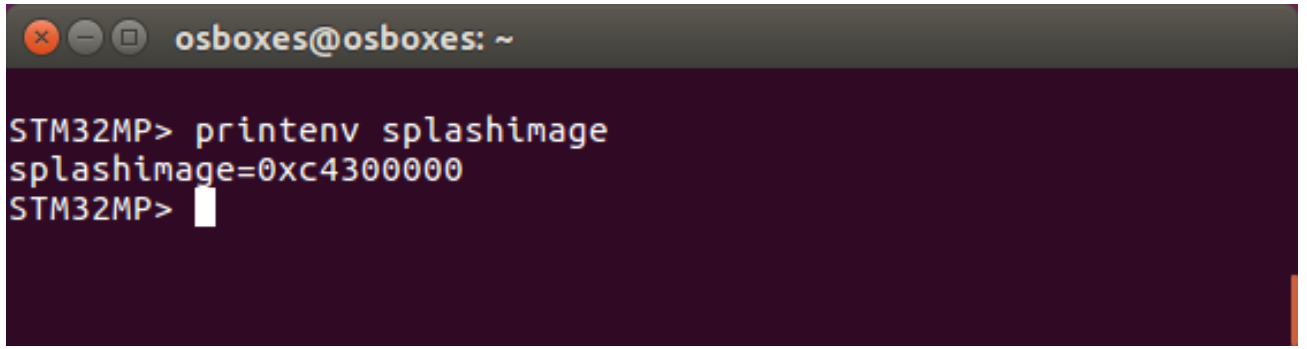
```
printenv
```

To set U-Boot variable

```
setenv <variable_name> <variable_value>
```

Check the value of splashimage (the splash image load address)

```
printenv splashimage
```

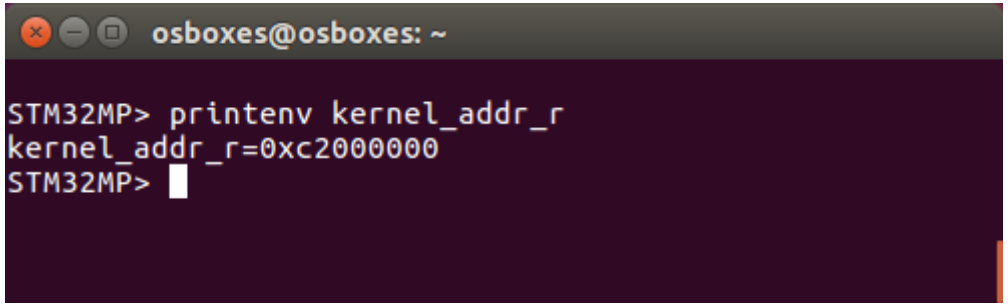


Lab *Boot Customization*

U-Boot – observe kernel load address

Observe kernel load address

```
printenv kernel_addr_r
```



kernel_addr_r must be same as **load_address** in kernel building command line (reminded here)

make uImage vmlinux dtbs LOADADDR=**load_address**

U-Boot command to run the kernel

```
run bootcmd
```

U-Boot – how to modify kernel command line (information only)

- **kernel command line** can be modified via the file ***extlinux.conf*** on the board in directory
  - : /boot/mmc0_stm32mp157c-dk2_extlinux

- **Kernel command line examples**

✓ how to configure only one core:

APPEND root=/dev/mmcblk0p6 rootwait rw console=ttySTM0,115200 **maxcpus=0**

✓ how to add dynamic debug or early debug:

APPEND root=/dev/ mmcblk0p6 rootwait rw **earlyprintk** console=ttyS3,115200 **loglevel=8 dyndbg="file drivers/pinctrl/* +p"**

Lab *Boot Customization*

U-Boot – add new kernel configuration in extlinux.conf

- In a Linux host terminal window

```
gedit $HOME/Desktop/InputLabMaterial/Lab-BspCustomization/extlinux.conf &
```

- Observe in extlinux.conf a new dtb configuration has been added in the following section:

```
LABEL stm32mp157c-dk2-iks01a2
KERNEL /uImage
FDT /stm32mp157c-dk2-iks01a2.dtb
APPEND root=/dev/mmcblk0p6 rootwait rw console=ttySTM0,115200
```

- This adds support for the X-NUCLEO-IKS01A2 - Motion MEMS and environmental sensor shield

Lab *Boot Customization*

U-Boot – add new kernel configuration in extlinux.conf

- **Set board as usb mass storage**

```
ums 0 mmc 0
```

- **Open Linux host Terminal , Go to lab directory**

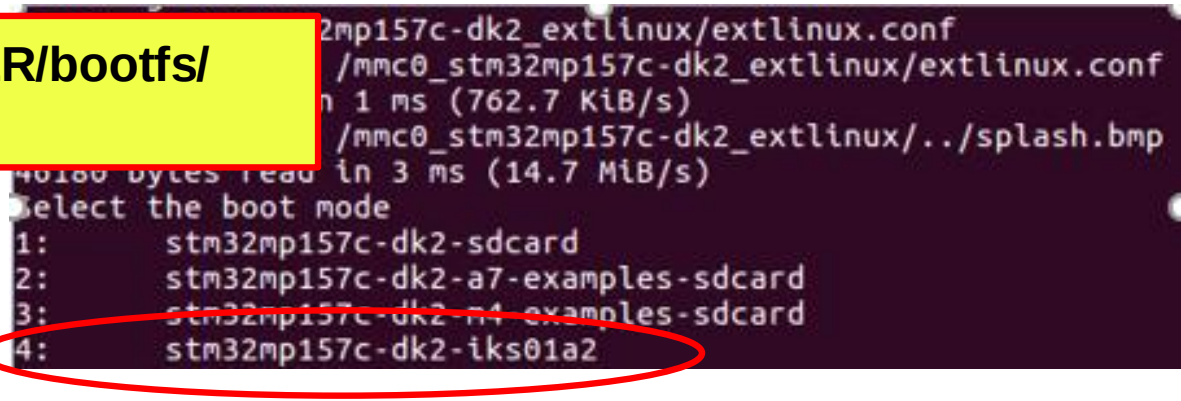
```
cd $HOME/Desktop/InputLabMaterial/Lab-BspCustomization/
```

- **Copy new kernel configuration to the board**:

```
sudo cp extlinux.conf /media/$USER/bootfs/mmc0_stm32mp157c-dk2_extlinux/extlinux.conf
```

- Add new kernel configuration dtb to the board and synchronize storage:

```
sudo cp stm32mp157c-dk2-iks01a2.dtb /media/$USER/bootfs/
sync
```



Reboot target (black button)

Observe new kernel configuration choice

Modify partition size (information only)

- Modify Offset in tsv file to modify size of partition (see tsv example below)

#Opt	Id	Name	Type	IP	**Offset**	Binary
-	0x01	fsbl1-boot	Binary	none	0x0	tf-a-stm32mp157c-ev1-trusted.stm32
-	0x03	ssbl-boot	Binary	none	0x0	u-boot-stm32mp157c-ev1-trusted.stm32
P	0x04	fsbl1	Binary	nor0	0x00000000	tf-a-stm32mp157c-ev1-trusted.stm32
P	0x05	fsbl2	Binary	nor0	0x00040000	tf-a-stm32mp157c-ev1-trusted.stm32
P	0x06	ssbl	Binary	nor0	0x00080000	u-boot-stm32mp157c-ev1-trusted.stm32
PE	0x20	logo	Binary	nor0	0x00280000	none
PE	0x10	empty	Binary	nor0	0x002C0000	none
P	0x21	bootfs	System	mmc1	0x00080000	st-image-bootfs-openstlinux-weston-stm32mp1.ext4
P	0x22	vendorfs	FileSystem	mmc1	0x04080000	st-image-vendorfs-openstlinux-weston-stm32mp1.ext4
P	0x23	rootfs	FileSystem	mmc1	0x05080000	st-image-weston-openstlinux-weston-stm32mp1.ext4
P	0x24	userfs	FileSystem	mmc1	0x33F00000	st-image-userfs-openstlinux-weston-stm32mp1.ext4

Boot time optimization (information only)

- **It is possible to optimize U-Boot time removing/modifying boot device list**

  Following macros are defined

  CONFIG_PREBOOT
  BOOT_TARGET_DEVICES
  CONFIG_EXTRA_ENV_SETTINGS

  By default U-Boot with parse all devices in given order from BOOT_TARGET_DEVICES list. It will look for extlinux.conf if it exists.

- **In u-boot-2018.11/include/configs/stm32mp1.h**

```
/* default order is eMMC (SDMMC 1)/ NAND / SDCARD (SDMMC 0) / SDMMC2 */
#define BOOT_TARGET_DEVICES(func) \
        func(MMC, mmc, 1) \
        func(UBIFS, ubifs, 0) \
        func(MMC, mmc, 0) \
        func(MMC, mmc, 2) \
        func(PXE, pxe, na)
```

life.augmented

You can apply patch to U-Boot to modify these macros and can force to boot directly from the device you want.

Boot time optimization (information only)

- Further information about methodology will delivered in end of June2019 in the wiki article called :

https://wiki.st.com/stm32mpu/wiki/How_to_optimize_the_boot_time

TF-A GDB Debug : configure debug

TF-A is pre-built (See wiki articles)

STM32MP1 Developer Package - TF-A
File:TF-A.README.HOW TO.txt

Debug scripts are already installed in **$HOME/gdbscripts**

Wiki user guide articles:

GDB
Setup.gdb.txt
Path_env.gdb

Modify debug configuration for TF-A in Setup.gdb

set $debug_phase = **1 <-debug TF-A fw**
set $debug_mode = **0 <- debug from reset**
set $debug_trusted_bootchain = **0**

TF-A Gdb Debug : start openOCD connection to JTAG

OpenOCD connects to the JTAG through STLink probe

Open a new Linux host terminal window and prepare the environment to launch OpenOCD :

```
cd /local/STM32MP15-Ecosystem-v1.0.0/Developer-Package/SDK

source ./environment-setup-cortexa7t2hf-neon-vfpv4-openstlinux_weston-linux-gnueabi

$OECORE_NATIVE_SYSROOT/usr/bin/openocd -s $OECORE_NATIVE_SYSROOT/usr/share/openocd/scripts \
-f ./sysroots/x86_64-openstlinux_weston_sdk-linux/usr/share/openocd/scripts/board/stm32mp15x_dk2.cfg
```

‘\’ character prevents command execution after new line
(command is too long to fit on single line)

Lab *Boot Customization*

TF-A Gdb Debug : start GDB connection to OpenOCD

Open new terminal window for GDB

GDB will connect to OpenOCD

```
cd $HOME/gdbscripts/

source /local/STM32MP15-Ecosystem-v1.0.0/Developer-Package/SDK/environment-setup-cortexa7t2hf-\
neon-vfpv4-openstlinux_weston-linux-gnueabi

$OECORE_NATIVE_SYSROOT/usr/bin/arm-openstlinux_weston-linux-gnueabi/arm-openstlinux_weston-\
linux-gnueabi-gdb -x=$HOME/gdbscripts/Setup.gdb
```

‘\’ character prevents command execution after new line
(command is too long to fit on single line)

Lab *Boot Customization*

TF-A GDB Debug :GDB debugging

GDB shows A7 is stopped at the beginning of “bl2_entrypoint()”

For all the command look in wiki at
https://wiki.st.com/stm32mpu/index.php/GDB_commands

```
/tf-a-stm32mp-2.0-r0/build/trusted/tf-a-bl32-trusted.elf" at
        .text_addr = 0x2fff0000
warning: section .text not found in /local/STM32MP15-Ecosystem-v1.0.0/Developer-Pack
age/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/arm-openstlinux_weston-linux
-gnueabi/tf-a-stm32mp-2.0-r0/build/trusted/tf-a-bl32-trusted.elf
force hard breakpoints
Hardware assisted breakpoint 1 at 0x2ffda000: file bl2/aarch32/bl2_el3_entrypoint.S,
 line 19.
stm32mp15x.cpu0 rev 5, partnum c07, arch f, variant 0, implementor 41

Temporary breakpoint 1, bl2_entrypoint () at bl2/aarch32/bl2_el3_entrypoint.S:19
19              bl      plat_dbg_attach_loop
(gdb)
```

Lab *Boot Customization*

TF-A Gdb Debug :GDB debugging

Set breakpoints:

hb bl2_main

```
(gdb) hb bl2_main
Hardware assisted breakpoint 8 at 0x2ffe1670: file bl2/bl2_main.c, line 32.
(gdb) info breakpoint
Num     Type           Disp Enb Address    What
7       breakpoint     keep y   0x2ffda000 bl2/aarch32/bl2_el3_entrypoint.S:19
8       hw breakpoint  keep y   0x2ffe1670 in bl2_main at bl2/bl2_main.c:32
```

Continue execution to next breakpoint:

cont

```
(gdb) cont
Continuing.
stm32mp15x.cpu0 rev 5, partnum c07, arch f, variant 0, implementor 41

Breakpoint 8, bl2_main () at bl2/bl2_main.c:32
32              NOTICE("BL2: %s\n", version_string);
(gdb)
```

Lab *Boot Customization*

TF-A Gdb Debug :GDB debugging

See the call stack with back trace:

bt

```
(gdb) bt
#0  bl2_main () at bl2/bl2_main.c:32
#1  0x2ffda108 in bl2_entrypoint () at bl2/aarch32/bl2_el3_entrypoint.S:51
```

Exit gdb :

q

TF-A Configure Peripheral assignments in ETZPC (security)

- **ETZPC device tree**
  https://wiki.st.com/stm32mpu/index.php/ETZPC_device_tree_configuration

look for etzpc configuration in
tf-a-stm32mp-2.0-r0/arm-trusted-firmware-2.0/fdts/stm32mp157c-security.dtsi

firewall is configured according to
tf-a-stm32mp-2.0-r0/arm-trusted-firmware-2.0/include/dt-bindings/soc/st,stm32-etzpc.h

Possible protections :

DECPROT_S_RW 0x0 -> Read/write Secure
DECPROT_NS_R_S_W 0x1 -> Non secure read / Read/write Secure
DECPROT_MCU_ISOLATION 0x2 -> MCU access only
DECPROT_NS_RW 0x3 -> Non secure read/write

TF-A Configure secure assignments in ETZPC example I2C4

I2C4 config is by default secure

from **RM0436 - rev1**

Table 92. Programmable options according to resource type

Resource type	0b00 (secured)	0b01 (Write secured)	0b10 (MCU isolation)	0b11 (non-secured)
1: Securable	Prog (default)	Prog	No	Prog
2: MCU isolation	No	No	Prog	Prog (default)
3: Securable and MCU Isolation	Prog	Prog	Prog	Prog (default)

Table 93. DECPROT assignment

Index	decprot bits	IP	Bus	Default	Bus master	Type	Attributes
0	DECPROT0[1:0]	STGENC	APB4	0b00	-	1	Securable
1	DECPROT0[3:2]	BKPSRAM	AHB5	0b00	-	1	Securable
2	DECPROT0[5:4]	IWDG1	APB5	0b00	-	1	Securable
3	DECPROT0[7:6]	USART1	APB5	0b00	-	1	Securable
4	DECPROT0[9:8]	SPI6	APB5	0b00	-	1	Securable
5	DECPROT0[11:10]	I2C4	APB5	0b00	-	1	Securable

TF-A Configure secure assignments in ETZPC example I2C4

If I2C4 config is let in secure configuration U-Boot would freeze while accessing to i2C4 clock

In tf-a-stm32mp-2.0-r0/arm-trusted-firmware-2.0/fdts/stm32mp157a-dk1.dts included by

tf-a-stm32mp-2.0-r0/arm-trusted-firmware-2.0/fdts/stm32mp157c-dk2.dts

I2C4 is set to unsecure by the *device tree* line

DECPROT(STM32MP1_ETZPC_I2C4_ID, **DECPROT_NS_RW, DECPROT_UNLOCK**)