# Linux fw\_printenv fw\_setenv to access U-Boot's environment variables

05/06/2020

Biyong Sun

### fw\_printenv/fw\_setenv

U-Boot provide fw\_printenv/fw\_setenv for Linux to access U-Boot environment variables.

It is complied in the U-Boot, but used in Linux.

### **Test Environment**

HW: i.MX8QXP MEK

SW: L4.14.98\_2.0.0\_ga

## Build fw\_printenv

source /opt/fsl-imx-xwayland/4.14-sumo/environment-setup-aarch64-poky-linux

make envtools CC="\$CC"

Now, you will find the u-boot/tools/env/fw\_printenv

Note: the fw\_setenv is applet of fw\_printenv as busybox does.

### fw\_env.config

fw\_env.config default directory is in the /etc/

/dev/mmcblk1

 $\mathbf{0x400000}$ 

0x2000

uboot/include/configs/imx8qxp\_mek.h

**0x2000**: #define CONFIG\_ENV\_SIZE 0x2000

**0x400000**: #define CONFIG\_ENV\_OFFSET (64 \* SZ\_64K)

### Add all the files in target rootfs

fw\_env.config copy to /etc/

fw\_printenv copy to /bin

Make symbol link fw\_setenv from fw\_printenv ln -s fw\_printenv fw\_setenv

fw\_setenv -> fw\_printenv

### Test fw\_printenv/fw\_setenv

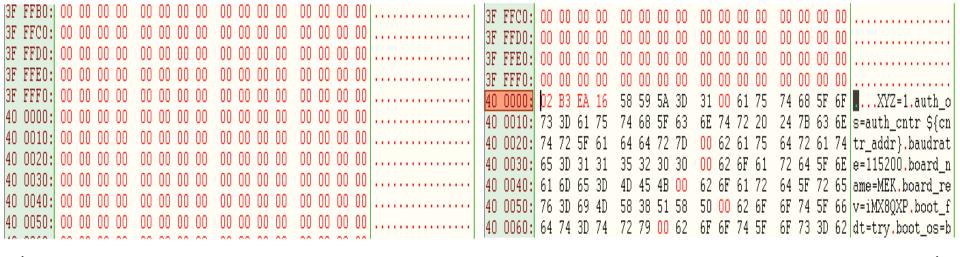
The first boot, the environment is generated by uboot C code.

i.MX uboot detects the which board it is, then give proper u-boot environment variables. It is in the memory not in the storage such as emmc, till you use saveenv to do save.

If it is the first time to do the test, get into uboot run saveenv first.

### Before run saveenv

### After run saveenv



### Test fw\_printenv/fw\_setenv(Cont.)

Get in to linux then test fw\_printenv

```
baudrate=115200
board name=MEK
board rev=iMX8QXP
boot fdt=try
poot os=booti ${loadaddr} - ${fdt addr};
pootcmd=mmc dev ${mmcdev}; if mmc rescan; then if run loadbootscript; then run bootscript; else if
if run loadimage; then run mmcboot; else run netboot; fi; fi; fi; else booti ${loadaddr} - ${fdt_a
pootcmd_mfg=run mfgtool_args;if iminfo ${initrd_addr}; then if test ${tee} = yes; then bootm ${tee_
lse echo "Run fastboot ...": fastboot 0: fi:
bootdelay=3
bootscript=echo Running bootscript from mmc ...: source
ontr addr=0x98000000
ontr tile=os ontr signed.bin
commit atf=1cb68fa
commit mkimage=dd023400
commit scfw=f83<u>a2bed</u>
commit secofw=92ef1143
console=ttyLPO
domOfdt_file=fsl-imx8qxp-mek-dom0.dtb
earlycon=lpuart32.0x5a060000
emmo dev=0
ethact=ethernet@5b040000
ethaddr=00:04:9f:05:8e:ea
ethprime=ethO
fastboot dev=mmc1
fdt addr=0x83000000
    file=fsl-imv8avn-mek-rnmsø dth
```

### Test fw\_printenv/fw\_setenv(Cont.)

The linux fw\_setenv to Set u-boot environment variables then verify in uboot fw\_setenv fdt\_file fsl-imx8qxp-mek.dtb (Linux Side)

The reboot board and stop at u-boot, using "printenv" to check if it is changed.

### Test fw\_printenv/fw\_setenv(Cont.)

Test Batch mode. Every time using fw\_setenv to set single variable will cause write to storage such as emmc. Batch mode provide a way to set several variables and store them to the storage just one write.

- Create scprit file call uboot\_vars uboot\_vars
   ABC 1 XYZ 2
- 2. fw\_setenv -s uboot\_vars (Linux Side)
- 3. Reboot board, Stop at uboot, printenv to check the changes

```
xenTinux_bootargs-
xenTinux_console=hvc0 earlycon=xen
xenmmchoot=setenv get_cmd "fatload
xennetboot=setenv get_cmd dhcp;set
ABC=1
XYZ=2
root@imx8qxpmek:~# U-Boot
```

# Demo on i.MX6ULL EVK

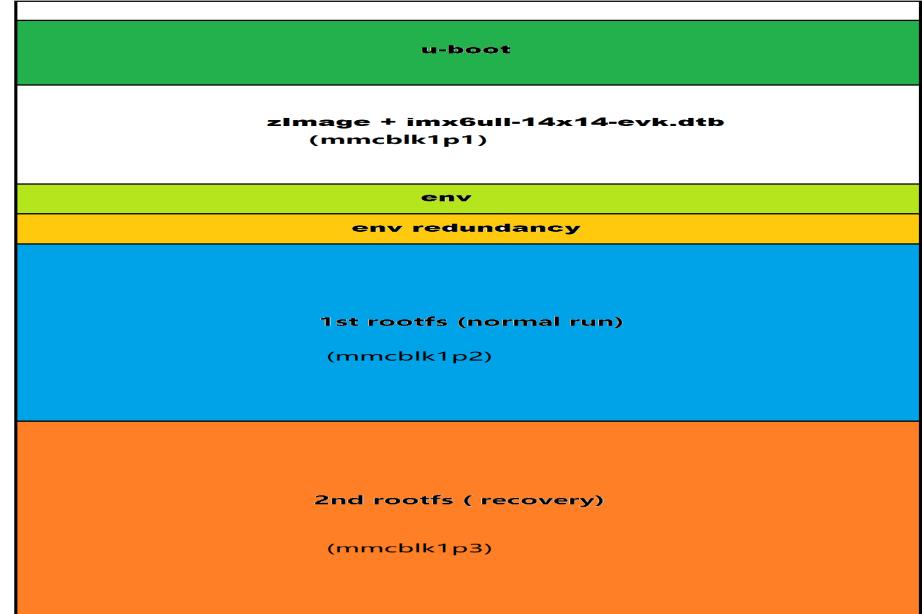
### Demo description

In this demo, the normal  $1^{st}$  rootfs will use fw\_setenv to set the u-boot bootargs to let the system to the  $2^{nd}$  recovery rootfs after reboot.

Once system goes into the  $2^{nd}$  recovery rootfs, it just prints out some information as a demo show. Then the recovery rootfs will also call the fw\_setenv to let the system mount the  $1^{st}$  normal rootfs after reboot.

Demo using the CONFIG\_ENV\_OFFSET\_REDUND to store the 2<sup>nd</sup> u-boot environment as backup.

# Demo image layout



## Major changes in u-boot code

```
u-boot/include/configs/mx6ullevk.h

296 #define CONFIG_ENV_SIZE SZ_8K

297 #if defined(CONFIG_ENV_IS_IN_MMC)

298 #define CONFIG_ENV_OFFSET (14 * SZ_64K)

299 #define CONFIG_ENV_OFFSET_REDUND (14 * SZ_64K + CONFIG_ENV_SIZE)

300 #elif defined(CONFIG_ENV_IS_IN_SPI_FLASH)
```

## Major changes in 1<sup>st</sup> rootfs(normal run)

```
-rwxr-xr-x 1 root root 33996 May 6 2020 fw_printenv lrwxrwxrwx 1 root root 11 May 6 2020 fw_setenv -> fw_printenv /etc/fw_env.config (configuration for fw_printenv/fw_setenv)

/dev/mmcblk1 0xE0000 0x2000 /dev/mmcblk1 0xE2000 0x2000 #redundancy
```

/usr/sbin/upg (script)
fw\_setenv mmcroot /dev/mmcblk1p3 rootwait rw
reboot

# Major changes in 2<sup>nd</sup> rootfs(recovery)

```
-rwxr-xr-x 1 root root
                                   33996 May 6 2020 fw_printenv
                                        11 May 6 2020 fw_setenv -> fw_printenv
lrwxrwxrwx 1 root
                           root
/etc/fw_env.config (configuration for fw_printenv/fw_setenv)
/dev/mmcblk1 0xE0000 0x2000
/dev/mmcblk1 0xE2000 0x2000 #redundancy
/etc/init.d/S50upg (script)
                                     echo "Upgrade Done! Reboot..."
 case "$1" in
                                        fw_setenv mmcroot /dev/mmcblk1p2 rootwait rw
 start)
                                        reboot
    echo "Upgrade Start..."
    echo "******"
                                      stop)
    sleep 1
    echo "*****
                                        exit 1
                                        ;;
    sleep 1
                                      restart | reload)
    echo "******"
                                        exit 1
    sleep 1
    echo "******"
    sleep 1
    echo "******"
                                     esac
```

exit \$

echo " "

### Demo screenshots

```
Starting syslogd: OK
Starting klogd: OK
Running sysctl: OK
Initializing random number generator: OK
Saving random seed: random: dd: uninitialized
     1. type upg to show the demo
Starti
      the upg script set u-boot env
<sup>₩e|com</sup>mmcroot=/dev/mmcblk1p3 rootwait rw
type u It is in the 1st rootfs(normal run)
the scrpit is at /usr/sbin/upg
fw setenv mmcroot /dev/mmcblk1p3 rootwait rw
reboot
Running sysctl: OK
Initializing random number
Saving random seed: random
OΚ
Starting network: OK
Upgrade Start...
*****
Upgrade Done! Reboot...
Stopping network: OK
3. run to 2nd rootfs(recovery)
the /etc/init.d/S50upg script print out demo upgrade
 set u-boot env mmcroot=/dev/mmcblk1p2 rootwait rw
 then reboot
```

```
# upg
# Stopping network: OK
Saving random seed: random: dd: uninitialized urandom read (51
OK
Stopping klogd: OK
Stopping syslogd: OK
umount: devtmpfs busy - remounted read-only
EXT4-fs (mmcblk 2. reboot and u-boot know
The system is go
Sent SIGTERM to mmcroot=/dev/mmcblk1p3 rootwait rw
Sent SIGKILL to
Requesting syste
imx-sdma 20ec000.sdma: external firmware not found, using ROM
ci_hdrc ci_hdrc.1: remove, state 4
usb usb1: USB disconnect, device number 1
ci_hdrc si_hdrc.1: USB bus 1 deregistered
reboot: Restarting system
```

FreeIng unused Kernel Memory: 1024K
EXT4-fs (mmcblk1p2): re-mounted. Opts: data=order
Starting syslogd: OK
Starting klogd: OK
Running sysctl: OK
Initializing random pumbor goporator: OK
Saving random seed 4. back to the 1st rootfs(normal ized ura
OK run)
Starting network:
Welcome to fw printenv/fw setenv demo

type upg to show the upgrade demo the scrpit is at /usr/sbin/upg fw\_setenv mmcroot /dev/mmcblk1p3 rootwait rw reboot

root login: root