EMBEDDED LINUX

U-Boot visit

Brief
U-Boot is the third bootloader stage. It most importantly is the first visible software component and the first user interface of the BeagleBone Black. We'll see how to interact with U-Boot and how it interacts with the Linux kernel.

Author
dimitri.boudier@ensicaen.fr
hugo.descoubes@ensicaen.fr

Summary
Discovery of U-Boot (Universal Boot loader) solution maintained by DENX. U-Boot is the most used bootloader solution in Embedded Linux systems and products.
U-Boot website.

Reminder
/!\ Understand all commands before running them! /!\

MANUAL BOOT - Discovery of main U-Boot features and manual boot

Enter the U-Boot menu

Make sure a functional system is deployed on your microSD card, that you have open and configured Minicom and then be ready to boot the BBB.
When you boot the BBB, quickly press the SPACE key multiple times, until you've managed to interrupt boot sequence. Note that you only have 2 seconds to do so.

When you see the following output, it means you have enterred the U-Boot menu. GG.

> Press SPACE to abort autoboot in 0 seconds
> =>

U-Boot program is now waiting for your commands.

Discovery of all commands and environment variables

Display all supported commands by this U-Boot build version

```
help
```

Display all supported environment variables by this U-Boot build version

```
printenv
```

Display selected variables

```
printenv bootfile
```

Modify selected variables

```
setenv bootfile xImage
```

Shortcut for `printenv`

```
pri bootfile
```

Run default boot procedure

```
boot
```

Manual boot

Restart the BBB and enter the U-Boot menu once again (using the SPACE key).

Set some environment variables.

```
setenv mmcpart 1
setenv bootfile vmlinuz-4.14.198-bone-rt-r40
setenv fdt_file am335x-boneblack.dtb

pri mmcdev mmcpart bootdir bootfile loadaddr fdt_file fdtaddr
```

- What is the purpose of those variables?

```
load mmc ${mmcdev}:${mmcpart} ${loadaddr} ${bootdir}/${bootfile}
load mmc ${mmcdev}:${mmcpart} ${fdtaddr}  ${bootdir}/${fdt_file}
```

- Fully explain these commands (and the options). Explain the output as well. Type `help` if needed.

Edit kernel command line parameters (string used by kernel to configure itself)
kernel.org - root/Documentation/admin-guide/kernel-parameters.txt

```
setenv bootargs console=ttyO0,115200n8 root=/dev/mmcblk0p1 ro rootfstype=ext4
rootwait init=/lib/systemd/systemd
```

- What do the `bootargs` fields mean? What are they used for?

U-Boot let go and run device tree and Linux kernel compressed image from main memory

```
bootz ${loadaddr} - ${fdtaddr}
```

- Explain the difference between `boot` and `bootz`. Type `help`, `help boot`, `help bootz` if needed.

Let Debian start. Log in when ready, and display the boot arguments and compare with the parameters you just gave to U-Boot

```
cat /proc/cmdline
```

- Copy the output below. What is it?

AUTOMATIC BOOT - /uEnv.txt file

Here we go again: restart the BBB and enter the Uboot menu (using the SPACE key). Show default file name used by U-Boot to automatize boot sequence.

```
pri bootenv
pri bootenvfile
```

- Write down the outputs so you can use them later.

Shut down the BBB down and plug the microSD card into your computer.

On your **computer**, you will use a new `uEnv.txt` file that will replace the one on the microSD card. First analyze its content, have a look at the U-boot Shell Commands and ask for help if you don't understand.

```
cat ${MEDIA}/rootfs/boot/uEnv.txt
cat ${DISCOPATH}/misc/uEnv.txt
```

- Compare the original file (still on the SD card) and the new file (in the `misc` directory).

Copy the new `uEnv.txt` file onto the microSD card and remove the default BBB `uEnv.txt` file.

```
lsblk -f
sudo cp -v ${DISCOPATH}/misc/uEnv.txt ${MEDIA}/rootfs/
sudo rm ${MEDIA}/rootfs/boot/uEnv.txt
```

Never forget to unmount your partition before removing the device

```
sync
sudo umount ${MEDIA}/rootfs
```

Verify that your mounting point `${MEDIA}/rootfs/` is correctly unmount

```
lsblk -f
```

Boot BBB GNU\Linux system, execute and verify the boot sequence on BBB. Analyse U-Boot output during boot.

Check that U-Boot have passed the arguments to the kernel.

```
cat /proc/cmdline
```

- Copy the output below.

UBOOT MMC SCRIPT

Before doing this last part, you must remove the custom `uEnv.txt` file and restore the original one. Plug the SD card into your **computer**.

```
lsblk -f
sudo rm ${MEDIA}/rootfs/uEnv.txt
export kernel_version=4.14.198-bone-rt-r40
sudo sh -c "echo 'uname_r=${kernel_version}' >> ${MEDIA}/rootfs/boot/uEnv.txt"

sync
sudo umount ${MEDIA}/rootfs
```

Now go back to the BBB and open Minicom. Power up the BBB and analyze the U-Boot output.

The script below is the default boot script embedded in the eMMC. One of the bootloader patches contain something very close to it (reminder: you've downloaded them in the `bootloarder/u-boot/` folder).

- Which patch? Which lines? Does it have any effect on the eMMC boot script?

```
mmc dev ${mmcdev};

 if mmc rescan;
    then gpio set 54;
    setenv bootpart ${mmcdev}:1;
    if test -e mmc ${bootpart} /etc/fstab;
        then setenv mmcpart 1;
    fi;
    echo Checking for: /uEnv.txt ...;
    if test -e mmc ${bootpart} /uEnv.txt;
        then if run loadbootenv;
            then gpio set 55;
            echo Loaded environment from ${bootenv};
            run importbootenv;
        fi;
        if test -n ${cape};
            then if test -e mmc ${bootpart} ${fdtdir}/${fdtbase}-${cape}.dtb;
                then setenv fdtfile ${fdtbase}-${cape}.dtb;
            fi;
            echo using: $fdtfile...;
        fi;
        echo Checking if uenvcmd is set ...;
        if test -n ${uenvcmd};
            then gpio set 56;
            echo Running uenvcmd ...;
            run uenvcmd;
        fi;
        echo Checking if client_ip is set ...;
        if test -n ${client_ip};
            then if test -n ${dtb};
                then setenv fdtfile ${dtb};
                echo using ${fdtfile} ...;
            fi;
            gpio set 56;
            if test -n ${uname_r};
                then echo Running nfsboot_uname_r ...;
                run nfsboot_uname_r;
            fi;
            echo Running nfsboot ...;
            run nfsboot;
        fi;
    fi;
    echo Checking for: /${script} ...;
    if test -e mmc ${bootpart} /${script};
        then gpio set 55;
        setenv scriptfile ${script};
        run loadbootscript;
        echo Loaded script from ${scriptfile};
        gpio set 56;
        run bootscript;
    fi;
    echo Checking for: /boot/${script} ...;
    if test -e mmc ${bootpart} /boot/${script};
        then gpio set 55;
        setenv scriptfile /boot/${script};
        run loadbootscript;
        echo Loaded script from ${scriptfile};
        gpio set 56;
        run bootscript;
    fi;
    echo Checking for: /boot/uEnv.txt ...;
    for i in 1 2 3 4 5 6 7 ;
        do setenv mmcpart ${i};
        setenv bootpart ${mmcdev}:${mmcpart};
        if test -e mmc ${bootpart} /boot/uEnv.txt;
            then gpio set 55;
            load mmc ${bootpart} ${loadaddr} /boot/uEnv.txt;
            env import -t ${loadaddr} ${filesize};
            echo Loaded environment from /boot/uEnv.txt;
            if test -n ${dtb};
                then setenv fdtfile ${dtb};
                echo Using: dtb=${fdtfile} ...;
            fi;
            echo Checking if uname_r is set in /boot/uEnv.txt...;
            if test -n ${uname_r};
                then gpio set 56;
                echo Running uname_boot ...;
                setenv mmcroot /dev/mmcblk${mmcdev}p${mmcpart} ro;
                run uname_boot;
            fi;
        fi;
    done;
fi;
```

- Now compare this script above with the U-Boot output during the start-up of the BBB.