WT-EDKSOM6ULX Kernel Compilation Method

Revise history:

Version	Date	Log
V1.0	2019/07/08	Create

Contents

WT-EDKSOM6ULX Kernel Compilation Method 1
1．Construction of Development Environment 3
1.1、Get source file 3
1.2、Get the development SDK 4
2. Compile u-boot and kernel 5
2.1、compile u-boot 5
2.2、Compile kernel 5
3. Update u-boot and kernel 5
3.1 update u-boot 5
3.2 update kernel 6
3.3 Use mfgtools 6

1．Construction of Development Environment

Create a work directory:

```
mkdir /home/industio_work
cd /home/industio_work
```

1.1、Get source file

- Download uboot source:

  Download link: git clone http://git.freescale.com/git/cgit.cgi/imx/uboot-imx.git

  After download, enter the source directory "uboot-imx":

  ```
  cd uboot-imx
  ```

  Checkout v2016 branch:

  ```
  git checkout imx_v2016.03_4.1.15_2.0.0_ga
  ```

  Download uboot patch:

  ```
  wget https://github.com/industio/WT-EDKSOM6ULX-Linux/raw/master/UBOOT-PATCH/0001-evk-board-first-commit.patch
  wget https://github.com/industio/WT-EDKSOM6ULX-Linux/raw/master/UBOOT-PATCH/0002-update-CONFIG_ENV_OFFSET.patch
  wget https://github.com/industio/WT-EDKSOM6ULX-Linux/raw/master/UBOOT-PATCH/0003-add-512M-ddr-cfg-and-config.patch
  wget https://github.com/industio/WT-EDKSOM6ULX-Linux/raw/master/UBOOT-PATCH/0004-update-fec-and-bootdelay.patch
  wget https://github.com/industio/WT-EDKSOM6ULX-Linux/raw/master/UBOOT-PATCH/0005-add-power-on-from-onoff-key-select-lcd-resolution.patch
  wget https://github.com/industio/WT-EDKSOM6ULX-Linux/raw/master/UBOOT-PATCH/0006-update-boot_evk.dtb-boot_edk.dtb-boot_evk.img-boot_e.patch
  wget https://github.com/industio/WT-EDKSOM6ULX-Linux/raw/master/UBOOT-PATCH/0007-imx6ull-14x14-evk.dtb-to-imx6ull-14x14-edk.dtb.patch
  ```

  Merge patch: use command "**git am -s**"

  ```
  git am -s 0001-evk-board-first-commit.patch
  git am -s 0002-update-CONFIG_ENV_OFFSET.patch
  ...
  git am -s 0007-imx6ull-14x14-evk.dtb-to-imx6ull-14x14-edk.dtb.patch
  ```

- Download kernel source file

  ```
  git clone http://git.freescale.com/git/cgit.cgi/imx/linux-imx.git
  ```

  Enter kernel directory : linux-imx

  ```
  cd linux-imx
  ```

  Checkout branch:

git checkout imx_4.1.15_2.0.0_ga

Download kernel patch:

wget https://github.com/industio/WT-EDKSOM6ULX-Linux/raw/master/KERNEL-PATCH/0001-edk-board-first-commit.patch

Merge patch:

git am -s 0001-edk-board-first-commit.patch

1.2、Get the development SDK

Enter the work directory:

cd /home/industio_work

Download the SDK ：

Download link:https://releases.linaro.org/components/toolchain/binaries/4.9-2017.01/arm-linux-gnueabihf/gcc-linaro-4.9.4-2017.01-i686_arm-linux-gnueabihf.tar.xz

Decompression:

sudo tar xjvf gcc-linaro-4.9.4-2017.01-i686_arm-linux-gnueabihf.tar.xz -C /opt/industio

Add environment variables to PATH:

Method 1:

Configure environment variables and edit configuration scripts,

#vi environment-setup_hf

```
GCC_PATH=/opt/industio/gcc-linaro-4.9.4-2017.01-i686_arm-linux-gnueabihf
GCC_CC=arm-linux-gnueabihf

export ARCH=arm
export CROSS_COMPILE=$GCC_CC-
export PATH=$GCC_PATH/bin:$GCC_PATH/bin/$GCC_CC:$PATH
```

Run: source environment-setup_hf

Method 2: add the environment variables to profile

eg:

vi .profile add the following contents to the end of the file.

```
export PATH=$PATH:/opt/industio/gcc-linaro-4.9.4-2017.01-i686_arm-linux-gnueabihf/bin
export ARCH=arm
export CORESS_COMPILE= arm-linux-gnueabihf
```

Note: If this method is used to compile QTs through qmake, make, it is necessary to ensure that the current system does not have qmake.

Check whether the environment is in effect:

which arm-linux-gnueabihf-gcc

2. Compile u-boot and kernel

2.1、compile u-boot

Enter uboot-imx:

```
cd /home/industio_work/uboot-imx
```

- For Nandflash version:

```
make mx6ull_14x14_evk_nand_defconfig
make
```

Final Generation: u-boot.imx

Note: If you want to compile the generated files into a directory, the method is as follows (for example, to generate the current "build" directory), here take Nand version as an example:

```
make mx6ull_14x14_evk_nand_defconfig O=build
make O=build
```

2.2、Compile kernel

```
cd /home/industio_work/linux-imx
make    wt_edksom6ull_defconfig
make    -j4
```

(J4 represents multithreaded compilation, and 4 is the number of host kernels)

Final Generation : zImage and dtb

zImage directory: arch/arm/boot/zImage

nand version dtb file directory: arch/arm/boot/dts/imx6ull-14x14-evk-gpmi-weim.dtb

emmc version dtb file directory: arch/arm/boot/dts/imx6ull-14x14-evk-emmc.dtb

Note: If you want to compile the generated files into a directory, the method is as follows (for example, to generate the current "build" directory),

```
make wt_edksom6ull_defconfig O=build
make O=build
```

3. Update u-boot and kernel

3.1 update u-boot

Firstly, put the u-boot.imx file generated at 2.1 placed to the SD card root directory, insert the SD card into the EVK board and mounted. Mount to the /mnt directory for example.

Enter the /mnt directory and then execute the following commands:

3.1.1 For Nandflash version

```
mount -t debugfs debugfs /sys/kernel/debug
flash_erase /dev/mtd0 0 0
kobs-ng init -x -v --chip_0_device_path=/dev/mtd0 u-boot.imx
```

3.2 update kernel

3.2.1 For Nandflash version:

After Kernel compilation is completed, executing make Zi generates boot_edk.img and boot_edk.dtb files in the /home/industio_work/linux-imx directory. Put these two files in the root directory of TF card, insert it into the EVK board, and power on, kernel files will be automatically updated.

3.3 Use mfgtools

We can also use “mfgtools” to update u-boot and kernel, but “uuu” is not supported. For the usage of “mfgtool”, please refer to the online materials.