

# Build U-Boot, Linux, Filesystem for imx8mq

(RGB-D Camera board)

2020. 6. 1

# Table of Contents

- ❑ Files
- ❑ Setup Build Environment
- ❑ Build U-Boot
- ❑ Make flash.bin
- ❑ Build Kernel
- ❑ Build Filesystem
- ❑ Setting up the system
- ❑ Write flash.bin to emmc
- ❑ Write kernel to emmc
- ❑ Write filesystem to emmc

# Files

U-Boot : u-boot-2018.03.tar.gz

Kernel : linux-4.14.98.tar.gz

Filesystem : buildroot-2019.02.1.tar.gz  
imx8mq\_i4vine\_defconfig

Toolchain : gcc-7.3.tar.gz

imx-tool : imx-mkimage.tar.gz

flash tool : **uuu.exe**

- > Download uuu.exe to your host PC
- > for Windows : [https://github.com/NXPmicro/mfgtools/releases/download/uuu\\_1.3.154/uuu.exe](https://github.com/NXPmicro/mfgtools/releases/download/uuu_1.3.154/uuu.exe)
- > for Linux : [https://github.com/NXPmicro/mfgtools/releases/download/uuu\\_1.3.154/uuu](https://github.com/NXPmicro/mfgtools/releases/download/uuu_1.3.154/uuu)

# Setup Build Environment

USER\_ID : ryu (modify your id)

1. Make build directory (Run following command)

```
> cd /home/USER_ID
```

```
> mkdir build
```

```
> cd build
```

```
> mkdir u-boot kernel out_uboot out_kernel out_filesystem
```

2. Copy files (u-boot-2018.03.tar.gz, linux-4.14.98.tar.gz, buildroot-2019.02.1.tar.gz,

imx8mq\_i4vine\_defconfig, gcc-7.3.tar.gz, imx-mkimage.tar.gz)

Copy files to build directory

# Setup Build Environment

USER\_ID : ryu (modify your id)

3. Untar source files & imx-tool (Run following command)

```
> cd /home/USER_ID/build/u-boot
```

```
> tar -xvf ../u-boot-2018.03.tar.gz
```

```
> cd /home/USER_ID/build/kernel
```

```
> tar -xvf ../linux-4.14.98.tar.gz
```

```
> cd /home/USER_ID/build
```

```
> tar -xvf buildroot-2019.02.1.tar.gz
```

```
> tar -xvf imx-mkimage.tar.gz
```

```
> mv imx8mq_i4vine_defconfig buildroot-2019.02.1/configs/
```

# Setup Build Environment

USER\_ID : ryu (modify your id)

TOOLCHAIN\_PATH : /opt/imx8mq/4.14-sumo/sysroots/x86\_64-pokysdk-linux/usr/bin/aarch64-poky-linux

4. Untar compiler (Run following command)

```
> sudo mkdir /opt/imx8mq
```

```
> cd /opt/imx8mq
```

```
> sudo tar -xvf /home/USER_ID/build/gcc-7.3.tar.gz
```

5. Setting compiler path (Run following command)

```
> source /opt/imx8mq/4.14-sumo/environment-setup-aarch64-poky-linux
```

# Build U-Boot

USER\_ID : ryu (modify your id)

U-BOOT\_SOURCE\_PATH : /home/USER\_ID/build/u-boot

U-BOOT\_OUTPUT\_PATH : /home/USER\_ID/build/out\_uboot

1. Build (Run following command)

```
> source /opt/imx8mq/4.14-sumo/environment-setup-aarch64-poky-linux
```

```
> cd /home/USER_ID/build/u-boot
```

```
> make O=../out_uboot imx8mq_i4vine_defconfig
```

# Build U-Boot

## 1. Build (Run following command)

```
> make O=../out_uboot menuconfig
```

If you get fatal error: curses.h: No such file or directory.

```
"ryu@ryu:~/workspace/uvc-gadget/build/u-boot$ make O=../out_uboot/ menuconfig
make[1]: Entering directory '/home/ryu/workspace/uvc-gadget/build/out_uboot'
  GEN      ./Makefile
  HOSTCC  scripts/kconfig/mconf.o
In file included from /home/ryu/workspace/uvc-gadget/build/u-boot/scripts/kconfig/mconf.c:23:0:
/home/ryu/workspace/uvc-gadget/build/u-boot/scripts/kconfig/lxdialog/dialog.h:26:20: fatal error: ncurses.h: No such file or directory
compilation terminated.
scripts/Makefile.host:116: recipe for target 'scripts/kconfig/mconf.o' failed
make[2]: *** [scripts/kconfig/mconf.o] Error 1
/home/ryu/workspace/uvc-gadget/build/u-boot/Makefile:479: recipe for target 'menuconfig' failed
make[1]: *** [menuconfig] Error 2
make[1]: Leaving directory '/home/ryu/workspace/uvc-gadget/build/out_uboot'
Makefile:150: recipe for target 'sub-make' failed
make: *** [sub-make] Error 2"
```

Typing the command below.

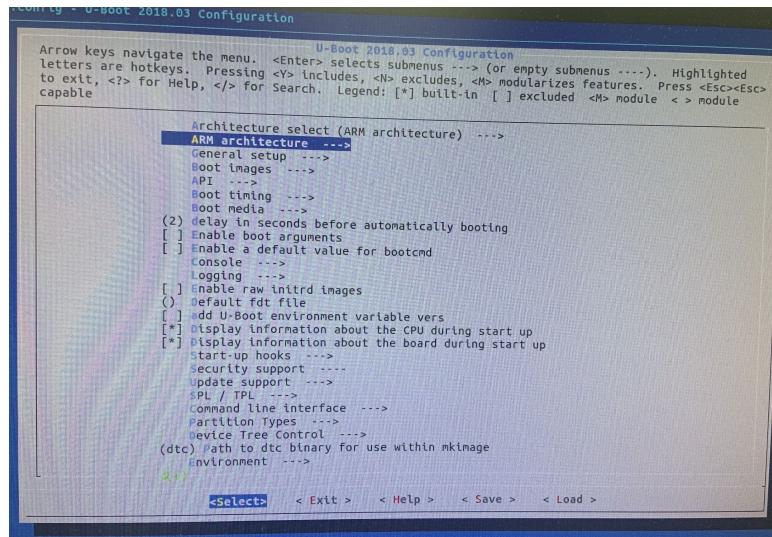
```
> sudo apt-get install libncurses5-dev libncursesw5-dev
```

# Build U-Boot

## 1. Build (Run following command)

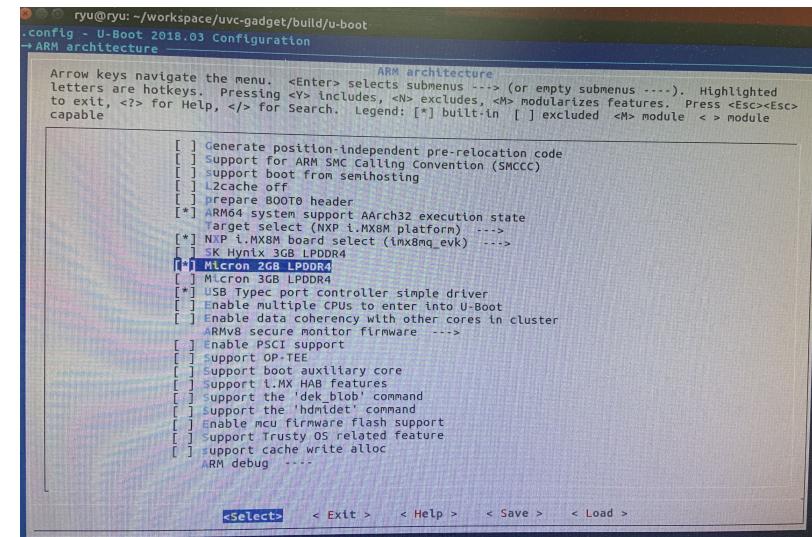
> make O=../out\_uboot menuconfig

Select ARM architecture



=> Uncheck Micron 3GB LPDDR4 &

Select Micron 2GB LPDDR4



# Build U-Boot

USER\_ID : ryu (modify your id)

U-BOOT\_SOURCE\_PATH : /home/USER\_ID/build/u-boot

U-BOOT\_OUTPUT\_PATH : /home/USER\_ID/build/out\_uboot

1. Build (Run following command)

> make -j8 O=../out\_uboot ( -j option : Specifies the number of jobs(commands) to run simultaneously.

2. Output files

U-BOOT\_OUTPUT\_PATH/spl/u-boot-spl.bin,

U-BOOT\_OUTPUT\_PATH/u-boot-nodtb.bin,

U-BOOT\_OUTPUT\_PATH/arch/arm/dts/fsl-imx8mq-evk.dtb

# Make flash.bin

USER\_ID : ryu (modify your id)

U-BOOT\_OUTPUT\_PATH : /home/USER\_ID/build/out\_uboot

1. Copy u-boot output files (Run following command)

```
> cd /home/USER_ID/build/imx-mkimage
```

```
> cp -a /home/USER_ID/build/out_uboot/tools/mkimage    iMX8M/mkimage_uboot
```

```
> cp -a /home/USER_ID/build/out_uboot/spl/u-boot-spl.bin    iMX8M/
```

```
> cp -a /home/USER_ID/build/out_uboot/u-boot-nodtb.bin    iMX8M/
```

```
> cp -a /home/USER_ID/build/out_uboot/arch/arm/dts/fsl-imx8mq-evk.dtb    iMX8M/
```

# Make flash.bin

2. Make flash.bin (Run following command)

```
> make SOC=iMX8M flash_spl_uboot
```

If you get 'git: not found'

```
ryu@ryu:~/workspace/uvc-gadget/build/imx-mkimage$ make SOC=iMX8M flash_spl_uboot
/bin/sh: 1: git: not found
Makefile:40: recipe for target 'buildinfo' failed
make[1]: *** [buildinfo] Error 127
Makefile:19: recipe for target 'flash_spl_uboot' failed
make: *** [flash_spl_uboot] Error 2
```

Typing the command below.

```
> sudo apt-get install git
```

# Make flash.bin

## 2. Make flash.bin (Run following command)

```
> make SOC=iMX8M flash_spl_uboot
```

If you get 'dtc: not found'

```
ryu@ryu:~/workspace/uvc-gadget/build/imx-mkimage$ make SOC=iMX8M flash_spl_uboot
Compiling mkimage_imx8
./mkimage_fit_atf.sh fsl-imx8mq-evk.dtb > u-boot.its
bl31.bin size:
44752
u-boot-nodtb.bin size:
687008
fsl-imx8mq-evk.dtb size:
28475
./mkimage_uboot -E -p 0x3000 -f u-boot.its u-boot-lpddr4.itb
sh: 1: dtc: not found
./mkimage_uboot: Can't open u-boot-lpddr4.itb.tmp: No such file or directory
soc.mak:56: recipe for target 'u-boot-lpddr4.itb' failed
make[1]: *** [u-boot-lpddr4.itb] Error 255
Makefile:19: recipe for target 'flash_spl_uboot' failed
make: *** [flash_spl_uboot] Error 2
```

Typing the command below.

```
> sudo apt-get install device-tree-compiler
```

# Make flash.bin

2. Make flash.bin (Run following command)

```
> make SOC=iMX8M flash_spl_uboot
```

3. Output files : iMX8M/flash.bin

# Build Kernel

USER\_ID : ryu (modify your id)

KERNEL\_SOURCE\_PATH : /home/USER\_ID/build/kernel

KERNEL\_OUTPUT\_PATH : /home/USER\_ID/build/out\_kernel

## 1. Build (Run following command)

```
> source /opt/imx8mq/4.14-sumo/environment-setup-aarch64-poky-linux  
> unset LDFLAGS  
> cd /home/USER_ID/build/kernel  
> make O=../out_kernel imx8mq_i4vine_defconfig  
> make -j8 O=../out_kernel Image dtbs LOADADDR=0x40480000
```

## 2. Output files

KERNEL\_OUTPUT\_PATH/arch/arm64/boot/Image,

KERNEL\_OUTPUT\_PATH/arch/arm64/boot/dts/freescale/fsl-imx8mq-evk.dtb

# Build Filesystem

USER\_ID : ryu (modify your id)

FILESYSTEM\_SOURCE\_PATH : /home/USER\_ID/build/buildroot-2019.02.1

FILESYSTEM\_OUTPUT\_PATH : /home/USER\_ID/build/out\_filesystem

1. Build (Run following command)

```
> cd /home/USER_ID/build/buildroot-2019.02.1
```

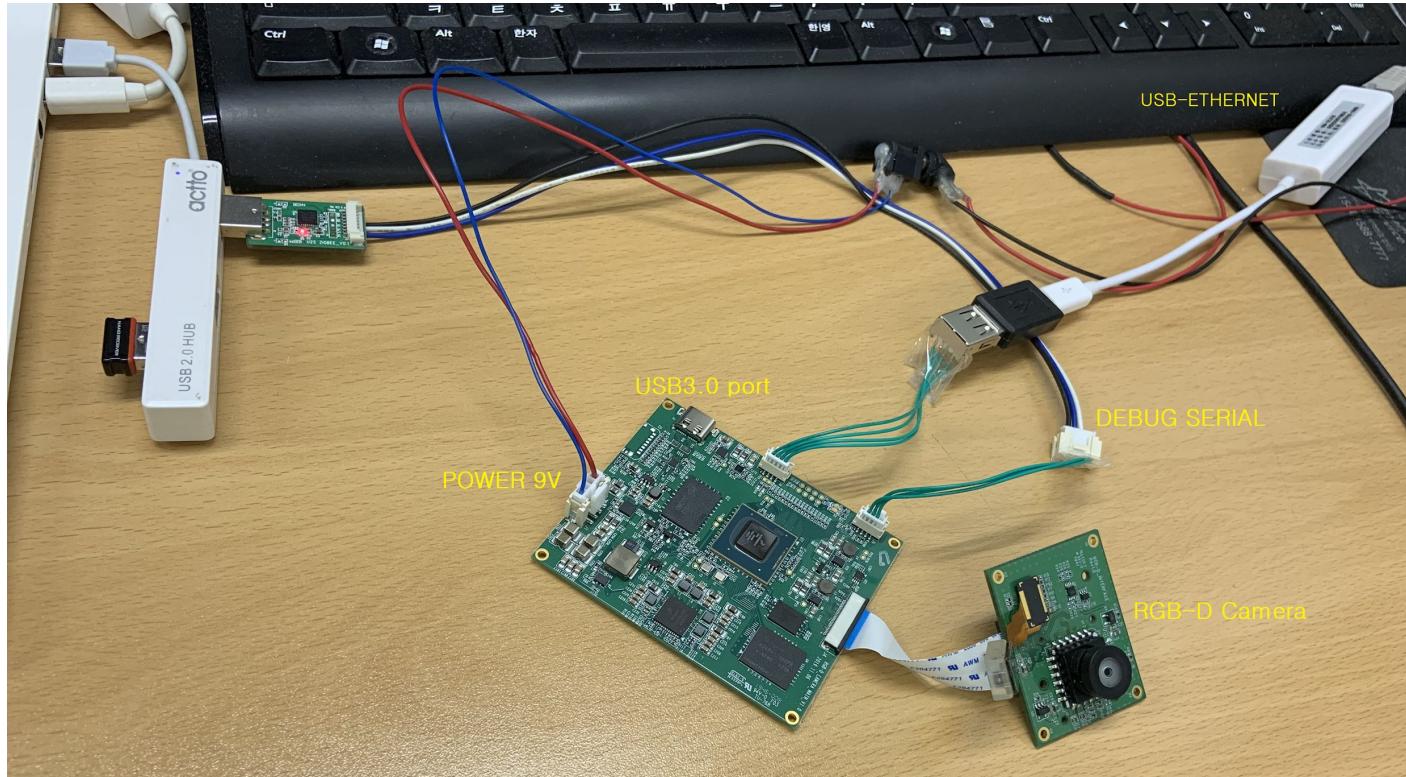
```
> make O=../out_filesystem imx8mq_i4vine_defconfig
```

```
> make O=../out_filesystem
```

2. Output files

FILESYSTEM\_OUTPUT\_PATH/images/rootfs.tar

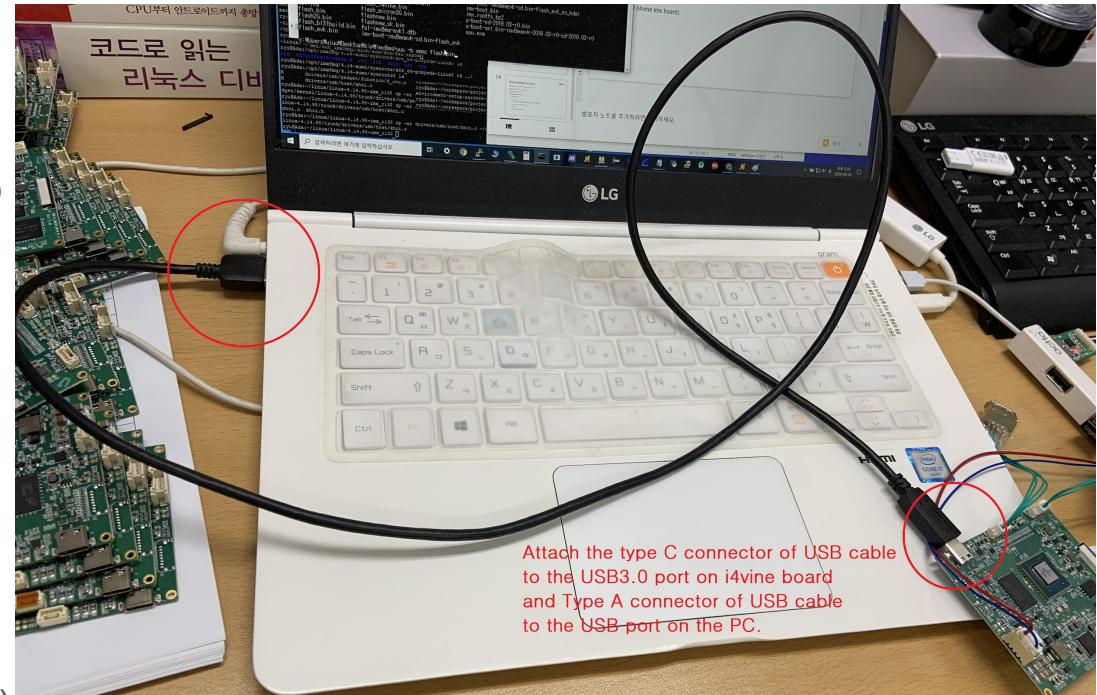
# Setting up the system



# Write flash.bin to emmc

FLASH\_PATH : C:/imx8mq/ (on Windows)

1. Copy uuu.exe to FLASH\_PATH.
2. Copy flash.bin to FLASH\_PATH.
3. Execute cmd.exe (C:\Windows\System32\cmd.exe)
4. Power on i4vine imx8mq board.
5. Enter u-boot mode. (press enter key)
6. Run following command  
> fastboot 0 (i4vine imx board)  
> cd FLASH\_PATH (cmd.exe)  
> uuu -b emmc flash.bin (cmd.exe)
7. Power off and on (i4vine imx8mq board)



# Write kernel to emmc

1. Power on i4vine imx8mq board.
2. Enter u-boot mode (press enter key)  
or After booting
3. Run following command (u-boot mode load file via tftp)

```
> usb reset  
> tftp $loadaddr imx8mq/Image  
> mmc write $loadaddr 0x5000 0xAEED (Image size)  
> tftp $fdt_addr imx8mq/fsl-imx8mq-evk.dtb  
> mmc write $fdt_addr 0x10000 0x800 (dtb size)  
> run bootcmd_mmc
```

# Write kernel to emmc

IP\_ADDR : 192.168.1.213 (modify your ipaddr)    USER\_ID : ryu (modify your id)

SERVER\_IP : 192.168.1.11 (modify your server ip)

SERVER\_FILE\_PATH : /home/test

IMAGE\_NAME : Image

DTB\_NAME : fsl-imx8mq-evk.dtb

3. Run following command (After booting)

```
> ifconfig eth0 IP_ADDR up  
> scp USER_ID@SERVER_IP:SERVER_FILE_PATH/IMAGE_NAME .  
> dd if=/home/test/Image of=/dev/mmcblk0p1 conv=fsync  
> scp USER_ID@SERVER_IP:SERVER_FILE_PATH/DTB_NAME .  
> dd if=/home/test/fsl-imx8mq-evk.dtb of=/dev/mmcblk0 bs=512 seek=65536 count=2048 conv=fsync  
> Power off and on (i4vine imx8mq board)
```

# Write filesystem to emmc

SERVER\_IP : 192.168.1.11 (modify your server ip)

SERVER\_FILE\_PATH : /home/test

FILE\_NAME : rootfs.tar

1. Power on i4inve imx8mq board.
2. After booting.
3. Run following command

## 1) Format filesystem

```
> fdisk /dev/mmcblk0
> d -> 1 -> d ->2
> n -> p -> 1 -> 20480 -> 1024000 -> n -> p -> 2 -> 1228800 -> enter -> p -> w
> mkfs.vfat /dev/mmcblk0p1
> mkfs.ext4 /dev/mmcblk0p2
```

# Write filesystem to emmc

SERVER\_IP : 192.168.1.11 (modify your server ip)    USER\_ID : ryu (modify your id)

SERVER\_FILE\_PATH : /home/test

FILE\_NAME : rootfs.tar

3. Run following command

2) Write filesystem

```
> cd /root  
> scp USER_ID@SERVER_IP:SERVER_FILE_PATH/FILE_NAME .  
> mount /dev/mmcblk0p2 /mnt  
> cd /mnt  
> tar -xvf /root/rootfs.tar  
> cd /  
> umount /mnt
```