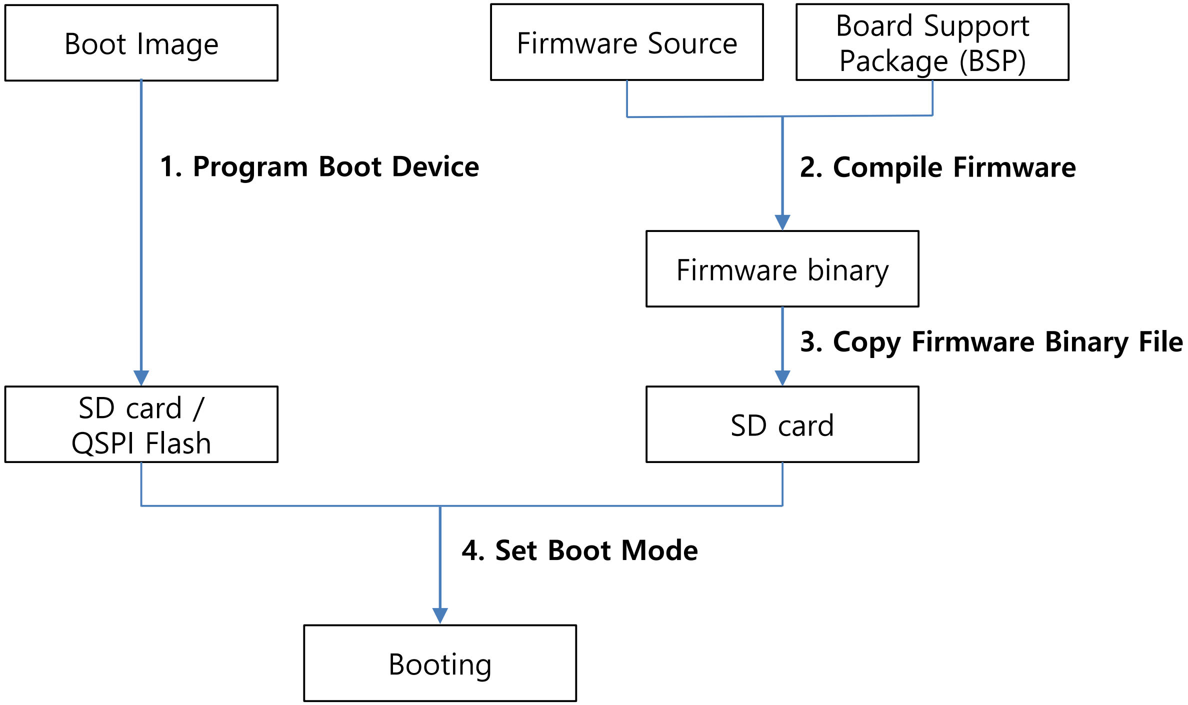
Cosmos+ OpenSSD Boot Guide for SW Developer

Boot Sequence



Materials

- **Firmware**
  - GreedyFTL-2.7.0.c: 8channel-8way configuration
  - GreedyFTL-2.7.1.c: 2channel-8way configuration
- **Boot image support package**
  - BootImageSupoort-0.0.0
    - Board support package (BSP)
    - Makefiles for BSP and firmware
    - Boot image
      - Contain first stage boot loader (FSBL) and FPGA bitstream
        - Xilinx default FSBL is modified for splitting a firmware binary file from a boot image
        - FPGA bitstream is extracted from Prebuild-3.0.0
      - Boot image file type
        - .bin for SD card boot mode
        - .mcs for QSPI flash boot mode

Program Boot Device

Download Boot Image Support Package

- **Download URL:**
  - https://github.com/Cosmos-OpenSSD/Cosmos-plus-OpenSSD/tree/master/support

- **Decompress the downloaded file**
  - cd /download_directory
  - tar xfvz BootImageSupport-0.0.0.tgz

Choose Boot device

- **SD card**
  - Specification
    - Standard SD or SDHC cards
    - FAT 16/32 file system
    - Up to 32GB card densities
    - Use card reader to access SD card
  - Boot image
    - Boot.bin of a boot image support package
- **QSPI flash**
  - Specification
    - Dual parallel configuration
    - More details are in Cosmos OpenSSD tutorial document
  - Boot image
    - Boot.mcs of a boot image support package

Program SD card

- **Format SD card**
  1. Check mounted devices
     - sudo fdisk –l
  2. Find a device name
     - Ex) /dev/sdb1
  3. Unmount the device
     - sudo umount /dev/sdb1
  4. Format the device
     - mkfs.fat -F 32 -I /dev/sdb1
- **Copy boot image**
  1. Make directory for mounting SD card
     - mkdir /home/usb
  2. Mount SD card
     - mount -t vfat /dev/sda1 /home/usb
  3. Copy a boot image file
     - cd /download_directory/BootImageSupport-0.0.0/BootImage
     - cp boot.bin /home/usb

Program QSPI Flash

*** Appendix A: Install Xilinx SDK**

- **Go to boot image directory**
  - cd /download_directory/BootImageSupport-0.0.0/BootImage
- **Program QSPI flash using Xilinx SDK tool***
  - export PATH=$PATH:opt/Xilinx/SDK/2017.1/bin
  - program_flash -f boot.mcs -offset 0 -flash_type qspi_dual_parallel -verify -cable type xilinx_tcf

Compile Firmware

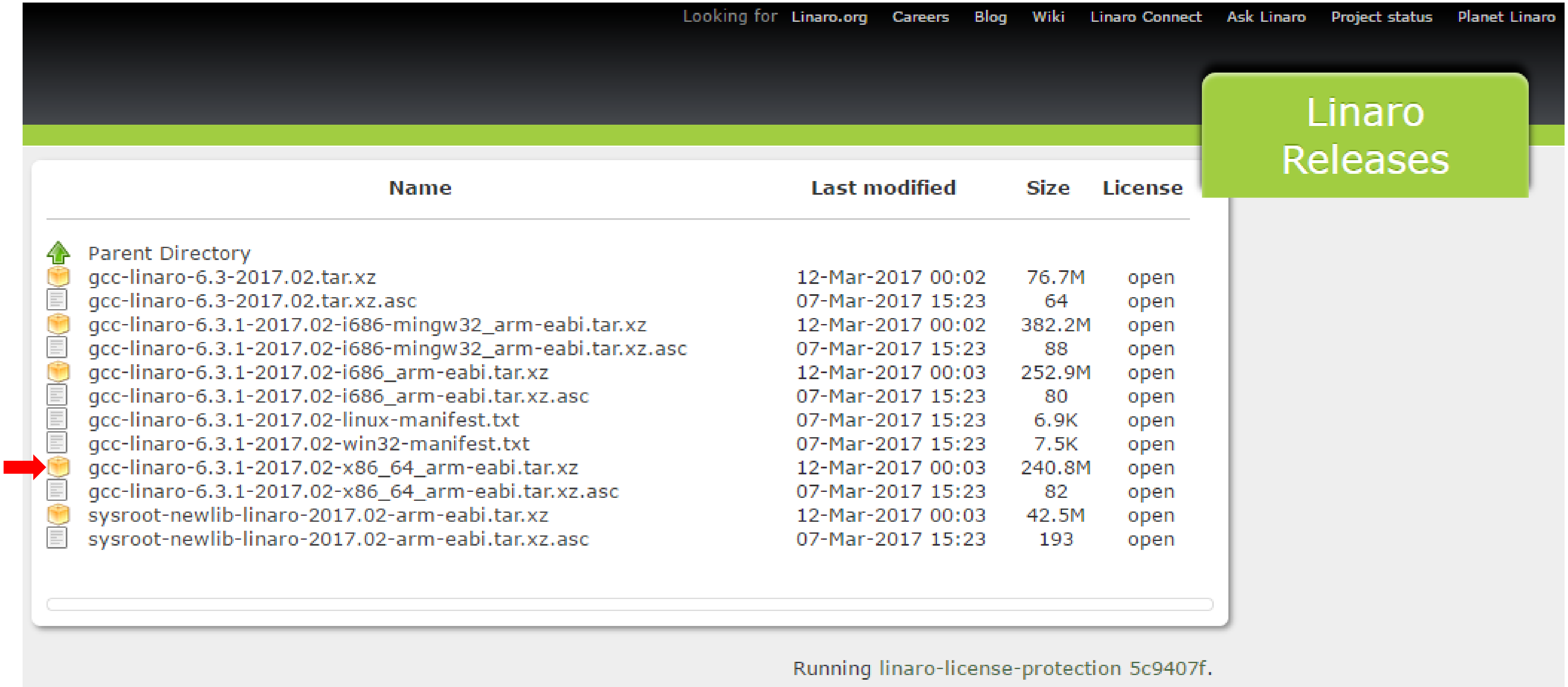
ARM Cross Compiler Toolchain

*** Appendix B: Makefile for Other Toolchain**

- **Target**
  - 32-bit ARMv7 Cortex-A
  - Bard-metal application
- **Tested toolchain***
  - Linaro GCC 6.3
    - https://releases.linaro.org/components/toolchain/binaries/latest/arm-eabi/

Download Toolchain

- Choose the one suited to your PC



Looking for Linaro.org Careers Blog Wiki Linaro Connect Ask Linaro Project status Planet Linaro
Linaro Releases

Name	Last modified	Size	License
Parent Directory			
gcc-linaro-6.3-2017.02.tar.xz	12-Mar-2017 00:02	76.7M	open
gcc-linaro-6.3-2017.02.tar.xz.asc	07-Mar-2017 15:23	64	open
gcc-linaro-6.3.1-2017.02-i686-mingw32_arm-eabi.tar.xz	12-Mar-2017 00:02	382.2M	open
gcc-linaro-6.3.1-2017.02-i686-mingw32_arm-eabi.tar.xz.asc	07-Mar-2017 15:23	88	open
gcc-linaro-6.3.1-2017.02-i686_arm-eabi.tar.xz	12-Mar-2017 00:03	252.9M	open
gcc-linaro-6.3.1-2017.02-i686_arm-eabi.tar.xz.asc	07-Mar-2017 15:23	80	open
gcc-linaro-6.3.1-2017.02-linux-manifest.txt	07-Mar-2017 15:23	6.9K	open
gcc-linaro-6.3.1-2017.02-win32-manifest.txt	07-Mar-2017 15:23	7.5K	open
gcc-linaro-6.3.1-2017.02-x86_64_arm-eabi.tar.xz	12-Mar-2017 00:03	240.8M	open
gcc-linaro-6.3.1-2017.02-x86_64_arm-eabi.tar.xz.asc	07-Mar-2017 15:23	82	open
sysroot-newlib-linaro-2017.02-arm-eabi.tar.xz	12-Mar-2017 00:03	42.5M	open
sysroot-newlib-linaro-2017.02-arm-eabi.tar.xz.asc	07-Mar-2017 15:23	193	open

Running linaro-license-protection 5c9407f.

Install Toolchain

1. **Copy the downloaded file to “/usr”**

```
cd /download_directory
cp gcc-linaro-6.3.1-2017.02-x86_64_arm-eabi.tar.xz /usr
```

2. **Decompress the downloaded file**

```
cd /usr
tar -xvf gcc-linaro-6.3.1-2017.02-x86_64_arm-eabi.tar.xz
```

3. **Add a path of binary files of toolchain**

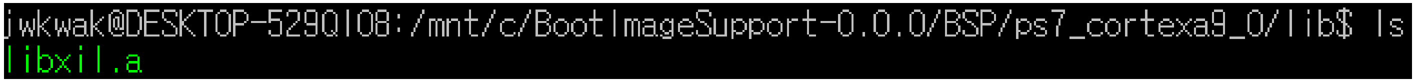
```
export PATH=$PATH:/usr/ gcc-linaro-6.3.1-2017.02-x86_64_arm-eabi/bin
```

Place Sources to Predefined Directory

1. **Download firmware**
   - https://github.com/Cosmos-OpenSSD/Cosmos-plus-OpenSSD/tree/master/source/software
2. **Decompress the downloaded file**
   - cd /download_directory
   - tar xfvz GreedyFTL-2.7.0.c.tgz
3. **Copy firmware sources to a predefined directory**
   - cp -r ./GreedyFTL-2.7.0.c/* ./BootImageSupport-0.0.0/CosmosPlusFirmware/src

Make Library File of BSP

1. **Go to a directory makefile of BSP exists**
   - cd /download_directory/BootImageSupport-0.0.0/BSP
2. **Run makefile**
   - make clean
   - make
3. **Check a created library file**
   - cd /download_directory/ BootImageSupport-0.0.0/BSP/ps7_cortexa9_0/lib
   - ls

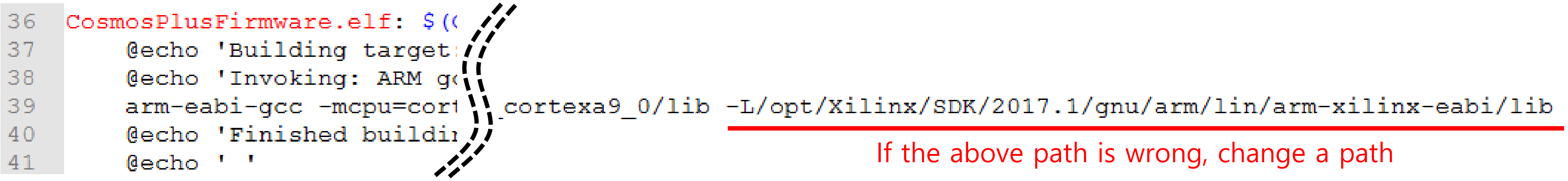


```
jwkwak@DESKTOP-529QI08:/mnt/c/BootImageSupport-0.0.0/BSP/ps7_cortexa9_0/lib$ ls
libxil.a
```

Make Executable File of Firmware

*** Appendix A: Install Xilinx SDK**

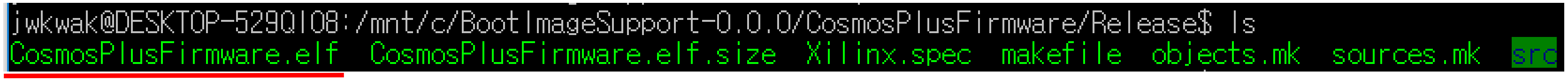
1. **Go to a directory makefile of firmware exists**
   - cd /download_directory/BootImageSupport-0.0.0/CosmosPlusFirmware/Release
2. **Change a path of C standard library of arm-xilinx-eabi-gcc***
   - Line 39 of makefile



```
36 CosmosPlusFirmware.elf: $(
37     @echo 'Building target
38     @echo 'Invoking: ARM g
39     arm-eabi-gcc -mcpu=cort  cortexa9_0/lib -L/opt/Xilinx/SDK/2017.1/gnu/arm/lin/arm-xilinx-eabi/lib
40     @echo 'Finished buildi
41     @echo ' '
```

If the above path is wrong, change a path

3. **Run makefile**
   - make clean
   - make
4. **Check a created library file**
   - ls



```
jwkwak@DESKTOP-529QI08:/mnt/c/BootImageSupport-0.0.0/CosmosPlusFirmware/Release$ ls
CosmosPlusFirmware.elf  CosmosPlusFirmware.elf.size  Xilinx.spec  makefile  objects.mk  sources.mk  src
```

Executable file of firmware

Make Binary File of Firmware

- **Transform executable file to binary file using objcopy**

  -
    ```
    arm-eabi-objcopy -S -O binary CosmosPlusFirmware.elf UserFW.bin
    ```
    - Name of a binary file should be UserFW

Copy Firmware Binary File

Copy Firmware Binary File

Format SD card

1. Check mounted devices
   - sudo fdisk –l
2. Find a device name
   - Ex) /dev/sdb1
3. Unmount the device
   - sudo umount /dev/sdb1
4. Format the device
   - mkfs.fat -F 32 -I /dev/sdb1

Copy firmware binary file

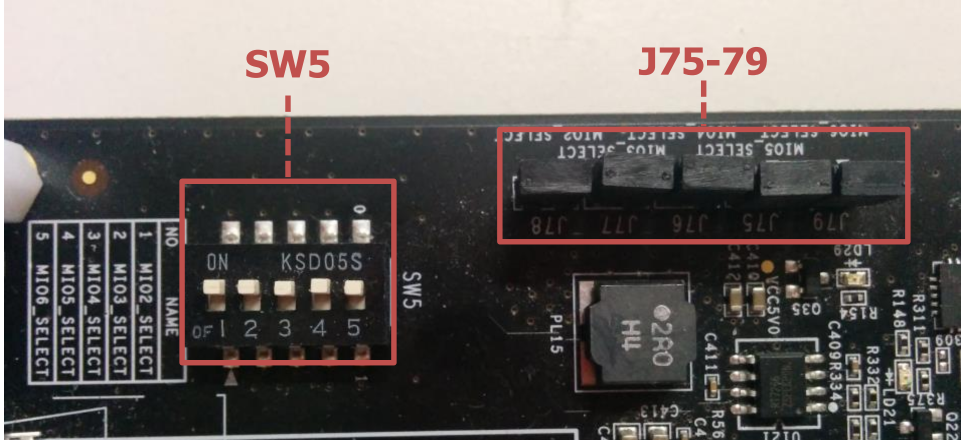
1. Make directory for mounting SD card
   - mkdir /home/usb
2. Mount SD card
   - mount -t vfat /dev/sda1 /home/usb
3. Copy a firmware binary file
   - cp /download_directory/BootImageSupport-0.0.0/CosmosPlusFirmware/Release
   - cp UserFW.bin /home/usb

These steps are not needed if you have
chosen SD card as a boot device
(the same SD card is used)

Set Boot Mode

Set Platform Board

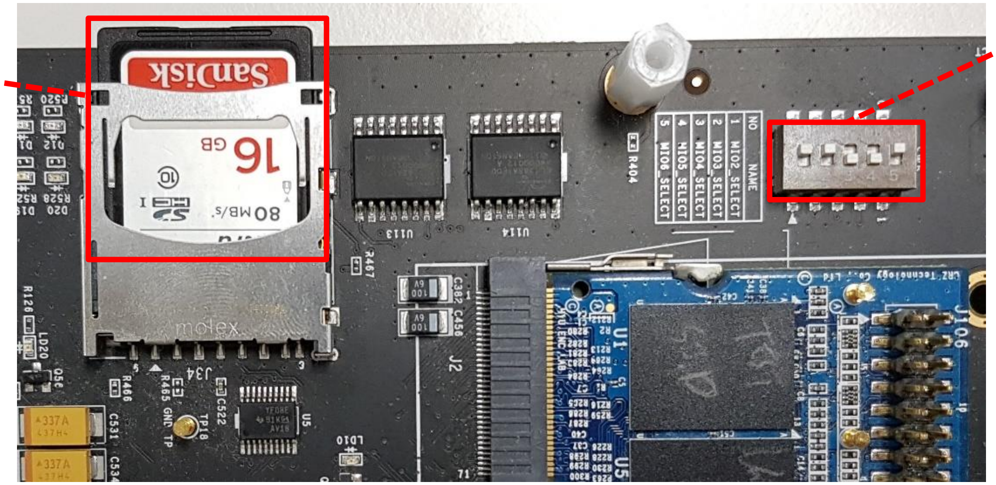
- **Set SW5 of platform board according to the boot device**



Mode	SW5					J75-79
	1	2	3	4	5	
JTAG	On	On	On	On	On	On
QSPI			On	Off		
SD card			Off	Off		

SD Card Boot Mode

- **Insert SD card to J34 of platform board**



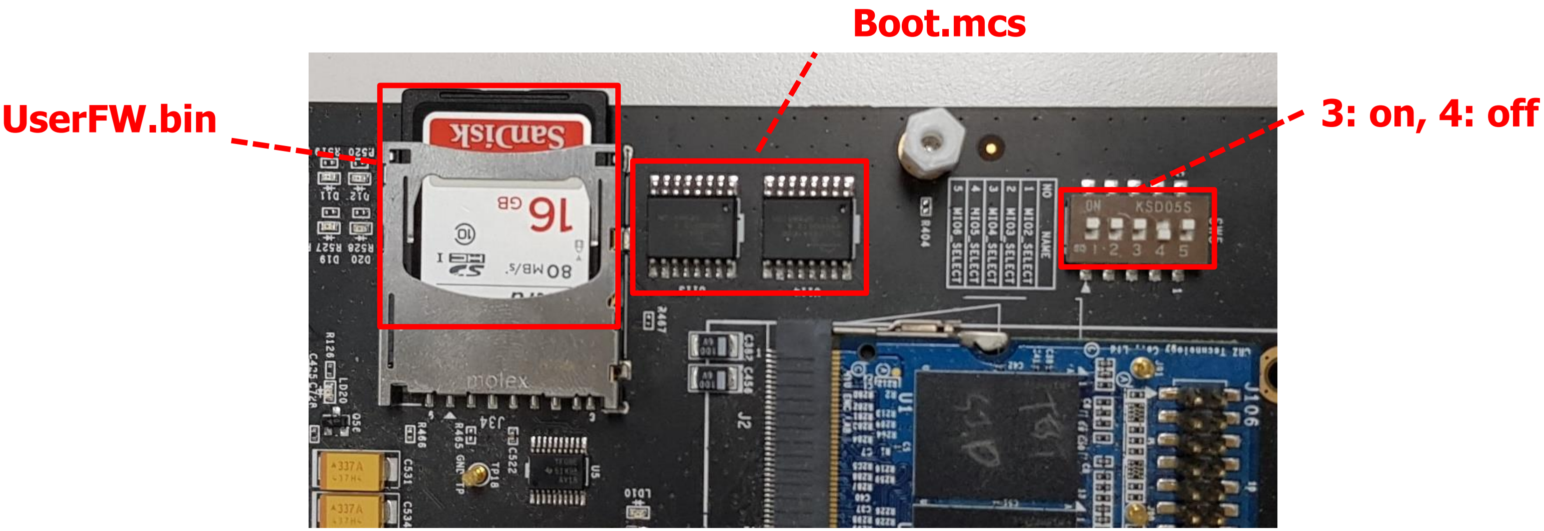
Boot.bin
UserFW.bin

3: off, 4: off

- **Power on the board**
  - FPGA booting is completed in a few seconds

QSPI Flash Boot Mode

- **Insert SD card to J34 of platform board**

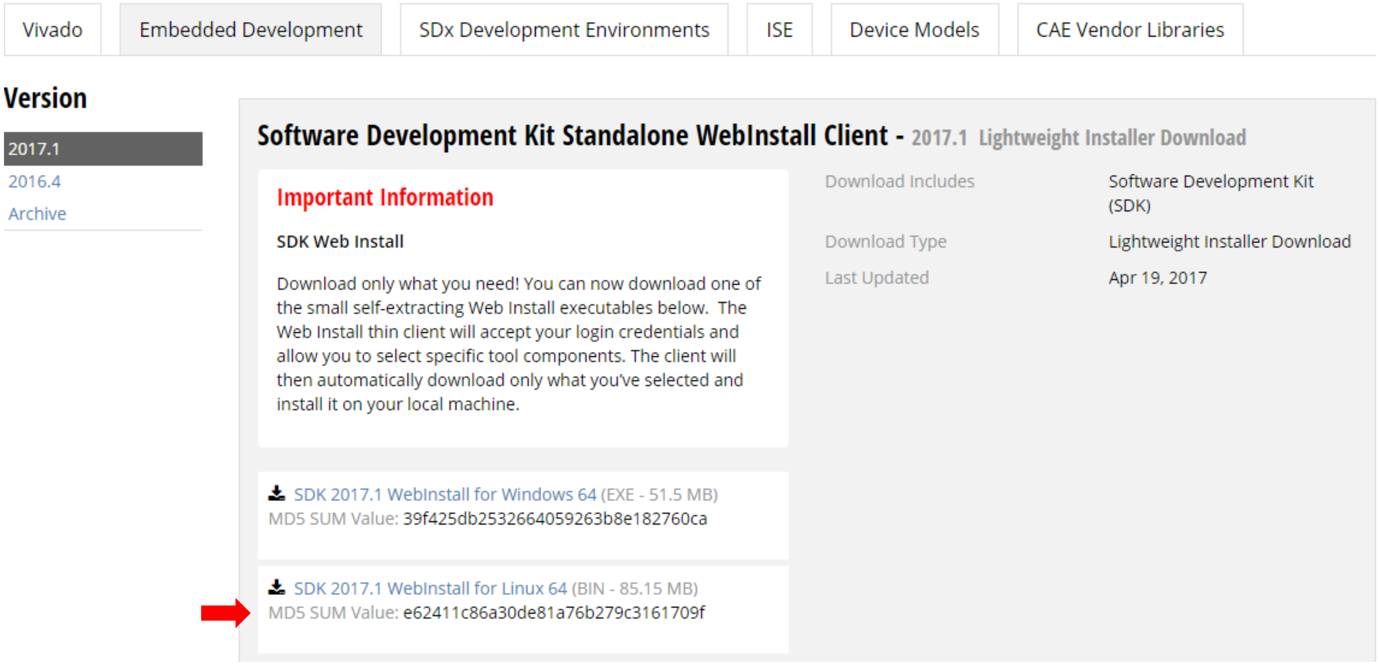


- **Power on the board**
  - FPGA booting is completed in a few seconds

Appendix A: Xilinx SDK Install

Xilinx SDK

- **Download URL:**
  - https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools.html



Software Development Kit Standalone WebInstall Client - 2017.1 Lightweight Installer Download

Important Information

SDK Web Install

Download only what you need! You can now download one of the small self-extracting Web Install executables below. The Web Install thin client will accept your login credentials and allow you to select specific tool components. The client will then automatically download only what you've selected and install it on your local machine.

Download Includes	Software Development Kit (SDK)
Download Type	Lightweight Installer Download
Last Updated	Apr 19, 2017

SDK 2017.1 WebInstall for Windows 64 (EXE - 51.5 MB)
MD5 SUM Value: 39f425db2532664059263b8e182760ca

SDK 2017.1 WebInstall for Linux 64 (BIN - 85.15 MB)
MD5 SUM Value: e62411c86a30de81a76b279c3161709f

Install Xilinx SDK

1. **Change the permissions of the downloaded file**
   - chmod +x ./Xilinx_Vivado_SDK_2014.1_0405_1_Lin64.bin
2. **Execute the downloaded file**
   - sudo ./Xilinx_Vivado_SDK_2014.1_0405_1_Lin64.bin

✓ **If there are errors**

- https://www.xilinx.com/support/answers/62241.html

Install Xilinx SDK to Linux Subsystem of Window 10

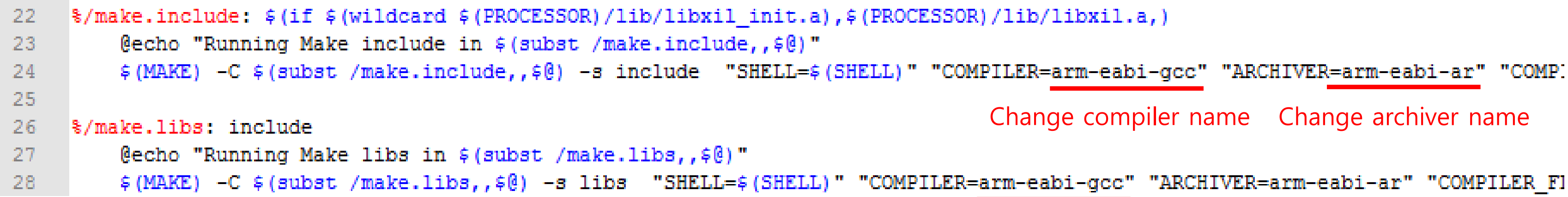
- **Window 10 supports linux subsystem**
  - https://msdn.microsoft.com/en-us/commandline/wsl/install_guide
- **Some programs are needed for graphical linux desktop application**
  - Install X server to window
    - https://sourceforge.net/projects/xming/
  - Install vim-gtk to linux subsystem
    - sudo apt-get install vim-gtk
  - Modify a environmental variable (DISPLAY)
    - export DISPLAY=:0
  - Reference
    - https://www.howtogeek.com/261575/how-to-run-graphical-linux-desktop-applications-from-windows-10s-bash-shell/
- **Change the permissions of the SDK install file**
  - chmod +x ./Xilinx_Vivado_SDK_2014.1_0405_1_Lin64.bin
- **Execute the SDK install file**
  - sudo ./Xilinx_Vivado_SDK_2014.1_0405_1_Lin64.bin

Appendix B: Makefile for Other Toolchain

Edit Makefile of BSP

Target makefile 1

- BootImageSupport-0.0.0/BSP/makefile
- Line 24, 28



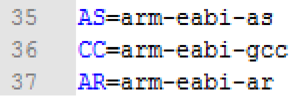
```
22 %/make.include: $(if $(wildcard $(PROCESSOR)/lib/libxil_init.a),$(PROCESSOR)/lib/libxil.a,)
23     @echo "Running Make include in $(subst /make.include,,$@)"
24     $(MAKE) -C $(subst /make.include,,$@) -s include  "SHELL=$(SHELL)" "COMPILER=arm-eabi-gcc" "ARCHIVER=arm-eabi-ar" "COMP
25 
26 %/make.libs: include
27     @echo "Running Make libs in $(subst /make.libs,,$@)"
28     $(MAKE) -C $(subst /make.libs,,$@) -s libs  "SHELL=$(SHELL)" "COMPILER=arm-eabi-gcc" "ARCHIVER=arm-eabi-ar" "COMPILER_F
```

Change compiler name Change archiver name

Change compiler name Change archiver name

Target makefile 2

- BootImageSupport-0.0.0/BSP/ps7_cortexa9_0/libsrc/standalone_v4_2/src/makefile
- Line 35~37



```
35 AS=arm-eabi-as
36 CC=arm-eabi-gcc
37 AR=arm-eabi-ar
```

← Change assembler name

← Change compiler name

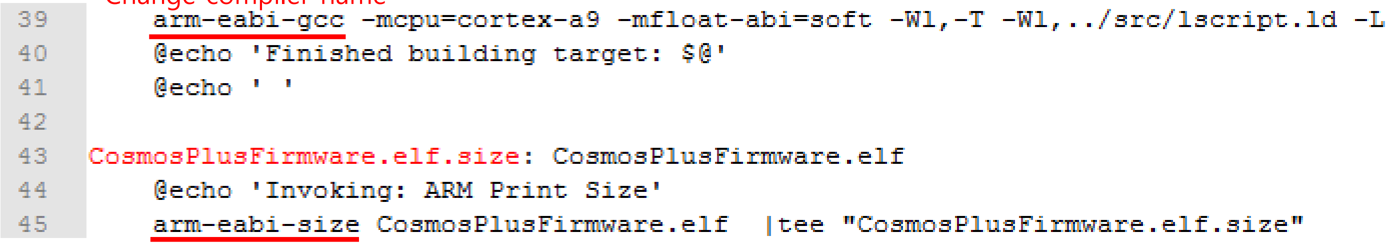
← Change archiver name

Edit Makefile of Firmware [1/2]

Target makefile 1

- BootImageSupport-0.0.0/CommosPlusFirmware/Release/makefile
- Line 39, 45

Change compiler name

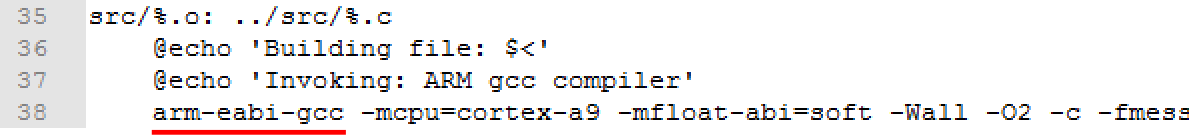


```
39      arm-eabi-gcc -mcpu=cortex-a9 -mfloat-abi=soft -Wl,-T -Wl,../src/lscript.ld -L
40      @echo 'Finished building target: $@'
41      @echo ' '
42
43  CosmosPlusFirmware.elf.size: CosmosPlusFirmware.elf
44      @echo 'Invoking: ARM Print Size'
45      arm-eabi-size CosmosPlusFirmware.elf  |tee "CosmosPlusFirmware.elf.size"
```

Change size print utility

Target makefile 2

- BootImageSupport-0.0.0/CommosPlusFirmware/Release/src/subdir.mk
- Line 38



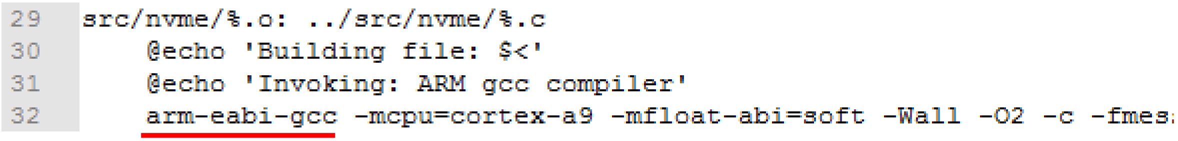
```
35  src/%.o: ../src/%.c
36      @echo 'Building file: $<'
37      @echo 'Invoking: ARM gcc compiler'
38      arm-eabi-gcc -mcpu=cortex-a9 -mfloat-abi=soft -Wall -O2 -c -fmess
```

Change compiler name

Edit Makefile of Firmware [2/2]

Target makefile 3

- BootImageSupport-0.0.0/CommosPlusFirmware/Release/src/nvme/subdir.mk
- Line 32



```
29 src/nvme/%.o: ../src/nvme/%.c
30     @echo 'Building file: $<'
31     @echo 'Invoking: ARM gcc compiler'
32     arm-eabi-gcc -mcpu=cortex-a9 -mfloat-abi=soft -Wall -O2 -c -fmes
```

Change compiler name

Thank You