

Xilinx Zynq FPGA, TI DSP, MCU 프로그래밍 및 회로 설계 전문가 과정

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

AM5728 Heterogeneous Architecture Based Linux Kernel Compile

Let's Build Custom Linux Kernel

먼저 기존의 TI 가 제공하고 있는 루트 파일 시스템을 활용할 필요가 있다.
그러므로 임시 파일 시스템을 저장할 공간을 아래와 같이 만들어두도록 한다.
향후 SD Card 에 구울때 파일 시스템으로 이 위치를 지정해주면 된다.

```
kwrg@kwrg-Samsung-DeskTop-System: ~  
kwrg@kwrg-Samsung-DeskTop-System:~$ mkdir tmp_fs
```

TI SDK 위치에서 filesystem 쪽으로 접근한다.

```
kwrg@kwrg-Samsung-DeskTop-System: ~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05/  
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/filesystem$ ls  
arago-base-tisdk-image-am57xx-evm.tar.xz  tisdk-rootfs-image-am57xx-evm.tar.xz  
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/filesystem$
```

우선 파일 시스템을 복사해둔다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/filesystem$ cp tisdk-rootfs-image-am57xx-evm.tar.xz ~/tmp_fs
```

복사한 녀석을 해당 위치에서 압축 해제한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/filesystem$ cd ~/tmp_fs  
kwrg@kwrg-Samsung-DeskTop-System:~/tmp_fs$ tar -xf tisdk-rootfs-image-am57xx-evm  
.tar.xz
```

모두 완료하였다면 지운다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/filesystem$ cd ~/tmp_fs  
kwrg@kwrg-Samsung-DeskTop-System:~/tmp_fs$ rm -rf tisdk-rootfs-image-am57xx-evm.  
tar.xz
```

실제 커널 디렉토리로 가서
경로상에 크로스컴파일러 경로를
아래와 같이 입력한다.

```
kwrq@kwrq-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05
/board-support$ ls
extra-drivers
linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8
prebuilt-images
u-boot-2016.05+gitAUTOINC+6c5519b6fc-g6c5519b6fc
kwrq@kwrq-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05
/board-support$ cd linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8/
kwrq@kwrq-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05
/board-support/linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8$ ls
COPYING          System.map        include           scripts
CREDITS          TISDK-README     init             security
Documentation    arch              ipc              sound
Kbuild           backported-features kernel            ti_config_fragments
Kconfig          block             lib              tools
MAINTAINERS      certs             mm               usr
Makefile         crypto            modules.builtin  virt
Module.symvers   drivers           modules.order    vmlinux
README           firmware         net              vmlinux.o
REPORTING-BUGS   fs                samples
kwrq@kwrq-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05
/board-support/linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8$ export PATH=~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05/linux-devkit/sysroots/x86_64-arago-li
nux/usr/bin/
```

기존에 컴파일 되어 있는 정보를 깔끔하게 모두 정리한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/board-support/linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8$ make ARCH=arm CRO  
SS_COMPILE=arm-linux-gnueabihf- distclean
```

TI SDK AM57XX EVM 에 적절한 형태로 설정을 한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/board-support/linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8$ make ARCH=arm CRO  
SS_COMPILE=arm-linux-gnueabihf- tisdsk_am57xx-evm_defconfig
```

컴파일을 수행하기 위해 Architecture 로 arm 을 지정해둔다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/board-support/linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8$ export ARCH=arm
```

실제 Kernel Image 를 컴파일 한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/board-support/linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- zImage
```

Kernel 의 Device Tree 부분을 컴파일 한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/board-support/linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- am57xx-evm-reva3.dtb
```

reva3 과 일반을 모두 해두도록 한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/board-support/linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- am57xx-evm.dtb
```

Linux Device Driver 에 해당하는 Module 을 컴파일 하기 시작한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/board-support/linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- modules
```

앞서서 미리 작업해봤던 임시 파일 시스템쪽에 Module 을 설치한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/board-support/linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8$ make ARCH=arm INSTALL_MOD_PATH=/home/kwrg/tmp_fs modules_install
```

U-Boot 컴파일을 수행하기 위해 Device Tree Compiler 를 설치하도록 한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/board-support/linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8$ sudo apt-get install all device-tree-compiler
```


이제 U-Boot 관련 부분을 작업하도록 위치를 이동시키도록 한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05
/board-support/linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8$ cd ..
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05
/board-support$ ls
extra-drivers
linux-4.4.32+gitAUTOINC+adde2ca9f8-gadde2ca9f8
prebuilt-images
u-boot-2016.05+gitAUTOINC+6c5519b6fc-g6c5519b6fc
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05
/board-support$ cd u-boot-2016.05+gitAUTOINC+6c5519b6fc-g6c5519b6fc/
```

U-Boot 에서도 기존의 정보를 깔끔하게 정리한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05
/board-support/u-boot-2016.05+gitAUTOINC+6c5519b6fc-g6c5519b6fc$ make ARCH=arm C
ROSS_COMPILE=arm-linux-gnueabihf- distclean
```

우리 Target EVM 에 적절하게 크로스 컴파일을 수행한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/board-support/u-boot-2016.05+gitAUTOINC+6c5519b6fc-g6c5519b6fc$ make ARCH=arm C  
ROSS_COMPILE=arm-linux-gnueabihf- O=am57xx_evm am57xx_evm_defconfig
```

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/board-support/u-boot-2016.05+gitAUTOINC+6c5519b6fc-g6c5519b6fc$ make ARCH=arm C  
ROSS_COMPILE=arm-linux-gnueabihf- O=am57xx_evm
```

모든 작업이 완료되면 am57xx_evm 디렉토리로 이동한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/board-support/u-boot-2016.05+gitAUTOINC+6c5519b6fc-g6c5519b6fc$ ls  
Kbuild      Makefile    arch        configs     examples    post  
Kconfig     README      board       disk        fs          scripts  
Licenses    TISDK-README cmd         doc         include     snapshot.commit  
MAINTAINERS am57xx_evm  common      drivers     lib         test  
MAKEALL     api         config.mk   dts         net         tools  
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05  
/board-support/u-boot-2016.05+gitAUTOINC+6c5519b6fc-g6c5519b6fc$ cd am57xx_evm/
```

다음으로 임시 u-boot 디렉토리를 아래와 같이 만든다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05/board-support/u-boot-2016.05+gitAUTOINC+6c5519b6fc-g6c5519b6fc/am57xx_evm$ mkdir ~/tmp_boot
```

u-boot.img 에 해당하는 부트 이미지를 임시 부트 디렉토리로 복사한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05/board-support/u-boot-2016.05+gitAUTOINC+6c5519b6fc-g6c5519b6fc/am57xx_evm$ cp u-boot.img ~/tmp_boot/
```

추가적으로 MLO 도 복사해줘야 한다.

```
kwrg@kwrg-Samsung-DeskTop-System:~/ti-processor-sdk-linux-am57xx-evm-03.02.00.05/board-support/u-boot-2016.05+gitAUTOINC+6c5519b6fc-g6c5519b6fc/am57xx_evm$ cp MLO ~/tmp_boot/
```

이제 SD Card 를 만드는 방법을 통해서 tmp_fs 와 tmp_boot 를 선택하여 SD Card 를 만들고 이를 통해서 DSP 를 부팅하면 우리가 개조한 Kernel 이 잘 올라감을 확인할 수 있을 것이다.