Embedded System Software Project

2020 Spring

20151556 변홍수

Goal

JNI를 사용한 택시 미터기 프로그램과 지금까지 진행한 숙제 (HW1 – HW3)를 외부와 연결을 통한 입력 없이 자체적으로 실행 할 수 있는 하나의 안드로이드 어플리케이션 구현

Big Contents

- Kernel Image 수정
- Android Application
- JNI Program
- Final Flowchart

Kernel Image 수정

[GPIO Button]

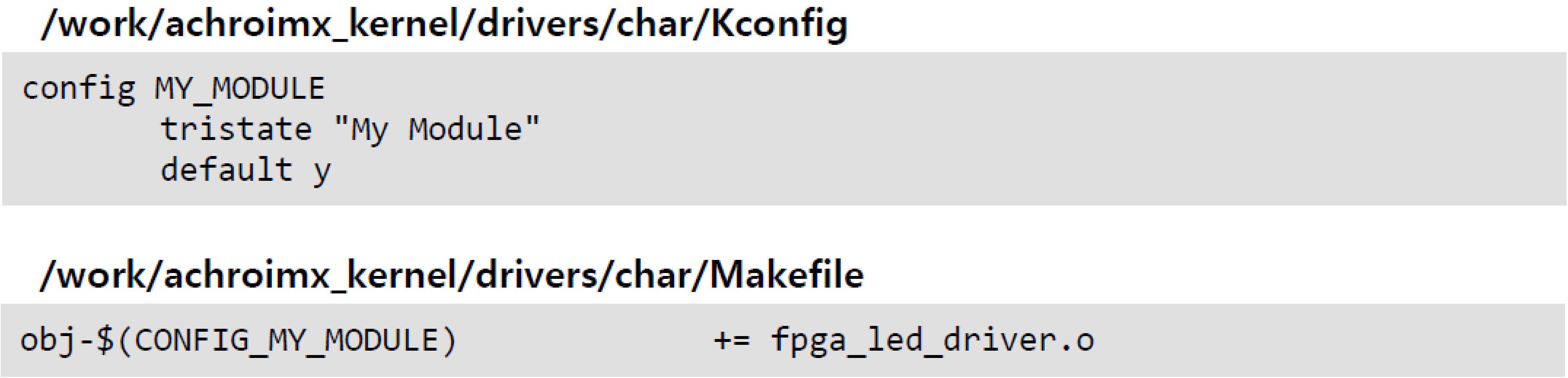
/work/achroimx_kernel/arch/arm/mach-mx6/board-achroimx.c

```
static struct gpio_keys_button ard_buttons[] = {
//      GPIO_BUTTON(SABREAUTO_ANDROID_HOME,
                    KEY_HOME,      1, "home",       0),
//      GPIO_BUTTON(SABREAUTO_ANDROID_BACK,
                    KEY_BACK,      1, "back",       0),
        GPIO_BUTTON(SABREAUTO_ANDROID_VOLUP,
                    KEY_VOLUMEUP,   1, "volume-up",   0),
        GPIO_BUTTON(SABREAUTO_ANDROID_VOLDOWN,
                    KEY_VOLUMEDOWN, 1, "volume-down", 0),
        GPIO_BUTTON(SABREAUTO_ANDROID_POWER,
                    KEY_POWER,     1, "power-key",   1),
};
```

Kernel Image 수정

[Custom Module 등록]

아래의 코드를 해당 파일에 추가



/work/achroimx_kernel/drivers/char/Kconfig

```
config MY_MODULE
        tristate "My Module"
        default y
```

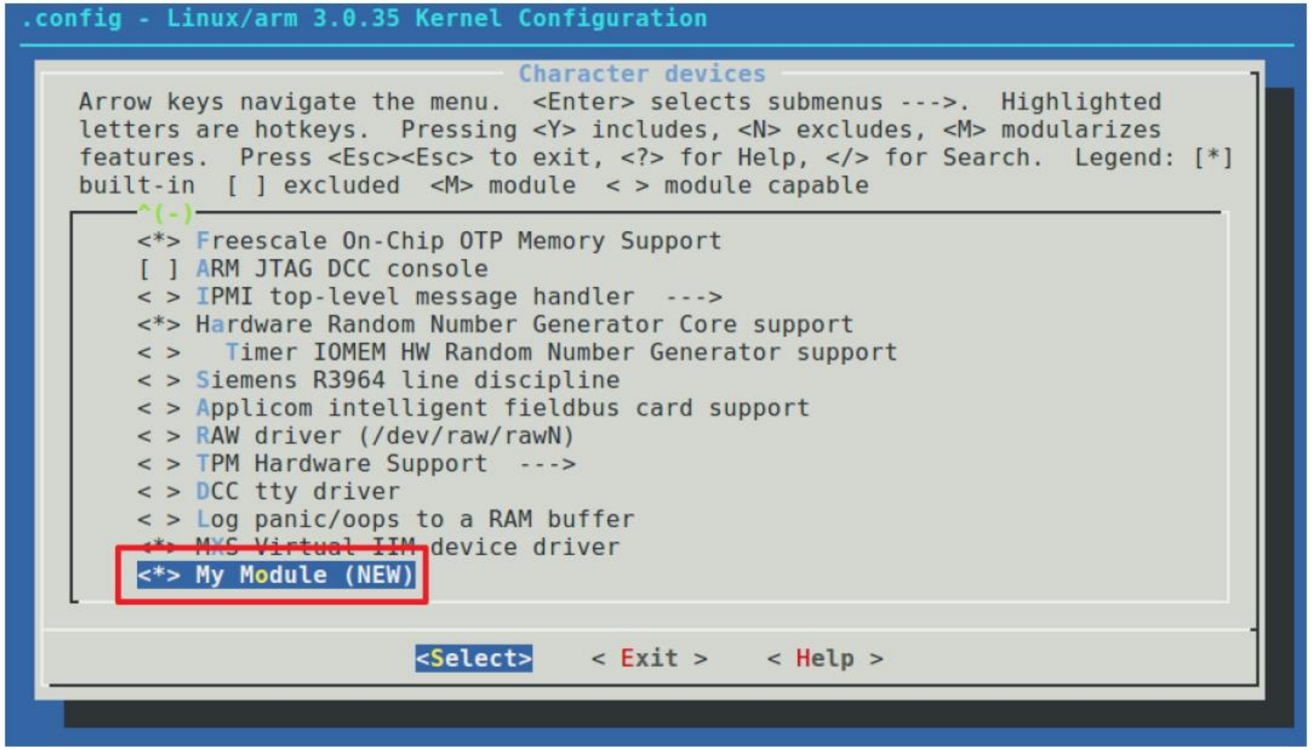
/work/achroimx_kernel/drivers/char/Makefile

```
obj-$(CONFIG_MY_MODULE)              += fpga_led_driver.o
```

Kernel Image 수정

[Custom Module 등록]

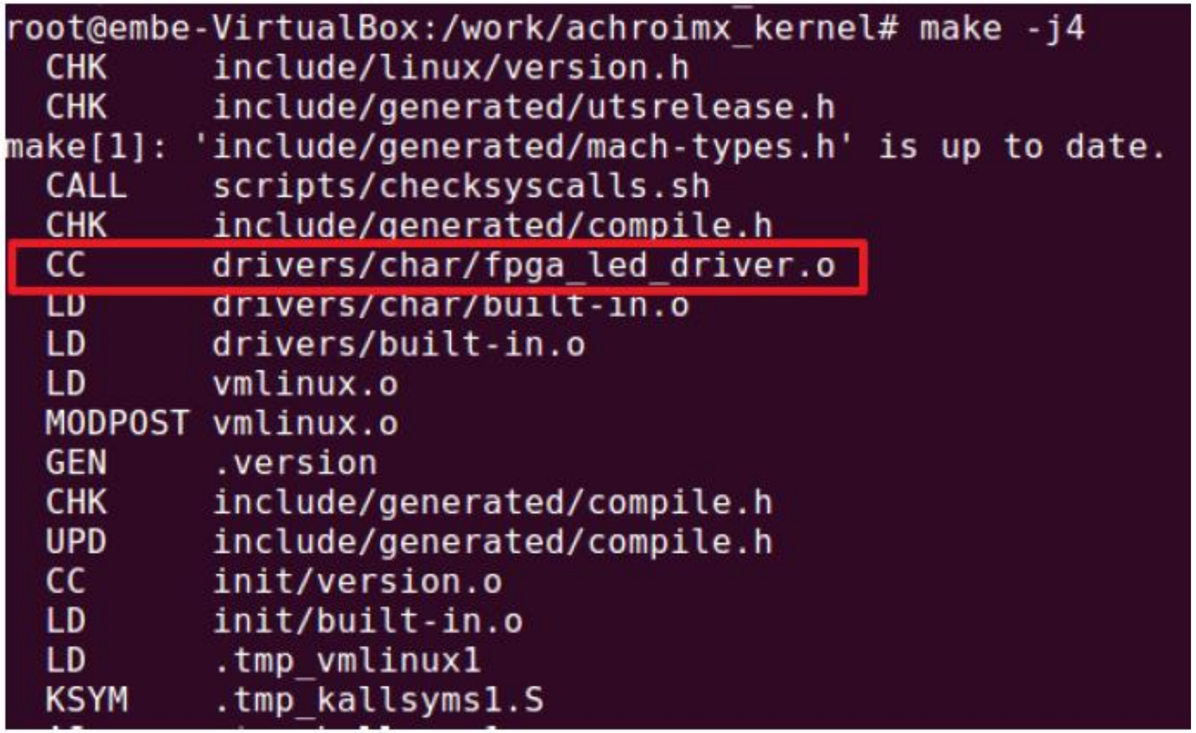
make menuconfig



```
.config - Linux/arm 3.0.35 Kernel Configuration
                         Character devices
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted
  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes
  features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*]
  built-in  [ ] excluded  <M> module  < > module capable
         ^(-)
        <*> Freescale On-Chip OTP Memory Support
        [ ] ARM JTAG DCC console
        < > IPMI top-level message handler  --->
        <*> Hardware Random Number Generator Core support
        < >   Timer IOMEM HW Random Number Generator support
        < > Siemens R3964 line discipline
        < > Applicom intelligent fieldbus card support
        < > RAW driver (/dev/raw/rawN)
        < > TPM Hardware Support  --->
        < > DCC tty driver
        < > Log panic/oops to a RAM buffer
        <*> MXS Virtual IIM device driver
        <*> My Module (NEW)

                  <Select>    < Exit >    < Help >
```

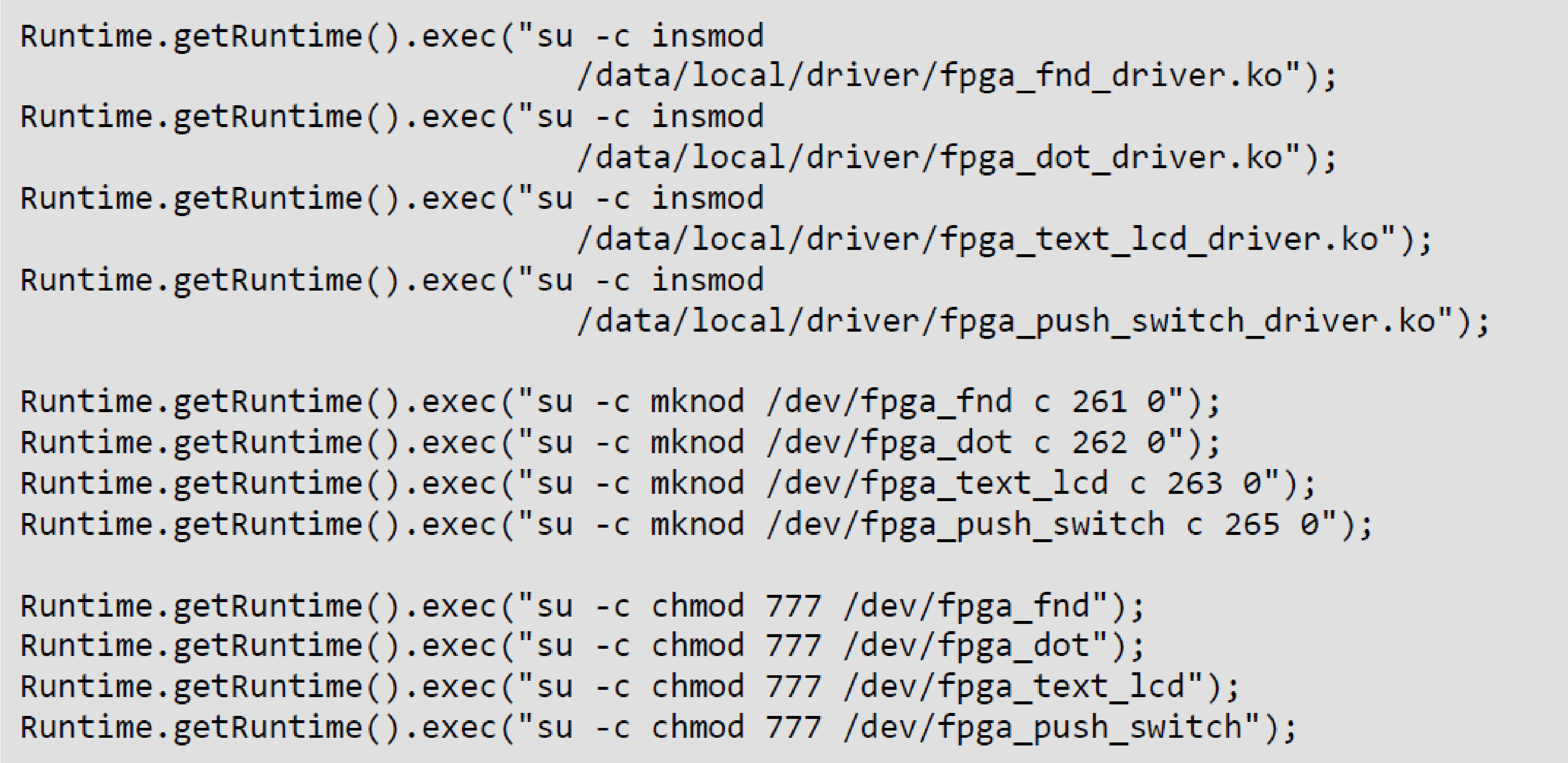
make –j4



```
root@embe-VirtualBox:/work/achroimx_kernel# make -j4
  CHK     include/linux/version.h
  CHK     include/generated/utsrelease.h
make[1]: 'include/generated/mach-types.h' is up to date.
  CALL    scripts/checksyscalls.sh
  CHK     include/generated/compile.h
  CC      drivers/char/fpga_led_driver.o
  LD      drivers/char/built-in.o
  LD      drivers/built-in.o
  LD      vmlinux.o
  MODPOST vmlinux.o
  GEN     .version
  CHK     include/generated/compile.h
  UPD     include/generated/compile.h
  CC      init/version.o
  LD      init/built-in.o
  LD      .tmp_vmlinux1
  KSYM    .tmp_kallsyms1.S
```

Android Application [Execute Linux Command Example]

Runtime.getRuntime().exec() Method 사용

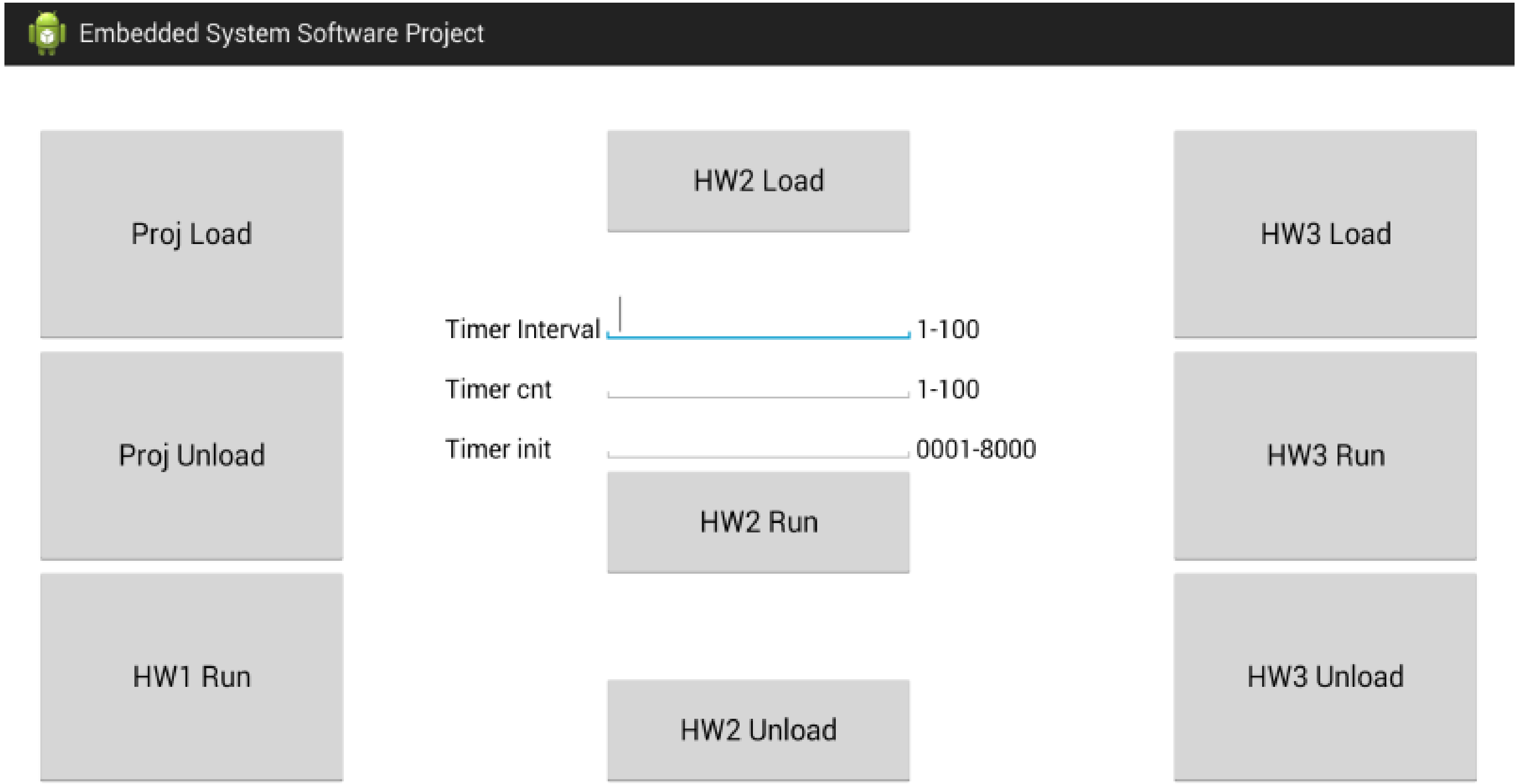


```
Runtime.getRuntime().exec("su -c insmod
                             /data/local/driver/fpga_fnd_driver.ko");
Runtime.getRuntime().exec("su -c insmod
                             /data/local/driver/fpga_dot_driver.ko");
Runtime.getRuntime().exec("su -c insmod
                             /data/local/driver/fpga_text_lcd_driver.ko");
Runtime.getRuntime().exec("su -c insmod
                             /data/local/driver/fpga_push_switch_driver.ko");

Runtime.getRuntime().exec("su -c mknod /dev/fpga_fnd c 261 0");
Runtime.getRuntime().exec("su -c mknod /dev/fpga_dot c 262 0");
Runtime.getRuntime().exec("su -c mknod /dev/fpga_text_lcd c 263 0");
Runtime.getRuntime().exec("su -c mknod /dev/fpga_push_switch c 265 0");

Runtime.getRuntime().exec("su -c chmod 777 /dev/fpga_fnd");
Runtime.getRuntime().exec("su -c chmod 777 /dev/fpga_dot");
Runtime.getRuntime().exec("su -c chmod 777 /dev/fpga_text_lcd");
Runtime.getRuntime().exec("su -c chmod 777 /dev/fpga_push_switch");
```

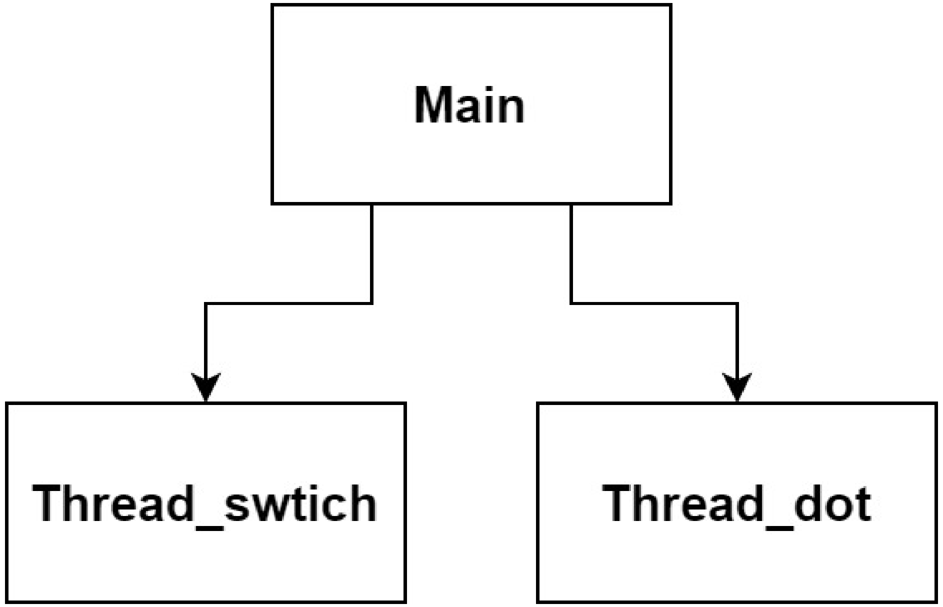
Android Application [Graphic Interface]



JNI Program
[Thread]

택시 미터기 구현 JNI JAVA 에서 FPGA 의 FND, DOT, Text LCD 를 계속해서 업데이트 하기 위하여 새로운 Thread 2 개를 생성

Thread Interrupt 로 인해 Thread 가 종료될 때까지 해당 기능을 수행



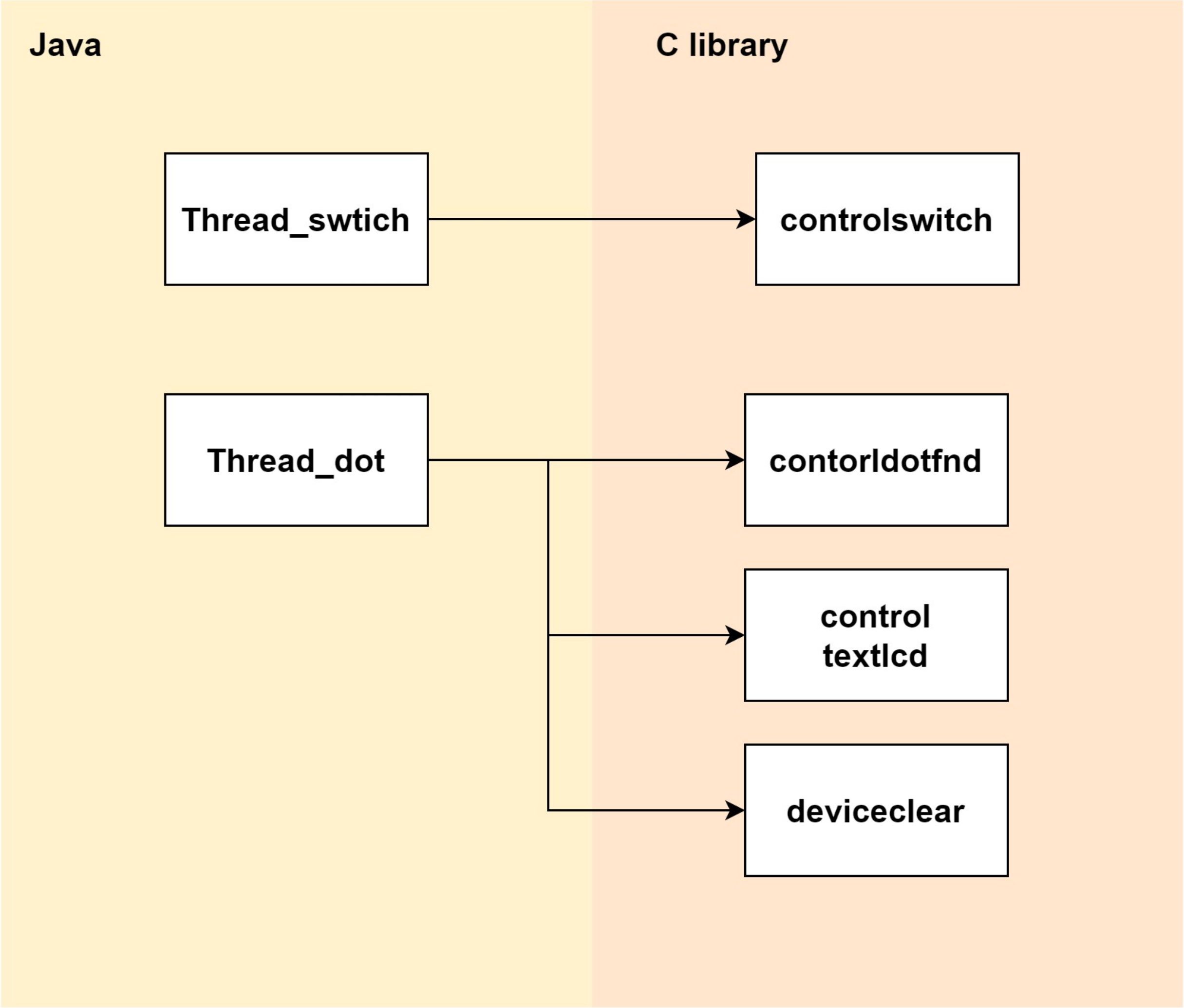
JNI Program
[C Function]

controlswitch : FPGA Switch 입력

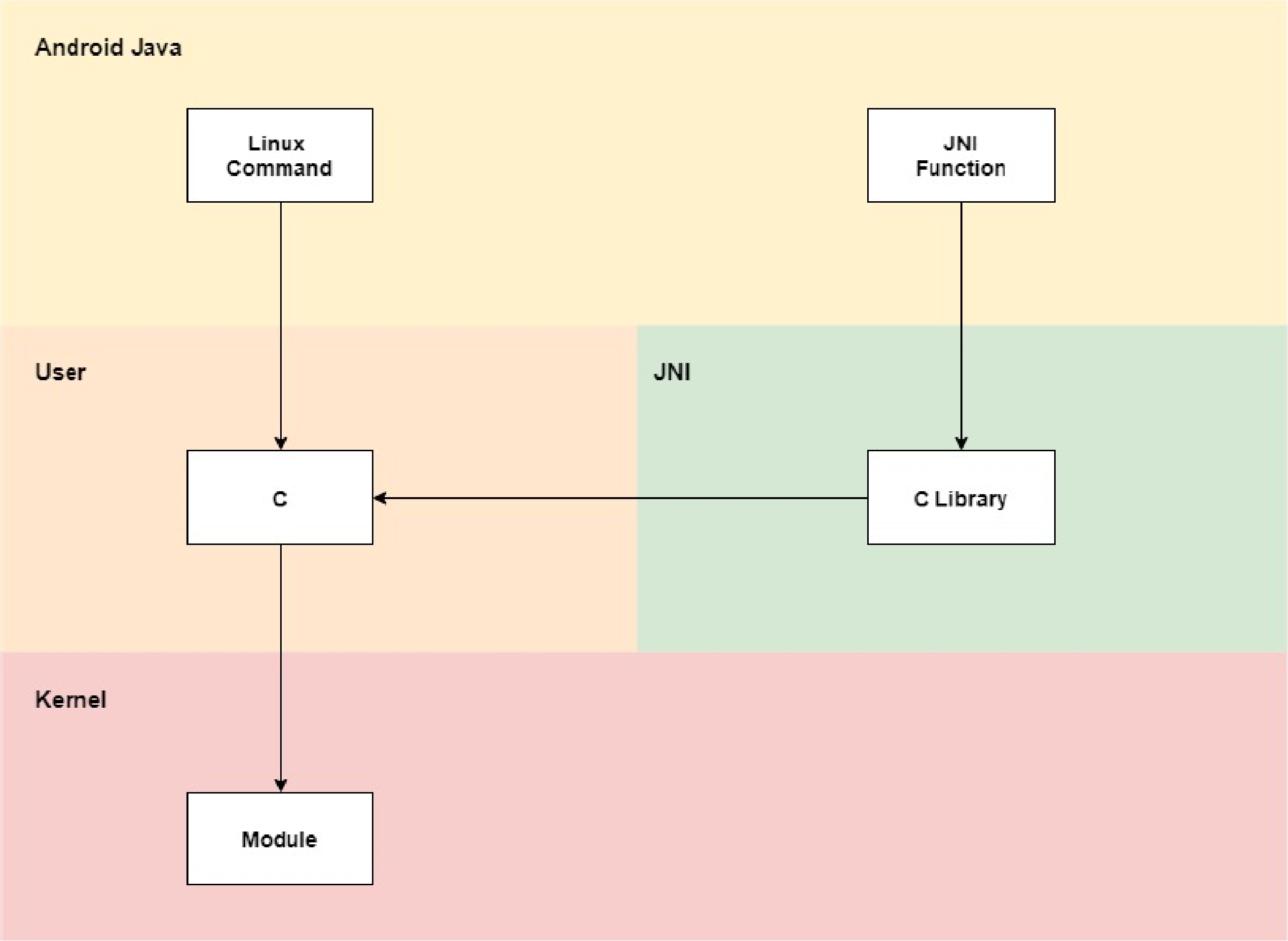
controldotfnd : FPGA DOT, FND 출력

controltextlcd : FPGA TextLCD 출력

deviceclear : 사용중인 모든 Device 초기화



Final Flowchart



Thank You