

SSE3052: Embedded Systems Practice

Jinkyu Jeong

jinkyu@skku.edu

Computer Systems Laboratory

Sungkyunkwan University

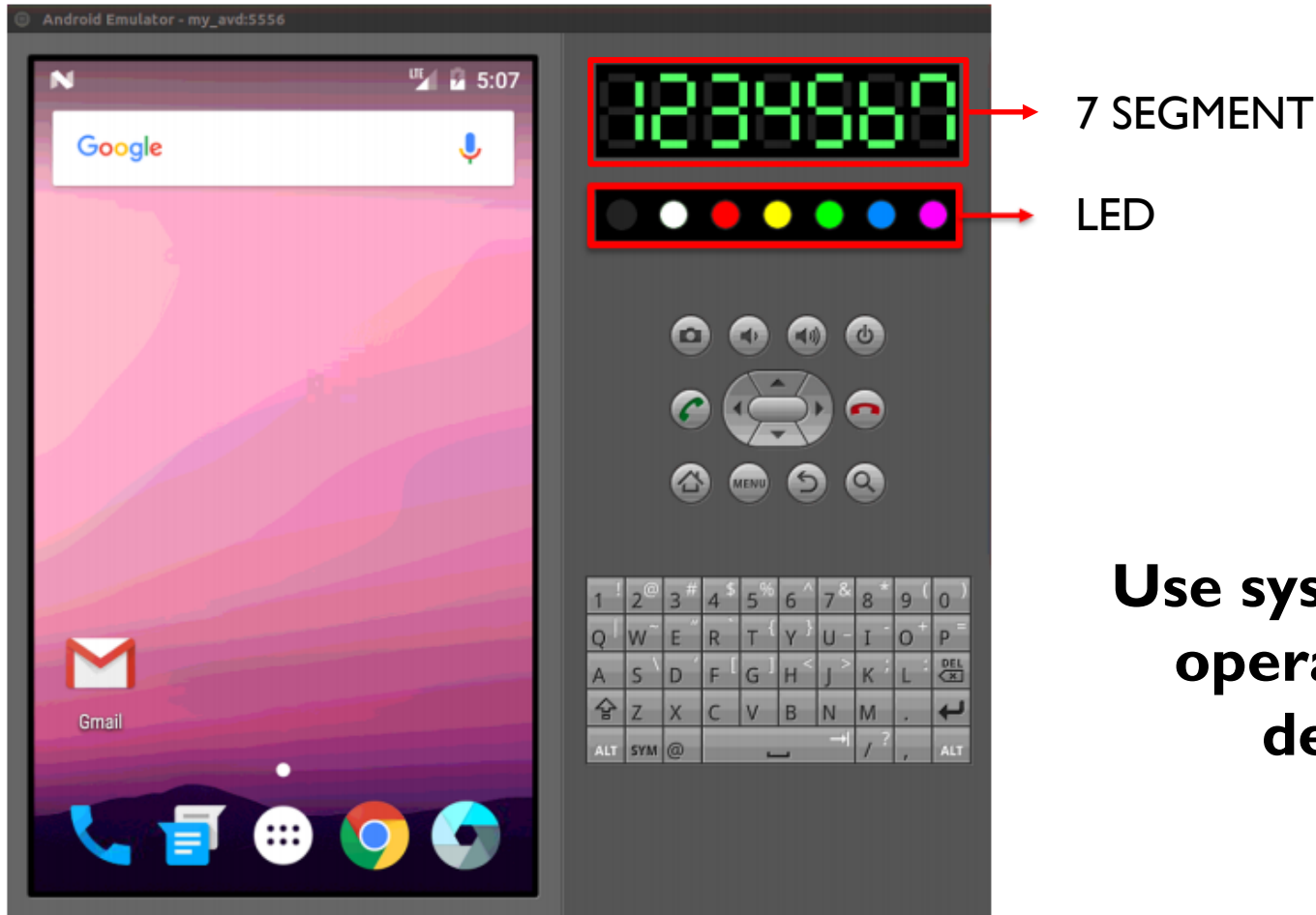
<http://csl.skku.edu>

Agenda

1. Attach 7 segment display to (virtual) device.
2. Write a device driver.
3. Write a system call & user program to manipulate the display.

Attach Virtual Device

- Attach new virtual devices (7 SEGMENT, LED)




**Use system call to
operate these
devices!**

Attach Virtual Device

- Download virtual_device.zip from i-Campus
- Unzip the downloaded file
 - goldfish_kernel_with_led.patch
 - Kernel patch (device drivers) for the virtual device
 - qemu_devices_with_led.patch
 - Emulator (QEMU) patch for the virtual device
 - skin_android-24_WVGA800_with_led.tgz
 - Skins of the virtual device

(Menu) Tools – AVD Manager

Virtual Device Configuration

 **Android Virtual Device (AVD)**
Android Studio

Verify Configuration

☒ Multi-Core CPU 4 ▼

Memory and Storage

RAM: 512 MB ▼

VM heap: 48 MB ▼

Internal Storage: 800 MB ▼

SD card: ☒ Studio-managed 100 MB ▼ ☐ External file ...

Device Frame ☒ Enable Device Frame

Custom skin definition WVGA800 ▼ ...

[How do I create a custom hardware skin?](#)

Keyboard ☒ Enable keyboard input

Hide Advanced Settings

Custom Device Frame

A collection of images and configuration data that indicates how to populate the window. Each skin can have several "layouts" (e.g. "landscape" and "portrait") corresponding to different orientation / physical configurations of the emulated device.

Previous Next Cancel Finish Help

Update Skin

- Untar "skin_android-24_WVGA800_with_led.tgz".
 - Note: The file contains image files of virtual devices' display.
- Untar & place all image files to relevant directory.
 - ~/Android/Sdk/platforms/android-24/skins/WVGA800/

External Emulator (QEMU)

- Download emulator source code .

– (We do not use the emulator in SDK.)

```
$mkdir ~/bin
```

```
$export PATH=~/bin:$PATH
```

```
$curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
```

```
$chmod a+x ~/bin/repo
```

```
$mkdir emu-2.2-release
```

```
$cd !$      -- this is equal to $cd emu-2.2-release
```

```
$repo init -u https://android.googlesource.com/platform/manifest -b emu-2.2-release
```

```
$repo sync
```

QEMU Patch for the new Devices

- Place “qemu_devices_with_led.patch” under emu-2.2-release/external/qemu

- Patch.

```
$cd ~/emu-2.2-release/external/qemu  
$patch -p1 < qemu_segment.patch
```

- Build.

```
$. /android-rebuild.sh
```


Run QEMU

- Run new emulator.

```
$export ANDROID_SDK_ROOT=/path/to/android-sdk
```

```
-ex) $export ANDROID_SDK_ROOT=~/.AndroidSdk
```

```
$/path/to/qemu/objs/emulator -avd [avd-name]
```

```
-ex) $~/emu-2.2-release/external/qemu/objs/emulator -avd my_avd
```

Device Driver

(You won't need to write a device driver.. yet..)

- Place “goldfish_kernel_with_led.patch” under goldfish.

- Patch

```
$patch -p1 < goldfish_kernel_with_led.patch
```

- Build (**modified! Read carefully**)

```
$export ARCH=x86_64
```

```
$export CROSS_COMPILE=~/.x86_64-linux-android-4.9/bin/x86_64-linux-android-
```

```
$make x86_64_emu_defconfig
```

```
$make menuconfig
```

=>**check the “device drivers/Misc devices/Android Goldfish 7 segment, LED”**

```
$make -j4
```

```
$cp arch/x86/boot/bzImage ~/Android/Sdk/system-images/android-24/googleApis/x86_64/kernel-qemu
```

Boot Error

- If your emulator shows below errors,

```
esp@esp:~$ ~/emu-2.2-release/external/qemu/objs/emulator -avd my_avd
ERROR: Invalid GPU mode 'software', use one of: on off host guest mesa swiftshader
```

```
esp@esp:~$ ~/emu-2.2-release/external/qemu/objs/emulator -avd my_avd
sh: 1: glxinfo: not found
sh: 1: glxinfo: not found
emulator: WARNING: Classic qemu does not support SMP. The hw.cpu.ncore option from your config file is ignored.
libGL error: unable to load driver: swrast_dri.so
libGL error: failed to load driver: swrast
X Error of failed request: BadValue (integer parameter out of range for operation)
  Major opcode of failed request: 154 (GLX)
  Minor opcode of failed request: 24 (X_GLXCreateNewContext)
  Value in failed request: 0x0
  Serial number of failed request: 21
  Current serial number in output stream: 22
QObject::~QObject: Timers cannot be stopped from another thread
```

- Modify the “~/android/avd/my_avd.avd/config.ini”
 - hw.gpu.mode=guest

goldfish/drivers/misc/goldfish_segment.c

Device Driver

goldfish_segment.c

```
static int goldfish_segment_probe(struct platform_device *pdev)
{
...
}

static int goldfish_segment_remove(struct platform_device *pdev)
{
...
}

static const struct of_device_id goldfish_segment_of_match[] = {
...
};
MODULE_DEVICE_TABLE(of, goldfish_segment_of_match);

static const struct acpi_device_id goldfish_segment_acpi_match[] = {
...
};
MODULE_DEVICE_TABLE(acpi, goldfish_segment_acpi_match);
```

goldfish_segment.c

```
static struct platform_driver goldfish_segment_device = {
    .probe      = goldfish_segment_probe,
    .remove     = goldfish_segment_remove,
    .driver     = {
        .name = "goldfish_segment",
        .of_match_table = goldfish_segment_of_match,
        .acpi_match_table = ACPI_PTR(goldfish_segment_acp
i_match),
    }
};

module_platform_driver(goldfish_segment_device);
```

Platform Driver

<linux/platform_device.h>

```
struct platform_driver {
    int (*probe)(struct platform_device *);
    int (*remove)(struct platform_device *);
    void (*shutdown)(struct platform_device *);
    int (*suspend)(struct platform_device *, pm_message_t state);
    int (*resume)(struct platform_device *);
    struct device_driver driver;
    const struct platform_device_id *id_table;
};
```

...

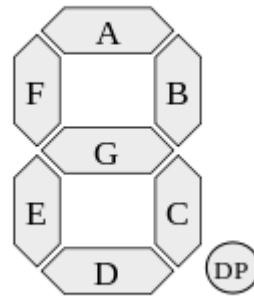
```
#define module_platform_driver(__platform_driver) \
    module_driver(__platform_driver, \
                  platform_driver_register, \
                  platform_driver_unregister)
```

goldfish_segment.c

```
#define GOLDFISH_SEGMENT_READ(data, addr) \  
    (readl(data->reg_base + addr))  
  
#define GOLDFISH_SEGMENT_WRITE(data, addr, x) \  
    (writel(x, data->reg_base + addr))  
  
...  
  
GOLDFISH_SEGMENT_WRITE( data, SEGMENT0, 0x24) ;    // "1"  
GOLDFISH_SEGMENT_WRITE( data, SEGMENT1, 0x5d) ;    // "2"  
GOLDFISH_SEGMENT_WRITE( data, SEGMENT2, 0x6d) ;    // "3"  
GOLDFISH_SEGMENT_WRITE( data, SEGMENT3, 0x2e) ;    // "4"  
GOLDFISH_SEGMENT_WRITE( data, SEGMENT4, 0x6b) ;    // "5"  
GOLDFISH_SEGMENT_WRITE( data, SEGMENT5, 0x7a) ;    // "6"  
GOLDFISH_SEGMENT_WRITE( data, SEGMENT6, 0x27) ;    // "7"
```


7 Segment Display

- 0x24? 0x5d? 0x6d?...



A = 0000 0001b
F = 0000 0010b
B = 0000 0100b
G = 0000 1000b
E = 0001 0000b
C = 0010 0000b
D = 0100 0000b

"1" = 0010 0100b = 0x24
"2" = 0101 1101b = 0x5d
"3" = 0110 1101b = 0x6d
...
...

Questions?

- If you have questions,
 - please use i-Campus (토론>수업 Q&A 토론) or email
 - minwoo.ahn@csi.skku.edu
 - bumsuk.kim@csi.skku.edu