Android kernal => HAL => framework => application 傳遞開發

RK3399 Android 7.0 增加

1. 首先自行創建一個device

增加

kernel/drivers/hello/hello.c

kernel/drivers/hello/hello.h

kernel/drivers/hello/Kconfig

kernel/drivers/hello/Makefile

Kconfig 可以先不管 因為 Makefile

obj-y += hello.o 我使用 y 強制編譯

接著修改

kernel/drivers/Kconfig

kernel/drivers/Makefile

Kconfig 的修改只是為了以後有用到config決定是否編譯預留可以先不管

source "drivers/hello/Kconfig"

Makefile 也是用強制編譯

obj-y += hello/

這樣子Kernel就可以先編譯出 hello.o

燒錄進系統後,可驗證

验证：

#cd dev

#ls

#cd proc

#ls

#cd sys/class/hello/hell0

#ls

#cat val

#echo '0' > val

#cat val

1. HAL層訪問驅動

增加

hardware/libhardware/include/hardware/hello.h

hardware/rockchip/hello/

修改

device/rockchip/common/device.mk

增加紅字部分

PRODUCT\_PACKAGES += \

sensors.$(TARGET\_BOARD\_HARDWARE) \

gralloc.$(TARGET\_BOARD\_HARDWARE) \

hwcomposer.$(TARGET\_BOARD\_HARDWARE) \

lights.$(TARGET\_BOARD\_PLATFORM) \

camera.$(TARGET\_BOARD\_HARDWARE) \

Camera \

libvpu \

libstagefrighthw \

libgralloc\_priv\_omx \

gpio.$(TARGET\_BOARD\_PLATFORM) \

hello.default \

akmd

編譯後會生成 hello.default.so

查看

#cd system/lib/hw/

#ls

#hello.default.so

1. 編寫 JNI 提供 Java 訪問HAL

增加

frameworks/base/services/core/jni/com\_android\_server\_HelloService.cpp

修改

增加紅字部分

frameworks/base/services/core/jni/Android.mk

LOCAL\_SRC\_FILES += \

$(LOCAL\_REL\_DIR)/com\_android\_server\_UsbHostManager.cpp \

$(LOCAL\_REL\_DIR)/com\_android\_server\_VibratorService.cpp \

$(LOCAL\_REL\_DIR)/com\_android\_server\_PersistentDataBlockService.cpp \

$(LOCAL\_REL\_DIR)/com\_android\_server\_GpioService.cpp \

$(LOCAL\_REL\_DIR)/com\_android\_server\_HelloService.cpp \

$(LOCAL\_REL\_DIR)/onload.cpp

frameworks/base/services/core/jni/onload.cpp

namespace android {

int register\_android\_server\_Watchdog(JNIEnv\* env);

int register\_android\_server\_HardwarePropertiesManagerService(JNIEnv\* env);

int register\_com\_android\_server\_rkdisplay\_RkDisplayModes(JNIEnv\* env);

int register\_android\_server\_GpioService(JNIEnv\* env);

int register\_android\_server\_HelloService(JNIEnv \*env);

};

register\_android\_server\_ConsumerIrService(env);

register\_android\_server\_BatteryStatsService(env);

register\_android\_server\_hdmi\_HdmiCecController(env);

register\_android\_server\_tv\_TvUinputBridge(env);

register\_android\_server\_tv\_TvInputHal(env);

register\_android\_server\_PersistentDataBlockService(env);

register\_android\_server\_Watchdog(env);

register\_android\_server\_HardwarePropertiesManagerService(env);

register\_com\_android\_server\_rkdisplay\_RkDisplayModes(env);

register\_android\_server\_GpioService(env);

register\_android\_server\_HelloService(env);

1. Framework 增加Service 提供application 控制

增加

framworks/base/core/java/android/os/HelloManager.java

frameworks/base/core/java/android/os/IHelloService.aidl

frameworks/base/services/core/java/com/android/server/HelloService.java

修改

紅字部分

frameworks/base/Android.mk

core/java/android/os/IDisplayDeviceManagementService.aidl \

core/java/android/os/IToolManager.aidl \

core/java/android/os/IGpioService.aidl \

core/java/android/os/IHelloService.aidl \

core/java/android/os/IRkDisplayDeviceManagementService.aidl \

frameworks/base/core/java/android/app/SystemServiceRegistry.java

import android.os.IHelloService;

import android.os.HelloManager;

registerService(Context.HELLO\_SERVICE, HelloManager.class,

new CachedServiceFetcher<HelloManager>() {

@Override

public HelloManager createService(ContextImpl ctx) {

IBinder b = ServiceManager.getService(Context.HELLO\_SERVICE);

IHelloService service = IHelloService.Stub.asInterface(b);

return new HelloManager(ctx, service);

}});

frameworks/base/core/java/android/content/Context.java

public static final String HELLO\_SERVICE = "helloservice";

frameworks/base/services/java/com/android/server/SystemServer.java

try {

HelloService helloservice = new HelloService(context);

ServiceManager.addService(Context.HELLO\_SERVICE, helloservice);

} catch (Throwable e) {

Slog.e(TAG, "Failure starting HelloService Service", e);

}

記得編譯 make update-api

1. 修改 /dev/hello

system/core/rootdir/init.rc

chmod 0777 /dev/hello

1. 增加 Seliunx 權限

修改

紅字部分

system/sepolicy/service.te

type hello\_service, system\_api\_service, system\_server\_service, service\_manager\_type;

system/sepolicy/service\_contexts

helloservice u:object\_r:hello\_service:s0

1. 編譯整個系統後

用附件 HelloDeviceDemo.apk

去控制即可看出變化