飛翔ソフトウェア

新人研修

Android結合編

|  |  |  |
| --- | --- | --- |
| 日付 | 変更内容 | 変更者 |
| 2022/04/13 | 新人研修向けに新規作成 | 庭 |
|  |  |  |

# 前提条件

本研修は、以下の研修が終了していることを前提とする。

飛翔ソフトウェア新人研修-4-Linuxドライバ開発編

飛翔ソフトウェア新人研修-4-Linuxドライバ開発基礎編

よって、仕様やヒントの提示に留まり、コードの詳細は指示しないこととする。

# 概要

本研修では飛翔ソフトウェア新人研修-4\_2-Linuxドライバ開発編で作成したドライバとAndroidを連携し、以下の動作が行えることを目標とする。

・AndroidアプリからTest Driverを介して拡張回路のLED点滅を行う

アプリはGoogle Playからダウンロードした一般アプリ

Light Flow Legacy :Led Control

(com.rageconsulting.android.lightflowlegacy\_83099\_apps.evozi.com.apk)

を使用する

・拡張回路のボタンを押すことでAndroidへ電源キーのイベントを送信する

## 構成図

全体の構成図を以下に示す。色付き部分が想定される改造箇所。

LED制御要求

application層

framework層

アプリ

LightsService

HAL層

lights HAL

PowerManager/SystemServer

kernel層

Test Driver

SysFs(LED Class)

WindowManager

native層

InputDispatcher

InputReader

EventHub

/dev/input/\*

LightsService JNI

キーイベント

# Driverの改造

## LED Classの登録

HALとのI/Fに使用する、sys/class/leds/配下にデバイスファイルを登録する

[改造内容]

・linux/leds.hをincludeする

・struct led\_classdevの型の領域を用意する

.nameにはsys/class/leds/配下に登録する名前を設定

例)”test”

.brightness\_setには輝度設定された際に実行する関数を設定

関数は以下のI/F

|  |
| --- |
| static void xxxx(struct led\_classdev \*class\_cdev, enum led\_brightness value) |

.brightnessには初期値としてLED\_OFFを設定

.flagsは0を設定

・led\_classdev\_registerでLED classデバイスを登録

・led\_classdev\_unregisterで登録したデバイスを削除

## Input Classの登録

/dev/inputXデバイスを登録する

[改造内容]

・linux/input.hをincludeする

・input\_allocate\_deviceを使用してstruct input\_dev \*領域を確保する

・name、physに適当に名前を入れる

・input\_set\_capabilityを使用して使用するイベントを登録

例)

info->key\_code = KEY\_POWER;

input\_set\_capability(info->input\_dev, EV\_KEY, info->key\_code);

・input\_register\_deviceを使用して登録

・input\_unregister\_deviceで登録したデバイスを削除

## 割り込み処理の変更

[改造内容]

・LED点灯スレッドの起床処理は削除

・代わりにinput\_report\_key・input\_syncを実行してキーイベントを送信する

例）

押したとき

input\_report\_key(info->input\_dev, info->key\_code, 1);

input\_sync(info->input\_dev);

放したとき

input\_report\_key(info->input\_dev, info->key\_code, 0);

input\_sync(info->input\_dev);

・wakelockはtest\_button\_irqtの開始と終了の範囲で行う

## LED点滅スレッドの変更

点滅要求があった際にwakelock、消灯要求があった際にwakeunlockするように変更

# 初期化スクリプト(init.rc)の改造

init.rcはAndroidのinitプロセスが実行する処理が記述されたスクリプトファイル。

system/core/rootdir/init.rc

が使用され、import文で他のrcファイルも読み込まれる。

フォーマットなどは以下を参照

<https://android.googlesource.com/platform/system/core/+/master/init/README.md>

device\qcom\common\rootdir

に適当なrcファイルを追加する。

HALからsysfsファイルにアクセスできるようにするため、rcファイルではon initのトリガでTest Driverのsysfsファイルの権限を変更する処理を記述する。

所有者：system

所有グループ：system

パーミッション：0664

例)

init.hisho.led.rc

|  |
| --- |
| on init  chown system system /sys/class/leds/test/brightness  chmod 0664 /sys/class/leds/test/brightness |

makeファイルを編集して追加したrcファイルをビルド対象とする

device\qcom\common\rootdir\Android.mk

例)

|  |
| --- |
| include $(CLEAR\_VARS)  LOCAL\_MODULE := init.hisho.led.rc  LOCAL\_MODULE\_TAGS := optional eng  LOCAL\_MODULE\_CLASS := ETC  LOCAL\_SRC\_FILES := etc/init.hisho.led.rc  LOCAL\_MODULE\_PATH := $(TARGET\_ROOT\_OUT)  include $(BUILD\_PREBUILT) |

device\qcom\common\ base.mk

例)

|  |
| --- |
| PRODUCT\_PACKAGES += init.hisho.led.rc |

既存の適当なrcファイルから追加したrcファイルをimportする

例) init.qcom.rc

|  |
| --- |
| import init.qcom.usb.rc  import init.target.rc  import init.qcom.factory.rc  import init.hisho.led.rc |

# HALの改造

hardware\qcom\display\liblight\lights.c

を改造してTest Driverのsysfsを操作する処理を追加する

注意点

HALはuserスペースのプログラムである

Android層のためログ出力はALOGIやALOGEマクロを使用し、logcatで取得する。