



2831P 修改、添加遥控按键

Revision History

Revision No.	Description	Updated By	Date
Version 1.0	Initial Release	张小钟	2017/8/22

Kernel 层:

1、当遥控器的红外信号发出后，首先接到信号的是 ir 的驱动程序，ir 设备对应的驱动程序 W:\2831p\kernel\linux\linux-4.4.3\drivers\rtk_kdriver\ir\ir_core.c 在这个文件的 examine_ir_avail 方法中检测收到的 ir 信号是否有效，这里会判断 ir 的协议和客户码是否支持，如果不是支持的协议和客户码就会直接将 ir 信号忽略。

2、ir 设备驱动将信号有效性检测完成后会交给 linux 输入子系统 input 设备处理，因此在驱动程序\kernel\linux\linux-3.10.0\arch\arm\bsp-realtek\mach-rtd294x\ir\ir_input.c 中添加如下代码：

为了避免新添加客户码的键值对原来的遥控器按键造成干扰新建如下连个结构体：

```
struct venus_key {
    u32 scancode;
    u32 keycode;
};

struct venus_key_table {
    struct venus_key *keys;
    int size;
};

static struct venus_key rtk_mk5_2_tv_keys[] = {
    {0x5f, KEY_POWER},
    {0x5b, KEY_MUTE},
    .....
}

static struct venus_key_table rtk_mk5_2_tv_key_table = {
    .keys = rtk_mk5_2_tv_keys,
    .size = ARRAY_SIZE(rtk_mk5_2_tv_keys),
};
```

这里 0x5b 就是遥控器通过 IR 传过来的 scancode

如果是新添加，添加了结构体后，还需要在初始化函数 vrn_irs_init 添加

```
#if defined(CONFIG_RTK_KDRV_TV001_IR)
    for (i = 0; i < rtk_mk5_2_tv_key_table.size; i++) {
        set_bit(rtk_mk5_2_tv_key_table.keys[i].keycode, data->input_dev->keybit);
    }
#endif
```

完成设备属性的初始化

scancode 是由 Linux 的 input 驱动框架定义的整数类型，也是 driver 收到遥控器打出来的值。Scancode 通过 venus_ir_scancode_to_keycode 函数转换成 keycode，这个 keycode 需要在 W:\2831p\kernel\linux\linux-4.4.3\include\linux\input-event-codes.h 提前定义好。

```
#define KEY_MUTE 113
```

到这里就实现了扫描码到 **keycode** 的转化，把按键的信号传递到了 **linux** 层，此时如果一切 **ok**，在串口中可以看到以下打印，

```
[ 2068.411718] (0)-0000 [IR] [DBG] RTK_MK5_2 Venus IR: report key 0x5b 113
```

3、Debug,

可参 W:\2831p\kernel\linux\linux-4.4.3\drivers\rtk_kdriver\rtk_verus_watch.c

```
#define DEV_DEBUG
```

```
W:\2831p\kernel\linux\linux-4.4.3\drivers\rtk_kdriver\ir\irrc_core.c
```

```
ir_decode_dbg_en = ture
```

```
W:\2831p\kernel\linux\linux-4.4.3\drivers\rtk_kdriver\ir\irrc_core.c
```

```
ir_decode_dbg_en = ture
```

另外需要 `echo 7 > /proc/sys/kernel/printk`

Android 层:

4、**kl** 文件是一个映射文件，是标准 **linux** 与 **anroid** 的键值映射文件，**kl** 文件可以有很多个，

```
key 113 VOLUME_MUTE
```

像 **113** 是在 **input-event-codes.h** 中定义的，**kernel** 报上来的键值就是 **113**，而 **kl** 文件是对这个键值的映射，在 **android** 系统中代表的是 **VOLUME_MUTE**。

如果需要添加 **kl** 文件，必须与 **input** 的 **device** 设备的 **name** 一致，否则 **EventHub** 在加载设备时因找不到对应的 **kl** 而加载默认的 **qwerty.kl**，导致键值转换错误。新增加 **kl** 文件，需要在 **kernel\android\nougat\device\realtek\RealtekATV\device-common.mk** 中加上你的 **kl** 文件

```
PRODUCT_COPY_FILES += \
```

```
device/realtek/common/venus_IR_input.kl:system/usr/keylayout/venus_IR_input.kl \
```

```
device/realtek/common/venus_IR_input.kcm:system/usr/keychars/venus_IR_input.kcm
```

这样才会使 **kl** 文件被打包进 **/system/usr/keylayout/**

还可以手动 **copy** 进板子。

```
/bin/busybox mount -o remount,rw /system
```

```
cp venus_IR_input.kl /system/usr/keylayout
```

Android 中添加按键

5、定义按键对应的 **keylabel**

W:\2831p\kernel\android\nougat\frameworks\native\include\input\ InputEventLabels.h

在数组 **InputEventLabel KEYCODES[]** 中添加

```
DEFINE_KEYCODE(MUTE),
```

定义 **keycode**

W:\2831p\kernel\android\nougat\frameworks\native\include\android\ keycodes.h

```
AKEYCODE_MUTE = 91,
```

此处 **keycodes.h** 里定义的值即是 上面 **key label** 定义在 **KEYCODES** 数组中的位置(**index**)

6、**JAVA** 定义

W:\2831p\kernel\android\nougat\frameworks\base\core\java\android\view\KeyEvent.java

定义键值

```
public static final int KEYCODE_VOLUME_MUTE = 164;
```

如果 KEYCODE_TV_INPUT 是最后一个，还需修改 LAST_KEYCODE

```
private static final int LAST_KEYCODE = 新按键;
```

7、资源文件（attrs.xml）添加 keycode

W:\2831p\kernel\android\nougat\frameworks\base\core\res\res\values\attrs.xml

在<attr name="keycode"> </attr> 里面

```
<enum name="KEYCODE_VOLUME_MUTE" value="164" />
```

修改完后，需要在根目录下先执行 make update-api 更新 current.xml

(W:\2831p\kernel\android\nougat\frameworks\base\api 目录下)文件，再进行编译

8、可通过 input event 来模拟按键。

Thanks