红外遥控器配置操作

目录

每个客户要求的红外遥控器应该都是不同的，因此就需要根据每个客户对 android 中红外遥控器的按键值进行修改适配。这里以 amlogic s912 平台 q201 为例，说明 android 中遥控器按键值的修改与添加。

1 红外遥控器配置

红外遥控器的配置是通过一个 service 来启动的，该 service 位于 device/amlogic/common/products/mbox/init.amlogic.rc 中，如下：

```
service remotecfg /system/bin/remotecfg /system/etc/remote.conf
    class main
    oneshot
    seclabel u:r:remotecfg:s0
```

启动机顶盒，在 kernel 启动完成之后就会创建用户空间的第一个进程——init 进程，实现由内核空间到用户空间的转变。init 进程启动后会读取各种 init.*.rc 配置文件，通过 fork 系统调用启动 init.*.rc 中配置的各种 service 进程。红外遥控器进程也就是在这个阶段通过 remotecfg 这个 service 来启动的，其具体的配置命令是 remotecfg，具体路径是 /system/bin/remotecfg；参数是红外配置文件 remote.conf，具体路径是 /system/etc/remote.conf。

remotecfg 的源码是 vendor/amlogic/frameworks/services/remoteconf/ 目录下的 irremote.c，编译出来的可执行文件位于 /system/bin 目录中，它将 remote.conf 中的配置通过 ioctl 赋值到 ir 驱动的变量中，从而实现按键值由用户空间到内核空间的映射，这样定制的遥控器配置才会生效。

在需要验证新的配置是否正确的时候可以通过如下命令来进行配置，并使新的配置生效，无须重启

remotecfg path/of/remote.conf

或者写成完整命令形式，如下

/system/bin/remotecfg path/of/remote.conf

注：s812 及之前的平台，remotecfg 的源码位于 external/remoteconf 目录下

2 红外驱动

amlogic android 平台的红外遥控器驱动程序源代码位于 common/drivers/amlogic/input/remote 目录，该驱动依照红外驱动配置文件 remote.conf 将红外按键值 ircode 映射为 Linux 标准键盘扫描码 scancode，该过程将红外遥控器的按键事件转换为 Linux 的标准 input keyevent。除非要改驱动，否则知道该驱动的作用即可。

红外驱动配置文件 /system/etc/remote.conf

Linux 标准按键扫描码在源码的位置 common/include/uapi/linux/input.h

3 android 层按键转换

在 android 系统中,windows manager 从红外驱动中读取 keyevent，再通过 /system/usr/keylayout/Vendor_0001_Product_0001.kl 文件（或者类似的其他 kl 文件，如 Generic.kl）把 Linux 标准 input 设备的 scancode 映射为 android api 按键 KEYCODE 字符串。最终 scancode 和 keycode 被 windows manager 发送到应用程序，被其 focus view 消化处理。

4 配置/映射文件说明

4.1 ircode —> scancode 的映射

配置文件为 remote.conf，一般修改该文件以适配不同客户的遥控器，其映射部分基本格式及说明如下图所示：

```
custom_begin    # 一款红外遥控器按键配置
    factory_infcode = 0     # 适配的红外遥控器序号，从 0 开始，这里表示第一个，若有多个要依次递增
    factory_code    = 0xf7080001    # factory code，逻辑上每款不同的遥控器都有一个对应的 factory code
                                    # 格式为 custom_code(16 bit) + index_code(16 bit)
                                    # 这里是：0xf7080001 = 0xf708(客户码) + 0001(可填任16进制数，一般填0001)
    mouse_begin # 鼠标方向映射表
        0 0x03      # 0 是鼠标的 left 方向，对应红外遥控器的 0x03
        1 0x02      # right
        2 0x00      # up
        3 0x01      # down
    mouse_end   # 鼠标方向映射表结束
    key_begin   # 按键映射表
        0x0A 116    # power
        0x1F 28     # enter     0x1F 为遥控器 enter 键的物理码，28 是 Linux 标准键盘 enter 的 scancode
        0x1C 158    # back      可多个物理按键对应一个 scancode
        0x00 103    # up
        0x01 108    # down
        0x03 105    # left
        0x02 106    # right
    key_end     # 按键映射表结束
    repeat_key_begin # 连发按键映射表
        0x0A 116    # power
        0x1F 28     # enter
        0x1C 158    # back
        0x00 103    # up
        0x01 108    # down
        0x03 105    # left
        0x02 106    # right
    repeat_key_end  # 连发按键映射表结束
custom_end  # 该款红外遥控器按键配置结束
```

以**电源、确定、返回、上、下、左、右** 7 个按键为例，红外按键值 ircode 与 Linux 标准键盘扫描码 scancode 之间的映射关系如下所示：

```
0x0A 116    # power
0x1F 28     # enter
0x1C 158    # back
0x00 103    # up
0x01 108    # down
0x03 105    # left
0x02 106    # right
```

其中左边红框内的 0x0A 0x1F 0x1C 0x00 0x01 0x03 0x02 分别为红外遥控器中的**电源、确定、返回、上、下、左、右** 7 个按键的物理码，其对应的 Linux

标准键盘扫描码分别为右边红框内的 116 28 158 103 108 105 106。而 116 28 158 103 108 105 106 在 Linux 标准键盘扫描码中分别表示 **KEY_POWER**、**KEY_ENTER**、**KEY_BACK**、**KEY_UP**、**KEY_DOWN**、**KEY_LEFT**、**KEY_RIGHT**，其定义在 common/include/uapi/linux/input.h

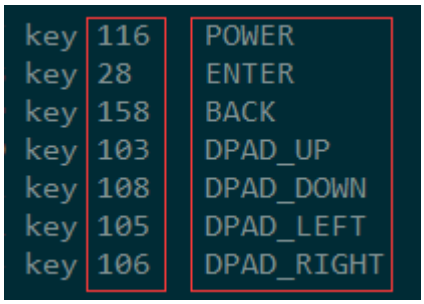
至此，我们把**电源**、**确定**、**返回**、**上**、**下**、**左**、**右** 7 个红外遥控器的物理码转成了 Linux 的标准键盘扫描码。

4.2 Linux 标准按键 scancode —> KEYCODE 字符串的映射

映射文件为：Vendor_0001_Product_0001.kl

其映射格式如 key 28 ENTER

同样，以**电源**、**确定**、**返回**、**上**、**下**、**左**、**右** 7 个 Linux 标准键盘扫描码为例，Linux 标准键盘扫描码与 android api 按键的 KEYCODE 字符串之间的映射关系如下所示：



其中左边红框内的 116、28、158、103、108、105、108 就是第 2 步中红外遥控器的**电源**、**确定**、**返回**、**上**、**下**、**左**、**右** 7 个按键的物理码经过 remote.conf 转换之后的 Linux 标准键盘扫描码。而右边红框内的 **POWER**、**ENTER**、**BACK**、**DPAD_UP**、**DPAD_DOWN**、**DPAD_LEFT**、**DPAD_RIGHT** 分别是 android api 的**电源**、**确定**、**返回**、**上**、**下**、**左**、**右** 7 个按键的 KEYCODE 字符串，也就是 KEYCODE_POWER、KEYCODE_ENTER 、 KEYCODE_BACK 、 KEYCODE_DPAD_UP 、 KEYCODE_DPAD_DOWN 、KEYCODE_DPAD_LEFT、KEYCODE_DPAD_RIGHT 字符串去掉 KEYCODE 之后的部分。

至此，红外遥控器的按键完成了从 物理码 —> Linux 标准键盘扫描码 —> android api 的按键 KEYCODE 的转换。

在实际应用中，上图 .kl 文件中的右边红框内的值是根据左边 Linux 标准按键扫描码的意义匹配的。

4.3 KEYCODE 字符串 —> android ui 键值

KEYCODE 字符串与 android UI 键值的关系定于在这个文件中

development/ndk/platforms/android-21/include/android/keycodes.h

截取其中部分如下所示：

```
enum {
    AKEYCODE_UNKNOWN         = 0,
    AKEYCODE_SOFT_LEFT       = 1,
    AKEYCODE_SOFT_RIGHT      = 2,
    AKEYCODE_HOME            = 3,
    AKEYCODE_BACK            = 4,
    AKEYCODE_CALL            = 5,
    AKEYCODE_ENDCALL         = 6,
    AKEYCODE_0               = 7,
    AKEYCODE_1               = 8,
    AKEYCODE_2               = 9,
    AKEYCODE_3               = 10,
    AKEYCODE_4               = 11,
    AKEYCODE_5               = 12,
    AKEYCODE_6               = 13,
    AKEYCODE_7               = 14,
    AKEYCODE_8               = 15,
    AKEYCODE_9               = 16,
};
```

4.4 Linux 标准键盘 scancode

Linux 标准键盘扫描码的定义在 common/include/uapi/linux/input.h 截取部分如下所示：

```
#define KEY_HOME        102
#define KEY_UP          103
#define KEY_PAGEUP      104
#define KEY_LEFT        105
#define KEY_RIGHT       106
#define KEY_END         107
#define KEY_DOWN        108
```