

密级状态：绝密（ ） 秘密（ ） 内部资料（ ☒ ） 公开（ ）

遥控器添加方法

文件状态： <input type="checkbox"/> 草稿 <input type="checkbox"/> 正在修改 <input checked="" type="checkbox"/> 正式发布	文件标识	遥控器添加方法
	文件编号	
	主要内容	
	当前版本	V1.0.0
	作 者	钟 斌
	完成日期	
	审 核	
	审核日期	

历史版本

时间	版本	内容	作者	备注
2016/9/27	V1.0.0	遥控器添加方法	钟斌	

Top-Tech

目录

一、	修改代码.....	4
1、	添加遥控器需要修改的文件.....	4
2、	代码修改方法.....	4
1)	kernel 修改.....	5
2)	android 修改.....	6
3)	supernova 修改	8
二、	遥控器调试.....	9
1、	非 POWER 键调试.....	9
2、	POWER 键调试.....	9
3、	关键节点调试.....	10

一、修改代码

1、添加遥控器需要修改的文件

- 1) msd6a638_asean\vendor\mstar\kernel\mstar2\drv\ir\IR_MSTAR_DTV.h
- 2) msd6a638_asean\vendor\mstar\kernel\mstar2\drv\ir\mdrv_ir.c
- 3) msd6a638_asean\vendor\mstar\kernel\3.10.40\drivers\media\rc\keymaps\toptech_config\config.h
- 4) msd6a638_asean\vendor\mstar\kernel\3.10.40\drivers\media\rc\keymaps\toptech_config\Toptech_At061.h
- 5) msd6a638_asean\device\mstar\arbutus\preinstall\keylayout\Vendor_3697_Product_0001.kl
- 6) msd6a638_asean\frameworks\base\core\java\android\view\KeyEvent.java
- 7) W:\szcode\638\msd638_asean\frameworks\native\include\input\InputEventLabels.h
- 8) msd6a638_asean\frameworks\base\core\res\res\values\attrs.xml
- 9) msd6a638_asean\frameworks\native\include\android\Keycodes.h

添加完成上面这些文件之后，框架层就已经 OK 了，就可以到自己的应用中调用你所加的按键并实现功能了。

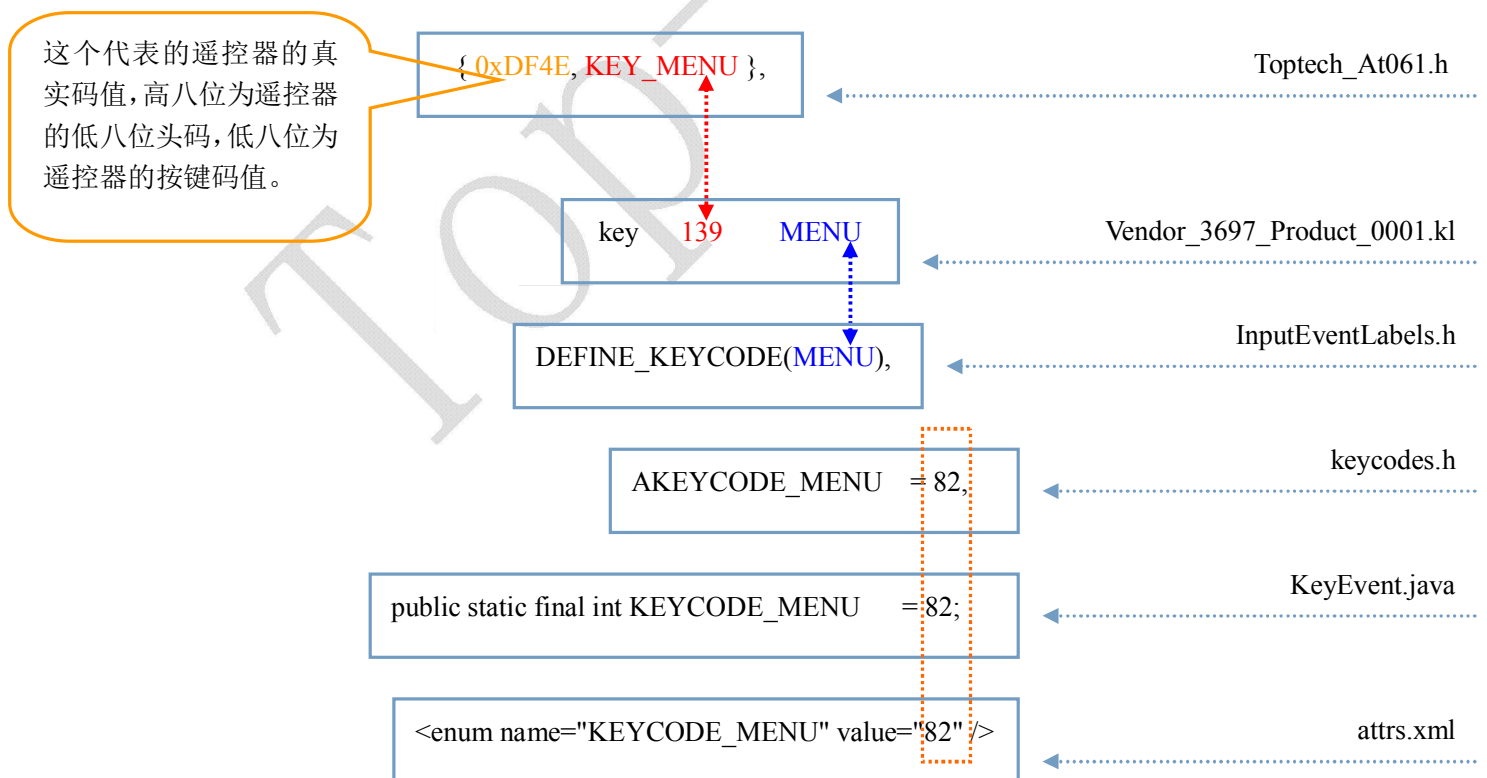
说明：1、第 2) 个文件只有在要改遥控器解码的时候才要修改。

2、若只是配置客户通用遥控器（即没有新增目前系统中没有的按键）时，只需修改第 1)、3)、4) 三个文件。

3、若要新增按键的话，上面除第 2) 的所有文件都要修改。

2、代码修改方法

按键从 kernel 到 android 传递的映射关系图如下（以 MENU 键为例）：



从这张图可以看出，各个部分的写法都是有严格的规定的，不能随便乱写，比如说：

A: .kl 文件中第三个参数，是把 keycodes.h 的中的 AKEYCODE_ 去掉的后半部分；

B: keycodes.h、keyevent.java 与 attrs.xml 三个文件中，同一个按键的值都是要保持一致的；

C: 若要新增按键时，宏的定义都是依次递增的，比如说前一个键定义为 81，那么再新一个按键时，要从 82 开始。

1) kernel 修改

步骤一：在 msd6a638_asean\vendor\mstar\kernel\mstar2\drv\ir\IR_MSTAR_DTV.h 添加遥控器头码，如下：

```
#define TOPTECH_AT061_J110_R41 7
#define TOPTECH_AT061_M043_Y096T 8

#define IR_INDEX TOPTECH_AT061_M043_Y096T
.....
#elif(IR_INDEX == TOPTECH_AT061_M043_Y096T)
#define IR2_HEADER_CODE0 0x40
#define IR2_HEADER_CODE1 0xBF
#else
#define IR2_HEADER_CODE0 0xFF
#define IR2_HEADER_CODE1 0xFF
#endif
```

我们在配置客户遥控器时，修改的头码都是 IR2_HEADER_CODE0 和 IR2_HEADER_CODE1，不要去修改 IR1_HEADER_CODE0 和 IR1_HEADER_CODE1，因为 IR1 是我们的公版遥控器，不要动它，否则会造成一些其它的问题。这里定义客户 IR 的宏也是有要求的，规定为 TOPTECH_公版 IR 型号_客户代号_客户 IR 型号，这个名字与 SN 下的客户脚本中的 ir_name 定义是一一对应的。

步骤二：在 msd6a638_asean\vendor\mstar\kernel\3.10.40\drivers\media\rc\keymaps\toptech_config\ 目录下添加客户遥控器文件，比如：Toptech_At061_M043_Y096T.h，可以从一个公版遥控器文件复制出一个来进行修改，要特别注意，在客户遥控器文件中，都必须要有公版遥控器码值表及工厂生产遥控器码值表，所以，一般来说，一个客户遥控器码值都有三个码值表，如下：

```
static struct rc_map_table mstar_tv[] = {
    //公版遥控器开始，主要用于工程师调试用
    //mstar STR key
    { 0x00FE, KEY_POWER2 }, //这个键必须保留，否则在 STR 模式下，关机无法开机
    { 0xDF0B, KEY_POWER },
    { 0xDF14, KEY_MUTE },
    .....
    //工厂生产遥控器开始，主要用于工厂生产测试用
    { 0xDF3C, KEY_FACTORY_AGING }, //aging
    { 0xDF3B, KEY_FACTORY_RESET }, //reset
```

```

.....
//客户遥控器
//M043 Y096T 40BF
{ 0xBF0B, KEY_POWER },
{ 0xBF0F, KEY_KP1 }, // TV_INPUTSOURCE
.....
};

```

这里举个例子，对一组值进行解释一下，如：{ 0xDF0B, KEY_POWER },
0xDF 是遥控器头码的低八位，比如我们公版遥控器的头是 0x20DF，这里就写 DF;
0B 是遥控器的真实键值；

KEY_POWER 是定义在\vendor\mstar\kernel\3.10.40\include\ui\linux\input.h 中，若要新添加按键的话，都要在这里添加，并强行要求添加到我们添加定义部分后面，并依次递增增加。KEY_POWER 值会被传递到安卓那边，可以在串口中输入 getevent 命令进行查看 kernel 传到 android 时的按键值是多少，如下：

```

/dev/input/event1: 0004 0004 0000df4e
/dev/input/event1: 0001 008b 00000001
/dev/input/event1: 0000 0000 00000000
/dev/input/event1: 0001 008b 00000000
/dev/input/event1: 0000 0000 00000000

```

步骤三：修改 msd6a638_asean\vendor\mstar\kernel\3.10.40\drivers\media\rc\keymaps\toptech_config\config.h 文件去调用客户文件。如下：

```

#define TOPTECH_AT061_J110_R41 12
#define TOPTECH_AT061_M043_Y096T 13

#define IR_INDEX TOPTECH_AT061_M043_Y096T
.....
#elif (IR_INDEX == TOPTECH_AT061_M043_Y096T)
#include "Toptech_At061_M043_Y096T.h"
#endif

```

这里的宏 TOPTECH_AT061_M043_Y096T 必须与 IR_MSTAR_DTV.h 中定义的宏名字一致，主要是为了方便查找及后期维护。

对于普通的添加客户遥控器的话，到这里 kernel 部分的修改就说完了，若要修改解码的话，涉及的东西较多，后面再讲。

2) android 修改

配置按键时，在 keyevent.java 中都会有说明要修改哪些文件，如下：

```

00827: // NOTE: If you add a new keycode here you must also add it to:
00828: // isSystem()
00829: // isWakeKey()
00830: // frameworks/native/include/android/keycodes.h
00831: // frameworks/native/include/input/InputEventLabels.h
00832: // frameworks/base/core/res/res/values/attrs.xml
00833: // emulator?
00834: // LAST_KEYCODE

```

一般来说，只有在新增加了按键才需要修改此部分，下面以新增加一个按键

(KEYCODE_TV_KEYMOUSE_MODE_SWITCH)为例来说明各处的修改方法及注意事项。

A: 在 KeyEvent.java 添加按键定义, 主要是用于应用中调用。

```
.....  
public      static      final      int      KEYCODE_TV_KEYMOUSE_MODE_SWITCH  
=TOPTECH_IRKEY_BEGIN+54;  
  
private static final int LAST_KEYCODE = KEYCODE_TV_KEYMOUSE_MODE_SWITCH;
```

这里 TOPTECH_IRKEY_BEGIN 的值是 290, 所以 KEYCODE_TV_KEYMOUSE_MODE_SWITCH 的值就是 344 了。添加按键时, 都必须依次这样添加下来, 方便后续方案移植, 并且要同时修改一下 LAST_KEYCODE 的值。说一下为什么要定义一个这样的宏 (TOPTECH_IRKEY_BEGIN)? 主要为了兼容不同的方案, 因为 mstar release 出来的公版 code, 不同的方案可能最后的一个按键不一致, 为了方便我们开发时做移植, 就添加了这么一个宏, 不用每个定义都要去写后面数值。

B: 修改 attrs.xml 文件。

```
.....  
<enum name="KEYCODE_TV_KEYMOUSE_DOWN" value="343" />  
<enum name="KEYCODE_TV_KEYMOUSE_MODE_SWITCH" value="344" />  
<enum name="KEY_PIC_SOUND" value="555" />  
.....
```

C: 修改 Keycodes.h 文件。

```
.....  
AKEYCODE_TV_KEYMOUSE_DOWN=ATOPTECH_IRKEY_BEGIN+53,  
KEYCODE_TV_KEYMOUSE_MODE_SWITCH=ATOPTECH_IRKEY_BEGIN+54,  
//toptech ir end
```

从安卓传递按键到 supernova 时用的到, 比 CEC 传键。

D: 修改 InputEventLabels.h 文件。

```
.....  
#DEFINE_KEYCODE(KEYCODE_TV_KEYMOUSE_MODE_SWITCH),
```

这个文件的主要作用是跟 .kl 文件匹配。

E: 修改 Vendor_3697_Product_0001.kl 文件。

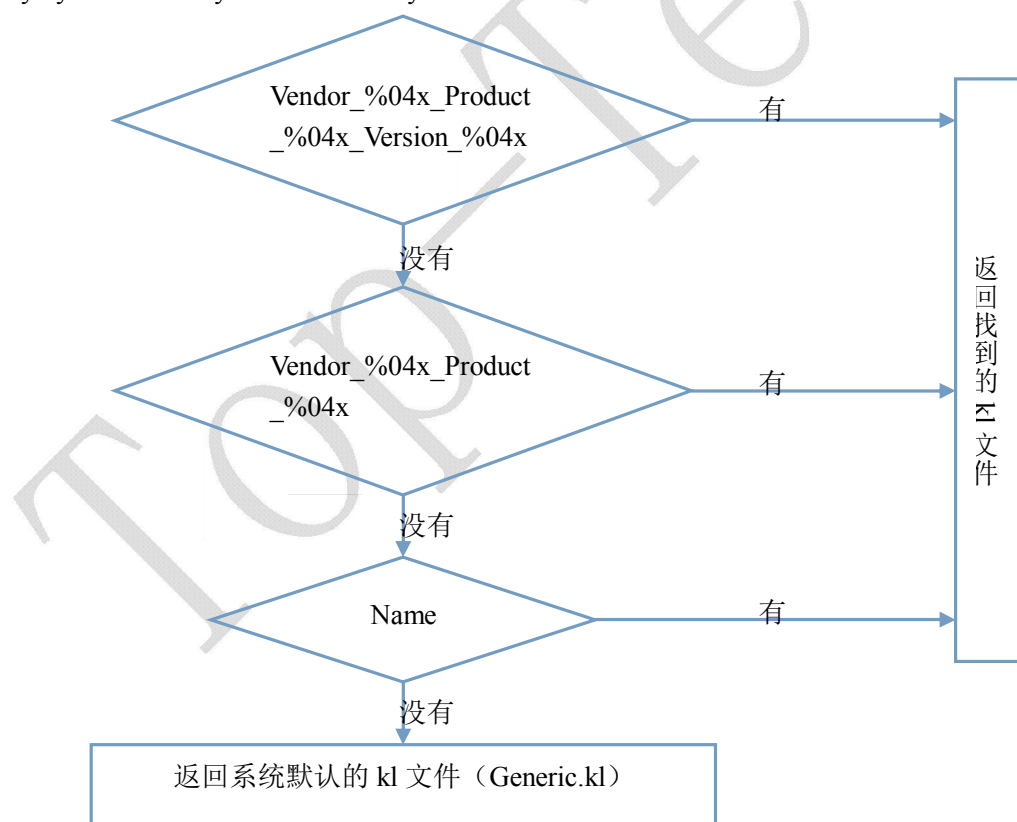
```
.....  
key 352      CUS_FAC_ADC  
key 353      CUS_FAC_OVER_SET  
key 518      TV_KEYMOUSE_MODE_SWITCH  
.....
```

这个文件是从 kernel 到 android 的唯一匹配文件, 这里的 518 值是 kernel 上报上来的值。为什么我们的遥控器是使用这个文件呢? 我们可以去看一下我们的遥控器驱动文件 (mdrv_ir.c), 可以查到有如下部分代码:

```
int MDrv_IR_Input_Init(void) {  
.....
```

```
#else
char *map_name = RC_MAP_MSTAR_TV;
char *input_name = "MStar Smart TV IR Receiver";
__u16 vendor_id = 0x3697UL;
#endif
.....
dev->input_id.vendor = vendor_id;
dev->input_id.product = 0x0001UL;
dev->input_id.version = 1;
.....
return 0;
}
```

我们任何一个设备在 kernel 中注册时，都会将一些重要的参数（比如设备名、VID、PID）等存到设备节点中，这些参数可以上层使用加以区分的。然而，在系统启动的时候，eventHub.cpp 的相关函数就会去通过设备节点去匹配这些 kl 文件，匹配顺序规则如下：匹配的路径有：/system/usr/idc/、/system/usr/keylayout/、/system/usr/keychars/，如果系统中没有这些路径的话，会去查找下面的这些路径：/data/system/devices/idc/、/data/system/devices/keylayout/、/data/system/devices/keychars/。



3) supernova 修改

这部分修改主要处理 POWER 键开机的，如：

msd638_asean\vendor\mstar\supernova\projects\customer_setting\customer\sample\ir\ 目录下的 ir_config.ini 文件。

ir_config.ini 内容如下:

```
[ir2_config]    //客户遥控器头码及 POWER 键配置
header_code0 = 0xff;
header_code1 = 0xff;
power_key = 0xff;

[ir1_config]    //公版遥控器配置, 禁止修改
header_code0 = 0x20;
header_code1 = 0xdf;
power_key = 0x0b;
```

虽然配置客户遥控器优化的很不错了, 修改的地方非常少, 但还是讲一下其中的原理。在上电进入待机时, POWER 键是通过 MBOOT 传到 PM 程序中; 按 POWER 键关机时, POWER 键是通过 Supernova 传到 PM 的, 所以有时候有可能会遇到 POWER 键在待机状态下时, 无法开机的问题, 这时候就要区分是从哪个状态下进入到的 PM, 然后去查相应部分的代码。

二、遥控器调试

1、非 POWER 键调试

在配置客户遥控器, 要新添加按键的情况还是比较少, 一般都是丝印或遥控器码值与公版不一致时, 才要配置的。所以在 kernel 中添加好码值表之后, 可以直接编译 kernel, 把编译好的 rc_mstar_tv.ko (这个文件路径是要看其相应的.c 文件在位置, 编译 kernel OK 后, 会在同一路径下生成这个文件) 直接通过串口命令把它拷贝到 system/lib/modules/目录下, 执行步骤如下:

```
su
mount -o remount,rw /system
cp /mnt/usb/sda1/rc_mstar_tv.ko /system/lib/modules/    //前半部分 U 盘路径依实际情况变动
chmod 644 /system/lib/modules/rc_mstar_tv.ko
reboot
```

重启之后就可以去测试遥控器功能是否正常了。

注: 在不同的方案中 rc_mstar_tv.ko 可能在系统中的名字有些差异, 可以先进入到目录去看看当前使用的是什麼名字, 然后再做相应的调整, 一般就两个名字, 分别为 rc_mstar_tv.ko 和 rc_keymap_tv.ko。

2、POWER 键调试

在配置好 rc_mstar_tv.c 之后, POWER 键是可以关机的, 但是有可能不能开机, 开机分为两种状态: 1、上电进入待机之后, 再按 POWER 键进行开机;

2、按 POWER 关机进入待机之后, 再按 POWER 键进行开机;

在 338 与 638 方案中, 做了很大的优化, 都是调用同一个文件。所以当发现能关机不能开机时, 要考虑这个文件配置对了没有。调试时, 可以将 ir_config.ini 直接拷贝到 tvconfig 目录下, 并注意好文件权限, 然后进入到 mboot 命令行下输入 set db_table 0;save;reset, 就可以去确认上电待机状态下的开机功能是否正常了。

3、关键节点调试

主要是在遇到配置好遥控器之后，遇到有部分按键无作用或整个遥控器都无作用时，为了提高工作效率，需要去对几个关键的节点进行分析。

1) getevent 信息分析

在串口下，在 root 权限时，可以输入 getevent，可以看到如下的信息：

```
/dev/input/event1: 0004 0004 0000df4e
/dev/input/event1: 0001 008b 00000001
/dev/input/event1: 0000 0000 00000000
/dev/input/event1: 0001 008b 00000000
/dev/input/event1: 0000 0000 00000000
```

0004 表示类型设备类型

Df4e 表示遥控器的真实码值部分，其中 df 是头码的低 8 位，4e 是键值

008b 表示 kernel 上报给 android 的值

00000001 表示按键按下

0000 0000 00000000 这组表示同步

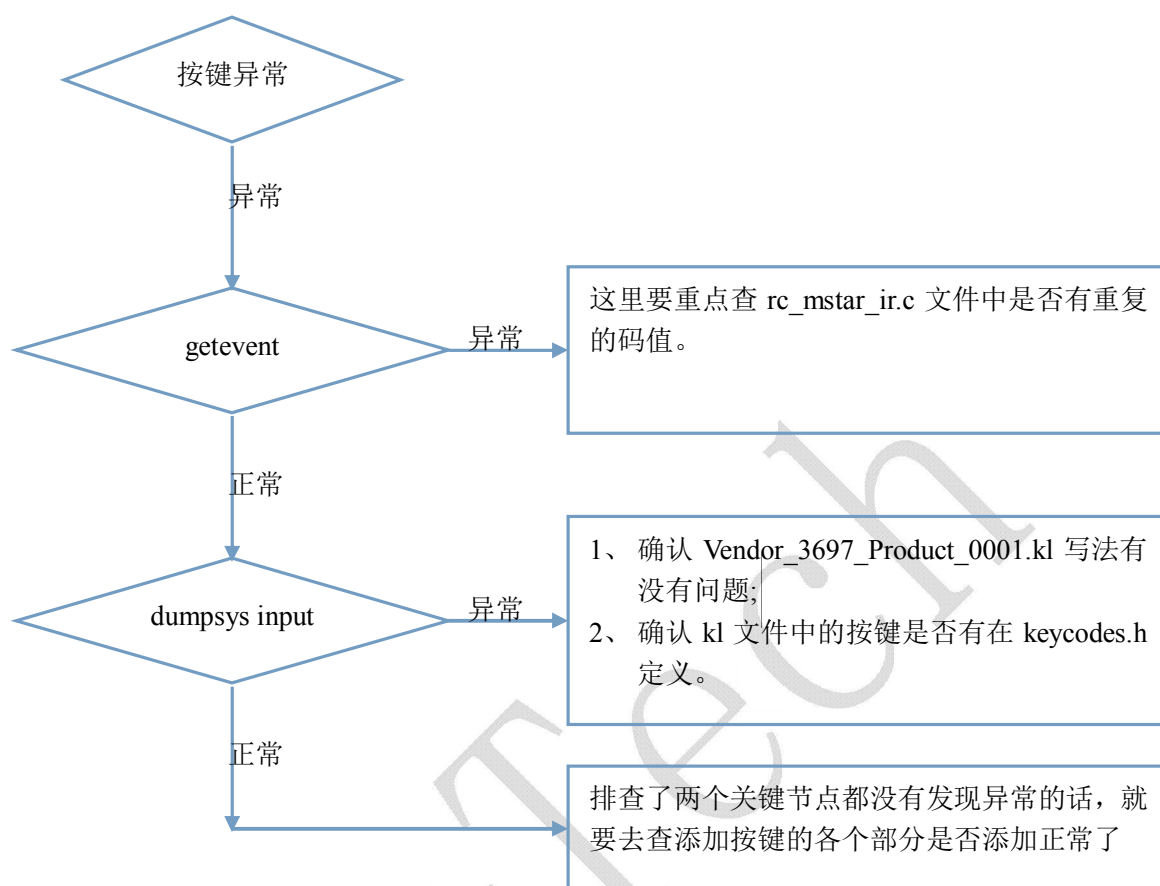
0001 008b 00000000 表示弹起

```
#define EV_SYN      0x00
#define EV_KEY      0x01
#define EV_REL      0x02
#define EV_ABS      0x03
#define EV_MSC      0x04
#define EV_SW       0x05
```

2) dumphs input 信息分析

```
HaveKeyboardLayoutOverlay: false
1: MStar Smart TV Keypad
Classes: 0x00000801
Path: /dev/input/event0
Descriptor: 8f43d929a9472e8dc54d48a6c41e2435e8eaff35
Location:
ControllerNumber: 0
UniqueId:
Identifier: bus=0x0006, vendor=0x3697, product=0x0002, version=0x0000
KeyLayoutFile: /system/usr/keylayout/vendor_3697_Product_0002.kl
KeyCharacterMapFile: /system/usr/keychars/Generic.kcm
ConfigurationFile:
HaveKeyboardLayoutOverlay: false
2: MStar Smart TV IR Receiver
Classes: 0x00000801
Path: /dev/input/event1
Descriptor: 0e50bdc18d3ae0b6f247100cbd99062d93c208eb
Location: /dev/ir
ControllerNumber: 0
UniqueId:
Identifier: bus=0x0018, vendor=0x3697, product=0x0001, version=0x0001
KeyLayoutFile: /system/usr/keylayout/vendor_3697_Product_0001.kl
KeyCharacterMapFile: /system/usr/keychars/Generic.kcm
ConfigurationFile:
HaveKeyboardLayoutOverlay: false
3: MCE IR Keyboard/Mouse (ir)
```

分析流程图如下：



当所有的按键都配置 OK 之后，还有最后一步工作要做，就是把 kernel 编译生成的 ko 文件拷贝到 supernova 下的客户目录文件夹下并提交的服务器。