

Rk3308 Key Interface Documentation

发布版本：1.0

作者：Jacky.Ge

日期：2019.3.29

文件密级：公开资料

概述

该文档旨在介绍RK3308 DeviceIo库中接口。

芯片名称

RK3308

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

日期	版本	作者	修改说明
2019-3-29	V1.0	Jacky.Ge	初始版本

Rk3308 Key Interface Documentation

- 1、概述
- 2、接口说明
- 3、使用示例

1、概述

该代码模块集成在libDeviceIo.so动态库里面，基于input_event输入子系统，对按键的常用需求，包括短按、长按、组合按键等需求做了封装处理，方便开发。

2、接口说明

- `Callback函数定义`

最基础的Callback回调，会回调每一次按键的up和down事件

```
typedef int (*RK_input_callback)(const int key_code, const int key_value);
```

经过处理的Callback回调，一次短按事件只会回调一次

```
typedef int (*RK_input_press_callback)(const int key_code);
```

长按事件回调接口

```
typedef int (*RK_input_long_press_callback)(const int key_code, const uint32_t time);
```

心跳长按事件回调接口（即满足长按条件后，若保持按下则定时回调接口）

```
typedef int (*RK_input_long_press_hb_callback)(const int key_code, const int times);
```

组合按键回调接口

```
typedef int (*RK_input_compose_press_callback)(const char compose, const uint32_t time);
```

事务按键回调接口

```
typedef int (*RK_input_transaction_press_callback)(const char trans, const uint32_t time);
```

多次点击回调接口

```
typedef int (*RK_input_multiple_press_callback)(const int key_code, const int times);
```

- `int RK_input_init(RK_input_callback input_callback_cb)`

按键模块初始化接口，需要传入一个基础的RK_input_callback 回调函数

- `int RK_input_register_press_callback(RK_input_press_callback cb)` 注册按键单击事件回调，按键单击触发

- `int RK_input_register_long_press_callback(RK_input_long_press_callback cb, const uint32_t time, const int key_code)`

为key_code按键注册时长为time ms的长按事件

- `int RK_input_register_long_press_hb_callback(RK_input_long_press_hb_callback cb, const uint32_t time, const int key_code)`

为key_code按键注册hb长按事件，每time ms触发一次

- `int RK_input_register_multiple_press_callback(RK_input_multiple_press_callback cb, const int key_code, const int times)`

为key_code按键注册times次多击事件（即单击key_code times次，两两相差不超过500ms）

- `int RK_input_register_compose_press_callback(RK_input_compose_press_callback cb, const uint32_t time, const int key_code, ...)`

为key_code按键集注册组合事件，key_code按键集同时按下达达到time ms触发

- `int RK_input_register_transaction_press_callback(RK_input_transaction_press_callback cb, const uint32_t time, int key_code, ...)`

为key_code按键集注册事务事件，按顺序依次按下key_code集后触发

- `int RK_input_exit(void)`

按键模块退出，并释放相关资源

3、使用示例

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <unistd.h>
4  #include <linux/input.h>
5  #include <DeviceIo/Rk_key.h>
6
7  static int _RK_input_callback(const int key_code, const int key_value)
8  {
9      printf("_RK_input_callback key_code:%d; key_value:%d\n", key_code, key_value);
10     return 0;
11 }
12
13 static int _RK_input_press_callback(const int key_code)
14 {
15     printf("_RK_input_press_callback key_code:%d;\n", key_code);
16     return 0;
17 }
18
19 static int _RK_input_long_press_callback(const int key_code, const uint32_t time)
20 {
21     printf("_RK_input_long_press_callback key_code:%d; time:%lu\n", key_code, time);
22     return 0;
23 }
24
25 static int _RK_input_long_press_hb_callback(const int key_code, const int times)
26 {
27     printf("_RK_input_long_press_hb_callback key_code:%d; times:%d\n", key_code,
28     times);
29     return 0;
30 }
31
32 static int _RK_input_multiple_press_callback(const int key_code, const int times)
33 {
34     printf("_RK_input_multiple_press_callback key_code:%d; times:%d\n", key_code,
35     times);
36     return 0;
37 }
38
39 static int _RK_input_transaction_press_callback(const char* trans, const uint32_t
40     time)
41 {
42     printf("_RK_input_transaction_press_callback trans:%s; time:%lu\n", trans, time);
43     return 0;
44 }
```

```
43 static int _RK_input_compose_press_callback(const char* compose, const uint32_t time)
44 {
45     printf("_RK_input_compose_press_callback compose:%s; time:%lu\n", compose, time);
46     return 0;
47 }
48
49 int main(int argc, char **argv)
50 {
51     // 初始化input模块
52     RK_input_init(_RK_input_callback);
53     // 注册单击回调
54     RK_input_register_press_callback(_RK_input_press_callback);
55     // 注册KEY_VOLUMEUP按键的5000ms长按事件
56     RK_input_register_long_press_callback(_RK_input_long_press_callback, 5000,
KEY_VOLUMEUP);
57     // 注册KEY_VOLUMEDOWN按键的hb长按事件，每500ms触发一次hb
58     RK_input_register_long_press_hb_callback(_RK_input_long_press_hb_callback, 500,
KEY_VOLUMEDOWN);
59     // 注册KEY_POWER的双击事件
60     RK_input_register_multiple_press_callback(_RK_input_multiple_press_callback,
KEY_POWER, 2);
61     // 注册KEY_VOLUMEUP->KEY_VOLUMEUP->KEY_VOLUMEDOWN->KEY_VOLUMEDOWN的事务事件
62     RK_input_register_transaction_press_callback(_RK_input_transaction_press_callback,
2000, 4, KEY_VOLUMEUP, KEY_VOLUMEUP, KEY_VOLUMEDOWN, KEY_VOLUMEDOWN);
63     // 注册KEY_VOLUMEUP + KEY_VOLUMEDOWN 5000ms的组合按键
64     RK_input_register_compose_press_callback(_RK_input_compose_press_callback, 5000,
2, KEY_VOLUMEUP, KEY_VOLUMEDOWN);
65
66
67     for (;;)
68
69     RK_input_exit();
70     return 0;
71 }
```