

# RockChip Devicelo 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RockChip Devicelo库中按键相关接口。

## 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

---

## RockChip Devicelo 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libDevicelo.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

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <linux/input.h>
#include <DeviceIo/Rk_key.h>

static int _RK_input_callback(const int key_code, const int key_value)
{
    printf("_RK_input_callback key_code:%d; key_value:%d\n", key_code, key_value);
    return 0;
}

static int _RK_input_press_callback(const int key_code)
{
    printf("_RK_input_press_callback key_code:%d;\n", key_code);
    return 0;
}

static int _RK_input_long_press_callback(const int key_code, const uint32_t time)
{
    printf("_RK_input_long_press_callback key_code:%d; time:%lu\n", key_code, time);
    return 0;
}

static int _RK_input_long_press_hb_callback(const int key_code, const int times)
{
    printf("_RK_input_long_press_hb_callback key_code:%d; times:%d\n", key_code, times);
    return 0;
}

static int _RK_input_multiple_press_callback(const int key_code, const int times)
{
    printf("_RK_input_multiple_press_callback key_code:%d; times:%d\n", key_code, times);
    return 0;
}

static int _RK_input_transaction_press_callback(const char* trans, const uint32_t time)
{
    printf("_RK_input_transaction_press_callback trans:%s; time:%lu\n", trans, time);
    return 0;
}

static int _RK_input_compose_press_callback(const char* compose, const uint32_t time)
{
    printf("_RK_input_compose_press_callback compose:%s; time:%lu\n", compose, time);
    return 0;
}

int main(int argc, char **argv)
{

```

```
// 初始化input模块
RK_input_init(_RK_input_callback);
// 注册单击回调
RK_input_register_press_callback(_RK_input_press_callback);
// 注册KEY_VOLUMEUP按键的5000ms长按事件
RK_input_register_long_press_callback(_RK_input_long_press_callback, 5000,
KEY_VOLUMEUP);
// 注册KEY_VOLUMEDOWN按键的hb长按事件，每500ms触发一次hb
RK_input_register_long_press_hb_callback(_RK_input_long_press_hb_callback, 500,
KEY_VOLUMEDOWN);
// 注册KEY_POWER的双击事件
RK_input_register_multiple_press_callback(_RK_input_multiple_press_callback, KEY_POWER,
2);
// 注册KEY_VOLUMEUP->KEY_VOLUMEUP->KEY_VOLUMEDOWN->KEY_VOLUMEDOWN的事务事件
RK_input_register_transaction_press_callback(_RK_input_transaction_press_callback,
2000, 4, KEY_VOLUMEUP, KEY_VOLUMEUP, KEY_VOLUMEDOWN, KEY_VOLUMEDOWN);
// 注册KEY_VOLUMEUP + KEY_VOLUMEDOWN 5000ms的组合按键
RK_input_register_compose_press_callback(_RK_input_compose_press_callback, 5000, 2,
KEY_VOLUMEUP, KEY_VOLUMEDOWN);

for (;;)

RK_input_exit();
return 0;
}
```