密级状态：绝密（　　）　秘密（　　）　内部（ √ ）　公开（　）

RK3399_VR 分体机_KEY_修改说明文档 _V1.0_2016.9.3

（技术部，第二系统产品部门）

文件状态： [　] 正在修改 [√] 正式发布	当前版本：	V1.0
	作　　者：	王剑辉
	完成日期：	2016-09-03
	审　　核：	张文平
	完成日期：	2016-09-03

福州瑞芯微电子股份有限公司

Fuzhou Rockchips Semiconductor Co . , Ltd

更新记录

版本	修改人	修改日期	修改说明	备注
V1.0	王剑辉	2016.9.3	初始版本	
V1.1	王剑辉	2017.1.4	更新 KEY 值修改说明和添加 TP 功能	

目 录

1 概述

本文档主要说明 RK3399 VR 分体机项目如何添加一个按键或者触摸板按键，主要包括头盔端 NanaoC 和主机端 RK3399 两边修改。

2 KEY 按键、触摸板添加或修改--NANOC 端

2.1 代码介绍

Nanoc 端的修改，相关代码如下：

Nanoc 对应 key 值的宏定义在 NanoC_VR_Release/Common/Driver/AD_KEY/AD_Key.h 中。Nanoc key 有定义一个 key 的结构体，用来存放 key 值的状态，结构体的路径 Common\Driver\USB\USBHidRkvrData.h

```
struct keymap_t{
    __u16 key_menu_up:1;
    __u16 key_menu_down:1;
    __u16 key_home_up:1;
    __u16 key_home_down:1;
    __u16 key_power_up:1;
    __u16 key_power_down:1;
    __u16 key_volup_up:1;
    __u16 key_volup_down:1;
    __u16 key_voldn_up:1;
    __u16 key_voldn_down:1;
    __u16 key_esc_up:1;
    __u16 key_esc_down:1;
    /*for touch screen **/
```

```
    __u16 key_up_pressed:1;
    __u16 key_up_released:1;
    __u16 key_down_pressed:1;
    __u16 key_down_released:1;
    __u16 key_left_pressed:1;
    __u16 key_left_released:1;
    __u16 key_right_pressed:1;
    __u16 key_right_released:1;
    __u16 key_enter_pressed:1;
    __u16 key_enter_released:1;
    __u16 key_pressed:1;
    __u16 psensor_on:1;
    __u16 psensor_off:1;
};

typedef union rkvr_data_t{
    struct rkvr_data{
        struct rkvr_sensor_data sensor_data;
        __u8 buf_reserve[10];
        struct keymap_t key_map;
    }rkvr_data;
    __u8 buf[62];
}RKVR_DATA_UN;
```

Nanoc 监听 key 按下的代码路径：NanoC_VR_Release/SDK/UI/USB/FunUSB.c

代码如下：

```
_ATTR_USB_UI_CODE_
void HID_ReportData()
{
    int16 accel_data[3] = {0};
    int16 gyro_data[3] ={0};
    uint8 temperature[2] ={0};
    float tempreature;
    UINT32    TempKeyVal;

    RKVR_DATA_UN rkvr_data_un;

    if(USBWriter_IsBusy(62)) {
        return;
    }

    memset(rkvr_data_un.buf, 0, sizeof(rkvr_data_un.buf));

    if (gSysConfig.UsbSensor == 1)
    {
        //获取 Sensor  数据
        MPU6500_Read_Data(accel_data,1,gyro_data,1);
        MPU6500_Read_Temperature(temperature);
        sensor_data_fill(&(rkvr_data_un.rkvr_data.sensor_data),accel_data,gyro_data,temperature);
    }
    //触摸板的键值上报
    if(key_map&KEY_UP_MASK_BIT) {
```

```
        rkvr_data_un.rkvr_data.key_map.key_up_pressed = 1;
        key_map &= ~KEY_UP_MASK_BIT;
        printf("menu UP down\n");
    }
    if(key_map&KEY_DOWN_MASK_BIT) {
        rkvr_data_un.rkvr_data.key_map.key_down_pressed = 1;
        key_map &= ~KEY_DOWN_MASK_BIT;
        printf("menu DOWN down\n");
    }
    if(key_map&KEY_LEFT_MASK_BIT) {
        rkvr_data_un.rkvr_data.key_map.key_left_pressed = 1;
        key_map &= ~KEY_LEFT_MASK_BIT;
        printf("menu LEFT down\n");
    }
    if(key_map&KEY_RIGHT_MASK_BIT) {
        rkvr_data_un.rkvr_data.key_map.key_right_pressed = 1;
        key_map &= ~KEY_RIGHT_MASK_BIT;
        printf("menu RIGHT down\n");
    }
    if(key_map&KEY_ENTER_MASK_BIT) {
        rkvr_data_un.rkvr_data.key_map.key_enter_pressed = 1;
        key_map &= ~KEY_ENTER_MASK_BIT;
        printf("menu ENTER down\n");
    }
```

//按键键值上报

```
TempKeyVal =    GetKeyVal();
switch (TempKeyVal)
{
    case KEY_VAL_MENU_DOWN:
        {
            printf("menu key down\n");
            rkvr_data_un.rkvr_data.key_map.key_menu_down = 1;
            rkvr_data_un.rkvr_data.key_map.key_pressed = 1;
            USBWriteEp(HID_IN_EP_SENSOR ,62, rkvr_data_un.buf);
        }
        break;

    case KEY_VAL_MENU_LONG_UP:
    case KEY_VAL_MENU_SHORT_UP: //Menu key
        {
            printf ("Menu key up\n");
            rkvr_data_un.rkvr_data.key_map.key_menu_up = 1;
            rkvr_data_un.rkvr_data.key_map.key_pressed = 1;
            USBWriteEp(HID_IN_EP_SENSOR ,62, rkvr_data_un.buf);     // 发送数据包给
3399 端
        }
        break;

    case KEY_VAL_UP_DOWN:
        {
            printf("volup key down\n");
```

```
            rkvr_data_un.rkvr_data.key_map.key_volup_down = 1;
            rkvr_data_un.rkvr_data.key_map.key_pressed = 1;
            USBWriteEp(HID_IN_EP_SENSOR ,62, rkvr_data_un.buf);
        }
        break;

    case KEY_VAL_UP_LONG_UP:        //volume increse
    case KEY_VAL_UP_SHORT_UP:
        {
            printf ("volup key up\n");
            rkvr_data_un.rkvr_data.key_map.key_volup_up = 1;
            rkvr_data_un.rkvr_data.key_map.key_pressed = 1;
            USBWriteEp(HID_IN_EP_SENSOR ,62, rkvr_data_un.buf);
        }
        break;

    case KEY_VAL_FFW_DOWN:
        {
            printf("voldn key down\n");
            rkvr_data_un.rkvr_data.key_map.key_voldn_down = 1;
            rkvr_data_un.rkvr_data.key_map.key_pressed = 1;
            USBWriteEp(HID_IN_EP_SENSOR ,62, rkvr_data_un.buf);
        }
        break;

    case KEY_VAL_FFW_LONG_UP:
```

```
case KEY_VAL_FFW_SHORT_UP:
    {
        printf ("voldn key up\n");
        rkvr_data_un.rkvr_data.key_map.key_voldn_up = 1;
        rkvr_data_un.rkvr_data.key_map.key_pressed = 1;
        USBWriteEp(HID_IN_EP_SENSOR ,62, rkvr_data_un.buf);
    }
    break;

case KEY_VAL_FFD_DOWN:
    {
        printf("FFD key down\n");
    }
    break;

case KEY_VAL_FFD_SHORT_UP:
case KEY_VAL_FFD_LONG_UP:
    {
        printf("FFD key up\n");
    }
    break;

case KEY_VAL_DOWN_DOWN:
    {
        printf("home key down\n");
        rkvr_data_un.rkvr_data.key_map.key_home_down = 1;
```

```
            rkvr_data_un.rkvr_data.key_map.key_pressed = 1;
            USBWriteEp(HID_IN_EP_SENSOR ,62, rkvr_data_un.buf);
        }
        break;

    case KEY_VAL_DOWN_LONG_UP:
    case KEY_VAL_DOWN_SHORT_UP:
        {
            printf ("home key up\n");
            rkvr_data_un.rkvr_data.key_map.key_home_up = 1;
            rkvr_data_un.rkvr_data.key_map.key_pressed = 1;
            USBWriteEp(HID_IN_EP_SENSOR ,62, rkvr_data_un.buf);
        }
        break;

    case KEY_VAL_PLAY_DOWN:
        {
            printf("PLAY key down\n");
        }
        break;

    case KEY_VAL_PLAY_LONG_UP:
    case KEY_VAL_PLAY_SHORT_UP:
        {
            printf ("Play key up\n");
        }
```

```
        break;

    case KEY_VAL_ESC_DOWN:
        {
            printf("power key down\n");
            rkvr_data_un.rkvr_data.key_map.key_power_down = 1;
            rkvr_data_un.rkvr_data.key_map.key_pressed = 1;
            USBWriteEp(HID_IN_EP_SENSOR ,62, rkvr_data_un.buf);
        }
        break;

    case KEY_VAL_ESC_LONG_UP:
    case KEY_VAL_ESC_SHORT_UP:
        {
            printf ("power key up\n");
            rkvr_data_un.rkvr_data.key_map.key_power_up = 1;
            rkvr_data_un.rkvr_data.key_map.key_pressed = 1;
            USBWriteEp(HID_IN_EP_SENSOR ,62, rkvr_data_un.buf);
        }
        break;

    default:
        {
            if (gSysConfig.UsbSensor == 1)
            {
                rkvr_data_un.rkvr_data.key_map.key_pressed = 0;
```

```
                    USBWriteEp(HID_IN_EP_SENSOR ,62, rkvr_data_un.buf);
                }
            }
            break;
        }
}
```

2.2 实例

下面举个例子：添加一个 play 按键

1.修改结构体：NanoC_VR_Release\Common\Include\RkvrInterface.h

```
struct keymap_t{
    __u16 key_menu_up:1;
    __u16 key_menu_down:1;
    __u16 key_home_up:1;
    __u16 key_home_down:1;
    __u16 key_power_up:1;
    __u16 key_power_down:1;
    __u16 key_volup_up:1;
    __u16 key_volup_down:1;
    __u16 key_voldn_up:1;
    __u16 key_voldn_down:1;
    __u16 key_play_up:1;
    __u16 key_play_down:1;
    __u16 key_pressed:1;
};
```

2.查找 play 按键的键值宏定义：NanoC_VR_Release/Common/Driver/AD_KEY/AD_Key.h

```
#define KEY_VAL_PLAY_DOWN              ((KEY_VAL_PLAY)|(KEY_STATUS_DOWN))
#define KEY_VAL_PLAY_SHORT_UP          ((KEY_VAL_PLAY)|(KEY_STATUS_SHORT_UP))
#define KEY_VAL_PLAY_LONG_UP           ((KEY_VAL_PLAY)|(KEY_STATUS_LONG_UP))
```

3.修改监听 key 事件的代码

```
_ATTR_USB_UI_CODE_
void HID_ReportData()
{
    。。。。。
    TempKeyVal =    GetKeyVal();
    switch (TempKeyVal)
    {
        case KEY_VAL_PLAY_DOWN:
            {
                printf("play key down\n");
                rkvr_data_un.rkvr_data.key_map.key_play_down = 1;
                rkvr_data_un.rkvr_data.key_map.key_pressed = 1;
                USBWriteEp(HID_IN_EP_SENSOR ,62, rkvr_data_un.buf);
            }
            break;
        case KEY_VAL_PLAY_LONG_UP:
        case KEY_VAL_PLAY_SHORT_UP: //Play    key
            {
                printf ("play key up\n");
                rkvr_data_un.rkvr_data.key_map.key_play_up = 1;
                rkvr_data_un.rkvr_data.key_map.key_pressed = 1;
```

```
                USBWriteEp(HID_IN_EP_SENSOR ,62, rkvr_data_un.buf);
            }
            break;
            。。。。。
    }
    。。。。。
}
```

3 KEY 按键、触摸板添加或修改--3399 端

结合上面 1 中的例子（添加 play 键）：

1.修改结构体，代码路径：kernel/drivers/hid/hid-rkvr.c

```
struct keymap_t {
    __u16 key_menu_up:1;
    __u16 key_menu_down:1;
    __u16 key_home_up:1;
    __u16 key_home_down:1;
    __u16 key_power_up:1;
    __u16 key_power_down:1;
    __u16 key_volup_up:1;
    __u16 key_volup_down:1;
    __u16 key_voldn_up:1;
    __u16 key_voldn_down:1;
    __u16 key_esc_up:1;
    __u16 key_esc_down:1;
    __u16 key_play_up:1;
    __u16 key_play_down:1;
```

```
    /*for touch panel **/
    __u16 key_up_pressed:1;
    __u16 key_up_released:1;
    __u16 key_down_pressed:1;
    __u16 key_down_released:1;
    __u16 key_left_pressed:1;
    __u16 key_left_released:1;
    __u16 key_right_pressed:1;
    __u16 key_right_released:1;
    __u16 key_enter_pressed:1;
    __u16 key_enter_released:1;
    __u16 key_pressed:1;
    __u16 psensor_on:1;
    __u16 psensor_off:1;
} __packed;
```

2.修改上报 key input 的代码，路径：kernel/drivers/hid/hid-rkvr.c

```
static unsigned int key_codes[] = {
    KEY_MENU,
    KEY_HOME,
    KEY_POWER,
    KEY_VOLUMEUP,
    KEY_VOLUMEDOWN,
    KEY_WAKEUP,
    KEY_PLAYER
};
```

```
static int rkvr_keys_event(struct hid_device *hdev, void *data, unsigned long len)
{
    struct input_dev *input = hdev->hiddev;
    union rkvr_data_t *rkvr_data = (union rkvr_data_t *)data;

    if (rkvr_data->rkvr_data.key_map.key_menu_up)
        rkvr_send_key_event(input, KEY_MENU, 0);
    else if (rkvr_data->rkvr_data.key_map.key_menu_down)
        rkvr_send_key_event(input, KEY_MENU, 1);
    else if (rkvr_data->rkvr_data.key_map.key_home_up)
        rkvr_send_key_event(input, KEY_HOME, 0);
    else if (rkvr_data->rkvr_data.key_map.key_home_down)
        rkvr_send_key_event(input, KEY_HOME, 1);
    else if (rkvr_data->rkvr_data.key_map.key_power_up)
        rkvr_send_key_event(input, KEY_POWER, 0);
    else if (rkvr_data->rkvr_data.key_map.key_power_down)
        rkvr_send_key_event(input, KEY_POWER, 1);
    else if (rkvr_data->rkvr_data.key_map.key_volup_up)
        rkvr_send_key_event(input, KEY_VOLUMEUP, 0);
    else if (rkvr_data->rkvr_data.key_map.key_volup_down)
        rkvr_send_key_event(input, KEY_VOLUMEUP, 1);
    else if (rkvr_data->rkvr_data.key_map.key_voldn_up)
        rkvr_send_key_event(input, KEY_VOLUMEDOWN, 0);
    else if (rkvr_data->rkvr_data.key_map.key_voldn_down)
        rkvr_send_key_event(input, KEY_VOLUMEDOWN, 1);
```

```
    else if (rkvr_data->rkvr_data.key_map.key_play_up)
        rkvr_send_key_event(input, KEY_PLAYER, 0);
    else if (rkvr_data->rkvr_data.key_map.key_play_down)
        rkvr_send_key_event(input, KEY_PLAYER, 1);
    return 0;
}
```