接近感应功能移植说明

Goodix

Revision History

日期	版本	说明
2014.11.05	V0.1	初始版本。

目录

一、概述

在 HAL 层创建 PSensor 设备，通过 sysfs 控制驱动 enable/disable 接近检测功能，驱动通过单独的 input 设备上报接近、离开的事件。

因此，功能实现和移植包含 HAL 层和 TP 驱动两个部分。

二、功能说明

A. 驱动说明

1、 上电初始化

在 sysfs 中创建/sys/goodix/gtp_ps/enable 和/sys/goodix/gtp_ps/state 供 HAL 层控制使能和查询状态；创建单独的 input 设备（gtp_proximity）用于上报接近和离开的事件；

2、 Suspend 处理

如果当前接近感应功能使能，并且检测到了接近，suspend 不做任何处理，直接返回；否则执行正常的 suspend 流程；

3、 Resume 处理

如果之前的 suspend 是在检测到接近后触发的，本次 resume 直接返回；否则，如果接近接近检测功能打开，在最终 resume 完成后，重新向 IC 发送进入接近感应模式的命令；

4、 中断处理

在中断处理中，判断接近检测功能是否使能，如果使能，则判断接近感应标志位，如果和上一次的状态不相同，就上报当前的状态；

5、 与 HAL 层接口

驱动使用 sysfs 与 HAL 层交互接口：

- /sys/goodix/gtp_ps/enable

  使能接近感应： 写入“1\n”

  禁用接近感应： 写入“0\n”

  查看当前使能状态： 读取

- /sys/goodix/gtp_ps/state

  查询的接近状态： 读取

  强制上报离开： 写入“1\n”

地址：深圳市福田保税区腾飞工业大厦B座13层 邮编：518000 电话：+86-755-3333 8828 传真：+86-755-3333 8788 Email：info@goodix.com

强制上报接近： 写入“0\n”

B. HAL 说明

HAL 层创建一个标准的 PSensor 设备与系统进行交互，将上层的使能/禁用命令发送给驱动；同时从/dev/input/eventX 中读取驱动上报的接近感应事件，上报给系统；

三、移植说明

A. 驱动移植

驱动中只需要打开接近感应模块的开关：

#define GTP_PROXIMITY 1

B. HAL 移植

（针对标准 Android 源码目录，根目录为 Android 源码目录）

1、 切换到 sensor HAL 代码的目录（例如/hardware/cm/libsensors，**不同的项目位置不一样，需要自行找到该目录**）；将 GtpPSensor.cpp 和 GtpPSensor.h 拷贝到当前目录；

2、 在 Android.mk 中添加 GtpPSensor.cpp；

```
LOCAL_SRC_FILES := \
                SensorBase.cpp \
                InputEventReader.cpp \
                sensors.cpp \
                AdxlSensor.cpp \
                GtpPSensor.cpp
```

3、 修改 sensors.cpp

- 添加 #include "GtpPSensor.h"
- enum {

```
        acc    = 0,
        akm    = 1,
        pxy    = 2,
        numSensorDrivers,
        numFds,
    };
```

- 定义 PSensor 的 handle，与其他 sensor 不同即可：

```
#define SENSORS_PROXIMITY_HANDLE     3
```

- 在 static const struct sensor_t sSensorList[]中添加 PSensor 的定义：

```
static const struct sensor_t sSensorList[] = {
        { "Analog Devices ADXL345/6 3-axis Accelerometer",
          "ADI",
          1, SENSORS_ACCELERATION_HANDLE,
```

```
            SENSOR_TYPE_ACCELEROMETER, (GRAVITY_EARTH * 16.0f),
                (GRAVITY_EARTH * 16.0f) / 4096.0f, 0.145f, 10000, 0, 0, { } },
        { "AK8975 Orientation sensor",
            "Asahi Kasei Microdevices",
            1, SENSORS_ORIENTATION_HANDLE,
            SENSOR_TYPE_ORIENTATION, 360.0f,
                CONVERT_O, 0.495f, 10000, 0, 0, { } },
        { "GTP P-Sensor",
            "Goodix",
            1, SENSORS_PROXIMITY_HANDLE,
            SENSOR_TYPE_PROXIMITY, 1, 1,
            5.0f, 0, 0, 0, { } }
    };
```

- sensors_poll_context_t()中增加 PSensor 的初始化：

```
mSensors[pxy] = new GtpPSensor();
mPollFds[pxy].fd = mSensors[pxy]->getFd();
mPollFds[pxy].events = POLLIN;
mPollFds[pxy].revents = 0;
```

- int sensors_poll_context_t::handleToDriver(int handle) {

```
        switch (handle) {
            case ID_A:
                return acc;
            case ID_M:
            case ID_O:
                return akm;
            case SENSORS_PROXIMITY_HANDLE:
                return pxy;
        }
        return -EINVAL;
    }
```

四、调试说明

1、 查看 sysfs 接口是否创建成功，如果不成功，查看驱动打印的 log；

adb shell ls -l /sys/goodix/gtp_ps/enable

adb shell ls -l /sys/goodix/gtp_ps/state

2、 在亮屏状态下，测试接近检测功能是否正常；

1）查看驱动上报的事件

打开一个新的命令行，执行 **adb shell getevent -p** ，找到 name 为 gtp_proximity 的设备

/dev/input/eventX (X 为数字)

执行 **adb shell getevent -l /dev/input/eventX 监视 input 设备的输出，正常情况下输出内容如下格式：**

EV_ABS ABS_DISTANCE 00000000

EV_SYN SYN_REPORT 00000000

EV_ABS ABS_DISTANCE 00000001

EV_SYN SYN_REPORT 00000000

2）使能接近感应功能

adb shell "echo 1 >/sys/goodix/gtp_ps/enable"

3）贴近手机顶部，触发接近感应，看输出有变化；

五、注意事项

1、 驱动注册的 input 设备的名称、sysfs 接口的位置是与 HAL 代码中对应的，单独修改驱动或者 HAL 将不能正常工作；