rtos-keyadc使用说明

本使用说明编写时使用的配置为：hichip_hc15xx_db_b200_defconfig，16xx芯片的板子更换对应的deconfig，也适用。

1、配置

keyadc电压值的设置

1、打开hc15xx-db-b200.dts文件，找到adc节点。

``` adc { devpath = "/dev/adc"; status = "okay"; key-num = <20>; /*key_map = <voltagemin voltagemax, code>*/ key-map = <0 20 0>, <40 80 1>, <90 140 2>, <160 210 3>, <230 270 4>, <310 360 5>, <390 450 6>, <470 520 7>, <580 640 8>, <680 710 9>, <740 790 10>, <840 880 11>, <900 990 12>, <1020 1070 13>, <1140 1180 14>, <1230 1280 15>, <1370 1430 16>, <1480 1540 17>, <1620 1660 18>, <1760 1810 19>;

```
};
```

```

key-num代表按键数量。

key-map成员的含义为不同按键按下时产生的<电压最小值 电压最大值 按键>。最小值和最大值为按键按下产生的电压容错范围，在这个电压范围内认为是该按键按下，可根据按下按键产生的电压，修改最小值和最大值。

2、编译

keyadc驱动编译

1、cd进入hcrtos路径下，执行make menuconfig，按如下路径进入：

```
c -> Components -> kernel -> Drivers -> input event -> saradc menu  (*) key adc driver  ( ) poll adc driver  ( ) touch adc driver 16xx
```

2、选择key adc driver选项

3、退出make menuconfig，然后执行make kernel-rebuild all

keyadc 测试命令编译：

1、cd进入hcrtos路径下，执行make menuconfig，按如下路径进入：

```
c -> Components -> Cmds -> adc test operations  ( ) touch adc test cmds  (*) key adc test cmds
```

2、选择key adc test cmds选项

3、退出make menuconfig，然后执行make cmds-rebuild all

3、测试

1、将编译好的out文件download到开发板，然后依次执行nsh、cd dev、cd input、ls命令，查看key adc对应的event号，此处是event0。

```
hc1512a@dbB200# hc1512a@dbB200#nsh hc1512a@dbB200(nsh)#cd dev hc1512a@dbB200(nsh)#cd input hc1512a@dbB200(nsh)#ls /dev/input: event0 event1
```

2、通过eixt命令回到起始路径，然后在串口终端下输入adc_test命令进入测试菜单；

```
hc1512a@dbB200# hc1512a@dbB200#adc_test hc1512a@dbB200(adc_test)#
```

3、输入key_adc_test -h获取帮助信息；

``` hc1512a@dbB200# adc*test hc1512a@dbB200(adc*test)# key*adc*test -h

---

key adc test cmds help for example : key*adc*test -i1 'i' 1 means event1

---
```

```

4、通过输入 key_adc_test -i数字 ，即可进行测试，数字为keyadc对应event号。此处输入key_adc_test -i0。按下对应的按键后即有对应的键值输出。

``` hc1512a@dbB200(adc*test)# key*adc_test -i0 type:1, code:10, value:1 key 10 Pressed type:1, code:10, value:1 key 10 Pressed type:1, code:10, value:1 key 10 Pressed type:1, code:10, value:0 key 10 Released type:1, code:14, value:1 key 14 Pressed type:1, code:14, value:1 key 14 Pressed type:1, code:14, value:0 key 14 Released type:1, code:15, value:1 key 15 Pressed type:1, code:15, value:0 key 15 Released type:1, code:17, value:1 key 17 Pressed type:1, code:17, value:0 key 17 Released

```

type：表示按键事件，code：表示按键号，value：1代表按下，0代表松开。

4、应用层使用

第1步：打开设备文件

```
c fd = open(/dev/input/event0, O_RDONLY);
```

第2步：读取一个event事件包

```
struct input_event t; read(fd, &t, sizeof(struct input_event));
```

第3步：解析event包

```
t.type：按键类型 t.code：按键号 t.value：按键值
```
```