



iMX6 矩阵键盘应用

Revision History

Date	Doc. Rev.	Linux BSP Version	Changes
2017-08-23	Rev. 0.1	V2.7b3 , Colibri iMX6	初始版本

触摸屏，无论是电阻式还是电容式，在嵌入式领域有着广泛的使用。配合设计优秀的 GUI，能够给用户带来很好的体验。很多时候，触摸屏在使用之前都需要进行校准，并且当外界的温度发生变化时，可能需要重新校准。对于电容屏还会受到静电的干扰。因此在工业设备上，为了保证输入的准确性，矩阵键盘仍被广泛地使用。

控制器通过轮询矩阵键盘的行列 GPIO 获取按键的情况。Linux 是一种多任务非实时操作系统，其他任务的执行会影响到矩阵键盘轮询，从而导致按键识别出现错误。iMX6 SoC 带有一个键盘控制器 KPP，除了能够实现轮询矩阵键盘外，还支持自带上拉电阻、消抖、多按键识别、长按等功能。用户可以直接 GPIO 连接到按键上，无需添加电阻和二极管。如果使用二极管，还可以支持更多的按键组合。默认的 KPP 最大支持 8*8 矩阵键盘，Toradex 的 iMX6 计算机模块，根据不同的系列支持部分按键，Colibri iMX6 为 5*5，Apalis iMX6 为 7*6。下面我们将以 Colibri iMX6 Linux 4.1.41 为例，介绍矩阵键盘的应用。

默认配置下，KPP 驱动并没有开启，需要修改 imx6dl-colibri-eval-v3.dts 和 imx6qdl-colibri.dtsi device tree 文件。你可以直接使用下面的补丁，修改矩阵键盘的配置。

ftp://ftp.toradex.cn/Documents/ApplicationNote/resources/matrix_keypad.patch

然后重新生成 dtb 文件，并更新到 Colibri iMX6 后重启。

```
make imx6dl-colibri-eval-v3.dtb
```

在启动日志中可以看到 KPP 驱动被加载

```
root@colibri-imx6:/dev/input# dmesg|grep kpp
[ 1.551426] input: 20b8000.kpp as /devices/soc0/soc/2000000.aips-bus/20b8000.kpp/input/input0
```

Linux 输入设备识别 KPP

```
root@colibri-imx6:~# xinput
┌ Virtual core pointer                      id=2                [master pointer (3)]
├   ↳ Virtual core XTEST pointer            id=4                [slave pointer (2)]
├   ↳ stmpe-ts                               id=8                [slave pointer (2)]
┌ Virtual core keyboard                    id=3                [master keyboard (2)]
├   ↳ Virtual core XTEST keyboard            id=5                [slave keyboard (3)]
├   ↳ gpio-keys                             id=6                [slave keyboard (3)]
├   ↳ 20b8000.kpp                           id=7                [slave keyboard (3)]
```

根据在 device tree 中的 GPIO 配置，短接任一对 ROW 和 KEY 的引脚，即可模拟按键。

```
pinctrl_kpp: kppgrp {
    fsl,pins = <
        MX6QDL_PAD_KEY_ROW2__KEY_ROW2 0x1b010    /*SODIMM 186*/
        MX6QDL_PAD_KEY_ROW4__KEY_ROW4 0x1b010    /* SODIMM 188*/
        MX6QDL_PAD_CSI0_DAT5__KEY_ROW5 0x1b010    /* SODIMM 155*/
        MX6QDL_PAD_GPIO_2__KEY_ROW6 0x1b010       /* SODIMM 180*/
        MX6QDL_PAD_GPIO_5__KEY_ROW7 0x1b010       /* SODIMM 176*/
        MX6QDL_PAD_KEY_COL2__KEY_COL2 0x110b0     /* SODIMM 184*/
        MX6QDL_PAD_KEY_COL4__KEY_COL4 0x110b0     /* SODIMM 178*/
        MX6QDL_PAD_GPIO_19__KEY_COL5 0x110b0      /* SODIMM 168*/
        MX6QDL_PAD_CSI0_DAT6__KEY_COL6 0x110b0    /* SODIMM 157*/
        MX6QDL_PAD_GPIO_4__KEY_COL7 0x110b0       /* SODIMM 174*/

    >;
};
```

通过 evtest 命令，查看按下的键值，以及是否属于长按等信息。

```
root@colibri-imx6:~# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0: 20b8000.kpp
/dev/input/event1: stmpe-ts
/dev/input/event2: gpio-keys
Select the device event number [0-2]: 0
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x0 product 0x0 version 0x0
Input device name: "20b8000.kpp"
.....
Testing ... (interrupt to exit)
Event: time 1502961213.623148, type 4 (EV_MSC), code 4 (MSC_SCAN), value 12
Event: time 1502961213.623148, type 1 (EV_KEY), code 2 (KEY_1), value 1
Event: time 1502961213.623148, ----- SYN_REPORT -----
Event: time 1502961213.873096, type 1 (EV_KEY), code 2 (KEY_1), value 2
Event: time 1502961213.873096, ----- SYN_REPORT -----
Event: time 1502961213.913117, type 1 (EV_KEY), code 2 (KEY_1), value 2
Event: time 1502961213.913117, ----- SYN_REPORT -----
Event: time 1502961213.953088, type 1 (EV_KEY), code 2 (KEY_1), value 2
Event: time 1502961213.953088, ----- SYN_REPORT -----
Event: time 1502961213.963156, type 4 (EV_MSC), code 4 (MSC_SCAN), value 12
Event: time 1502961213.963156, type 1 (EV_KEY), code 2 (KEY_1), value 0
```

在应用程序中，用 struct input_event 读取按键的信息，

```
struct input_event {  
    struct timeval time;  
    __u16 type; //类型  
    __u16 code; //键值  
    __s32 value; //按下还是释放  
};
```

具体请参考下面的源码

ftp://ftp.toradex.cn/Documents/ApplicationNote/resources/maxtrix_keypad_test.c

input_event 中读取按下键值以及按下、保持、释放动作。

```
root@colibri-imx6:~# ./maxtrix_keypad_test  
25 keys.  
Press any key.  
Key press  
Code: 7  
Key release  
Code: 7  
Key press  
Code: 7  
Key hold  
Code: 7  
Key hold  
Code: 7  
Key hold  
Code: 7  
Key hold  
Code: 7  
Key hold  
Code: 7  
Key hold  
Code: 7  
Key release  
Code: 7
```

KPP 还具有其他功能，利于识别组合键、消除 ghost key 等，矩阵键盘电路也需要做一些修改，如使用二极管，具体的参考 IMX6SDLRM 文档中 Chapter 39 Keypad Port (KPP) 章节。