

EC2x&AG35-QuecOpen

按键开发指导

LTE Standard/Automotive Module Series

版本：EC2x&AG35-QuecOpen_按键开发指导_V1.0

日期：2019-04-02

状态：临时文件

上海移远通信技术股份有限公司始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司

上海市徐汇区虹梅路 1801 号宏业大厦 7 楼 邮编：200233

电话：+86 21 51086236 邮箱：info@quectel.com

或联系我司当地办事处，详情请登录：

<http://www.quectel.com/cn/support/sales.htm>

如需技术支持或反馈我司技术文档中的问题，可随时登陆如下网址：

<http://www.quectel.com/cn/support/technical.htm>

或发送邮件至：support@quectel.com

前言

上海移远通信技术股份有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范、参数来设计其产品。由于客户操作不当而造成的人身伤害或财产损失，本公司不承担任何责任。在未声明前，上海移远通信技术股份有限公司有权对该文档进行更新。

版权申明

本文档版权属于上海移远通信技术股份有限公司，任何人未经我司允许而复制转载该文档将承担法律责任。

版权所有 ©上海移远通信技术股份有限公司 2019，保留一切权利。

Copyright © Quectel Wireless Solutions Co., Ltd. 2019.

文档历史

修订记录

版本	日期	作者	变更表述
1.0	2019-04-02	张奚坤	初始版本

目录

文档历史	2
目录	3
1 引言	4
2 驱动层	5
2.1. 设备树配置	5
2.1.1. 配置 mdm9607-mtp.dtsi 文件	5
2.1.2. 配置 mdm9607-pinctrl.dtsi 文件	5
2.2. 驱动配置	6
2.2.1. 配置 mdm9607-perf_deconfig 和 mdm9607_deconfig 文件	6
3 用户层调试	8
3.1. 编写应用层测试程序	8
3.2. 更改可执行文件权限	8

1 引言

本文档介绍的按键方案使用的是标准 Linux Input 子系统框架，主要适用于 Global 市场，支持的模块有 EC2X、AG35 系列，以下介绍如何在模块上支持按键。

Quectel
Confidential

2 驱动层

2.1. 设备树配置

2.1.1. 配置 mdm9607-mtp.dtsi 文件

soc 节点内添加 gpio_keys 节点，然后添加需要的按键，如下 key_test，主要需要配置的有如下几项：

1. gpios: 对应原理图中的引脚；
2. linux,input-type: 上报事件的类型，我们这里选择 1（EV_KEY）按键事件；
3. linux,code: 上报的事件的 code。

```
&soc {
    gpio_keys {
        compatible = "gpio-keys";
        input-name = "gpio-keys";
        pinctrl-names = "tlmm_gpio_key_active","tlmm_gpio_key_suspend";
        pinctrl-0 = <&gpio_key_active>;
        pinctrl-1 = <&gpio_key_suspend>;

        key_test {
            label = "key test";
            gpios = <&tlmm_pinmux 15 0x1>;
            linux,input-type = <1>;
            linux,code = <201>;
            //gpio-key,wakeup;
            debounce-interval = <15>;
        };
    };
};
```

可参考：/kernel/Documentation/devicetree/bindings/input/gpio-keys.txt 文档。

2.1.2. 配置 mdm9607-pinctrl.dtsi 文件

如下图所示，添加 gpio_keys 节点。

```
/*key button*/
gpio_keys {
    gpio_key_active: gpio_key_active {
        mux {
            pins = "gpio15";
            function = "gpio";
        };

        config {
            pins = "gpio15";
            drive-strength = <2>;
            bias-pull-up;
        };
    };

    gpio_key_suspend: gpio_key_suspend {
        mux {
            pins = "gpio15";
            function = "gpio";
        };

        config {
            pins = "gpio15";
            drive-strength = <2>;
            bias-pull-up;
        };
    };
};
```

2.2. 驱动配置

2.2.1. 配置 mdm9607-perf_deconfig 和 mdm9607_deconfig 文件

在 mdm9607-perf_deconfig 和 mdm9607_deconfig 两个文件中均添加如下内容：

CONFIG_KEYBOARD_GPIO=y

如下图所示：

```

391 CONFIG_FAULT_INJECTION_STACKTRACE_FILTER=y
392 CONFIG_MSM_RTB=y
393 CONFIG_IPC_LOGGING=y
394 CONFIG_BLK_DEV_IO_TRACE=y
395 CONFIG_DEBUG_USER=y
396 CONFIG_CRYPTODEV_QCRYPTO=y
397 CONFIG_CRYPTODEV_QCOM_MSM_QCE=y
398 CONFIG_CRYPTODEV_QCEDEV=y
399 CONFIG_QMI_ENCDEC=y
400 CONFIG_KEYBOARD_GPIO=y
401 #will.shao, add wakelocks for sleep
402 CONFIG_PM_WAKELOCKS=y
403

```

配置完成后，需要重新编译内核，然后将生成的镜像烧写进模块。

可使用 fastboot 下载：

```

adb reboot bootloader           // 进入 BootLoader 模式
astboot flash boot mdm9607-boot.img // 烧写内核镜像
fastboot reboot                 // 重启开发板

```


3 用户层调试

3.1. 编写应用层测试程序



key_test.

以上附件中为测试程序，仅供参考！

编译已写好的应用层测试程序，生成可执行文件 `key_test`，然后下载进模块。可使用 `adb` 下载：可使用 `adb` 下载：

`adb push ./key_test/data` (./为 `key_test` 文件所在路径，`/data` 为下载到开发板的路径)

3.2. 更改可执行文件权限

使用 `chmod +x key_test` 更改权限。最后使用 `./key_test` 执行。然后将 `gpio` 端口分别设置成高电平和低电平，查看超级终端是否有正常打印输出。