Build a Smarter World

EC2x&AG35-QuecOpen Press Key Development Guide

LTE Standard/Automotive Module Series

Rev. EC2x&AG35-QuecOpen_Press_Key_Development_Guide_V1.0

Date: 2019-04-02

Status: Preliminary

www.quectel.com

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://www.quectel.com/support/sales.htm

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm
Or email to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2019-04-02	Carola ZHANG	Initial

Contents

1 Introduction

This document uses the standard Linux Input subsystem framework to descript press key solution.

This document is mainly applicable to the Global market. The supported modules includes EC2x and AG35 series. The following chapters describes how to support press key on the module.

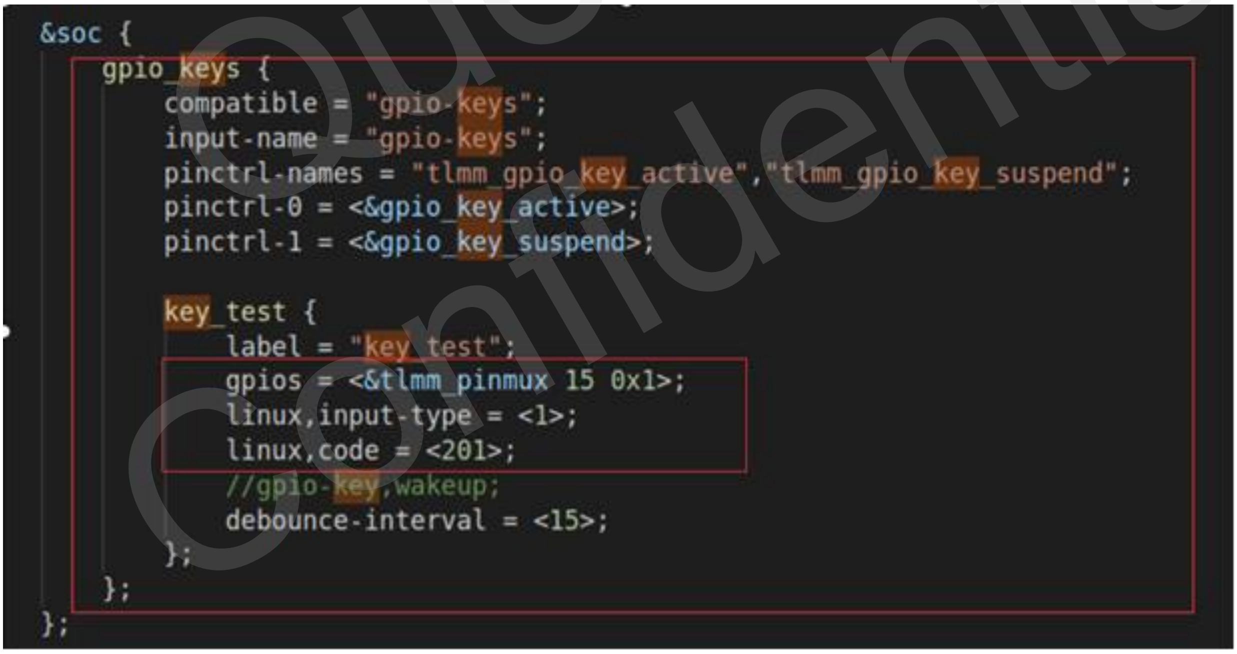
2 Drive Layer

2.1. Device Tree Configuration

2.1.1. Configuring mdm9607-mtp.dtsi File

Add the gpio_keys node to soc node, and then add the required keys, as follows key_test, there are mainly the following three items need to be configured:

1. Gpios: it corresponds to a pin in the schematic diagram.
2. Linux, input-type: report event type, user select 1 (EV_KEY) press key event here.
3. Linux, code: The event code that have been reported.



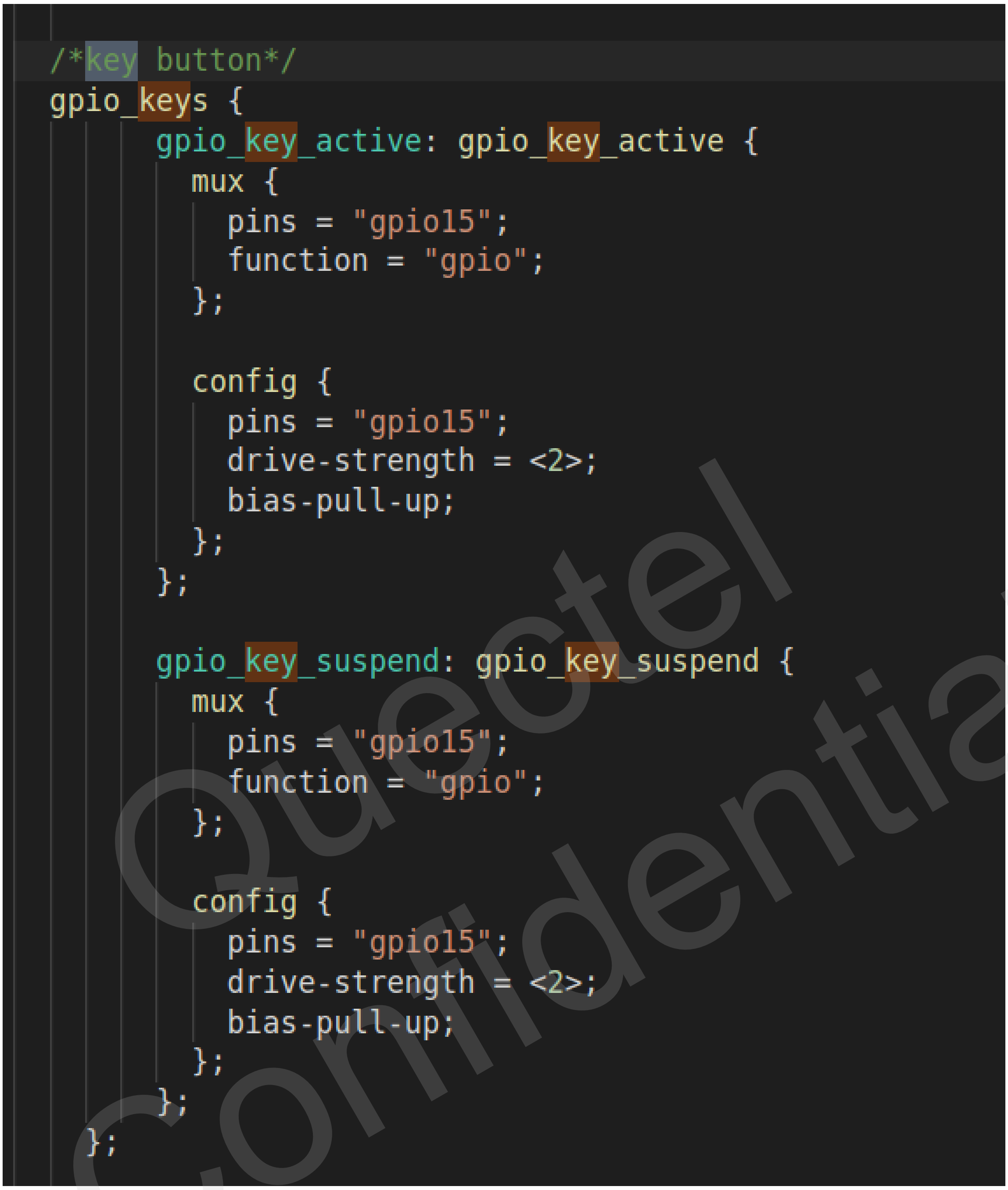
```
&soc {
    gpio_keys {
        compatible = "gpio-keys";
        input-name = "gpio-keys";
        pinctrl-names = "tlmm_gpio_key_active","tlmm_gpio_key_suspend";
        pinctrl-0 = <&gpio_key_active>;
        pinctrl-1 = <&gpio_key_suspend>;

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

For more information, please refer to*/kernel/Documentation/devicetree/bindings/input/gpio-keys.txt*

2.1.2. Configuring the mdm9607-pinctrl.dtsi File

Add *gpio_keys node* as shown below.



```
/*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;
			};
		};
	};
```

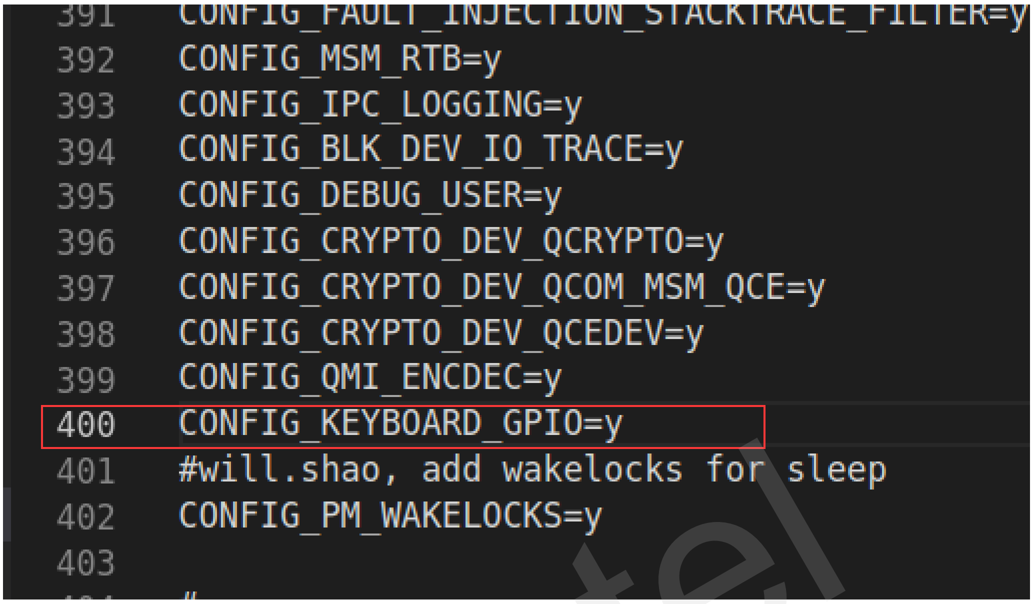
2.2. Drive Configuration

2.2.1. Configuring mdm9607-perf_deconfig and mdm9607_deconfig Files

Add the following content to mdm9607-perf_deconfig and mdm9607_deconfig files:

CONFIG_KEYBOARD_GPIO=y

As shown below:



```
CONFIG_FAULT_INJECTION_STACKTRACE_FILTER=y
CONFIG_MSM_RTB=y
CONFIG_IPC_LOGGING=y
CONFIG_BLK_DEV_IO_TRACE=y
CONFIG_DEBUG_USER=y
CONFIG_CRYPTO_DEV_QCRYPTO=y
CONFIG_CRYPTO_DEV_QCOM_MSM_QCE=y
CONFIG_CRYPTO_DEV_QCEDEV=y
CONFIG_QMI_ENCDEC=y
CONFIG_KEYBOARD_GPIO=y
#will.shao, add wakelocks for sleep
CONFIG_PM_WAKELOCKS=y

```

After the configuration, users need to recompile kernel and then download the generated image into the module.

It can be downloaded by fastboot:

```
adb reboot bootloader                        // Enter BootLoader mode
astboot flash boot mdm9607-boot.img          // download kernel image
fastboot reboot                             // reboot development board
```

3 User Layer Debugging

3.1. Write Application Layer Test Program

key_test.c

The above-mentioned attachment is the test program is for reference only.
Compile the written application layer test program to generate executable file *key_test*, and then download it into the module. It can be downloaded by adb:
Adb push./key_test/data (./ is the path where the key_test file is located, */data* is the path to download to development board)

3.2. Change Executable Permission

Change permission by *chmod +x key_test* and finally execute by./*key_test*. Then set gpio port to high level and low level respectively to see if the HyperTerminal has normal print output.