

N1 Smart Module GPIO Configuration

Version 1.0



Copyright © Neoway Technology Co., Ltd 2016. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Shenzhen Neoway Technology Co., Ltd.

Neoway 有方 is the trademark of Shenzhen Neoway Technology Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

This document provides guide for users to use the N1.

This document is intended for system engineers (SEs), development engineers, and test engineers.

The information in this document is subject to change without notice due to product version update or other reasons.

Every effort has been made in preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Shenzhen Neoway provides customers complete technical support. If you have any question, please contact your account manager or email to the following email addresses:

Sales@neoway.com

Support@neoway.com

Website: <http://www.neoway.com>

Change History		
Issue	Changes	Date
V1.0	Initial version	2016-08



Contents

1 Overview	1
2 LK	1
3 Kernel	1
3.1 DTSL.....	1
3.2 Driver Configuration.....	2



1 Overview

GPIO configuration includes LK part and kernel part.

2 LK

```
gpio_tlmm_config(31, 0, GPIO_OUTPUT, GPIO_PULL_DOWN, GPIO_8MA, GPIO_DISABLE);
gpio_set(31, 2);
```

`gpio_tlmm_config` configures one pin.

Output: `gpio_set`, sets output value

Input: `gpio_status`, reads input value

3 Kernel

3.1 DTSI

File: N1_msm8909-pinctrl.dtsi

Location: kernel/arch/arm/boot/dts/jsr

```
&soc {
    tlmm_pinmux: pinctrl@1000000 {
        compatible = "qcom,msm-tlmm-8916";
        reg = <0x1000000 0x300000>;
        interrupts = <0 208 0>;

        /*General purpose pins*/
        gp: gp {
            qcom,num-pins = <113>;
            #qcom,pin-cells = <1>;
            msm_gpio: msm_gpio {
                compatible = "qcom,msm-tlmm-gp";
                gpio-controller;
                #gpio-cells = <2>;
                interrupt-controller;
                #interrupt-cells = <2>;
                num_irqs = <113>;
            };
        };
    };
};
```

The above figure shows how to add GPIO configurations to the `tlmm_pinmux` node, wherein the `gp` node is the general GPIO configurations and there are 113 GPIOs.

```
pmx_i2c_5 {
```

```

/* CLK, DATA */
qcom,pins = <&gp 19>, <&gp 18>;      ---A. Used pins
qcom,num-grp-pins = <2>;          ---B. Total pin number
qcom,pin-func = <2>;             ---C. Function selection
label = "pmx_i2c_5";
i2c_5_active: i2c_5_active {      ---D. Initial status
    drive-strength = <2>; /* 2 MA */
    bias-disable = <0>; /* No PULL */
};
i2c_5_sleep: i2c_5_sleep {
    drive-strength = <2>; /* 2 MA */
    bias-pull-down; /* PULL DOWN */
};
};

```

3.2 Driver Configuration

```

if (gpio_is_valid(data->pdata->irq_gpio)) {
    err = gpio_request(data->pdata->irq_gpio,
        "ft5x06_irq_gpio");
    if (err) {
        dev_err(&data->client->dev,
            "irq gpio request failed");
        goto err_irq_gpio_req;
    }
    err = gpio_direction_input(data->pdata->irq_gpio);
    if (err) {
        dev_err(&data->client->dev,
            "set_direction for irq gpio failed\n");
        goto err_irq_gpio_dir;
    }
}
}

```

gpio_is_valid: determines whether the pin is valid.

gpio_request: requests pins.

gpio_direction_input: sets pin input.

gpio_direction_output: sets pin output.

 NOTE

Pin ID (GPIO_N) cannot be operated directly at kernel layer. You can operate the following parameters of a pin: data -> pdata -> irq_gpio, which are extracted through DTSL.

