Rockchip Pinctrl Developer Guide

ID: RK-KF-YF-136

Release Version: V1.0.0

Release Date: 2022-05-10

Security Level: □Top-Secret □Secret □Internal ■Public

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian,PRC

Website: www.rock-chips.com

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

This document introduces the dts usage and gpio APIs for Rockchip SoCs.

Product Version

Chipset	Kernel Version
RK3568/RK3399/RK3368/RK3288/PX30/RK3128/RK3126/RV1126	Linux-4.19
RK3588/RV1106	Linux-5.10

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Revision History

Version	Author	Date	Change Description
V1.0.0	Jianqun Xu	2022-05-10	Initial version

Contents

1. Introduce Pin Index Rules

Rockchip Pin Index includes three parts, bank、port and index.

1.1 GPIO

- bank: the number of banks should keep same with the gpio controllers number.
- port: fixed to 'A', 'B', 'C', 'D'.
- index: fixed range from 0 to 31.

For example, the rk3588 trm shows that the soc has 5 gpio controllers:

> There are five GPIOs (GPIO0 in PD_PMU,GPIO1/GPIO2/GPIO3/GPIO4 in PD_BUS)

each controller has 32 IO pads control by gpio registers.

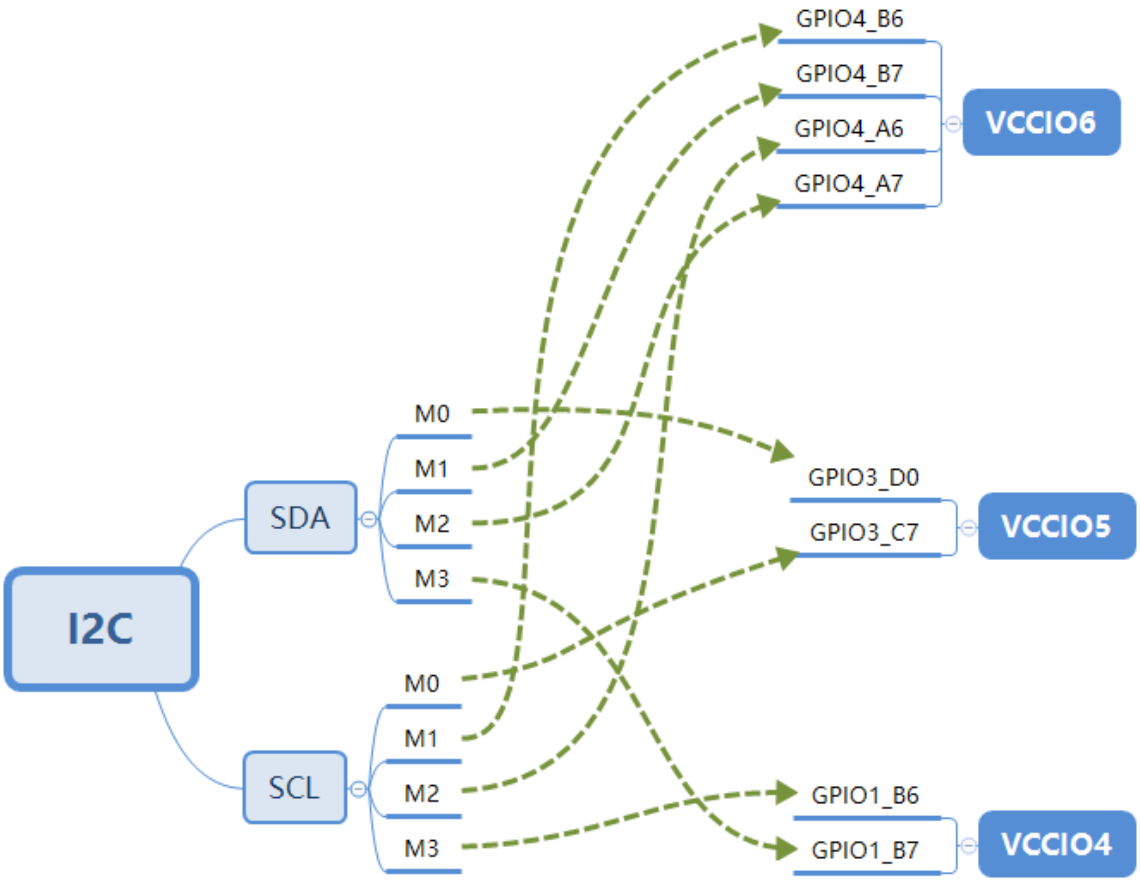
1.2 IOMUX

Rockchip Pin has multi-function routes, name m0, m1, m2 and so on.

For example RK3588 BUS_IOC_GPIO1B_IOMUX_SEL_H Address: Operational Base + offset (0x002C)

> gpio1b7_sel
> 4'h0: GPIO
> 4'h2: MIPI_CAMERA2_CLK_M0
> 4'h3: SPDIF1_TX_M0
> 4'h4: PCIE30X2_PERSTN_M3
> 4'h5: HDMI_RX_CEC_M2
> 4'h6: SATA2_ACT_LED_M1
> 4'h9: I2C5_SDA_M3
> 4'ha: UART1_RX_M1
> 4'hb: PWM13_M2

The rk3588 i2c5 iomux shows as following:



1.3 PULL

Rockchip Pin supports 3 bias states

- bias-disable
- bias-pull-up
- bias-pull-down

They are both work on GPIO and IOMUX since the bias designed on IO PAD.

1.4 DRIVE-STRENGTH

For the soc older than rk1808, the drive strength is set by mA, then from rk3399 it set by level, which is actually the register value ususally.

For example rk3588 GPIO0_C7

> gpio0c7_ds
> GPIO0C7 DS control Driver Strength Selection
> 3'b000: 100ohm
> 3'b100: 66ohm
> 3'b010: 50ohm
> 3'b110: 40ohm
> 3'b001: 33ohm
> 3'b101: 25ohm

From driver sight, it treats as:

> 3'b000: Level0
> 3'b100: Level4
> 3'b010: Level2
> 3'b110: Level6
> 3'b001: Level1
> 3'b101: Level5

From dts `drive-strength=<5>` means write '5' to register.

1.5 SMT

Usually the smitter is used by sda and scl of i2c, it can improve signal quality.

2. Introduce driver

2.1 Pinctrl driver

2.2 GPIO driver

3. Introduce DTS

For example `arch/arm64/boot/dts/rockchip/rk3588s.dtsi`

```
{
    pinctrl: pinctrl {
        compatible = "rockchip,rk3588-pinctrl";
        rockchip,grf = <&ioc>;
        #address-cells = <2>;
        #size-cells = <2>;
        ranges;

        gpio0: gpio@fd8a0000 {
            compatible = "rockchip,gpio-bank";
            reg = <0x0 0xfd8a0000 0x0 0x100>;
            interrupts = <GIC_SPI 277 IRQ_TYPE_LEVEL_HIGH>;
            clocks = <&cru PCLK_GPIO0>, <&cru DBCLK_GPIO0>;

            gpio-controller;
            #gpio-cells = <2>;
            interrupt-controller;
            #interrupt-cells = <2>;
        };

        gpio1: gpio@fec20000 {
            compatible = "rockchip,gpio-bank";
            reg = <0x0 0xfec20000 0x0 0x100>;
```

```
            interrupts = <GIC_SPI 278 IRQ_TYPE_LEVEL_HIGH>;
            clocks = <&cru PCLK_GPIO1>, <&cru DBCLK_GPIO1>;

            gpio-controller;
            #gpio-cells = <2>;
            interrupt-controller;
            #interrupt-cells = <2>;
        };

        gpio2: gpio@fec30000 {
            compatible = "rockchip,gpio-bank";
            reg = <0x0 0xfec30000 0x0 0x100>;
            interrupts = <GIC_SPI 279 IRQ_TYPE_LEVEL_HIGH>;
            clocks = <&cru PCLK_GPIO2>, <&cru DBCLK_GPIO2>;

            gpio-controller;
            #gpio-cells = <2>;
            interrupt-controller;
            #interrupt-cells = <2>;
        };

        gpio3: gpio@fec40000 {
            compatible = "rockchip,gpio-bank";
            reg = <0x0 0xfec40000 0x0 0x100>;
            interrupts = <GIC_SPI 280 IRQ_TYPE_LEVEL_HIGH>;
            clocks = <&cru PCLK_GPIO3>, <&cru DBCLK_GPIO3>;

            gpio-controller;
            #gpio-cells = <2>;
            interrupt-controller;
            #interrupt-cells = <2>;
        };

        gpio4: gpio@fec50000 {
            compatible = "rockchip,gpio-bank";
            reg = <0x0 0xfec50000 0x0 0x100>;
            interrupts = <GIC_SPI 281 IRQ_TYPE_LEVEL_HIGH>;
            clocks = <&cru PCLK_GPIO4>, <&cru DBCLK_GPIO4>;

            gpio-controller;
            #gpio-cells = <2>;
            interrupt-controller;
            #interrupt-cells = <2>;
        };
    };
};
```

Another file named `arch/arm64/boot/dts/rockchip/rk3588s-pinctrl.dtsi` will be include.

3.1 New pinctrl handle

The `rk3588s-pinctrl` file has already listed almost instances, module only needs to do selection.

But if there need to create a new pinctrl instance, some rules should be following:

- the node should under pinctrl node

- the node should has function and at least on group under it
- the node format

```
function {
    group {
        rockchip,pin = <bank gpio func &ref>;
    };
};
```

- common dt node rules

3.2 Reference to pinctrl handle

The module links to pinctrl driver by `pinctrl-names` and `pinctrl-0`.

For example rk3588 uart2:

```
{
    uart2: serial@feb50000 {
        compatible = "rockchip,rk3588-uart", "snps,dw-apb-uart";
        reg = <0x0 0xfeb50000 0x0 0x100>;
        interrupts = <GIC_SPI 333 IRQ_TYPE_LEVEL_HIGH>;
        clocks = <&cru SCLK_UART2>, <&cru PCLK_UART2>;
        clock-names = "baudclk", "apb_pclk";
        reg-shift = <2>;
        reg-io-width = <4>;
        dmas = <&dmac0 10>, <&dmac0 11>;
        pinctrl-names = "default";
        pinctrl-0 = <&uart2m1_xfer>;
        status = "disabled";
    };
};
```

The `uart2m1_xfer` is a pin group phandle.

The dt node also suppot mult-groups, for example rk3588 pdm1:

```
{
    pdm1: pdm@fe4c0000 {
        compatible = "rockchip,rk3588-pdm";
        reg = <0x0 0xfe4c0000 0x0 0x1000>;
        clocks = <&cru MCLK_PDM1>, <&cru HCLK_PDM1>;
        clock-names = "pdm_clk", "pdm_hclk";
        assigned-clocks = <&cru MCLK_PDM1>;
        assigned-clock-parents = <&cru PLL_AUPLL>;
        dmas = <&dmac1 4>;
        dma-names = "rx";
        power-domains = <&power RK3588_PD_AUDIO>;
        pinctrl-names = "default";
        pinctrl-0 = <&pdm1m0_clk
                 &pdm1m0_clk1
                 &pdm1m0_sdi0
                 &pdm1m0_sdi1
                 &pdm1m0_sdi2
```

```
                &pdm1m0_sdi3>;
/* another format as followings: */
/*
 *      pinctrl-0 = <&pdm1m0_clk>,
 *                  <&pdm1m0_clk1>,
 *                  <&pdm1m0_sdi0>,
 *                  <&pdm1m0_sdi1>,
 *                  <&pdm1m0_sdi2>,
 *                  <&pdm1m0_sdi3>;
 */
        #sound-dai-cells = <0>;
        status = "disabled";
    };
};
```

The pinctrl key states includes:

```
#define PINCTRL_STATE_DEFAULT "default"
#define PINCTRL_STATE_INIT "init"
#define PINCTRL_STATE_IDLE "idle"
#define PINCTRL_STATE_SLEEP "sleep"
```

And the `pinctrl-names` support user-defined, and driver should parse it.

4. QA

4.1 Userland IOMUX

build command

`gcc tools/testing/selftests/rkpinctrl/iomux.c -o iomux`

Example to set GPIO0_B7 iomux to func1

```
[root@RK3588:/]# iomux 0 15 1
```

Example to get GPIO0_B7 iomux value

```
[root@RK3588:/]# iomux 0 15
mux get (GPIO0-15) = 1
```