

# Rockchip Power Discrete DCDC Development Guide

---

ID:RK-KF-YF-114

Release Version: V1.4

Release Date: 2021-12-29

Security Level: Top-Secret Secret Internal Public

## DISCLAIMER

THIS DOCUMENT IS PROVIDED "AS IS". ROCKCHIP ELECTRONICS CO., LTD. ("ROCKCHIP") DOES NOT PROVIDE ANY WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, WITH RESPECT TO THE ACCURACY, RELIABILITY, COMPLETENESS, MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT OF ANY REPRESENTATION, INFORMATION AND CONTENT IN THIS DOCUMENT. THIS DOCUMENT IS FOR REFERENCE ONLY. THIS DOCUMENT MAY BE UPDATED OR CHANGED WITHOUT ANY NOTICE AT ANY TIME DUE TO THE UPGRADES OF THE PRODUCT OR ANY OTHER REASONS.

## Trademark Statement

"Rockchip", "瑞芯微", "瑞芯" shall be Rockchip's registered trademarks and owned by Rockchip. All the other trademarks or registered trademarks mentioned in this document shall be owned by their respective owners.

**All rights reserved. ©2021. Rockchip Electronics Co., Ltd.**

Beyond the scope of fair use, neither any entity nor individual shall extract, copy, or distribute this document in any form in whole or in part without the written approval of Rockchip.

Rockchip Electronics Co., Ltd.

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

Website: [www.rock-chips.com](http://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](mailto:fae@rock-chips.com)

---

## Preface

## Overview

## Product version

Chipset name	Kernel version
All Socs	Linux4.4 and above

## Applicable object

This document(guide) is mainly suitable for below engineers:

Field Application Engineer

Software Development Engineer

### Revision history

Date	Version	Author	Revision description
2017-07-24	V1.0	ZhangQing	The first version
2019-11-12	V1.1	ZhangQing	support linux 4.19 version
2021-02-24	V1.2	ZhangQing	support TCS452X
2021-10-14	V1.3	ZhangQing	support RK860X
2021-12-29	V1.4	ZhangQing	support linux 5.10 version

### Rockchip Power Discrete DCDC Development Guide

PWM Voltage Regulator

Driver

DTS node

SYR8XX Voltage Regulator

Driver

DTS node

XZ321X Voltage Regulator

Driver

DTS node

TCS452X Voltage Regulator

Driver

DTS node

RK860X Voltage Regulator

RK860X

Driver

DTS node

DEBUG Interface

Get Power Tree

Set voltage

---

## PWM Voltage Regulator

### Driver

The driver files location:

```
drivers/regulator/pwm-regulator.c
```

### DTS node

```
vdd_center: vdd-center {
    compatible = "pwm-regulator";
    rockchip,pwm_id = <2>;
    rockchip,pwm_voltge = <900000>;
    pwms = <&pwm2 0 25000 1>;
    regulator-name = "vdd_center";
    regulator-min-microvolt = <800000>;
    regulator-max-microvolt = <1400000>;
    regulator-always-on;
    regulator-boot-on;
};
```

The parameter description:

### Pwm Parameter

```
rockchip,pwm_id = <2>;//pwm2
rockchip,pwm_voltge = <900000>;//Init voltage in U-Boot
```

These two parameters are mainly used by U-Boot but not kernel.

```
pwms = <&pwm2 0 25000 1>;
```

PWM2 is using pwm2 node, 25000 is PWM cycle and 1 means PWM circuit polarity is reversed.

PWM circuit polarity:

Positive polarity: The larger the PWM duty ratio, the higher the output voltage

Reversed polarity: The larger the PWM duty ratio, the lower the output voltage

### Regulator Parameter

```
regulator-name = "vdd_center";
```

The name of the PWM output power, invoked for voltage regulating.

```
regulator-min-microvolt = <800000>;
regulator-max-microvolt = <1400000>;
```

The max and min voltages supported by PWM circuit hardware. They must be the actual hardware value. (Test method: The corresponding output voltage after pull PWM port up or down forcedly)

```
regulator-always-on;
```

Whether the power always on or not. You can delete the attribute if need to manage the switch by yourself.

```
regulator-boot-on;
```

Used in U-Boot if need to set the power on in U-Boot stage.

# SYR8XX Voltage Regulator

## Driver

The driver files location:

```
drivers/regulator/fan53555.c
```

## DTS node

```
vdd_cpu_b: syr827@40 {
    compatible = "silergy,syr827";
    reg = <0x40>;
    vin-supply = <&vcc5v0_sys>;
    regulator-compatible = "fan53555-reg";
    pinctrl-0 = <&vsel1_gpio>;
    vsel-gpios = <&gpio1 17 GPIO_ACTIVE_HIGH>;

    regulator-name = "vdd_cpu_b";
    regulator-min-microvolt = <712500>;
    regulator-max-microvolt = <1500000>;
    regulator-ramp-delay = <1000>;
    fcs,suspend-voltage-selector = <1>;
    regulator-always-on;
    regulator-boot-on;
    regulator-initial-state = <3>;
    regulator-state-mem {
        regulator-off-in-suspend;
    };
};
```

The parameter description:

### Supply Parameter

```
vin-supply = <&vcc5v0_sys>;
```

The hardware input voltage, no actual meaning, mainly used for constructing the power tree.

### Pinctrl Parameter

```
pinctrl-0 = <&vsel1_gpio>;
vsel-gpios = <&gpio1 17 GPIO_ACTIVE_HIGH>;
fcs,suspend-voltage-selector = <1>;
```

Pay attention to this: This IO is used to change two groups of different voltages, but currently it is used to quickly change the switch.

```
fcs,suspend-voltage-selector = <1>;
```

Enable voltage when VSEL pin is low, disable the voltage when it is high. IO is pulled down by default.

```
fcs,suspend-voltage-selector = <0>;
```

Enable voltage when VSEL pin is high, disable the voltage when it is low. IO is pulled up by default. The value should match with the actual hardware.

#### Note:

VSEL pin function can also be used to change voltage for sleep-resume instead of quickly changing the switch. Only need to delete:

```
pinctrl-0 = <&vsel1_gpio>;  
vsel-gpios = <&gpio1 17 GPIO_ACTIVE_HIGH>;
```

Now VSEL pin is connected to pmic\_sleep. The function:

```
fcs,suspend-voltage-selector = <1>;
```

Output running voltage when VSEL pin is low and output standby voltage when it is high(also can set to off for standby). IO is pulled down by default.

```
fcs,suspend-voltage-selector = <0>;
```

Output running voltage when VSEL pin is high and output standby voltage when it is low(also can set to off for standby). IO is pulled up by default.

#### Regulator Parameter

```
regulator-name = "vdd_cpu_b";
```

The name of the PWM output power, invoked for voltage regulating.

```
regulator-min-microvolt = <712500>;  
regulator-max-microvolt = <1500000>;
```

The max and min values limited by software, it is not allowable to set the values out of the range.

```
regulator-always-on;
```

Whether the power always on or not. You can delete the attribute if need to manage the switch by yourself.

```
regulator-boot-on;
```

Used in U-Boot to if need to set the power on in U-Boot stage.

```
regulator-ramp-delay = <1000>;
```

It is to control the ascending speed of voltage regulating. Normally no need to change as it is already the optimal value.

# XZ321X Voltage Regulator

## Driver

The driver files location:

```
drivers/regulator/xz3216.c
```

## DTS node

```
xz3216: xz3216@60 {
    compatible = "xz3216";
    reg = <0x60>;
    status = "okay";
    regulators {
        #address-cells = <1>;
        #size-cells = <0>;
        xz3216_dc1: regulator@0 {
            reg = <0>;
            regulator-compatible = "xz_dcdc1";
            regulator-name = "vdd_cpu_1";
            regulator-min-microvolt = <712500>;
            regulator-max-microvolt = <1400000>;
            regulator-always-on;
            regulator-boot-on;
            regulator-initial-state = <3>;
            regulator-state-mem {
                regulator-off-in-suspend;
                regulator-suspend-microvolt = <1100000>;
            };
        };
    };
};
```

The parameter description:

### Regulator Parameter

```
regulator-name = "vdd_cpu_1";
```

The name of the output power, invoked for voltage regulating.

```
regulator-min-microvolt = <712500>;
regulator-max-microvolt = <1500000>;
```

The max and min values limited by software, it is not allowable to set the values out of the range.

```
regulator-always-on;
```

Whether the power always on or not. You can delete the attribute if need to manage the switch by yourself.

### Cpu Parameter

Pay attention to the changes for frequency and voltage regulating:

If it is used for CPU little core, also need to modify:

```
&cpu_10 {
    cpu-supply = <&xz3216_dc1>;
};
&cpu_11 {
    cpu-supply = <&xz3216_dc1>;
};
&cpu_12 {
    cpu-supply = <&xz3216_dc1>;
};
&cpu_13 {
    cpu-supply = <&xz3216_dc1>;
};
```

If it is used for CPU big core, also need to modify:

```
&cpu_b0 {
    cpu-supply = <&xz3216_dc1>;
};
&cpu_b1 {
    cpu-supply = <&xz3216_dc1>;
};
```

If it is used for GPU, also need to modify:

```
&gpu {
    status = "okay";
    mali-supply = <&xz3216_dc1>;
};
```

The configuration depends on the actual power supply situation of XZ3126.(configured according to the released hardware circuit by default)

## TCS452X Voltage Regulator

### Driver

The driver files location:

```
drivers/regulator/fan53555.c
```

### DTS node

```
vdd_cpu: tcs4525@1c {
    compatible = "tcs,tcs452x";
    reg = <0x1c>;
    vin-supply = <&vcc5v0_sys>;
    regulator-compatible = "fan53555-reg";
    regulator-name = "vdd_cpu";
    regulator-min-microvolt = <712500>;
    regulator-max-microvolt = <1390000>;
```

```
regulator-ramp-delay = <2300>;
fcs,suspend-voltage-selector = <1>;
regulator-boot-on;
regulator-always-on;
regulator-state-mem {
    regulator-off-in-suspend;
};
};
```

The parameter description:

### Supply Parameter

```
vin-supply = <&vcc5v0_sys>;
```

The hardware input voltage, no actual meaning, mainly used for constructing the power tree.

### Pinctrl Parameter

```
pinctrl-0 = <&vsel1_gpio>;/* may be not used */
vsel-gpios = <&gpio1 17 GPIO_ACTIVE_HIGH>;/* may be not used */
fcs,suspend-voltage-selector = <1>;
```

Pay attention to this:This IO is used to change two groups of different voltages, but currently it is used to quickly change the switch.

```
fcs,suspend-voltage-selector = <1>;
```

Enable voltage when VSEL pin is low, disable the voltage when it is high. IO is pulled down by default.

```
fcs,suspend-voltage-selector = <0>;
```

Enable voltage when VSEL pin is high, disable the voltage when it is low. IO is pulled up by default.

The value should match with the actual hardware.

### Note:

VSEL pin function can also be used to change voltage for sleep-resume instead of quickly changing the switch. Only need to delete:

```
pinctrl-0 = <&vsel1_gpio>;
vsel-gpios = <&gpio1 17 GPIO_ACTIVE_HIGH>;
```

Now VSEL pin is connected to pmic\_sleep. The function:

```
fcs,suspend-voltage-selector = <1>;
```

Output running voltage when VSEL pin is low and output standby voltage when it is high(also can set to off for standby). IO is pulled down by default.

```
fcs,suspend-voltage-selector = <0>;
```



Output running voltage when VSEL pin is high and output standby voltage when it is low(also can set to off for standby). IO is pulled up by default.

### Regulator Parameter

```
regulator-name = "vdd_cpu";
```

The name of the PWM output power, invoked for voltage regulating.

```
regulator-min-microvolt = <712500>;  
regulator-max-microvolt = <1390000>;
```

The max and min values limited by software, it is not allowable to set the values out of the range.

```
regulator-always-on;
```

Whether the power always on or not. You can delete the attribute if need to manage the switch by yourself.

```
regulator-boot-on;
```

Used in U-Boot to if need to set the power on in U-Boot stage.

```
regulator-ramp-delay = <2300>;
```

It is to control the ascending speed of voltage regulating. Normally no need to change as it is already the optimal value.

## RK860X Voltage Regulator

### RK860X

CHIP	ADDR	Volt Range	DEFAULT	Die Id
RK8600	0x40	0.7125~1.5V(12.5MV)	1.0V	0x8
RK8601	0x41	0.7125~1.5V(12.5MV)	1.0V	0x8
RK8602	0x42	0.5~1.5V(6.25MV)	0.8V	0xa
RK8603	0x43	0.5~1.5V(6.25MV)	0.8V	0xa

### Driver

The driver files location:

```
drivers/regulator/rk860x-regulator.c
```

### DTS node

```
vdd_cpu: rk8600@40{  
    compatible = "rockchip,rk860x";
```

```

reg = <0x40>;
vin-supply = <&vcc5v0_sys>;
regulator-compatible = "rk860x-reg";
regulator-name = "vdd_cpu";
regulator-min-microvolt = <712500>;
regulator-max-microvolt = <1390000>;
regulator-init-microvolt = <900000>;
regulator-ramp-delay = <2300>;
rockchip,suspend-voltage-selector = <1>;
regulator-boot-on;
regulator-always-on;
regulator-state-mem {
    regulator-off-in-suspend;
};
};

```

The parameter description:

### Supply Parameter

```
vin-supply = <&vcc5v0_sys>;
```

The hardware input voltage, no actual meaning, mainly used for constructing the power tree.

### Pinctrl Parameter

```

pinctrl-0 = <&vsel1_gpio>, <&en_gpio>;
vsel-gpios = <&gpio1 17 GPIO_ACTIVE_HIGH>;
en-gpios = <&gpio1 18 GPIO_ACTIVE_LOW>;
rockchip,suspend-voltage-selector = <1>;

```

Pay attention to this: This IO is used to change two groups of different voltages, but currently it is used to quickly change the switch.

```
rockchip,suspend-voltage-selector = <1>;
```

Enable voltage when VSEL pin is low, disable the voltage when it is high. IO is pulled down by default.

```
rockchip,suspend-voltage-selector = <0>;
```

Enable voltage when VSEL pin is high, disable the voltage when it is low. IO is pulled up by default.

The value should match with the actual hardware.

### Note:

VSEL pin function can also be used to change voltage for sleep-resume instead of quickly changing the switch. Only need to delete:

```

pinctrl-0 = <&vsel1_gpio>;
vsel-gpios = <&gpio1 17 GPIO_ACTIVE_HIGH>;

```

Now VSEL pin is connected to pmic\_sleep. The function:

```
rockchip,suspend-voltage-selector = <1>;
```

Output running voltage when VSEL pin is low and output standby voltage when it is high(also can set to off for standby). IO is pulled down by default.

```
rockchip,suspend-voltage-selector = <0>;
```

Output running voltage when VSEL pin is high and output standby voltage when it is low(also can set to off for standby). IO is pulled up by default.

EN pin function can also be used to power up:

```
pinctrl-0 = <&en_gpio>;  
en-gpios = <&gpio1 18 GPIO_ACTIVE_LOW>;
```

Now EN pin is output high in uboot.

### Regulator Parameter

```
regulator-name = "vdd_cpu";
```

The name of the output power, invoked for voltage regulating.

```
regulator-min-microvolt = <712500>;  
regulator-max-microvolt = <1500000>;
```

The max and min values limited by software, it is not allowable to set the values out of the range.

```
regulator-always-on;
```

Whether the power always on or not. You can delete the attribute if need to manage the switch by yourself.

```
regulator-boot-on;
```

Used in U-Boot to if need to set the power on in U-Boot stage.

```
regulator-ramp-delay = <1000>;
```

It is to control the ascending speed of voltage regulating. Normally no need to change as it is already the optimal value.

## DEBUG Interface

---

### Get Power Tree

```
cat /sys/kernel/debug/regulator/regulator_summary
```

### Set voltage

Set the voltage interface:

```
echo 1000000 > /sys/kernel/debug/regulator/vdd_cpu/voltage
```

Get the voltage interface:

```
cat /sys/kernel/debug/regulator/vdd_cpu/voltage
```