

# Rockchip Developer Guide Linux SARADC

---

文件标识: RK-KF-YF-079

发布版本: V1.0.0

日期: 2019-12-23

文件密级: 公开资料

---

## 免责声明

本文档按“现状”提供, 福州瑞芯微电子股份有限公司 (“本公司”, 下同) 不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

## 商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自所有者所有。

版权所有© 2019福州瑞芯微电子股份有限公司

超越合理使用范畴, 非经本公司书面许可, 任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部, 并不得以任何形式传播。

福州瑞芯微电子股份有限公司

Fuzhou Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: [www.rock-chips.com](http://www.rock-chips.com)

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: [[fae@rock-chips.com](mailto:fae@rock-chips.com)]

---

## 前言

SARADC是一个6通道10bit有效位的数模转换器, 当输入频率为13MHz, 转换速度为1MSPS。

## 产品版本

芯片名称	内核版本
ROCKCHIP 芯片	4.4/4.19

读者对象 本文档(本指南)主要适用于以下工程师: 技术支持工程师 软件开发工程师

## 修订记录

日期	版本	作者	修改说明
2019.12.23	V1.0	薛小明	初始发布

---

## Rockchip Developer Guide Linux SARADC

SARADC 驱动

    驱动文件

    DTS 节点配置

SARADC 使用

内核配置

SARADC常用接口

---

# SARADC 驱动

---

## 驱动文件

驱动文件所在位置: `drivers/iio/adc/rockchip_saradc.c`

## DTS 节点配置

DTS 配置参考文档为 `Documentation/devicetree/bindings/iio/adc/rockchip-saradc.txt`, 本文主要说明如下参数:

- `interrupts = <GIC_SPI 62 IRQ_TYPE_LEVEL_HIGH 0>;` 转换完成, 产生中断信号。
- `io-channel-cells = <1>;`  
必须为1, 详见iio-bindings.txt。
- `vref-supply = <&vccadc_ref>;` saradc值对应的参考电压, 需要根据具体的硬件环境设置, 最大为1.8V, 对应的saradc值为1024, 电压和adc值成线性关系。

# SARADC 使用

1. 依赖"iio"框架，需要初始化 `struct iio_dev` 结构体，具体请看 `rockchip_saradc_probe` 函数当中的 `indio_dev`，最后调用 `iio_device_register(indio_dev)` 注册 `indio_dev`，等待"input"框架使用。
2. 以"adc-key"为例，需要初始化 `struct input_polled_dev`，具体请看 `drivers/input/keyboard/adc-keys.c` 当中的 `adc_keys_probe` 函数，调用 `input_register_polled_device(poll_dev)`；将 `poll_dev` 注册进"input"框架。
3. 当使用 `getevent` 测试时候，假设 `adc-key` 为 `event0`，则 `getevent -s /dev/input/event0`，会有如下调用关系：`adc_keys_poll -> iio_read_channel_processed -> iio_channel_read -> chan->indio_dev -> info->read_raw(rockchip_saradc_read_raw) -> iio_convert_raw_to_processed_unlocked`

`rockchip_saradc_read_raw` 是重要函数，逐条分析：

1. `writel_relaxed(8, info->regs + SARADC_DLY_PU_SOC)`；设置power up到开始采样的间隔为8个sclk周期。
2. `writel(SARADC_CTRL_POWER_CTRL | (chan->channel & SARADC_CTRL_CHN_MASK) | SARADC_CTRL_IRQ_ENABLE, info->regs + SARADC_CTRL)`；a) "power up saradc" b) 设置采样通道 c) 使能中断，开始采样
3. `wait_for_completion_timeout(&info->completion, SARADC_TIMEOUT)` 等待saradc完成采样，并产生中断。
4. `*val = info->last_val`；将采样数据存放在val中。
5. 最后调用 `iio_convert_raw_to_processed_unlocked` 将采样数据转换成对应的电压值。

中断处理过程：`rockchip_saradc_isr` 函数：

1. `info->last_val = readl_relaxed(info->regs + SARADC_DATA)`；保存数据，提供给上面的第4步使用。
2. `writel_relaxed(0, info->regs + SARADC_CTRL)`；清中断，并且"power down saradc"，关闭saradc。

一个完整的采样过程是 `rockchip_saradc_read_raw` 配置saradc,打开saradc,开始采样，等待中断，中断函数中清除中断，关闭saradc。

## 内核配置

---

```
1 Symbol: ROCKCHIP_SARADC [=y]
2 Type : tristate
3 Prompt: Rockchip SARADC driver
4   Location:
5     -> Device Drivers
6       -> Industrial I/O support (IIO [=y])
7 (1)   -> Analog to digital converters
8   Defined at drivers/iio/adc/Kconfig:319
9   Depends on: IIO [=y] && (ARCH_ROCKCHIP [=y] || ARM && COMPILE_TEST [=n])
   &&
10  RESET_CONTROLLER [=y]
```

# SARADC常用接口

---

1.可以通过用户态接口获取adc值，其中\*表示adc第多少通道:

```
cat /sys/bus/iio/devices/iio\:device0/in_voltage*_raw
```

例如 channle0:

```
cat /sys/bus/iio/devices/iio\:device0/in_voltage0_raw
```

2.内核常用接口:

获取adc值: `iio_read_channel_raw()`

获取电压: `iio_read_channel_processed()`