Rockchip Developer Guide Linux SARADC

ID: RK-KF-YF-084

Release Version: V 1.0.0

Release Date: 2019-12-23

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

DISCLAIMER

THIS DOCUMENT IS PROVIDED “AS IS”. FUZHOU 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.

Fuzhou 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

SARADC is a 6-channel 10-bit effective digital-to-analog converter, when the input frequency is 13MHz, the conversion speed is 1MSPS

Product Version

Chipset	Kernel Version
ALL	4.4&4.19

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	Simon.Xue	2019-12-23	Initial version

Content

Rockchip Developer Guide Linux SARADC

1. SARADC Driver

The driver file is in:
`drivers/iio/adc/rockchip_saradc.c`

1.1 DTS Configuration

The reference document for DTS configuration is `Documentation/devicetree/bindings/iio/adc/rockchip-saradc.txt`, this document mainly introduces the following parameter:

- `interrupts = <GIC_SPI 62 IRQ_TYPE_LEVEL_HIGH 0>;`

  Finish conversion and generate interrupt signal.

- `io-channel-cells = <1>;`

  Here value must be 1, and the explanation refer to iio-bindings.txt

- `vref-supply = <&vccadc_ref>;`

  The reference voltage matched to the SARADC value needs to be set according to the specific hardware environment, the maximum is 1.8V, the corresponding SARADC value is 1024, and the voltage and the ADC value have a linear relationship.

2. Usage of SARADC

1. SARADC depends on the "iio" framework, you need to initialize the structure `struct iio_dev`, please see `indio_dev` of the `rockchip_saradc_probe` function, finally call `iio_device_register (indio_dev)` to register `indio_dev`, waiting for the "input" framework to use.
2. Take "adc-key" as an example, you need to initialize `struct input_polled_dev`, please see the `adc_keys_probe` function in `drivers/input/keyboard/adc-keys.c` for detail, call `input_register_polled_device (poll_dev);` and register to the" input "framework.
3. When using `getevent` test, assuming that `adc-key` is event0, then `getevent -s /dev/input/event0` will have the following calling relationship:
   `adc_keys_poll-> iio_read_channel_processed-> iio_channel_read-> rockchip_saradc_read_raw-> iio_convert_raw_to_processed_unlocked`

`rockchip_saradc_read_raw` is an important function, analyzed one by one:

1. `writel_relaxed(8, info->regs + SARADC_DLY_PU_SOC);`

   Set the interval from power up to start sampling to 8 sclk cycles.

2. `writel(SARADC_CTRL_POWER_CTRL | (chan->channel & SARADC_CTRL_CHN_MASK) | SARADC_CTRL_IRQ_ENABLE,info->regs + SARADC_CTRL);`

   1. Power up saradc
   2. Set the sampling channel
   3. Enable interrupt and start sampling

3. `wait_for_completion_timeout(&info->completion, SARADC_TIMEOUT)`

Wait for SARADC to complete sampling and generate an interrupt.

4. `*val = info->last_val;`

   Store the sampled data into val.

5. Convert the sampled data to the corresponding voltage value by calling `iio_convert_raw_to_processed_unlocked`.

Interrupt processing: function `rockchip_saradc_isr` :

1. `info->last_val = readl_relaxed(info->regs + SARADC_DATA);`

   Save the data for usage in step 4 above.

2. `writel_relaxed(0, info->regs + SARADC_CTRL);`
   Clear the interruption, and "power down saradc", shut down the SARADC.

A complete sampling process is `rockchip_saradc_read_raw` configure SARADC, open SARADC, start sampling, wait for interrupt, clear interrupt in interrupt function and finally close SARADC.

3. Kernel Configuration

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

4. Common Interfaces of SARADC

1.The ADC value can be obtained through the user mode interface, where * indicates the number of ADC channels:

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

For example, channle0:

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

2.Commonly used interfaces of the kernel:

Obtain ADC Value: `iio_read_channel_raw()`

Obtain Voltage Value: `iio_read_channel_processed()`