

Rockchip FLEXBUS ADC 和 DAC 模式开发指南

文件标识: RK-KF-YF-C11

发布版本: V1.0.0

日期: 2024-06-11

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

本文档按“现状”提供, 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 © 2024 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。

瑞芯微电子股份有限公司

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

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

前言

概述

本文档介绍了如何在 Linux 使用 FLEXBUS ADC 模式和 DAC 模式。

产品版本

芯片名称	内核版本
RK3576	6.1

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

版本号	作者	修改日期	修改说明
V1.0.0	姚旭伟	2024-06-11	初始版本

目录

Rockchip FLEXBUS ADC 和 DAC 模式开发指南

1. FLEXBUS ADC 模式
 - 1.1 概述
 - 1.2 配置
 - 1.2.1 硬件配置
 - 1.2.2 内核配置
 - 1.2.3 dtsi 配置
 - 1.2.4 驱动文件
 - 1.3 常用接口
 - 1.3.1 确认 FLEXBUS ADC 对应的 device
 - 1.3.2 获取 ADC 值
 - 1.3.3 获取和修改时钟频率
2. FLEXBUS DAC 模式
 - 2.1 概述
 - 2.2 配置
 - 2.2.1 硬件配置
 - 2.2.2 内核配置
 - 2.2.3 dtsi 配置
 - 2.2.4 驱动文件
 - 2.3 常用接口
 - 2.3.1 确认 FLEXBUS DAC 对应的 device
 - 2.3.2 向 DAC 发送数据
 - 2.3.3 获取和修改时钟频率

1. FLEXBUS ADC 模式

1.1 概述

FLEXBUS ADC 模式指的是 **FLEXBUS1** 对接高速并行 **ADC** 器件，例如 TI 的 ADS6144 (<https://www.ti.com.cn/product/cn/ADS6144>)、ADI 的 LTC2207 (<https://www.analog.com/cn/products/ltc2207.html>)。不支持 **SPI**、**I2C** 等接口类型的 **ADC** 器件。

- 时钟频率最高 100MHz
- 分辨率最高 16-Bit

1.2 配置

1.2.1 硬件配置

- 只支持将 ADC 器件的 LSB 连接到 FLEXBUS1_D0，例如 10-Bit ADC 连接至 FLEXBUS1_D[9:0]、16-Bit ADC 连接至 FLEXBUS1_D[15:0]
- slave 模式为器件向 FLEXBUS1 提供 CLK，master 模式为 FLEXBUS1 向器件提供 CLK

1.2.2 内核配置

FLEXBUS ADC 模式依赖 iio/adc 框架。

Device Drivers -> Multifunction device drivers -> Rockchip Flexbus

Device Drivers -> Industrial I/O support -> Analog to digital converters -> Rockchip Flexbus ADC opmode driver

1.2.3 dtsti 配置

以 RK3576 平台和 RK3576 TEST1 板子，FLEXBUS1 对接 ADC 为例。

rk3576.dtsi 中：

```
flexbus: flexbus@2a2f0000 {
    .....

    flexbus_adc: adc {
        compatible = "rockchip,flexbus-adc";
        #io-channel-cells = <0>;
        rockchip,slave-mode;    // 配置 slave 模式，不配置则为 master 模式
        rockchip,free-sclk;    // 配置时钟保持输出，不配置则时钟跟随数据
        rockchip,auto-pad;    // 默认配置
        rockchip,cpol;    // 配置 CPOL = 1，不配置则为 0
        rockchip,cpha;    // 配置 CPHA = 1，不配置则为 0
        rockchip,dfs = <16>;    // data frames
```

```

        status = "disabled";
    };

    flexbus_dac: dac {
        .....
    };
};

```

- rockchip,slave-mode: 配置了为 slave 模式为器件向 FLEXBUS1 提供 CLK；不配置为 master 模式，FLEXBUS1 向器件提供 CLK
- rockchip,free-sclk: 配置了则时钟一直保持输出，不配置则只在数据传输时输出时钟（该配置仅在 master 模式有效）
- rockchip,cpol、rockchip,cpha: 与 SPI 协议的定义相同，根据器件手册的时序图配置
- rockchip,dfs: 数据的位宽，仅支持配置成 4、8、16；如果器件分辨率不是 4、8 或 16，需要向上取整；例如对接 14-Bit 分辨率的 ADC 器件时，需要把 rockchip,dfs 配成 16，接收数据时通过软件处理只取 14bit 的数据

arch/arm64/boot/dts/rockchip/rk3576-test1.dtsi 中：

```

&flexbus {
    rockchip,flexbus0-opmode = <ROCKCHIP_FLEXBUS0_OPMODE_XXX>;
    rockchip,flexbus1-opmode = <ROCKCHIP_FLEXBUS1_OPMODE_ADC>; // FLEXBUS1 选择
ADC 模式
    status = "okay"; // 使能 FLEXBUS
};

&flexbus_adc {
    pinctrl-names = "default";
    pinctrl-0 = <&flexbus1m4_csn &flexbus1_clk
        &flexbus1_d0 &flexbus1_d1 &flexbus1_d2 &flexbus1_d3
        &flexbus1_d4 &flexbus1_d5 &flexbus1_d6 &flexbus1_d7
        &flexbus1_d8 &flexbus1_d9 &flexbus1_d10 &flexbus1_d11
        &flexbus1m1_d12 &flexbus1m1_d13 &flexbus1m1_d14 &flexbus1m1_d15>;
    // 配置 FLEXBUS1 ADC 模式需要的 IOMUX
    status = "okay"; // 使能 ADC 模式
};

```

1.2.4 驱动文件

驱动文件为 drivers/iio/adc/rockchip-flexbus-adc.c。

其中 rockchip_flexbus_adc_read_block() 是 read ADC 器件数据的函数，主要操作有：

1. rockchip_flexbus_writel(rkfb, FLEXBUS_RX_NUM, num_of_dfs);
配置 RX 数量，单位是 dfs（dtsi 中的 rockchip,dfs）
2. rockchip_flexbus_writel(rkfb, FLEXBUS_DMA_DST_ADDR0, (ulong)dst_phys >> 2);
配置 dst buffer 的物理地址
3. rockchip_flexbus_writel(rkfb, FLEXBUS_DMA_DST_LEN0, dst_len);
配置 dst buffer 的长度
4. rockchip_flexbus_writel(rkfb, FLEXBUS_ENR, FLEXBUS_RX_ENR);
Enable RX 传输

5. wait_for_completion_timeout(&rkfb_adc->completion, FLEXBUS_ADC_TIMEOUT)

等待 RX 传输完成，产生中断（中断处理函数为 rockchip_flexbus_adc_isr()）

6. rockchip_flexbus_writel(rkfb, FLEXBUS_ENR, FLEXBUS_RX_DIS);

Disable RX 传输

1.3 常用接口

1.3.1 确认 FLEXBUS ADC 对应的 device

例如：

```
root@rk3576-buildroot:/# cat /sys/bus/iio/devices/iio\:device0/name
2a2f0000.flexbus:adc
```

说明 FLEXBUS ADC 对应 iio:device0

1.3.2 获取 ADC 值

```
root@rk3576-buildroot:/# cd /sys/bus/iio/devices/iio\:device0
root@rk3576-buildroot:/sys/bus/iio/devices/iio:device0# cat in_voltage_raw
33004
```

1.3.3 获取和修改时钟频率

获取时钟频率：

```
root@rk3576-buildroot:/# cd /sys/bus/iio/devices/iio\:device0
root@rk3576-buildroot:/sys/bus/iio/devices/iio:device0# cat
in_voltage_sampling_frequency
99000000
```

说明当前时钟频率为 99MHz。

修改时钟频率：

```
root@rk3576-buildroot:/sys/bus/iio/devices/iio:device0# echo 25000000 >
in_voltage_sampling_frequency
```

时钟频率就被改为 25MHz。

Note:

- 只有 master 模式支持获取和修改时钟频率，slave 模式的 CLK 来源于 ADC 器件
- 时钟频率最高 100MHz

2. FLEXBUS DAC 模式

2.1 概述

FLEXBUS DAC 模式指的是 **FLEXBUS0** 对接高速并行 **DAC** 器件，例如 ADI 的 MAX5885 (<https://www.analog.com/cn/products/max5885.html>)、ADI 的 AD9744 (<https://www.analog.com/cn/products/ad9744.html>)。不支持 **SPI**、**I2C** 等接口类型的 **DAC** 器件。

- 时钟频率最高 100MHz
- 分辨率最高 16-Bit

2.2 配置

2.2.1 硬件配置

- 只支持将 DAC 器件的 LSB 连接到 FLEXBUS0_D0，例如 10-Bit DAC 连接至 FLEXBUS0_D[9:0]、16-Bit DAC 连接至 FLEXBUS0_D[15:0]

2.2.2 内核配置

FLEXBUS DAC 模式依赖 iio/dac 框架。

Device Drivers -> Multifunction device drivers -> Rockchip Flexbus

Device Drivers -> Industrial I/O support -> Digital to analog converters -> Rockchip Flexbus DAC opmode driver

2.2.3 dtsti 配置

以 RK3576 平台和 RK3576 TEST1 板子，FLEXBUS0 对接 DAC 为例。

rk3576.dtsi 中：

```
flexbus: flexbus@2a2f0000 {
    .....

    flexbus_adc: adc {
        .....
    };

    flexbus_dac: dac {
        compatible = "rockchip,flexbus-dac";
        #io-channel-cells = <0>;
        rockchip,free-sclk; // 配置时钟保持输出，不配置则时钟跟随数据
        rockchip,cpol; // 配置 CPOL = 1，不配置则为 0
        rockchip,cpha; // 配置 CPHA = 1，不配置则为 0
        rockchip,dfs = <16>; // data frames
        status = "disabled";
    };
};
```

```
};
```

- rockchip,free-sclk: 配置了则时钟一直保持输出，不配置则只在数据传输时输出时钟
- rockchip,cpol、rockchip,cpha: 与 SPI 协议的定义相同，根据器件手册的时序图配置
- rockchip,dfs: 数据的位宽，仅支持配置成 4、8、16；如果器件分辨率不是 4、8 或 16，需要向上取整；例如对接 14-Bit 分辨率的 DAC 器件时，需要把 rockchip,dfs 配成 16，发送数据的 bit15、14 补 0

arch/arm64/boot/dts/rockchip/rk3576-test1.dtsi 中:

```
&flexbus {
    rockchip,flexbus0-opmode = <ROCKCHIP_FLEXBUS0_OPMODE_DAC>; // FLEXBUS0 选择
DAC 模式
    rockchip,flexbus1-opmode = <ROCKCHIP_FLEXBUS1_OPMODE_XXX>;
    status = "okay"; // 使能 FLEXBUS
};

&flexbus_dac {
    pinctrl-names = "default";
    pinctrl-0 = <&flexbus0m4_csn &flexbus0_clk
                &flexbus0_d0 &flexbus0_d1 &flexbus0_d2 &flexbus0_d3
                &flexbus0_d4 &flexbus0_d5 &flexbus0_d6 &flexbus0_d7
                &flexbus0_d8 &flexbus0_d9 &flexbus0_d10 &flexbus0_d11
                &flexbus0_d12 &flexbus0m0_d13 &flexbus0m0_d14 &flexbus0m0_d15>;
    // 配置 FLEXBUS0 DAC 模式需要的 IOMUX
    status = "okay"; // 使能 DAC 模式
};
```

2.2.4 驱动文件

驱动文件为 drivers/iio/dac/rockchip-flexbus-dac.c。

其中 rockchip_flexbus_dac_write_block() 是向 DAC 器件发送数据的函数，主要操作有：

1. rockchip_flexbus_writel(rkfb, FLEXBUS_TX_NUM, num_of_dfs);
配置 TX 数量，单位是 dfs（dtsi 中的 rockchip,dfs）
2. rockchip_flexbus_writel(rkfb, FLEXBUS_TXWAT_START, val);
配置水线
3. rockchip_flexbus_writel(rkfb, FLEXBUS_DMA_SRC_ADDR0, (ulong)src_phys >> 2);
配置 src buffer 的物理地址
4. rockchip_flexbus_writel(rkfb, FLEXBUS_DMA_SRC_LEN0, src_len);
配置 src buffer 的长度
5. rockchip_flexbus_writel(rkfb, FLEXBUS_ENR, FLEXBUS_TX_ENR);
Enable TX 传输
6. wait_for_completion_timeout(&rkfb_dac->completion, FLEXBUS_DAC_TIMEOUT)
等待 TX 传输完成，产生中断（中断处理函数为 rockchip_flexbus_dac_isr()）
7. rockchip_flexbus_writel(rkfb, FLEXBUS_ENR, FLEXBUS_TX_DIS);
Disable TX 传输

2.3 常用接口

2.3.1 确认 FLEXBUS DAC 对应的 device

例如：

```
root@rk3576-buildroot:/# cat /sys/bus/iio/devices/iio\:device2/name
2a2f0000.flexbus:dac
```

说明 FLEXBUS DAC 对应 iio:device2

2.3.2 向 DAC 发送数据

```
root@rk3576-buildroot:/# cd /sys/bus/iio/devices/iio\:device0
root@rk3576-buildroot:/sys/bus/iio/devices/iio:device2# echo 16383 >
out_voltage_raw
```

2.3.3 获取和修改时钟频率

获取时钟频率：

```
root@rk3576-buildroot:/# cd /sys/bus/iio/devices/iio\:device2
root@rk3576-buildroot:/sys/bus/iio/devices/iio:device2# cat
out_voltage_sampling_frequency
99000000
```

说明当前时钟频率为 99MHz。

修改时钟频率：

```
root@rk3576-buildroot:/sys/bus/iio/devices/iio:device2# echo 25000000 >
out_voltage_sampling_frequency
```

时钟频率就被改为 25MHz。

Note:

- 时钟频率最高 100MHz