Rockchip RT-Thread SPI2APB

文件标识：RK-KF-YF-498

发布版本：V1.0.0

日期：2024-03-28

文件密级：□绝密 □秘密 □内部资料 ■公开

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

本文主要描述了 ROCKCHIP RT-Thread SPI 驱动的使用方法。

产品版本

芯片名称	内核版本
所有使用 RK RT-Thread SDK 的芯片产品	RT-Thread

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
2024-03-28	V1.0.0	林鼎强	初始版本

目录

1. Rockchip SPI2PB 功能特点

1.1 支持特性

- 支持2种SPI模式，SPI mode 0/2，推荐使用 mode 0
- 支持8bits 传输
- 支持中断
- 支持读写控制器内部寄存器

SPI master 应遵从 SPI2APB 协议发起特定数据的传输，协议基础框架 `<CMD-32bits> [ADDR-32bits] [DUMMY-32bits] [READ_BEGING-32bits] [DATA]`

支持以下具体数据传输协议：

简介	协议简称	协议
读数据	read data	`<0x00000077> <ADDR> <DUMMY> <READ_BEGING> <DATA>`
写数据	write data	`<0x00000011> <ADDR> <DATA>`
写控制器内部 REG0 寄存器	write msg reg0	`<0x00010011> <DATA-32bits>`
写控制器内部 REG1 寄存器，触发中断	write msg reg1	`<0x00020011> <DATA-32bits>`
读取前一笔传输状态	query state	`<0x000000ff> <DATA-32bits>`
读控制器内部 REG2 寄存器	query msg reg2	`<0x000001ff> <DATA-32bits>`

2. 软件

2.1 代码路径

框架代码：

```
components/drivers/include/drivers/spi.h
components/drivers/spi/spi_core.c
components/drivers/spi/spi_dev.c
```

驱动适配层：

```
bsp/rockchip/common/drivers/drv_spi2apb.c
bsp/rockchip/common/hal/lib/hal/src/hal_spi2apb.c
```

测试命令，用户程序完全可以参照以下驱动：

```
bsp/rockchip-common/tests/spi2apb_test.c
```

2.2 配置

- 打开配置带有 SPI2APB 字样的选项

```
RT-Thread bsp drivers  --->
    RT-Thread rockchip "project" drivers  --->
        [*] Enable SPI
        [*]   Enable SPI0 (SPI2APB)
        [ ]   Enable SPI1
        [ ]   Enable SPI2
```

- 板级配置 iomux。

2.3 SPI2APB 使用配置

默认配置：

- LSB、Little Endian

drv_spi2apb.c 源码修改配置参考：

```
diff --git a/bsp/rockchip/common/drivers/drv_spi2apb.c
b/bsp/rockchip/common/drivers/drv_spi2apb.c
index 15045270fe..ef055d8d03 100644
--- a/bsp/rockchip/common/drivers/drv_spi2apb.c
+++ b/bsp/rockchip/common/drivers/drv_spi2apb.c
@@ -45,7 +45,7 @@ struct rockchip_spi2apb

 static struct rockchip_spi2apb rk_spi2apb =
 {
-    .device.config.mode = (SPI2APB_LITTLE_ENDIAN | SPI2APB_LSB),
+    .device.config.mode = (SPI2APB_LITTLE_ENDIAN | SPI2APB_MSB),
     .device.config.clock_polarity = SPI2APB_TXCP_INVERT,
     .base = SPI2APB,
     .irq = SPISLV0_IRQn,
```

说明：

- 建议只针对 LSB/MSB 做修改，其余配置默认配置

3. SPI2APB 测试

3.1 配置

```
RT-Thread bsp test case  --->
    [*] RT-Thread Common Test case  --->
      [*] Enable BSP Common TEST
    [*]  Enable BSP Common SPI2APB TEST
```

3.2 测试命令

```
spi2apb_test config spi2apb 1 1          # <mode> <polarity>
spi2apb_test cb spi2apb                  # 注册 SPI2APB->REG1 中断回调
spi2apb_test read spi2apb 0              # 读取 SPI2APB->REG0 信息，建议调试方式，先主设备端参考
'write msg reg0' 协议写，后从设备端执行该命令读
spi2apb_test read spi2apb 1              # 读取 SPI2APB->REG1 信息，建议调试方式，先主端参考
'write msg reg1' 协议写，后从端执行该命令读
spi2apb_test query spi2apb               # 读取 SPI2APB->SR 信息
spi2apb_test write spi2apb 0x12345678    # 写入 SPI2APB->REG2 信息，建议调试方式，先从端执行该命
令，后主端参考 'query msg reg2' 协议读
```

说明：

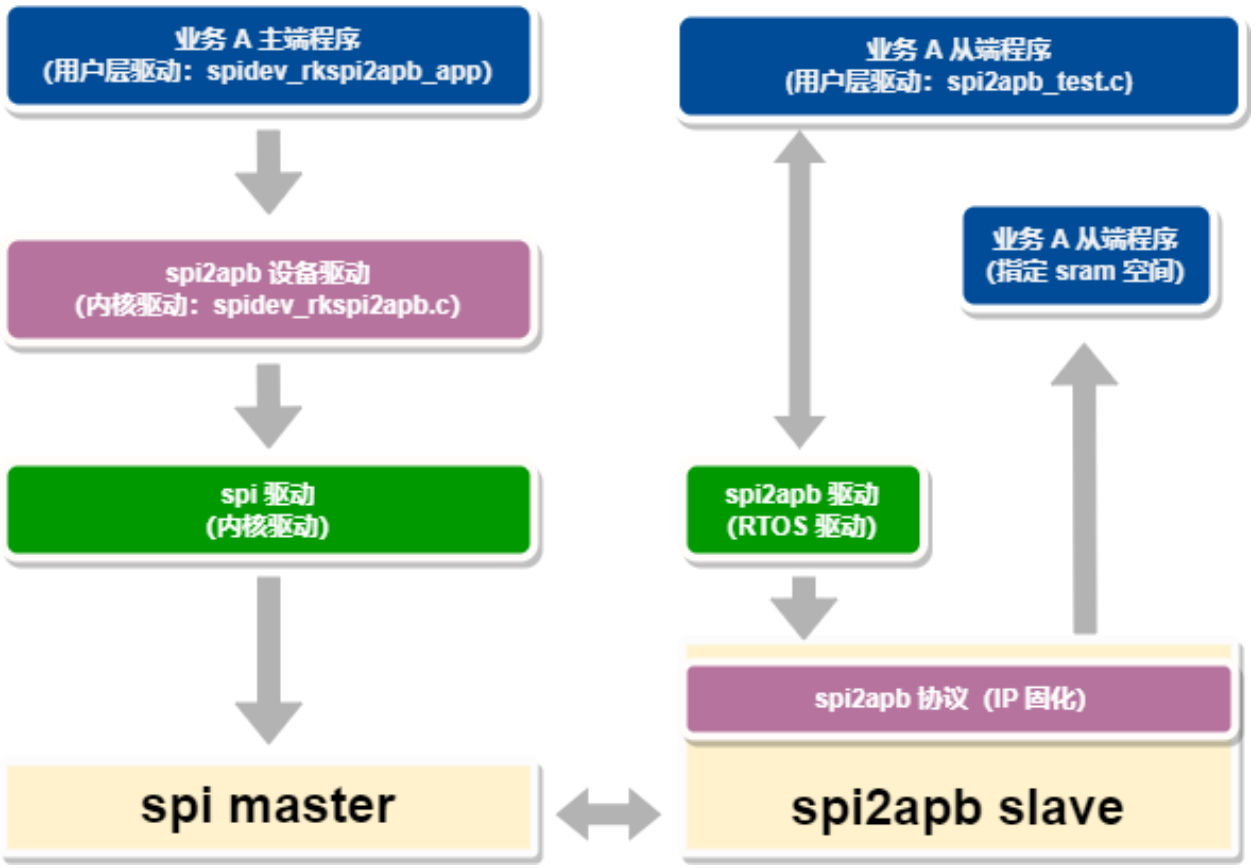
- 主设备端 - SPI master，从端 - RK SPI2APB slave

- mode，支持可配：
  - bit0：0-LSB 1-MSB
  - bit1：0-LittleEndian 1-BigEndian
- polarity，支持可配（仅做调试使用，实际请保持 TX invert/RX nornal 的默认配置）：
  - bit0：0-RX normal 1-RX invert
  - bit1：0-TX normal 1-TX invert

4. SPI2APB 业务搭建说明

SPI2APB 业务要求主从两端基于 SPI2APB 协议原理搭建数据流、控制信息等业务行为，其中运行在 SPI master 主设备端的业务代码、设备驱动可以基于 RK 参考代码进一步开发，源码获取：SPI开发资料 - FAE 项目 - Rockchip Redmine (rock-chips.com)

4.1 业务基础原理



以 RV1106 SPI master 对接 RK2118 SPI2APB 为例，从源码认识业务实现：

所属层次	主设备（Linux 5.10）	从设备（HAL + RTOS）
用户层业务	spidev_rkspi2apb_app c 程序，调用 spi2apb 设备节点的 ioctl/read/write 接口	test_spi2apb.c 程序，获取 REG 寄存器信息
内核态设备驱动	spidev_rkspi2apb.c 源码，依旧 spi2apb 协议实现 ioctl/read/write 接口	
控制器驱动	spi-rockchip.c 源码	drv_spi2apb.c hal_spi2apb.c
控制器	spi master	spi2apb slave

说明：

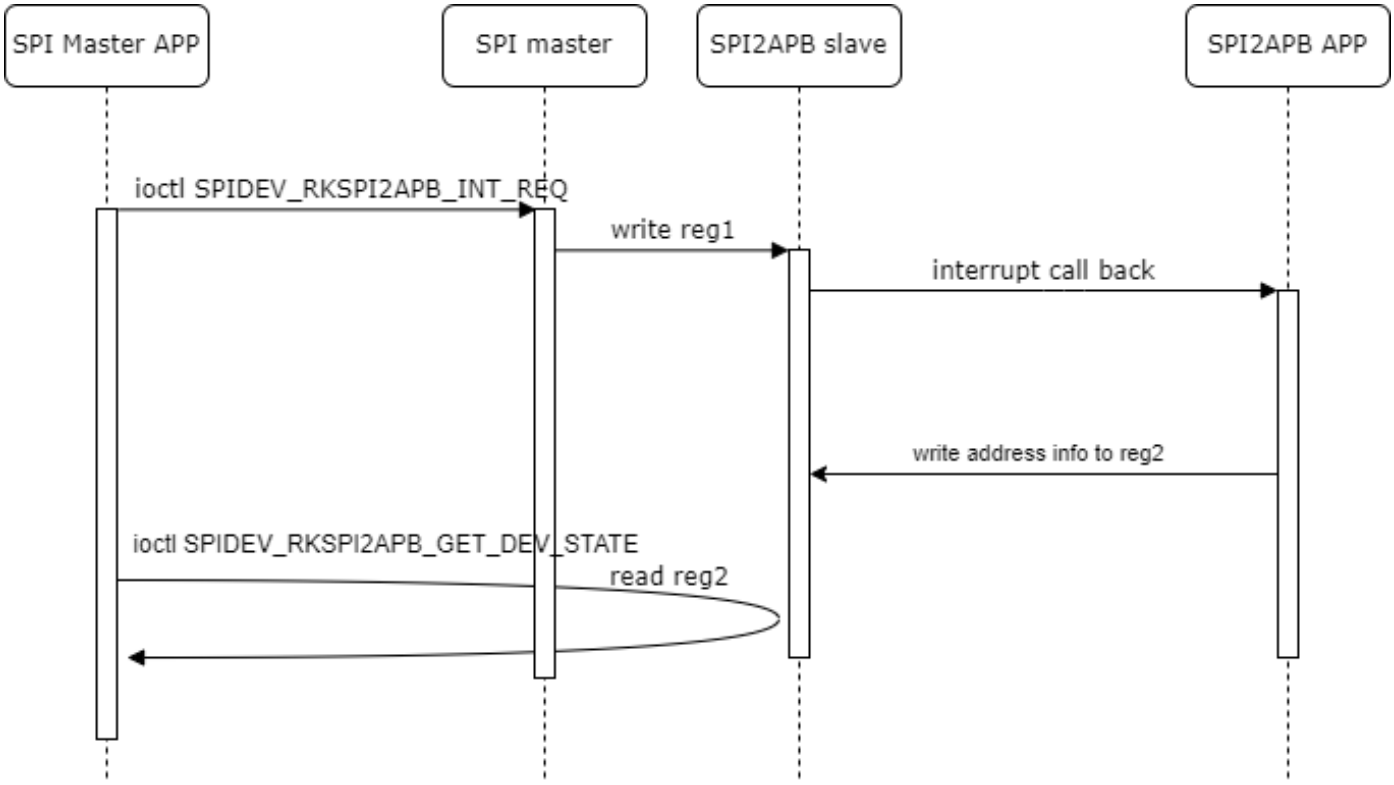
- spidev_rkspi2apb.c ioctl 接口说明：

```
#define SPIDEV_RKSPI2APB_GET_DEV_STATE    _IOR(SPIDEV_RKSPI2APB_BASE, 0, __u32) /* 参考 'query msg reg2' 协议 */
#define SPIDEV_RKSPI2APB_SET_DST_ADDR   _IOW(SPIDEV_RKSPI2APB_BASE, 1, __u32) /* 设置传输的目标内存地址，指向从设备端内存地址 */
#define SPIDEV_RKSPI2APB_SEND_MSG   _IOW(SPIDEV_RKSPI2APB_BASE, 2, __u32) /* 参考 'write msg reg0' 协议 */
#define SPIDEV_RKSPI2APB_INT_REQ    _IOW(SPIDEV_RKSPI2APB_BASE, 3, __u32) /* 参考 'write msg reg1' 协议 */
```

- spidev_rkspi2apbc.c 读写说明：

```
write(spidev_fd, buffer_w, size); /* 参考 'write data' 协议，addr 由 ioctl SPIDEV_RKSPI2APB_SET_DST_ADDR 设定 */
read(spidev_fd, buffer_r, size); /* 参考 'read data' 协议，addr 由 ioctl SPIDEV_RKSPI2APB_SET_DST_ADDR 设定 */
```

- 如何协定 SPI2APB 传输的目标内存地址：
  - 预设定（推荐使用该方案），通常需要在从设备预留一块已知的目标 sram 地址作为传输地址，主设备端使用 ioctl SPIDEV_RKSPI2APB_SET_DST_ADDR 设定
  - 自定义，从设备端申请一段 sram 地址空间，两端自定义业务，将地址信息最终传输到主设备端，例如以下参考流程：


SPI Master APP
SPI master
SPI2APB slave
SPI2APB APP
ioctl SPIDEV_RKSPI2APB_INT_REQ
write reg1
interrupt call back
write address info to reg2
ioctl SPIDEV_RKSPI2APB_GET_DEV_STATE
read reg2