Rockchip RT-Thread SPI

文件标识：RK-KF-YF-093

发布版本：V2.0.0

日期：2024-03-04

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
2019-07-13	V1.0.0	赵仪峰	初始发布
2020-05-27	V1.0.1	赵仪峰	修订格式
2024-03-04	V2.0.0	林鼎强	更新 RK2118 支持

目录

1. Rockchip SPI 功能特点

SPI（Serial Peripheral Interface）

- 支持4种SPI模式
- 支持2个片选
- 支持8bits 和 16bits 传输
- 支持中断传输模式和DMA传输模式
- FIFO深度 32级或64级
- 数据采样时钟RXD可配置
- 支持 slave mode，部分芯片支持 slave mode 外部时钟采样

1.1 SPI 接口特性

SOC	Master Mode 接口最高速率	Slave Mode 接口最高速率
RK2108	50MHz	20MHz
RK2118	50MHz	50MHz

说明：

- 接口最高速率为理论速率，受设备走线 PCB 质量影响，以实测为准
- 部分平台由于 PLL 策略原因无法准确分频到上限值，实际以最大分频值为准

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
components/drivers/spi/qspi_core.c
```

驱动适配层：

```
bsp/rockchip-common/drivers/drv_spi.c
bsp/rockchip-common/drivers/drv_spi.h
```

测试命令，用户程序完全可以参照以下驱动：

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

2.2 配置

- 打开配置，同时会生成/dev/spi0..2设备。

```
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 SPI 使用配置

SPI控制器作为MASTER时可以支持0-50MHz（个别平台可以配置更高频率），作为SLAVE时可以支持0-20Mhz。

框架提供的配置函数 rt_spi_configure() 可以配置频率、模式和传输位宽等。

SPI支持4种模式，具体使用哪种模式，参考设备手册.

4种模式定义如下：

```
#define RT_SPI_MODE_0       (0 | 0)                         /* CPOL = 0, CPHA = 0 */
#define RT_SPI_MODE_1       (0 | RT_SPI_CPHA)               /* CPOL = 0, CPHA = 1 */
#define RT_SPI_MODE_2       (RT_SPI_CPOL | 0)               /* CPOL = 1, CPHA = 0 */
#define RT_SPI_MODE_3       (RT_SPI_CPOL | RT_SPI_CPHA)     /* CPOL = 1, CPHA = 1 */
```

配置代码示例：

```
struct rt_spi_configuration cfg;

cfg.data_width = 8; /* 配置8bits传输模式 */
cfg.mode = RT_SPI_MASTER | RT_SPI_MSB | RT_SPI_MODE_0;
cfg.max_hz = 20 * 1000 * 1000; /* 配置频率 20Mhz */
rt_spi_configure(spi_device, &cfg); /* 配置 SPI*/
```

3. SPI测试

3.1 配置

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

3.2 测试命令

```
msh >spi_test
1. spi_test config <spi_device> <is_slave> <spi_mode> <is_msb> <speed>
2. spi_test read   <spi_device> <loops> <size>
3. spi_test write  <spi_device> <loops> <size>
4. spi_test duplex <spi_device> <loops> <size>
5. spi_test cs     <spi_device> <cs_state:1-take, 0-release>
6. spi_test bus    <spi_device> <bus_state:1-take, 0-release>
7. spi_test rate   <spi_device>
8. spi_test loop_thread   <spi_device> <loops> <size> <gap_ms> <random_pattern>
like:
        spi_test config spi1_0 0 3 0 24000000
        spi_test read   spi1_0 1 256
        spi_test write  spi1_0 1 256
        spi_test duplex spi1_0 1 256
        spi_test cs     spi1_0 1
        spi_test bus    spi1_0 1
        spi_test rate   spi1_0
        spi_test loop_thread   spi1_0 1 256 10 1
```

说明：

- random_pattern：0-递增序列，1-随机序列

3.3 测试实例

3.3.1 回环通路

短接 MOSI/MISO 后使用全双工传输，配置 spi1_0、master_mode、spi_mode_3、lsb、24MHz

```
spi_test config spi1_0 0 3 0 24000000
spi_test duplex spi1_0 1 256
```

3.3.2 测速

配置 spi1_0、master_mode、spi_mode_3、lsb、50MHz

```
spi_test config spi1_0 0 3 0 50000000
spi_test rate   spi1_0
```

3.3.3 设备互联

spi 从设备守护进程

- 配置 spi2_0、slave_mode、spi_mode_3、lsb、50MHz
- 传输 10 loops、每次传输 256B、每次传输之间无延时、递增序列（要求对端也是递增序列）

```
spi_test config spi2_0 1 3 0 50000000
spi_test duplex_thread spi2_0 10 256 0 0
```

spi 主设备测试进程

- 配置 spi1_0、master_mode、spi_mode_3、lsb、50MHz
- 传输 10 loops、每次传输 256B、每次传输延时 10 ms 等待 slave ready、递增序列（要求对端也是递增序列）

```
spi_test config spi1_0 0 3 0 50000000
spi_test duplex_thread spi1_0 10 256 10 0
```

3.3.4 信号测试

测试要求:

- 互联 SPI 信号线，其中 MOSI 接 MOSI
- 共地
- 测试命令验证按照以下次序，先从后主

测试命令:

- spi 从设备守护进程
  - 配置 spi2_0、slave_mode、spi_mode_3、lsb、50MHz
  - 传输 100000000 loops、每次传输 4096B、每次传输之间无延时、随机序列（不做校验）

```
spi_test config spi2_0 1 3 0 50000000
spi_test duplex_thread spi2_0 100000000 4096 0 1
```

- spi 主设备测试进程

- 配置 spi1_0、master_mode、spi_mode_3、lsb、50MHz
- 传输 100000000 loops、每次传输 4096B、每次传输之间 10 ms 延时、随机序列（不做校验）

```
spi_test config spi1_0 0 3 0 50000000
spi_test duplex_thread spi1_0 100000000 4096 10 1
```

测试判断测试命令成功

- MSH 确认是否有两个 duplex 进程：

```
msh >list_thread
thread   pri  status      sp     stack size max used left tick  error
-------- ---  ------- ---------- ----------  ------  ---------- ---
spi_dupl   5  suspend 0x000000c4 0x00000400    50%   0x00000009 OK
spi_dupl   5  suspend 0x00000120 0x00000400    38%   0x00000009 OK
```

- 信号上确认 MOSI/MISO 同时有输出