

# SDIO开发指南

---

文件标识: RK-KF-YF-102

发布版本: V1.2.0

日期: 2020-06-28

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，由其各自拥有者所有。

## 版权所有© 2020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](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

概述

产品版本

芯片名称	RT Thread 版本
全部采用 RT Thread 的芯片	

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	作者	日期	修改说明
V1.0.0	林涛	2019-07-11	初始发布
V1.0.1	林涛	2019-09-03	更新文件路
V1.0.2	林涛	2020-02-21	更新磁盘挂载
V1.1.0	林涛	2020-03-13	更新格式化操作
V1.2.0	林涛	2020-06-28	新增板级配置

## SDIO开发指南

1. RT-Thread Rockchip SDIO 功能特点
2. 软件
  - 2.1 代码路径
  - 2.2 配置
  - 2.3 调试
  - 2.4 EMMC或者SD Card挂载

# 1. RT-Thread Rockchip SDIO 功能特点

---

SDIO (Secure Digital Input and Output)

- 兼容 SDIO/SD card/eMMC
- 仅支持 Highspeed 模式, 1/4/8 线宽
- 支持 DMA 传输模式

## 2. 软件

---

### 2.1 代码路径

SDIO 框架:

```
components/drivers/include/drivers/mmc*_*.h
components/drivers/sdio/block_dev.c SD卡、eMMC与RTT block层对接的代码
components/drivers/sdio/mmc*_core.c SD卡、SDIO、eMMC协议栈公共部分
components/drivers/sdio/sdio.c SDIO协议栈
components/drivers/sdio/sd.c SD卡协议栈
components/drivers/sdio/mmc.c eMMC协议栈
```

SDIO 驱动适配层:

```
bsp/rockchip/common/drivers/drv_sdio.c
bsp/rockchip/common/drivers/drv_sdio.h
```

### 2.2 配置

打开 SDIO 配置, 如果是 SD 卡和 eMMC, 会生成/dev/sdioX 设备; 如果是 SDIO 设备, 需要 SDIO 的 function driver(例如 wifi 驱动)里面调用 sdio\_register\_driver 后才会出现对应的 wifi 驱动节点

```
RT-Thread bsp drivers --->
  RT-Thread rockchip common drivers --->
    [*] Enable SDIO
```

```

RT-Thread components --->
  Device Drivers --->
    [*] Using SD/MMC device drivers
    (512) The stack size for sdio irq thread
    (15) The priority level value of sdio irq thread
    (1024) The stack size for mmcsd thread
    (22) The priority level value of mmcsd thread
    (16) mmcsd max partition
    [ ] Enable SDIO debug log output

```

如果需要修改线宽或者通信频率，请在板级文件中调整。freq\_max为最大运行频率，不得配置高于50MHz；flags为支持的模式，如果需要在支持8线，则改为MMCSD\_BUSWIDTH\_8，如果仅支持1线，则直接去掉MMCSD\_BUSWIDTH\_4 属性即可。

```

RT_WEAK struct rk_mmc_platform_data rk_mmc_table[] =
{

#ifdef RT_USING_SDIO0

    {
        .flags = MMCSD_BUSWIDTH_4 | MMCSD_MUTBLKWRITE | MMCSD_SUP_SDIO_IRQ |
MMCSD_SUP_HIGHSPEED,
        .irq = SDIO_IRQn,
        .base = SDIO_BASE,
        .clk_id = CLK_SDIO_PLL,
        .freq_min = 100000,
        .freq_max = 50000000,
        .control_id = 0,
    },

#endif

}

```

执行命令可以看到已经生成的串口设备：

```

msh >list_device
device          type          ref count
-----
sd0             Block Device    0
sd1             Block Device    0

```

## 2.3 调试

- 将 RT\_SDIO\_DEBUG 配置选上后，则会输出更多 SDIO/SD/EMMC 协议栈的执行流程信息。
- 将 bsp/rockchip-common/drivers/drv\_sdio.c 中的 RK\_MMC\_DBG 配置后，则会输出更多控制器驱动执行信息。
- 请务必在板级配置中设置好 IOMUX，IO 电源设置，卡电源设置，对应 gpio bank 的 io domain，以及确保控制器的输出时钟为偶数分频所得。
- 如果wifi驱动有多个线程同时对function进行读写，请使用mmcsd\_host\_lock和mmcsd\_host\_unlock接口进行互斥保护。

## 2.4 EMMC或者SD Card挂载

(1) 需要开启的DFS的支持与elmfatfs文件系统格式的支持

(2) 在shell中输入list\_device，即可看到 SD Card 或者EMMC注册成了块设备了

```
\ | /
- RT -      Thread Operating System
/ | \      4.0.2 build Jul 31 2019
2006 - 2019 Copyright by rt-thread team
msh />[I/app.card] sd card mount to '/'

msh />list_device
device          type          ref count
-----
spi20          SPI Device          0
sd0            Block Device        1
spi10          SPI Device          0
spi2           SPI Bus           0
spi1           SPI Bus           0
uart1          Character Device    2
pin            Miscellaneous Device 0
msh />
```

(3)需要在各自板级目录中的mnt.c文件配置挂载格式与路径, 以RK2108b为例, 将sd0块设备, 以elm文件系统格式, 挂载在“/sdcard目录”。

```
bsp/rockchip/rk2108/board/rk2108b-evb/mnt.c
const struct dfs_mount_tbl mount_table[] = {
    {PARTITION_ROOT, "/", "elm", 0, 0},
+   {"sd0", "/sdcard", "elm", 0, 0},
};
```

(4) 如果eMMC颗粒或者SD card未经过格式化, 开机后会出现如下log

```
=== name = pcm capture = 1 multi_dais = 0 mclkfs = 0
pcm_name = es8311c
=== name = es8311 capture = 1 multi_dais = 0 mclkfs = 256
[E/DFS] mount fs[elm] on /sdcard failed.

testing sleep 1s:
msh />rk_mmc_request,get response returns -2, cmd: 8
rk_mmc_request,get response returns -2, cmd: 5
rk_mmc_request,get response returns -2, cmd: 55
rk_mmc_request,get response returns -2, cmd: 55
rk_mmc_request,get response returns -2, cmd: 55
rk_mmc_request,get response returns -2, cmd: 55
[I/SDIO] emmc card capacity 30535680 KB.
found part[0], begin: 32256, size: 29.123GB
[I/SDIO] try to mount file system!
[E/DFS] open failed
[E/DFS] mount fs[elm] device[sd0] to /sdcard failed.
```

那么请开启config配置项中的RT\_USING\_DEVICE\_OPS 这个宏，并进行格式化操作。我们以将sd0块设备

格式化fat分区(即elm)为例，在shell中执行 `mkfs -t elm sd0`，成功后重启设备即可完成挂载。

(5) 如果出现如下异常log：“There is no space to mount this file system”

请修改config配置项中的DFS\_FILESYSTEMS\_MAX的限制，可以尝试调到3或者以上。

```
RT-Thread components --->
Device virtual file system --->
(2) The maximal number of mounted file system
```