SPI屏开发指南

文件标识：RK-KF-YF-351

发布版本：V1.0.0

日期：2020-03-27

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，福州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。

福州瑞芯微电子股份有限公司

Fuzhou 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

SPI屏是指仅通过SPI线传输显示数据和配置参数的屏幕，这种屏分辨率一般不高于QVGA(320x240)， 主控只需将显示Buffer发送至屏内的RAM中，屏有自刷新保持显示，使用SPI屏可以大大节省GPIO资源。

产品版本

芯片名称	内核版本
通用	RT-Thread 3.1.x

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	钟勇汪	2020-03-27	初始版本

目录

1 Rockchip SPI 屏接口特点

SPI屏幕通常分为两种类型：3线9bit 和 4线8bit。3线9bit屏的第一个bit代表数据（DATA）或者命令（CMD），而4线8bit屏，需要多使用一个D/CX脚来确定发送的是数据还是命令。

Rockchip SPI的不支持发送9bit数据，因此仅支持4线8bit屏。对于320x240 16bit屏来说，一帧数据量是320x240x16=1228800， 因此50Mhz的SPI屏对应的最高帧率40fps。

2 软件

2.1 代码路径

驱动代码：

```
bsp/rockchip/common/drivers/drv_spi_screen.c
```

SPI屏配置文件：

```
bsp/rockchip/common/drivers/panel_cfg/kgm281g44pvaa_panel_cfg.h
bsp/rockchip/common/drivers/panel_cfg/h20b1301a_panel_cfg.h
```

SPI屏测试代码：

```
bsp/rockchip/common/tests/spi_screen_test.c
```

2.2 编译配置

打开SPI屏的开关RT_USING_SPI_SCREEN：

```
RT-Thread rockchip rk2108 drivers --->
     [*] Enable Display
         Display Controller (Enable SPI Transfer)  --->
```

选中一个屏，如：

```
RT-Thread rockchip common drivers --->
     SPI Panel Type (KGM281G44PVAA SPI panel, resolution is 80x160)  --->
```

2.3 SPI 屏测试

使能SPI 屏测试程序：

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

SPI屏测试命令：

```
spi_screen_test
```

屏上会显示一张color bar测试图像。

3 配置

3.1 SPI 屏使用配置

RK2108支持SPI1_M0，SPI1_M1，SPI2_M0，SPI2_M1，另外SPI2还支持CS0和CS1，共计有6种接法。实际产品中，需要先确认自己的板子是哪个SPI口连接到屏上。在板级的iomux.c中调整SPI的配置函数，如连接的是SPI2_M0，可在board/rk2108_evb/iomux.c 中调用spi2_m0_iomux_config：

```
void rt_hw_iomux_config(void)
{
    ...
    spi2_m0_iomux_config();
    ...
}
```

spi2_m0_iomux_config函数的实现在bsp/rockchip/rk2108/board/common/iomux_base.c 中：

```
RT_UNUSED static void spi2_m0_iomux_config(void)
{
    HAL_PINCTRL_SetIOMUX(GPIO_BANK1,
                         GPIO_PIN_A0 |  // SPI_MST2_CS0_M0
                         GPIO_PIN_A1 |  // SPI_MST2_CLK_M0
                         GPIO_PIN_A2 |  // SPI_MST2_MISO_M0
                         GPIO_PIN_A3 |  // SPI_MST2_MOSI_M0
                         GPIO_PIN_A5,   // SPI_MST2_CS1
                         PIN_CONFIG_MUX_FUNC3);

    /* set SPI master 2 IOMUX selection to M0 */
    WRITE_REG_MASK_WE(GRF->SOC_CON5,
                      GRF_SOC_CON5_GRF_CON_SPIMST2_IOMUX_SEL_MASK,
                      (0 << GRF_SOC_CON5_GRF_CON_SPIMST2_IOMUX_SEL_SHIFT));

#ifdef RT_USING_SPI_SCREEN
    /*
     * set GPIO0_C4 to be GPIO function, it is used as the A0(DCX) pin of
     * the SPI screen
     */
    HAL_PINCTRL_SetIOMUX(GPIO_BANK0,
                         GPIO_PIN_C4,
                         PIN_CONFIG_MUX_FUNC0);
#endif
}
```

另外，在board/common/board_base.h文件中定义屏使用的SPI接口，CS脚，传输频率和GPIO口：

```
#ifdef RT_USING_SPI_SCREEN
#define HAL_SPI_PANEL_SCLK      50000000  // SPI频率设置为50Mhz
#define HAL_PANEL_SPI           "spi2_1"  // SPI2 的 cs1，具体是M0还是M1，有上面的iomux函数决定
#define GPIO_DCX_GPIO_PORT      GPIO0     // D/CX 脚使用的是GPIO0_C4，需要在spi2_m0_iomux_config 把此PIN设置为GPIO功能，即FUNC0
#define GPIO_DCX_GPIO_PIN_OUT   GPIO_PIN_C4
#endif
```

3.2 如何新增 SPI 屏配置文件

以 bsp/rockchip/common/drivers/panel_cfg/kgm281g44pvaa_panel_cfg.h 为例

```
#define RT_HW_SPI_SCREEN_XRES           80      /* 屏宽 80列 */
#define RT_HW_SPI_SCREEN_YRES           160     /* 屏高 160行 */
#define RT_HW_SPI_SCREEN_BPP            16      /* 屏的显示位数 */
#define RT_HW_SPI_SCREEN_BUS_FORMAT     RTGRAPHIC_PIXEL_FORMAT_RGB565 /* RT-Thread支持的显示格式 */

const static struct rockchip_cmd spi_screen_cmd_on[] =
{
    {0x00, 0x78, 0x01, {0x11}}, // 00：这是一条命令；78：发送后delay 120ms; 01：命令的个数是1；11：发送的内容
    ...
    {0x01, 0x00, 0x01, {0x05}}, // 01：这是一条数据；00：发送后不延时；01：数据个数是1；11：发送的内容
};
const static struct rockchip_cmd spi_screen_cmd_off[] =
{}
```

修改 bsp/rockchip/common/drivers/drv_panel_cfg.h， 加入新的屏配置文件。