FREQCHIP
富芮坤

AN2609
Application Note

DMA 链表实现双屏显示不同内容

简介

本文档介绍通过 DMA 链表实现双屏显示不同内容的方法，以两块 160×80 分辨率的屏幕为例，渲染后分别提取左右半屏的数据，并通过 DMA 输出到对应屏幕。

Freqchip Confidential

Version 1.0

目录

1. 概述 3
2. 系统实现说明 4
2.1. LVGL 分辨率配置 4
2.2. SPI 与 DMA 初始化 4
2.3. DMA 链表配置 5
2.4. LVGL 渲染刷新回调配置 6
3. 功能测试 7
4. 联系信息 8

Freqchip Confidential

修订版本

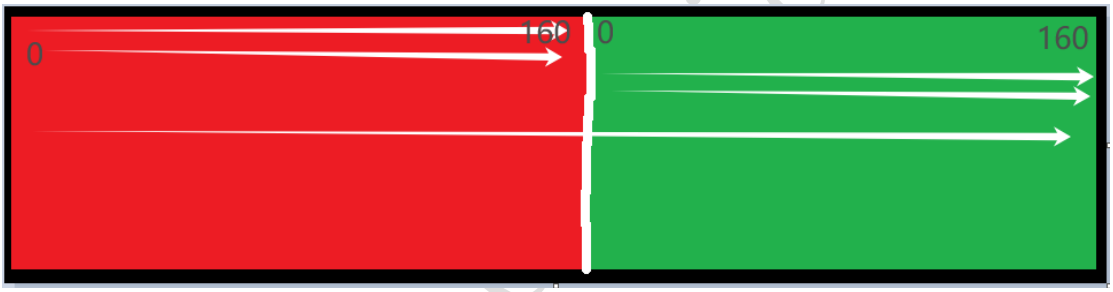
版本	日期	更新内容	修订人员
V1.0	2026.10.08	首版	刘浩

Freqchip Confidential

1. 概述

本文档说明通过 DMA 链表实现双屏显示不同内容的方法。以两块 160×80 分辨率的屏幕为例，将 LVGL 逻辑显示区域设置为 320×80，渲染后分别提取左右半屏的数据，并通过 DMA 输出到对应屏幕。

逻辑画面的左半区对应左屏，右半区对应右屏。每行包含 320 个像素，左右半屏各占 160 个像素；DMA 链表按行连接同一半屏的数据，实现从整幅渲染缓冲区中提取对应屏幕的画面。硬件传输共用一路硬件 SPI/QSPI，通过切换两个屏幕的 CS 脚来切换选中的屏幕。该方案将两块物理屏幕映射为一个逻辑屏幕，应用层只需按单屏方式进行界面 UI 制作，无需分别管理两块屏幕。底层驱动自动拆分左右半屏数据，通过 DMA 传输和 CS 切换，将对应内容发送到各屏幕。



以下依次给出 LVGL 分辨率、SPI 与 DMA 初始化、DMA 链表和刷新回调配置。

Freqchip Confidential

2. 系统实现说明

2.1. LVGL 分辨率配置

将 LVGL 水平分辨率设置为 320，垂直分辨率设置为 80。两个 160×80 显示区域在逻辑坐标中左右排列，应用可分别在左、右区域绘制不同内容。

```
#ifndef LV_CONF_H
#define LV_CONF_H

#include <stdint.h>

/* Maximal horizontal and vertical resolution to support by the library.*/
//#define LV_HOR_RES_MAX          (360)      //(320)
//#define LV_VER_RES_MAX          (360)      //(385)
#define LV_HOR_RES_MAX          (320)      //(320)
#define LV_VER_RES_MAX          (80)       //(385)
//#define LV_HOR_RES_MAX          (320)
//#define LV_VER_RES_MAX          (320)
/*=====================
```

2.2. SPI 与 DMA 初始化

先初始化 SPI 显示接口，再开启 DMA 时钟、配置 SPI 发送请求并初始化 DMA 通道，最后使能 DMA 中断。源地址递增、外设目标地址保持不变，具体传输位宽与请求配置见下图。

```
static void nv3030b_init_spi(void)
{
    __SYSTEM_SPI0_MASTER_CLK_SELECT_96M();
    __SYSTEM_SPI0_MASTER_CLK_ENABLE();

    nv3030b_SPI_Handle.SPIx                    = SPIM0;
    nv3030b_SPI_Handle.Init.Work_Mode          = SPI_WORK_MODE_0;
    nv3030b_SPI_Handle.Init.Frame_Size         = SPI_FRAME_SIZE_8BIT;
    nv3030b_SPI_Handle.Init.BaudRate_Prescaler = 2;
    nv3030b_SPI_Handle.Init.TxFIFOEmpty_Threshold = 15;
    nv3030b_SPI_Handle.Init.RxFIFOFull_Threshold  = 0;

    spi_master_init(&nv3030b_SPI_Handle);
}

static void nv3030b_init_dma(void)
{
    __SYSTEM_DMA_CLK_ENABLE();
    __DMA_REQ_ID_SPI0_MASTER_TX(1);

    nv3030b_DMA_Channel.Channel = DMA_Channel0;
    nv3030b_DMA_Channel.Init.Data_Flow        = DMA_M2P_DMAC;
    nv3030b_DMA_Channel.Init.Request_ID       = 1;
    nv3030b_DMA_Channel.Init.Source_Inc       = DMA_ADDR_INC_INC;
    nv3030b_DMA_Channel.Init.Desination_Inc   = DMA_ADDR_INC_NO_CHANGE;
    nv3030b_DMA_Channel.Init.Source_Width     = DMA_TRANSFER_WIDTH_32;
    nv3030b_DMA_Channel.Init.Desination_Width = DMA_TRANSFER_WIDTH_16;
    dma_init(&nv3030b_DMA_Channel);

    NVIC_EnableIRQ(DMA_IRQn);
}
```

Copyright© 2021 上海富芮坤微电子有限公司

2.3. DMA 链表配置

nv3030b_display_pro()根据 right_half 参数选择左右半屏。每行的起始像素偏移为 i×w 加上半屏偏移；right_half 为 true 时偏移 w/2，为 false 时偏移 0。每行建立一个 DMA 节点，下一节点指向下一行，最后一个节点结束链表。

```
void nv3030b_display_pro(uint32_t w, uint32_t h, uint16_t *data, bool right_half, void (*callback)(void))
{
    uint32_t i;
    uint32_t num_lines = h;
    uint32_t half_width = w / 2;

    if(dma_transfer_done == false)
    {
        return;
    }
    else
    {
        dma_transfer_done = false;
    }

    dma_trans_done_callback = callback;

    for (i = 0; i < num_lines; i++)
    {
        uint32_t offset = i * w + (right_half ? half_width : 0);
        LCD1_LinkParameter.SrcAddr          = (uint32_t)&data[offset];
        LCD1_LinkParameter.DstAddr          = (uint32_t)&nv3030b_SPI_Handle.SPIx->DR;
        LCD1_LinkParameter.NextLink         = (i == num_lines - 1) ? 0 : (uint32_t)&LCD1_Link_Channel[i + 1];
        LCD1_LinkParameter.Data_Flow        = DMA_M2P_DMAC;
        LCD1_LinkParameter.Request_ID       = nv3030b_DMA_Channel.Init.Request_ID;
        LCD1_LinkParameter.Source_Inc       = DMA_ADDR_INC_INC;
        LCD1_LinkParameter.Desination_Inc   = DMA_ADDR_INC_NO_CHANGE;
        LCD1_LinkParameter.Source_Width     = DMA_TRANSFER_WIDTH_32;
        LCD1_LinkParameter.Desination_Width = DMA_TRANSFER_WIDTH_8;
        LCD1_LinkParameter.Burst_Len        = DMA_BURST_LEN_4;
        LCD1_LinkParameter.Size             = half_width / 2;
        LCD1_LinkParameter.gather_enable    = 0;
        LCD1_LinkParameter.scatter_enable   = 0;

        dma_linked_list_init(&LCD1_Link_Channel[i], &LCD1_LinkParameter);
    }

    if(right_half) // 右边屏幕显示
    {
        nv3030b_set_window(0,half_width-1,0,h-1);
        LCD_SET_CS();

    }else{ //左边屏幕显示
        nv3030b_set_window(0,half_width-1,0,h-1);
        LCD_SET_CS();
    }
    LCD_RELEASE_DC();
    spi_master_transmit_X1_DMA(&nv3030b_SPI_Handle);
    dma_linked_list_start_IT(LCD1_Link_Channel, &LCD1_LinkParameter, nv3030b_DMA_Channel.Channel);
}
```

本例 w=320、h=80，half_width=160；每块屏幕使用 80 个行节点。链表将同一半屏各行的数据依次发送到 SPI 数据寄存器，并通过 LCD_SET_CS_0()和 LCD_SET_CS_1()选择对应屏幕。

图中的 Size=half_width/2 与源传输宽度 32 位、像素数据类型 uint16_t 配套使用，不能直接把 Size 理解为像素数。调整分辨率或像素格式时，应同步核对源地址偏移、节点长度和传输位宽。

Freqchip

2.4. LVGL 渲染刷新回调配置

在 my_disp_flush()中先调用 nv3030b_display_pro(320, 80, …, true, NULL)刷新右屏，等待传输完成后，再以 right_half=false 刷新左屏，并传入 my_disp_flush_done 作为完成回调。

```
static void my_disp_flush(lv_disp_drv_t * disp, const lv_area_t * area, lv_color_t * color_p)
{
    #if 1
    if(last_disp != NULL) {
        display_wait_transfer_done();
    }
    last_disp = disp;
    printf("flush1 x1:%d x2:%d y1:%d y2:%d buf:%x\r\n",area->x1,area->x2,area->y1,area->y2,(void *)color_p);
    //display_set_window(area->x1, area->x2, area->y1, area->y2);
    //display_update((area->x2+1-area->x1)*(area->y2+1-area->y1)/2, (void *)color_p, my_disp_flush_done);
    //display_set_window(area->x1, ((area->x2+1)/2)-1, area->y1, area->y2);
    uint32_t last_time = system_get_curr_time();
    nv3030b_display_pro(320,80,(void *)&color_p->full,true, NULL);//右边屏幕
    display_wait_transfer_done();
    printf("t:%d\r\n",system_get_curr_time()-last_time);
    nv3030b_display_pro(320,80,(void *)&color_p->full,false, my_disp_flush_done);//左边屏幕
    lv_disp_flush_ready(last_disp);
    #endif

}
```

两块屏幕共用渲染缓冲区，刷新需要按顺序进行；前一次 DMA 传输完成前，不应修改其正在读取的数据。刷新完成通知应与 DMA 完成时序保持一致，避免 LVGL 提前复用缓冲区。

本例按 320×80 整幅画面拆分左右半屏。若工程采用局部刷新，应同时处理 area、缓冲区跨度和半屏边界，不能直接套用固定宽高的地址计算。

Freqchip Confidential

3. 功能测试

1. 完成上述配置后编译工程并烧录到开发板，在 LVGL 左、右显示区域放置可区分的界面内容，确认两个屏幕分别显示对应半区。



2. 上图为双屏刷新效果，创建了两个圆弧控件分别放置到不同的位置显示效果。若出现内容错位，优先核对 320×80 逻辑分辨率、每行 320 像素跨度及左右半屏偏移；若出现刷新异常，核对 DMA 完成等待、片选切换和 LVGL 刷新完成通知。

Freqchip Confidential

4. 联系信息

公司：上海富芮坤微电子有限公司

地址：中国(上海)自由贸易试验区碧波路 912 弄 8 号 501-A 室

电话：+86-21-5027-0080

Website: www.freqchip.com

Sales Email: sales@freqchip.com

本文档的所有部分，其著作产权归上海富芮坤微电子有限公司（简称富芮坤）所有，未经富芮坤授权许可，任何个人及组织不得复制、转载、仿制本文档的全部或部分。富芮坤保留在不另行通知的情况下随时对产品或本文档进行更改、修正、增强的权利。购买者应在订购前获得富芮坤产品的最新相关资料。

Freqchip Confidential