N32H7xx系列GPU移植到LVGL使用指南

简介

此文档的目的在于让使用者能够快速熟悉 N32H7xx 系列微控制器（MCU）的 2.5D GPU 移植到 LVGL 的方法，用于指导用户更容易地使用 GPU 方式运行 LVGL 软件。

国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

目录

国民技术股份有限公司 NSING TECHNOLOGIES INC.

地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900　　传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

1. 概述

欢迎使用国民技术 N32H7xx 系列芯片，本文介绍了 N32H7xx 系列微控制器（MCU）的 2.5D GPU（简称为 NSGPU）移植到 LVGL 的方法，用于指导用户更容易地使用 GPU 方式运行 LVGL。

国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

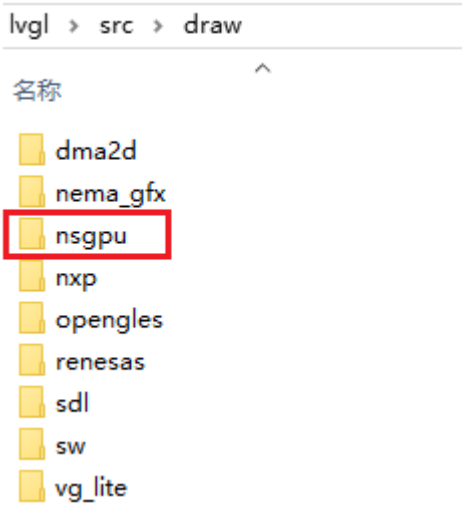
2. LVGL 移植指南

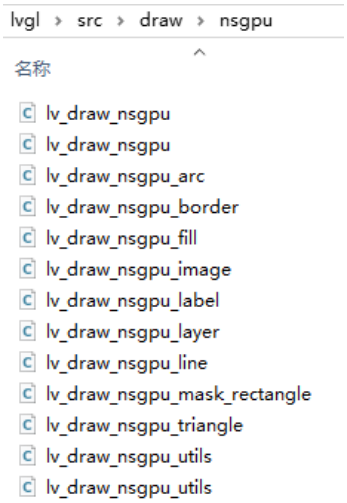
LVGL 默认为纯软件方式运行，若更改为 GPU 方式实现图形的渲染、数据的搬运等，将较大地提升系统运行效率。LVGL 移植指南介绍将 NSGPU 嵌入 LVGL 中，按照必要的配置，使用 GPU 方式运行 LVGL。使用的 LVGL 版本为 V9.3.0，其他版本可能出现 API 不兼容等问题。

2.1. 添加文件

2.1.1. LVGL 目录

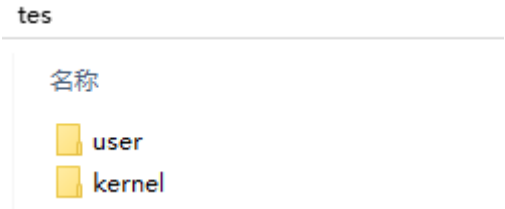
如下图所示，在 lvgl/src/draw 目录下加入 nsgpu 文件夹，其中包含了 nsgpu 与 LVGL 相关控件的衔接支持文件：





2.1.2. GPU 驱动

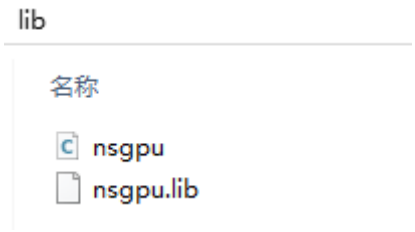
在 middlewares/gpu 目录下加入 tes 文件夹，包含 GPU 相关驱动。



2.1.3. LIB 库

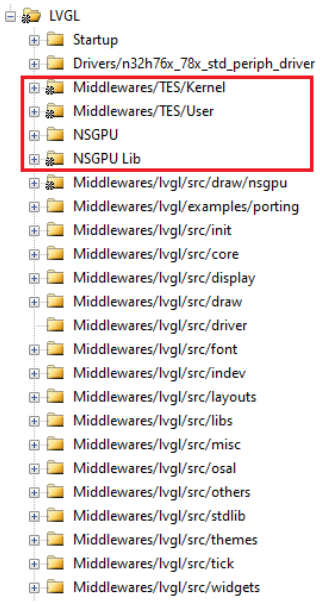
在 middlewares/gpu 目录下加入 lib 文件夹，包含 GPU 相关 API 封装库。

国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057



2.2. 加入工程

工程框图如下所示，需分别加入上述 nsgpu、tes、lib 文件夹中的文件和 LVGL 相关的文件。



2.3. 包含路径

..\..\middlewares\gpu\tes\kernel\inc
..\..\middlewares\gpu\tes\kernel\src
..\..\middlewares\gpu\tes\user\inc
..\..\middlewares\gpu\tes\user\src
..\..\middlewares\gpu\lib
..\..\middlewares\lvgl\src\draw\nsgpu

在需要使用的工程文件（如 main.c）中加入头文件#include "nsgpu.h"。

2.4. 代码适配

2.4.1. lvgl/src/init.c

（1）在该文件中加入包含头文件

国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
#if LV_USE_DRAW_VG_LITE
    #include "draw/vg_lite/lv_draw_vg_lite.h"
#endif
#if LV_USE_DRAW_NSGPU
  #include "../draw/nsgpu/lv_draw_nsgpu.h"
#endif
#if LV_USE_DRAW_DMA2D
    #include "draw/dma2d/lv_draw_dma2d.h"
#endif
#if LV_USE_DRAW_OPENGLES
    #include "draw/opengles/lv_draw_opengles.h"
#endif
#if LV_USE_WINDOWS
    #include "drivers/windows/lv_windows_context.h"
#endif
#if LV_USE_UEFI
    #include "drivers/uefi/lv_uefi_context.h"
#endif
#if LV_USE_EVDEV
    #include "drivers/evdev/lv_evdev_private.h"
#endif
```

（2）void lv_init(void)中加入初始化函数 lv_draw_nsgpu_init()

```
#if LV_USE_DRAW_VGLITE
    lv_draw_vglite_init();
#endif

#if LV_USE_PXP
#if LV_USE_DRAW_PXP || LV_USE_ROTATE_PXP
    lv_draw_pxp_init();
#endif
#endif

#if LV_USE_DRAW_G2D
    lv_draw_g2d_init();
#endif

#if LV_USE_DRAW_DAVE2D
    lv_draw_dave2d_init();
#endif

#if LV_USE_DRAW_SDL
    lv_draw_sdl_init();
#endif

#if LV_USE_DRAW_NSGPU
    lv_draw_nsgpu_init();
#endif

#if LV_USE_DRAW_DMA2D
    lv_draw_dma2d_init();
#endif

#if LV_USE_DRAW_OPENGLES
    lv_draw_opengles_init();
#endif
```

（3）void lv_deinit(void)中加入重置函数 lv_draw_nsgpu_deinit()

```
#if LV_USE_DRAW_VGLITE
    lv_draw_vglite_deinit();
#endif

#if LV_USE_DRAW_G2D
    lv_draw_g2d_deinit();
#endif

#if LV_USE_DRAW_VG_LITE
    lv_draw_vg_lite_deinit();
#endif

#if LV_USE_DRAW_NSGPU
    lv_draw_nsgpu_deinit();
#endif

#if LV_USE_DRAW_DMA2D
    lv_draw_dma2d_deinit();
#endif

#if LV_USE_DRAW_OPENGLES
    lv_draw_opengles_deinit();
#endif

#if LV_USE_DRAW_SW
    lv_draw_sw_deinit();
#endif
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900　　传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

2.4.2. lvgl/src/lv_conf_internal.h

加入 LV_USE_DRAW_NSGPU 宏定义：

```
#ifndef LV_USE_DRAW_NSGPU
    #ifdef CONFIG_LV_USE_GPU_NATIONS_GPU
        #define LV_USE_DRAW_NSGPU CONFIG_LV_USE_GPU_NATIONS_GPU
    #else
        #define LV_USE_DRAW_NSGPU 0
    #endif
#endif
#if LV_USE_DRAW_NSGPU
    #ifndef LV_GPU_NATIONS_CMSIS_INCLUDE
    #ifdef CONFIG_LV_GPU_NATIONS_INCLUDE
            #define LV_GPU_DMA2D_CMSIS_INCLUDE CONFIG_LV_GPU_NATIONS_INCLUDE
        #else
            #define LV_GPU_DMA2D_CMSIS_INCLUDE
        #endif
    #endif
#endif

/** Accelerate blends, fills, etc. with STM32 DMA2D */
#ifndef LV_USE_DRAW_DMA2D
    #ifdef CONFIG_LV_USE_DRAW_DMA2D
        #define LV_USE_DRAW_DMA2D CONFIG_LV_USE_DRAW_DMA2D
    #else
        #define LV_USE_DRAW_DMA2D 0
    #endif
#endif

#if LV_USE_DRAW_DMA2D
    #ifndef LV_DRAW_DMA2D_HAL_INCLUDE
        #ifdef CONFIG_LV_DRAW_DMA2D_HAL_INCLUDE
            #define LV_DRAW_DMA2D_HAL_INCLUDE CONFIG_LV_DRAW_DMA2D_HAL_INCLUDE
        #else
            #define LV_DRAW_DMA2D_HAL_INCLUDE "stm32h7xx_hal.h"
        #endif
    #endif

    /* if enabled, the user is required to call `lv_draw_dma2d_transfer_complete_interrupt_handler`
     * upon receiving the DMA2D global interrupt
     */
    #ifndef LV_USE_DRAW_DMA2D_INTERRUPT
        #ifdef CONFIG_LV_USE_DRAW_DMA2D_INTERRUPT
            #define LV_USE_DRAW_DMA2D_INTERRUPT CONFIG_LV_USE_DRAW_DMA2D_INTERRUPT
        #else
            #define LV_USE_DRAW_DMA2D_INTERRUPT 0
        #endif
    #endif
#endif
```

2.4.3. lvgl/lv_conf.h

（1）在该文件中加入宏定义 LV_USE_DRAW_NSGPU 或含有该宏定义的头文件，例如#include <user_config.h>，宏定义如下：

#define LV_USE_DRAW_NSGPU 1

（2）使用 NSGPU 时，LV_USE_DRAW_SW 需置为 0，即关闭默认的纯软件方式，改为 GPU 方式运行 LVGL。

国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
#if LV_USE_DRAW_NSGPU
#define LV_USE_DRAW_SW 0
#else
#define LV_USE_DRAW_SW 1
#endif

#if LV_USE_DRAW_SW == 1
    /*
     * Selectively disable color format support in order to reduce code size.
     * NOTE: some features use certain color formats internally, e.g.
     * - gradients use RGB888
     * - bitmaps with transparency may use ARGB8888
     */
    #define LV_DRAW_SW_SUPPORT_RGB565       1
    #define LV_DRAW_SW_SUPPORT_RGB565_SWAPPED       1
    #define LV_DRAW_SW_SUPPORT_RGB565A8     1
    #define LV_DRAW_SW_SUPPORT_RGB888       1
    #define LV_DRAW_SW_SUPPORT_XRGB8888     1
    #define LV_DRAW_SW_SUPPORT_ARGB8888     1
    #define LV_DRAW_SW_SUPPORT_ARGB8888_PREMULTIPLIED 1
    #define LV_DRAW_SW_SUPPORT_L8           1
    #define LV_DRAW_SW_SUPPORT_AL88         1
    #define LV_DRAW_SW_SUPPORT_A8           1
    #define LV_DRAW_SW_SUPPORT_I1           1

    /* The threshold of the luminance to consider a pixel as
     * active in indexed color format */
    #define LV_DRAW_SW_I1_LUM_THRESHOLD 127
```

（3）添加 LV_GPU_NATIONS_CMSIS_INCLUDE 宏定义

```
/*Use Nations's GPU*/
#if LV_USE_DRAW_NSGPU
    /*Must be defined to include path of CMSIS header of target processor*/
    #define LV_GPU_NATIONS_CMSIS_INCLUDE
#endif

/** Accelerate blends, fills, etc. with STM32 DMA2D */
#define LV_USE_DRAW_DMA2D 0

#if LV_USE_DRAW_DMA2D
    #define LV_DRAW_DMA2D_HAL_INCLUDE "stm32h7xx_hal.h"

    /* if enabled, the user is required to call `lv_draw_dma2d_transfer_complete_interrupt_handler`
     * upon receiving the DMA2D global interrupt
     */
    #define LV_USE_DRAW_DMA2D_INTERRUPT 0
#endif

/** Draw using cached OpenGLES textures */
#define LV_USE_DRAW_OPENGLES 0
```

（4）如果代码和内存空间不够，可以考虑关闭部分控件功能（宏定义 1 改为 0）

```
#define LV_USE_CANVAS     1

#define LV_USE_CHART      1

#define LV_USE_CHECKBOX   1

#define LV_USE_DROPDOWN   1

#define LV_USE_IMAGE      1

#define LV_USE_IMAGEBUTTON

#define LV_USE_KEYBOARD   1

#define LV_USE_LABEL      1
#if LV_USE_LABEL
    #define LV_LABEL_TEXT_SELI
    #define LV_LABEL_LONG_TXT_
    #define LV_LABEL_WAIT_CHAI
#endif

#define LV_USE_LED        1

#define LV_USE_LINE       1

#define LV_USE_LIST       1

#define LV_USE_LOTTIE     0

#define LV_USE_MENU       1

#define LV_USE_MSGBOX     1

#define LV_USE_ROLLER     1

#define LV_USE_SCALE      1

#define LV_USE_SLIDER     1

#define LV_USE_SPAN       1
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900　　传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

2.4.4. lvgl/examples/porting/lv_port_disp_template.c

（1） void lv_port_disp_init(void)选择 LVGL 缓存支持单缓存部分刷新、双缓存部分刷新、双缓存整屏刷新三种模式，用户根据应用需要选择缓存模式，部分刷新模式的缓存大小一般建议使用 1/10 屏大小。

以部分刷新模式为例，如下将 LVGL 的双缓存 buf_2_1[]和 buf_2_2[]指定到 sdram_area 空间中：

```
static uint8_t buf_2_1[MY_DISP_HOR_RES * 48 * BYTE_PER_PIXEL] __attribute__((section("sdram_area")));
static uint8_t buf_2_2[MY_DISP_HOR_RES * 48 * BYTE_PER_PIXEL] __attribute__((section("sdram_area")));
lv_display_set_buffers(disp, buf_2_1, buf_2_2, sizeof(buf_2_1), LV_DISPLAY_RENDER_MODE_PARTIAL);
```

sdram_area 空间是在分散加载文件(.sct)中定义的，具体配置方法请查看分散加载文件相关文档。

```
RW_IRAM1 0xC0000000 0x00600000  {
    .ANY (sdram_area)
}
```

（2） static void disp_init(void)加入 LCD 初始化函数

例如：LCDC_DispImage((u32)0xC0000000);

（3） static void disp_flush(lv_display_t * disp_drv, const lv_area_t * area, uint8_t * px_map)加入刷新函数

例如 800 x 480 分辨率的屏，使用部分刷新方式，刷新缓存大小为 800 x 48，格式为 RGB565，屏显示缓存的基地址为 0xC0000000。通过 nsgpu 拷贝调用参考如下：

```
static void disp_flush(lv_display_t * disp_drv, const lv_area_t * area, uint8_t * px_map){
    if(disp_flush_enabled) {
        lv_coord_t width = lv_area_get_width(area);
        lv_coord_t height = lv_area_get_height(area);

        nsgpu_data_copy((u32 *)0xC0000000, 800, 480, area->x1, area->y1, (u32 *)px_map, width, height, width, 2);
    }
    lv_display_flush_ready(disp_drv);
}
```

2.4.5. lvgl/examples/porting/lv_port_indev_template.c

如果需使用屏的触控功能，需完善触控与 LVGL 之间的衔接函数，参考举例如下：

（1） static void touchpad_init(void)

```
static void touchpad_init(void)
{
        tp_dev.init();
 }
```

（2） static bool touchpad_is_pressed(void)

```
static bool touchpad_is_pressed(void)
{
        tp_dev.scan(0);
        if (tp_dev.sta & TP_PRES_DOWN)
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
        {
            return true;
        }
        return false;
    }
```

（3） static void touchpad_get_xy(int32_t * x, int32_t * y)

```
    static void touchpad_get_xy(int32_t * x, int32_t * y)
    {
        (*x) = tp_dev.x[0];
        (*y) = tp_dev.y[0];
    }
```

2.5. 中断调用

2.5.1. LVGL 心跳

通过 SysTick 或定时器 1ms 周期调用一次 lv_tick_inc(1)，该计时为 LVGL 正常工作提供必需的嘀嗒心跳。

2.5.2. LCD 中断

根据应用需要，可使用 LCD 相关的中断，常用的中断主要是 EV 和 ER 中断。

（1）EV 中断支持某行刷新完成后产生中断，用户可以在中断时做一些简单的应用处理。

```
void LCD_EV_IRQ_Configuration(void)
{
    NVIC_InitType NVIC_InitStructure;

    NVIC_InitStructure.NVIC_IRQChannel                    = LCD_EV_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority   = 0x00;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority          = 0x01;
    NVIC_InitStructure.NVIC_IRQChannelCmd                 = ENABLE;
    NVIC_Init(&NVIC_InitStructure);
}
//
void LCD_EV_IRQHandler(void)
{
    __LCDC_CLEAR_FLAG(&LCDC_Handler, LCDC_FLAG_LI);
}
```

（2）ER 中断是在发生某种错误时产生中断。

```
void LCD_ER_IRQ_Configuration(void)
{
    NVIC_InitType NVIC_InitStructure;

    NVIC_InitStructure.NVIC_IRQChannel                    = LCD_ER_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority  = 0x00;
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
    NVIC_InitStructure.NVIC_IRQChannelSubPriority                = 0x01;
    NVIC_InitStructure.NVIC_IRQChannelCmd                        = ENABLE;
    NVIC_Init(&NVIC_InitStructure);
}
//
void LCD_ER_IRQHandler(void)
{
    __LCDC_CLEAR_FLAG(&LCDC_Handler, LCDC_IT_BE|LCDC_IT_RFE|LCDC_IT_FU);
}
```

2.5.3. 触控中断

如果使用屏的触控功能，且采用中断方式扫描，需先初始化相应中断，再调用中断处理函数。参考举例如下：

（1）初始化相应的中断（一般在触控初始化完成后调用）

```
static void touch_isr_enable(void)
{
     EXTI_InitType EXTI_InitStructure;
    NVIC_InitType NVIC_InitStructure;

     RCC_EnableAHB5PeriphClk2(RCC_AHB5_PERIPHEN_M7_GPIOI, ENABLE);
    GPIO_ConfigEXTILine(EXTI_LINE5, EXTI_GPIOI_Pin_5);

    EXTI_InitStructure.EXTI_Line = EXTI_LINE5;
    EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
    EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Falling;
    EXTI_InitStructure.EXTI_LineCmd = ENABLE;
    EXTI_InitPeripheral(&EXTI_InitStructure);

    NVIC_InitStructure.NVIC_IRQChannel                    = EXTI9_5_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority  = 0x0;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority         = 0x0;
    NVIC_InitStructure.NVIC_IRQChannelCmd                 = ENABLE;
    NVIC_Init(&NVIC_InitStructure);
}
```

（2）中断处理函数

```
  void EXTI15_10_IRQHandler(void)
  {
     if (EXTI_GetITStatus(EXTI_LINE5) != RESET)
       {
            tp_dev.scan(0);
            EXTI_ClrITPendBit(EXTI_LINE5);
       }
  }
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

2.5.4. GPU 中断

GPU 中断处理函数已在 GPU 驱动中实现，但要使 GPU 正常工作，在调用 GPU 初始化函数配置之前，需要在 main 函数中对 GPU 的 PWR、RCC、NVIC 进行配置。

2.6. main 中调用

2.6.1. 基础初始化

系统运行需要先在 main 函数中对基础模块进行初始化，包括 PWR、RCC、GPIO、NVIC 等。
如果使用分散加载文件，并在其中定义了 SDRAM 空间，则需要在进入 main 函数之前，初始化 PWR、RCC、GPIO、SDRAM 等，建议在 system_n32h76x_78x.c 中 void SystemInit(void)以寄存器方式初始化：

```
671 #ifdef CORE_CM7
672
673   /* Configure the Vector Table location add offset address ------------------*/
674 #ifdef VECT_TAB_SRAM
675   SCB->VTOR = 0x24000000;                          /* Vector Table Relocation in Internal SRAM */
676 #else
677   SCB->VTOR = FLASH_BANK1_BASE | VECT_TAB_OFFSET;       /* Vector Table Relocation in Internal FLASH */
678 #endif
679
680   /*defaule TCM_SIZE=0x3f,All TCMSRAM are AXI_SRAM,if you want to use ITCM/DTCM, define INIT_TCM_SIZE*/
681 #ifdef INIT_TCM_SIZE
682     ConfigTcmSize(TCM_SIZE_VALUE);
683 #endif
684     /*User can change the way of power supply */
685     //PWR_ConfigSupply(PWR_SUPPLY_SELECTION);
686
687     PreInitialize();
688
689 #else
690 #error Please #define CORE_CM4 or CORE_CM7
691 #endif
692 #endif
```

PreInitialize(void)函数及其调用的宏定义和函数参考代码如下（需结合实际应用调整 LCDC 像素时钟频率等）：

```
#define DBG_CPU_DELAY_INTI()      do{ DEM_CR |= DEM_CR_TRCENA;       \
                                      DWT_CYCCNT = 0U;                  \
                                      DWT_CR |= DWT_CR_CYCCNTENA; \
                                    }while(0)
#define DBG_CPU_DELAY_DISABLE() do{                                     \
                                      DEM_CR &= (uint32_t)(~(uint32_t)DEM_CR_TRCENA);
                                    }while(0)
#define __DBG_RCC_DELAY_US(usec)     do{ uint32_t delay_end;                        \
                                      DBG_CPU_DELAY_INTI();                         \
                                      delay_end=DWT_CYCCNT + (usec * (600000000/1000000)); \
                                      while(DWT_CYCCNT < delay_end){};          \
                                      DBG_CPU_DELAY_DISABLE();                      \
                                    }while(0)
void RCC_ConfigLCDPixelClkTemp(uint32_t CLK_source, uint32_t CLK_divider)
{
    uint32_t reg_value;
    uint32_t reg_value1;

    reg_value = RCC->AXISEL1;
    reg_value &= RCC_LCDPIXELCLK_SRC_MASK;
    reg_value |= CLK_source;
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
    RCC->AXISEL1 = reg_value;

    if(CLK_source == RCC_LCDPIXELCLK_SRC_AXIDIV)
    {
        reg_value1 = RCC->AXIDIV1;
        reg_value1 &=   RCC_LCDPIXELCLK_AXIDIV_MASK;
        reg_value1 |=   CLK_divider;
        RCC->AXIDIV1 = reg_value1;
    }
}
//
void RCC_ConfigPLL2BDividerTemp(uint32_t CLK_divider)
{
    uint32_t reg_value   = 0;

    reg_value = RCC->PLL2DIV;
    reg_value &=   RCC_PLLB_DIV_MASK;
    reg_value |=   CLK_divider;
    RCC->PLL2DIV = reg_value;
}
//
void PreInitialize(void)
{
    uint32_t temp_value1,temp_value2;

    //PWR for LCDC and GPU
    PWR->IPMEMCTRL &= (~GRAPHIC_LCDC_PWRCTRL);
    PWR->IPMEMCTRL &= (~GRAPHIC_GPU_PWRCTRL);
    PWR->SYSCTRL3 |= (PWR_SYSCTRL3_GRC_PSWACK1<< GRAPHIC_Domain);
    PWR->SYSCTRL3 |= (PWR_SYSCTRL3_GRC_PWREN<< GRAPHIC_Domain);
    while((PWR->SYSCTRL3&(PWR_SYSCTRL3_GRC_PWRRDY<<GRAPHIC_Domain))!=
(PWR_SYSCTRL3_GRC_PWRRDY<< GRAPHIC_Domain));
    PWR->SYSCTRL3 |= (PWR_SYSCTRL3_GRC_FUCEN<< GRAPHIC_Domain);
    PWR->SYSCTRL3 |= (PWR_SYSCTRL3_GRC_ISNEN<< GRAPHIC_Domain);

    //Configure RCC
    RCC->SYSBUSDIV1=0x01001100;
    __DBG_RCC_DELAY_US(1);
    //APB1=APB2=APB5=APB6=150MB
    RCC_ConfigAPBclkDivider(RCC_APB1CLK_DIV2,RCC_APB2CLK_DIV2,RCC_APB5CLK_DIV2,RCC_APB6CLK_DIV2)
    ;
    RCC->SYSBUSDIV2=0x04040404;

    // configure HSI to PLL1 source, frequency is 600M
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
    temp_value1 = RCC->PLL1CTRL1;
    temp_value2 = RCC->PLL1CTRL2;
    temp_value1 &= RCC_PLL_BWAJ_MASK;
    temp_value2 &= RCC_PLL_CLKR_CLKF_MASK;
    temp_value1 |= 0x3;
    temp_value2 |= 0x25800;
    RCC->PLL1CTRL1    = temp_value1;
    RCC->PLL1CTRL2    = temp_value2;
    RCC->PLL1CTRL1 |= RCC_PLL_LDO_ENABLE;
    __DBG_RCC_DELAY_US(10);
    RCC->PLL1CTRL1 &= (~RCC_PLL_POWER_DOWN);
    temp_value1 = RCC->PLL1CTRL1;
    temp_value1 &= RCC_PLL_SRC_MASK;
    temp_value1 |= RCC_PLL_SRC_HSI;
    RCC->PLL1CTRL1 = temp_value1;
    __DBG_RCC_DELAY_US(10);
    RCC->PLL1CTRL1 &= (~RCC_PLL_RESET_ENABLE);
    while((RCC->PLL1CTRL1&RCC_PLL_LOCK_FLAG) != RCC_PLL_LOCK_FLAG){}
    RCC->PLL1CTRL1 |= RCC_PLL_ENABLE;
    __DBG_RCC_DELAY_US(1);
    RCC->PLL1DIV=0x00020201;
    RCC->SRCCTRL1=0x6F000103;
    while((RCC->SRCCTRL1 & RCC_SYSCLK_STS_MASK) != RCC_SYSCLK_STS_PLL1A){}

     //Enable RCC of GPU/LCDC/DVP/GPIO and etc.
     RCC->AXIEN3   |= (RCC_AXI_PERIPHEN_M7_GPU | RCC_AXI_PERIPHEN_M7_GPULP);//GPU
     RCC->AXIEN2   |=   (RCC_AXI_PERIPHEN_M7_LCD          |          RCC_AXI_PERIPHEN_M7_LCDAPB          |
     RCC_AXI_PERIPHEN_M7_DVP1 | RCC_AXI_PERIPHEN_M7_DVP1APB);//LCDC and DVP
     RCC->AHB5EN1|=(RCC_AHB5_PERIPHEN_M7_GPIOA|RCC_AHB5_PERIPHEN_M7_GPIOB|RCC_AHB5_PERIPHEN_
M7_GPIOC|RCC_AHB5_PERIPHEN_M7_GPIOD);
     RCC->AHB5EN1|=(RCC_AHB5_PERIPHEN_M7_GPIOE|RCC_AHB5_PERIPHEN_M7_GPIOF|RCC_AHB5_PERIPHEN_
M7_GPIOG|RCC_AHB5_PERIPHEN_M7_GPIOH);
     RCC->AHB5EN2|=(RCC_AHB5_PERIPHEN_M7_GPIOI|RCC_AHB5_PERIPHEN_M7_AFIO|RCC_AHB5_PERIPHEN_M7
_GPIOJ |RCC_AHB5_PERIPHEN_M7_GPIOK | RCC_AHB5_PERIPHEN_PWR);
     RCC->APB5EN2 |=   RCC_APB5_PERIPHEN_EXTI;//EXTI
     RCC->APB1EN1 |=   RCC_APB1_PERIPHEN_M7_BTIM1;//BTIM

    RCC->SRCCTRL1 &= (~RCC_HSE_RDCNTEN);
    RCC->SRCCTRL1 |= RCC_HSE_ENABLE;
    __DBG_RCC_DELAY_US(50);
    RCC->SRCCTRL1 |= (RCC_HSE_RDCNTEN);
    while ((RCC->SRCCTRL1&RCC_HSE_STABLE_FLAG) == 0){}
    temp_value1 = RCC->PLL3CTRL1;
    temp_value2 = RCC->PLL3CTRL2;
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
temp_value1 &= RCC_PLL_BWAJ_MASK;
temp_value2 &= RCC_PLL_CLKR_CLKF_MASK;
temp_value1 |= 0x41;
temp_value2 |= 0x10214000;
RCC->PLL3CTRL1    = temp_value1;
RCC->PLL3CTRL2    = temp_value2;
RCC->PLL2CTRL1 |= RCC_PLL_LDO_ENABLE;
__DBG_RCC_DELAY_US(10);
RCC->PLL3CTRL1 &= (~RCC_PLL_POWER_DOWN);
temp_value1 = RCC->PLL3CTRL1;
temp_value1 &= RCC_PLL_SRC_MASK;
temp_value1 |= RCC_PLL_SRC_HSE;
RCC->PLL3CTRL1 = temp_value1;
__DBG_RCC_DELAY_US(10);
RCC->PLL3CTRL1 &= (~RCC_PLL_RESET_ENABLE);
while((RCC->PLL3CTRL1&RCC_PLL_LOCK_FLAG) != RCC_PLL_LOCK_FLAG){}
RCC->PLL3CTRL1 |= RCC_PLL_ENABLE;
RCC->PLL3DIV = 0x00020205;// SDRAM kernel clock is 133M
RCC->AXISEL2 = 0x00000300;
RCC->AXIEN4 |= (RCC_AXI_PERIPHEN_M7_SDRAM | RCC_AXI_PERIPHEN_M7_SDRAMLP);
RCC->PLLSFTLK = 0x01000020;

//Configure GPIO for SDRAM
GPIOA->AFH= 0xFFFFFFFF;
GPIOA->AFL= 0x0F1FFFFF;
GPIOA-> POTYPE= 0x00000000;
GPIOA-> PMODE= 0xABFFFFFF;
GPIOA-> PUPD= 0x64000000;
GPIOA->SR= 0x0000FFFF;
GPIOA->DS= 0xAAAAAAAA;

GPIOC->AFH= 0xFFFFFFFF;
GPIOC->AFL= 0xFF00FFFF;
GPIOC-> POTYPE= 0x00000000;
GPIOC-> PMODE= 0xFFFFFAFF;
GPIOC-> PUPD= 0x00000000;
GPIOC->SR= 0x0000FFCF;
GPIOC->DS= 0xAAAAA0AA;

GPIOD->AFH= 0x00FFF000;
GPIOD->AFL= 0xFFFFFF00;
GPIOD-> POTYPE= 0x00000000;
GPIOD-> PMODE= 0xAFEAFFFA;
GPIOD-> PUPD= 0x00000000;
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
GPIOD->SR= 0x000038FC;
GPIOD->DS= 0x0A80AAA0;

GPIOE->AFH= 0x00000000;
GPIOE->AFL= 0x0FFFFF00;
GPIOE-> POTYPE= 0x00000000;
GPIOE-> PMODE= 0xAAAABFFA;
GPIOE-> PUPD= 0x00000000;
GPIOE->SR= 0x0000007C;
GPIOE->DS= 0x00002AA0;

GPIOF->AFH=0x00000FFF;
GPIOF->AFL= 0xFF000000;
GPIOF-> POTYPE= 0x00000000;
GPIOF-> PMODE= 0xAABFFAAA;
GPIOF-> PUPD= 0x00000000;
GPIOF->SR= 0x000007C0;
GPIOF->DS= 0x002AA000;

GPIOG->AFH= 0x0FFFFFF0;
GPIOG->AFL= 0xFF00F000;
GPIOG-> POTYPE= 0x00000000;
GPIOG-> PMODE= 0xBFFEFAEA;
GPIOG-> PUPD= 0x00000000;
GPIOG->SR= 0x00007EC8;
GPIOG->DS= 0x2AA8A080;

GPIOH->AFH= 0xFFFFFFFF;
GPIOH->AFL= 0xFF0FFFFF;
GPIOH-> POTYPE= 0x00000000;
GPIOH-> PMODE= 0xFFFFFBFF;
GPIOH-> PUPD= 0x00000000;
GPIOH->SR= 0x0000FFDF;
GPIOH->DS= 0xAAAAA2AA;

SDRAM->BADD1       = 0xC0000000;
SDRAM->ADDMASK1 = 0xFE000000;
SDRAM->RI = 0x00000300; // 133M
SDRAM->RAT       = 0x00000007;
SDRAM->RCT       = 0x00000009;
SDRAM->RRDLY    = 0x00000003;
SDRAM->PT         = 0x00000004;
SDRAM->WRT       = 0x00000003;
SDRAM->RFCT      = 0x00000009;
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900　　传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
    SDRAM->RCDLY    = 0x00000004;
    SDRAM->CFG1 = 0xC02F0105;
    SDRAM->OS = 0x52000000;
    SDRAM->OR = 0x52000000;
    SDRAM->OS = 0x62000000;
    SDRAM->OR = 0x62000000;
    SDRAM->OS = 0x62000000;
    SDRAM->OR = 0x62000000;
    SDRAM->OS = 0x72000032;
    SDRAM->OR = 0x72000032;

    RCC->PLL3DIV = 0x00021405;// RCC_LCDPIXELCLK_SRC_PLL3B=33M for LCDC
    RCC_ConfigLCDPixelClkTemp(RCC_LCDPIXELCLK_SRC_PLL3B,RCC_LCDPIXELCLK_AXIDIV1);
}
```

2.6.2. 定时初始化

系统中常用 SysTick 或定时器来计时，以 SysTick 定时 100us 为例：

```
void SysTickInit(void)
{
     SysTick->LOAD = SystemCoreClock/10000 - 1;
    SysTick->VAL    = 0;
    SysTick->CTRL = 0x0007;
     NVIC_SetPriority(SysTick_IRQn, 0);
     NVIC_EnableIRQ(SysTick_IRQn);
}
```

2.6.3. GPU 初始化

在 LVGL 调用 GPU 初始化函数配置之前，需对 GPU 的 PWR、RCC、NVIC 进行配置（已在基础初始化中实现的，不需再调用），相关实现分别如下所示：

（1）PWR 使能

```
PWR_MoudlePowerEnable(GRAPHIC_GPU_PWRCTRL,ENABLE);
```

（2）RCC 使能

```
RCC_EnableAXIPeriphClk3(RCC_AXI_PERIPHEN_M7_GPU|RCC_AXI_PERIPHEN_M7_GPULP,ENABLE);
```

系统主频建议使用最高，通过调用 RCC 套件中的 RCC_SetSysClkToMode0()实现。

（3）中断使能

```
void NVIC_Configuration(void)
{
     NVIC_InitType NVIC_InitStructure;

    NVIC_InitStructure.NVIC_IRQChannel                    = GPU_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority  = 0x00;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority         = 0x00;
    NVIC_InitStructure.NVIC_IRQChannelCmd                 = ENABLE;
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
    NVIC_Init(&NVIC_InitStructure);
}
```

用户可根据自己的需要，在调用 NVIC_Configuration(void)之前，选择配置中断优先级分组，例如：NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);

2.6.4. 系统初始化

LVGL 系统初始化，调用函数 lv_init()。

2.6.5. 显示初始化

显示初始化，调用函数 lv_port_disp_init()。

2.6.6. 设备初始化

触控等输入设备初始化，调用函数 lv_port_indev_init()。

2.6.7. 周期调用

在主循环中调用 lv_task_handler()函数，运行周期至少为 1ms。

2.7. 应用示例

完成前面的移植后，就可以通过 GPU 方式运行 LVGL，实现用户的界面显示和事件处理。

2.7.1. 界面显示

如下所示，为一个简单的界面显示 demo：

```
uint8_t display_bg_img_map[64*2] __attribute__((section("sdram_area"))) = {
0x12,0x23,0x34,0x45,0x56,0x67,0x78,0x89,0x90,0x89,0x78,0x67,0x56,0x45,0x34,0x23,
0x12,0x23,0x34,0x45,0x56,0x67,0x78,0x89,0x90,0x89,0x78,0x67,0x56,0x45,0x34,0x23,
0x12,0x23,0x34,0x45,0x56,0x67,0x78,0x89,0x90,0x89,0x78,0x67,0x56,0x45,0x34,0x23,
0x12,0x23,0x34,0x45,0x56,0x67,0x78,0x89,0x90,0x89,0x78,0x67,0x56,0x45,0x34,0x23,
0x12,0x23,0x34,0x45,0x56,0x67,0x78,0x89,0x90,0x89,0x78,0x67,0x56,0x45,0x34,0x23,
0x12,0x23,0x34,0x45,0x56,0x67,0x78,0x89,0x90,0x89,0x78,0x67,0x56,0x45,0x34,0x23,
0x12,0x23,0x34,0x45,0x56,0x67,0x78,0x89,0x90,0x89,0x78,0x67,0x56,0x45,0x34,0x23,
0x12,0x23,0x34,0x45,0x56,0x67,0x78,0x89,0x90,0x89,0x78,0x67,0x56,0x45,0x34,0x23,
};
const   lv_img_dsc_t   display_bg_img = {
   .header.cf = LV_COLOR_FORMAT_RGB565,
   .header.w = 8,
   .header.h = 8,
   .data_size = 64*2,
   .data = display_bg_img_map,
};
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
static lv_obj_t * screen, *bg_map;
void DemoUI(void)
{
        lv_obj_clear_flag(lv_scr_act(), LV_OBJ_FLAG_SCROLLABLE);
        screen=lv_tileview_create(lv_scr_act());
        lv_obj_set_style_bg_color(screen,lv_color_hex(0x000000),LV_PART_MAIN);
        lv_obj_clear_flag(screen, LV_OBJ_FLAG_SCROLLABLE);
        lv_obj_add_event_cb(screen,click_screen_cb,LV_EVENT_RELEASED,0);

        bg_map=lv_img_create(screen);
        lv_obj_set_size(bg_map,lv_disp_get_hor_res(lv_disp_get_default()),lv_disp_get_ver_res(lv_disp_get_default()));
        lv_obj_set_style_bg_image_src(bg_map, &display_bg_img, LV_PART_MAIN);
        lv_obj_set_style_bg_image_tiled(bg_map, true, LV_PART_MAIN);
}
```

2.7.2. 事件处理

通过调用 lv_obj_add_event_cb(screen, click_screen_cb, LV_EVENT_RELEASED, 0)，加入了 click_screen_cb 事件处理函数，在 LV_EVENT_RELEASED 时执行相应的操作,比如切换不同的界面。

```
static lv_obj_t * white_screen;
static uint32_t count=0;

static void click_screen_cb(lv_event_t * e)
{
    lv_event_code_t code = lv_event_get_code(e);
    switch(code)
    {
        case LV_EVENT_RELEASED:
            if(count%2 == 0){
                white_screen=lv_tileview_create(lv_scr_act());
                lv_obj_set_style_bg_color(white_screen,lv_color_hex(0xffffff), LV_PART_MAIN);
                lv_obj_clear_flag(white_screen, LV_OBJ_FLAG_SCROLLABLE);
                lv_obj_add_event_cb(white_screen,click_screen_cb,LV_EVENT_RELEASED,0);
            }
            else
            {
                screen=lv_tileview_create(lv_scr_act());
                lv_obj_set_style_bg_color(screen,lv_color_hex(0x000000),LV_PART_MAIN);
                lv_obj_clear_flag(screen, LV_OBJ_FLAG_SCROLLABLE);
                lv_obj_add_event_cb(screen,click_screen_cb,LV_EVENT_RELEASED,0);
            }
            count++;
            break;
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
            default:
                break;
        }
}
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

3. GPU 使用指南

GPU 使用指南，主要介绍 GPU 的 API 库使用，包括通用 API 和专用 API（只适用于某种平台）。下述针对 LVGL 的专用 API 对应的 LVGL 版本为 V9.3.0，其他版本可能出现 API 不兼容等问题。

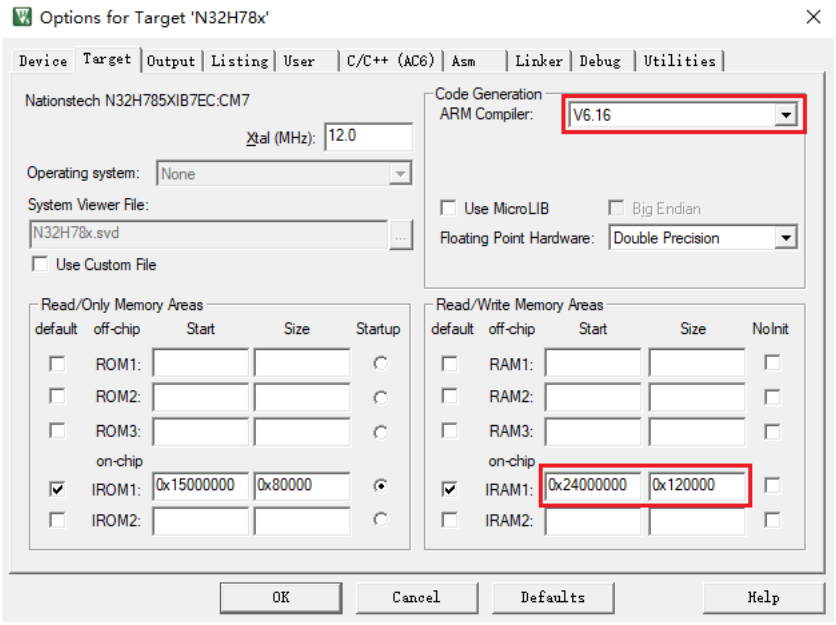
3.1. 基础环境

3.1.1. KEIL 配置

KEIL/Project/Options for target…的部分配置界面如下：

（1） 编译器建议使用 V6（需注意 V6 中若使用到 C++，与 MicroLIB 不能兼容）。

RAM 尽可能配置大一些，需保证 GPU 初始化配置的 framebuffer 大小需求，否则编译会出错。

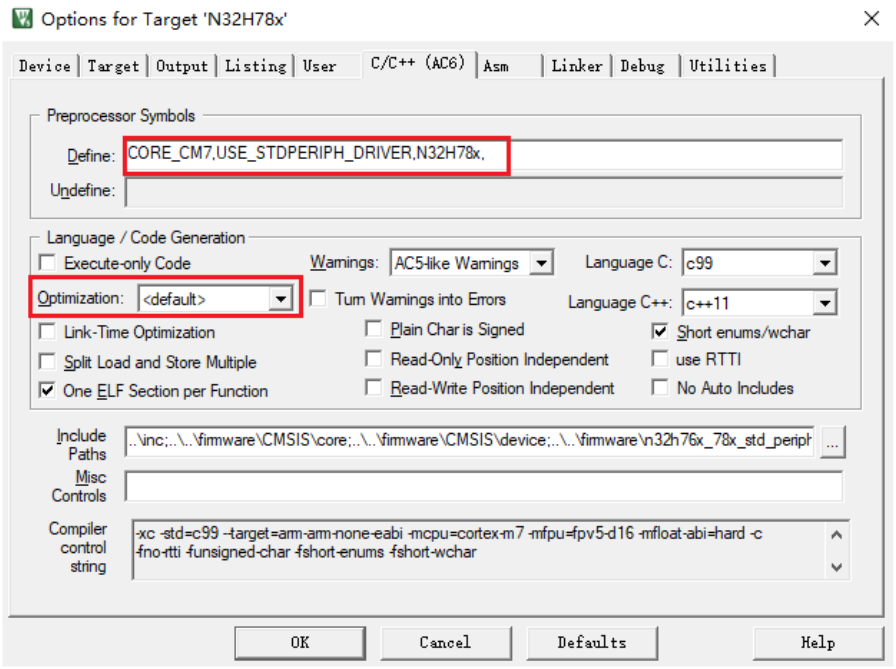


也可通过分散加载文件配置，具体方法参考分散加载文件相关的说明文档。

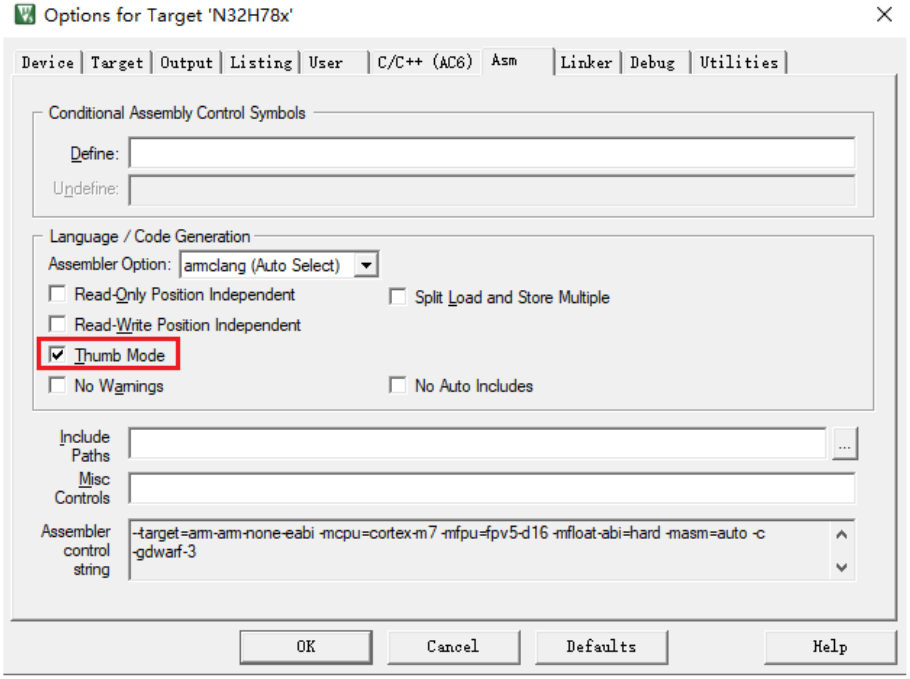
（2） 宏定义和优化项等

宏定义是为了支持基础套件正常使用，优化项使用<default>或-O0。

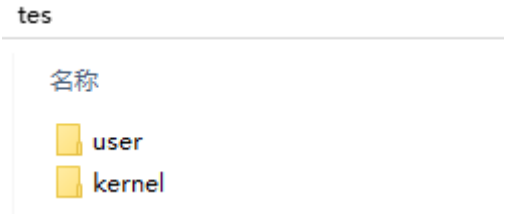
国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057



（3）指令集使用 Thumb Mode



3.1.2. 底层驱动

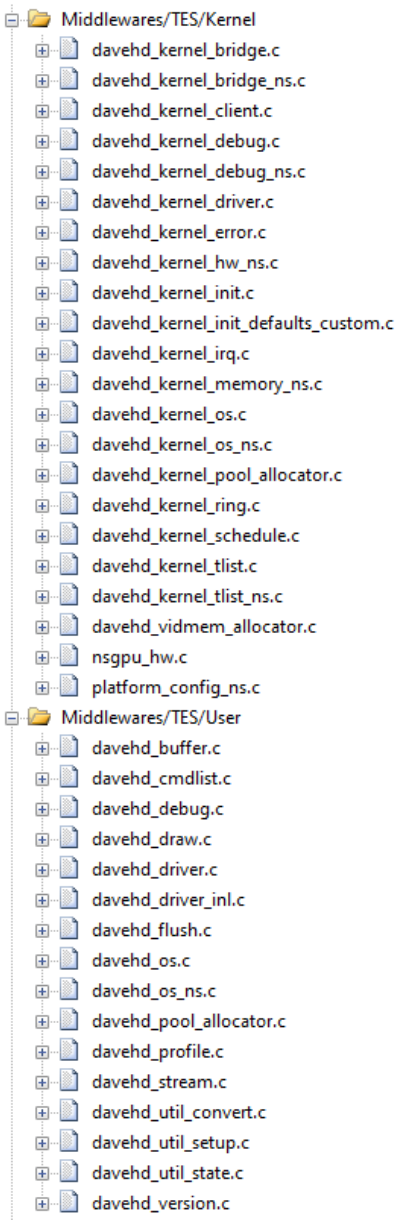


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

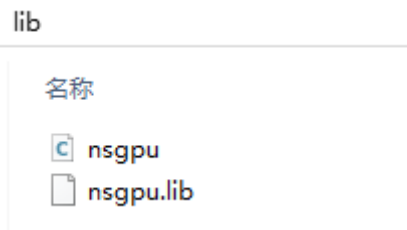
如下图所示，需要在 KEIL 工程的 Options 中添加底层驱动文件的包含路径。

```
..\..\middlewares\gpu\tes\kernel\inc
..\..\middlewares\gpu\tes\kernel\src
..\..\middlewares\gpu\tes\user\inc
..\..\middlewares\gpu\tes\user\src
```

并在 KEIL 工程中加入相应的代码：



3.1.3. API 库



在 KEIL 工程的 Options 中将 nsgpu.h 也加入包含路径,将 nsgpu.lib 加入代码工程中。在需要使用的

国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

工程文件（如 main.c）中加入头文件#include "nsgpu.h"。

3.1.4. 堆栈配置

在 startup_n32h78x_cm7.s 中配置堆和栈的大小， Heap_Size 至少需 0x2000，最好能 0x10000，Stack_Size 可配 0x2000。

3.2. 基础模块

系统主频建议使用最高，通过调用 RCC 套件中的 RCC_SetSysClkToMode0()实现。

3.2.1. 基础外设

要使 GPU 正常工作，在调用 GPU 初始化函数配置之前，需要对 GPU 的 PWR、RCC、NVIC 进行配置，相关实现分别如下所示：

（1）PWR 使能

```
PWR_MoudlePowerEnable(GRAPHIC_GPU_PWRCTRL,ENABLE);
```

（2）RCC 使能

```
RCC_EnableAXIPeriphClk3(RCC_AXI_PERIPHEN_M7_GPU|RCC_AXI_PERIPHEN_M7_GPULP,ENABLE);
```

（3）中断使能（中断处理函数 GPU_IRQHandler(void)已在驱动库中实现）

```
void NVIC_Configuration(void)
{
    NVIC_InitType NVIC_InitStructure;

    NVIC_InitStructure.NVIC_IRQChannel                    = GPU_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority  = 0x00;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority         = 0x00;
    NVIC_InitStructure.NVIC_IRQChannelCmd                 = ENABLE;
    NVIC_Init(&NVIC_InitStructure);
}
```

用户可根据自己的需要，在调用 NVIC_Configuration(void)之前，选择配置中断优先级分组，例如：

```
NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);
```

3.2.2. 扩展内存

GPU 工作所需内存由分辨率、像素字节数、缓存个数等决定，内部 SRAM 足够时，可以使用未被占用的 0x30000000 或 0x24040000 等 SRAM 地址，否则需使用外部 SDRAM 地址。

在 nsgpu.h 中 GPU_MEMORY_BASE_ADDRESS 默认为 0xC0C00000，可结合实际情况修改。未在分散加载文件中定义 SDRAM 空间的情况下，在 main 中对 SDRAM 初始化后即可使用外部扩展内存，具体参考 SDRAM 的 DEMO。

3.2.3. 显示模块

多数情况下，应用中需要用到显示模块 LCDC，相关功能的使用参考 LCDC 的 DEMO。

国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

3.3. 通用 API 库说明

3.3.1. 初始化

int nsgpu_init (u32 gpu_memory_base, u32 fb_num, u32 width, u32 height, u32 pixel_bytes)

（1）参数说明

gpu_memory_base：分配给 GPU 内存的基地址，width*height*pixel_bytes*fb_num 小于 300KB 时，可以考虑使用未被占用的 0x30000000 或 0x24040000 等内部 SRAM 地址，否则需使用外部 SDRAM 地址。在 nsgpu.h 中 GPU_MEMORY_BASE_ADDRESS 默认为 0xC0C00000，可结合实际情况修改。

fb_num：申请 framebuffer 的个数，会占用内存，一般设置为 1 即可，不能超过 FRAME_BUF_MAX；

width：申请 framebuffer 的宽度；

height：申请 framebuffer 的高度；

pixel_bytes：根据像素格式设置像素字节数，A8、L8 等类似设置为 1，RGB565 等类似设置为 2，RGB565A8 等类似设置为 3，ARGB8888 等类似设置为 4。

（2）返回值说明

返回初始化状态如下：

-1：参数设置异常

-2：平台错误

-3：设备错误或程序卡死

0：内存不足

1：正常

（3）示例

int result = nsgpu_init (GPU_MEMORY_BASE_ADDRESS, 1, SCREEN_WIDTH_SIZE, SCREEN_HEIGHT_SIZE, 2);

其中 SCREEN_WIDTH_SIZE 和 SCREEN_HEIGHT_SIZE 分别为屏宽和屏高。

3.3.2. 重置函数

void nsgpu_deinit(void)

GPU 重置到默认状态，一般只在系统异常时使用。

3.3.3. 获取就绪缓存

u32 nsgpu_get_ready_framebuffer (u8 index)

nsgpu_init 初始化完成后，通过该函数获取缓存序号对应的已初始化就绪的渲染缓存。一般情况下不需调用。

（1）参数说明

index：为 framebuffer 的缓存序号，取值范围在 0～(fb_num-1)之间，fb_num 是 nsgpu_init 函数初始化时配置的。

（2）返回值说明

返回值为缓存序号对应的已初始化就绪的渲染缓存首地址。

（3）示例

int result = nsgpu_init (GPU_MEMORY_BASE_ADDRESS, 2, SCREEN_WIDTH_SIZE, SCREEN_HEIGHT_SIZE, 2);

u32 address0 = nsgpu_get_ready_framebuffer (0);

u32 address1 = nsgpu_get_ready_framebuffer (1);

国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

3.3.4. 选择就绪缓存

void nsgpu_set_framebuffer (u8 index)

nsgpu_init 初始化完成后，通过该函数设置缓存序号对应的已初始化就绪的缓存作为当前工作的渲染缓存。一般情况下不需调用。

（1）参数说明

index：为 framebuffer 的缓存序号，取值范围在 0～(fb_num-1)之间，fb_num 是 nsgpu_init 函数初始化时配置的。

（2）示例

```
int result = nsgpu_init (GPU_MEMORY_BASE_ADDRESS, 2, SCREEN_WIDTH_SIZE, SCREEN_HEIGHT_SIZE, 2);
nsgpu_get_ready_framebuffer (0);或 nsgpu_get_ready_framebuffer (1);
```

3.3.5. 获取当前缓存

u32 nsgpu_get_framebuffer(void)

nsgpu_init 初始化完成后，调用该函数获取当前工作的渲染缓存。

（1）返回值说明

返回值为当前渲染缓存的首地址。

（2）示例

```
int result = nsgpu_init (GPU_MEMORY_BASE_ADDRESS, 1, SCREEN_WIDTH_SIZE, SCREEN_HEIGHT_SIZE, 2);
nsgpu_set_framebuffer (0xC0000000);
u32 address = nsgpu_get_framebuffer ();
```

3.3.6. 改变当前缓存

void nsgpu_set_framebuffer (u32 addr)

nsgpu_init 初始化完成后，可通过该函数改变当前工作的已初始化就绪的渲染缓存为未初始化的缓存。一般情况下不建议改变。

（1）参数说明

addr：为要配置为渲染缓存的首地址。

（2）示例

```
int result = nsgpu_init(GPU_MEMORY_BASE_ADDRESS, 1, SCREEN_WIDTH_SIZE, SCREEN_HEIGHT_SIZE, 2);
nsgpu_set_framebuffer(0xC0000000);
```

3.3.7. 数据批量填充

void nsgpu_data_fill (u32 *dst, u32 dst_width, u32 dst_height, u32 dst_startx, u32 dst_starty, u32 color, u8 alpha, u32 width, u32 height)

（1）参数说明

dst：颜色数据填充的目标地址；

dst_width：区域总宽度，一般和填充区域的宽度 width 相同；

国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

dst_height：区域总高度，一般和填充区域的高度 height 相同；
dst_startx：目标地址基础上的 x 方向偏移值，根据需要设置，一般为 0；
dst_starty：目标地址基础上的 y 方向偏移值，根据需要设置，一般为 0；
color：填充的颜色值，颜色始终为 ARGB8888 格式的值，如果 nsgpu_init ()初始化时 pixel_bytes=4，填充的值和 color 值一样，如果 pixel_bytes=2，填充的值则是 color 转换为 RGB565 的值, pixel_bytes 其他值也会转换成相应格式的值；
alpha：填充的透明度，取值范围为 0～255，0 表示全透明，255 表示不透明。
width：填充区域的宽度；
height：填充区域的高度。
（2）示例
从 0xC0000000 开始填充 800 x 480 矩形块，颜色为 0xFFFF0000（红色），不透明。

```
int result = nsgpu_init(GPU_MEMORY_BASE_ADDRESS, 1, 800, 480, 2);
nsgpu_data_fill((u32 *)0xC0000000, 800, 480, 0, 0, 0xFFFF0000, 255，800, 480);
```

3.3.8. 数据批量拷贝

```
void nsgpu_data_copy (u32 *dst, u32 dst_width, u32 dst_height, u32 dst_startx, u32 dst_starty, u32 *src, u32 width, u32 height, u32 pitch, u32 pixel_bytes)
```

（1）参数说明
dst：数据拷贝的目标地址；
dst_width：目标区域的宽度；
dst_height：目标区域的高度；
dst_startx：目标地址基础上的 x 方向偏移值，根据需要设置，一般为 0；
dst_starty：目标地址基础上的 y 方向偏移值，根据需要设置，一般为 0；
src：数据拷贝的源地址；
width：数据源区域的宽度；
height：数据源区域的高度;
pitch：数据源区域的行与行之间的间距，一般和宽度相同；
pixel_bytes：根据像素格式设置像素字节数，A8、L8 等类似设置为 1，RGB565 等类似设置为 2，RGB565A8 等类似设置为 3，ARGB8888 等类似设置为 4。
（2）示例
目标区域大小为 300 x 180，从偏移坐标（20,5）开始拷贝 300 x 180 的矩形块数据。

```
int result = nsgpu_init (GPU_MEMORY_BASE_ADDRESS, 1, 800, 480, 2);
memset((void*)0x30000000,0,0x20000);//dst 区域全部清零
memset((void*)0x30020000,0x55,0x20000);//src 区域写入 0x55
nsgpu_data_copy ((u32 *)0x30000000, 300, 180, 20, 5, (u32 *)0x30020000, 300, 180, 300, 2);
```

调试查看内存，可观察到从 0x30000000 +（300*5 + 20）*2 =0x30000BE0 开始写入 0x55。
（3）应用
例如在 LVGL 刷新函数中，800 x 480 分辨率的屏，使用部分刷新方式，刷新缓存大小为 800 x 48，格式为 RGB565，屏显示缓存的基地址为 0xC0000000。拷贝调用如下：

```
static void disp_flush (lv_display_t * disp_drv, const lv_area_t * area, uint8_t * px_map) {
    if(disp_flush_enabled) {
        lv_coord_t width = lv_area_get_width(area);
        lv_coord_t height = lv_area_get_height(area);
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
            nsgpu_data_copy ((u32 *)0xC0000000, 800, 480, area->x1, area->y1, (u32 *)px_map, width, height, width, 2);
        }
        lv_display_flush_ready(disp_drv);
}
```

注意：受 GPU 数据搬运机制决定，搬运前、后的数据可能会有细微字节不同，对于图像应用是没有问题的，但对于通信等数据要求完全保真的可能不适用。

3.3.9. 设置绘制颜色

void nsgpu_set_draw_color (u32 color)

（1）参数说明

color：设置绘制的颜色，颜色为 ARGB8888 格式。

（2）示例

```
nsgpu_set_draw_color (0xFF0000FF); //设置绘制颜色为蓝色。
```

3.3.10. 设置绘制 Alpha

void nsgpu_set_draw_alpha (u8 alpha)

（1）参数说明

alpha：设置绘制的透明度，取值范围为 0～255，0 表示全透明，255 表示不透明。

（2）示例

```
nsgpu_set_draw_alpha (255); //设置绘制透明度为 255。
```

3.3.11. 绘制基础图形

如果要画点，可以直接通过软件方式填充像素，也可以通过下面画线或矩形等方式实现。

3.3.11.1. 线段

void nsgpu_vgdraw_line (float x1, float y1, float x2, float y2, float width)

（1）参数说明

(x1,y1)和(x2,y2)：为实心线的两端点坐标。

width：为实心线的线宽，最小值可设置为 0，最小变化值为 0.1。

（2）示例

```
nsgpu_set_draw_color(0xFFFF0000); //设置红色
nsgpu_vgdraw_line (60, 100, 360, 300，1.0);
```

3.3.11.2. 矩形

void nsgpu_vgdraw_rect (float x, float y, float width, float height)

（1）参数说明

(x,y)：为矩形的左上角顶点坐标。

width：为实心矩形的水平宽度。

height：为实心矩形的垂直高度。

（2）示例

```
nsgpu_set_draw_color(0xFFFF0000);//设置红色
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
nsgpu_vgdraw_rect (20, 20, 100, 50);
```

3.3.11.3. 三角形

void nsgpu_vgdraw_triangle (float x1, float y1, float x2, float y2, float x3, float y3)

（1）参数说明

(x1,y1)、(x2,y2)、(x3,y3)：为实心三角形的三个顶点坐标，函数调用时，配置三个顶点无顺序要求。

（2）示例

```
nsgpu_set_draw_color(0xFFFF0000); //设置红色
nsgpu_vgdraw_triangle (10, 50, 40, 40, 70, 70) ;
```

3.3.11.4. 圆

void nsgpu_vgdraw_circle (float cx, float cy, float r)

（1）参数说明

(cx,cy)：为实心圆的圆心坐标。

r：为圆的半径。

（2）示例

```
nsgpu_set_draw_color(0xFFFF0000);//设置红色
nsgpu_vgdraw_circle (60, 60, 30);
```

3.3.11.5. 圆弧

void nsgpu_vgdraw_arc(float cx, float cy, float r, u32 width, u32 start_angle, u32 arc_angle)

（1）参数说明

(cx,cy)：为圆弧的圆心坐标。

r：为圆弧的半径。

width:为圆弧的宽度。

start_angle：为圆弧的起始角度。

arc_angle：为圆弧对应的角度（水平右侧为 0 度，顺时针递增）。

（2）示例

```
nsgpu_set_draw_color(0xFFFF0000);//设置红色
nsgpu_vgdraw_arc (60, 60, 30, 1, 0, 45);
```

3.3.11.6. 扇形

void nsgpu_draw_wedge(float cx, float cy, float r, u32 start_angle, u32 arc_angle)

（1）参数说明

(cx,cy)：为扇形的圆心坐标。

r：为扇形的半径。

start_angle：为扇形的起始角度。

arc_angle：为扇形对应的角度（水平右侧为 0 度，顺时针递增）。

（2）示例

```
nsgpu_set_draw_color(0xFFFF0000);//设置红色
nsgpu_draw_wedge (60, 60, 30,0, 45);
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

3.4. LVGL 专用 API 库说明

3.4.1. 填充处理

void nsgpu_lvgl_fill_proc (lv_draw_task_t *t, const lv_draw_fill_dsc_t * draw_dsc, lv_area_t draw_area, lv_area_t coordinates)

该函数适用于在 lv_draw_nsgpu_fill.c 的 void lv_draw_nsgpu_fill(lv_draw_task_t * t, const lv_draw_fill_dsc_t * draw_dsc, const lv_area_t * coords)中调用。

（1）参数说明

t：绘制任务相关的指针，来自于上层参数传递；

draw_dsc：填充相关的指针，来自于上层参数传递；

draw_area：绘制区域相关信息，由上层相关参数获得；

coordinates：坐标相关信息，由上层相关参数获得。

（2）示例

```
void lv_draw_nsgpu_fill (lv_draw_task_t * t, const lv_draw_fill_dsc_t * draw_dsc, const lv_area_t * coords)
{
    lv_area_t coordinates;
    lv_area_copy (&coordinates, coords);

    bool flag;
    lv_area_t clip_area;
    flag = lv_area_intersect (&clip_area, &coordinates, &t->clip_area);
    if (! flag) return;

    lv_area_move (&coordinates, -t->target_layer->buf_area.x1, -t->target_layer->buf_area.y1);
    lv_area_move (&clip_area, -t->target_layer->buf_area.x1, -t->target_layer->buf_area.y1);

    nsgpu_lvgl_fill_proc (t, draw_dsc, clip_area, coordinates);
}
```

3.4.2. 字符处理

void nsgpu_lvgl_label_proc (lv_draw_task_t *t, lv_draw_glyph_dsc_t * glyph_draw_dsc, lv_area_t draw_area, lv_area_t letter_area)

该函数适用于在 lv_draw_nsgpu_label.c 的 void lv_draw_nsgpu_draw_letter_cb (lv_draw_task_t *t, lv_draw_glyph_dsc_t *glyph_draw_dsc, lv_draw_fill_dsc_t * fill_draw_dsc, const lv_area_t * fill_area)中调用。

（1）参数说明

t：绘制任务相关的指针，来自于上层参数传递；

glyph_draw_dsc：字符相关的指针，来自于上层参数传递；

draw_area：绘制区域相关信息，由上层相关参数获得；

letter_area：字符区域相关信息，由上层相关参数获得。

（2）示例

```
void lv_draw_nsgpu_draw_letter_cb(lv_draw_task_t *t, lv_draw_glyph_dsc_t * glyph_draw_dsc,lv_draw_fill_dsc_t * fill_draw_dsc,
const lv_area_t * fill_area)
{
    lv_area_t letter_area;
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
letter_area = *glyph_draw_dsc->letter_coords;

bool flag;
lv_area_t clip_area;
flag = lv_area_intersect(&clip_area, &letter_coords, &clip_area);
if (! flag) return;

lv_area_move(&letter_area, -t->target_layer->buf_area.x1, -t->target_layer->buf_area.y1);
lv_area_move(&clip_area, -t->target_layer->buf_area.x1, -t->target_layer->buf_area.y1);

if(glyph_draw_dsc) {
        switch(glyph_draw_dsc->format) {
            case LV_FONT_GLYPH_FORMAT_NONE: {
#if LV_USE_FONT_PLACEHOLDER
                 /* Draw a placeholder rectangle*/
                lv_draw_border_dsc_t border_draw_dsc;
                 lv_draw_border_dsc_init(&border_draw_dsc);
                border_draw_dsc.opa = glyph_draw_dsc->opa;
                border_draw_dsc.color = glyph_draw_dsc->color;
                border_draw_dsc.width = 1;
                lv_draw_nsgpu_border (t, &border_draw_dsc, glyph_draw_dsc->bg_coords);
#endif
            }
            break;
        case LV_FONT_GLYPH_FORMAT_A1:
        case LV_FONT_GLYPH_FORMAT_A2:
        case LV_FONT_GLYPH_FORMAT_A3:
        case LV_FONT_GLYPH_FORMAT_A4:
        case LV_FONT_GLYPH_FORMAT_A8:
            {
                nsgpu_lvgl_label_proc (t, glyph_draw_dsc, clip_area, letter_area);
            }
            break;
        case LV_FONT_GLYPH_FORMAT_IMAGE: {
#if LV_USE_IMGFONT
         glyph_draw_dsc->glyph_data = lv_font_get_glyph_bitmap(glyph_draw_dsc->g, glyph_draw_dsc->_draw_buf);
         lv_draw_image_dsc_t img_dsc;
        lv_draw_image_dsc_init(&img_dsc);
         img_dsc.rotation = 0;
        img_dsc.scale_x = LV_SCALE_NONE;
        img_dsc.scale_y = LV_SCALE_NONE;
         img_dsc.opa = glyph_draw_dsc->opa;
        img_dsc.src = glyph_draw_dsc->glyph_data;
        t->draw_dsc = &img_dsc;
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
                    t->area = *glyph_draw_dsc->letter_coords;
                    nsgpu_lvgl_path_proc (t, fill_draw_dsc, blend_area);
    #endif
                    }
                    break;
                default:
                    break;
            }
        }
}
```

3.4.3. 图片处理

void nsgpu_lvgl_image_proc (lv_draw_task_t *t, const lv_draw_image_dsc_t * image_draw_dsc, lv_area_t draw_area, lv_area_t image_area)

该函数适用于在 lv_draw_nsgpu_image.c 的 void img_draw_core (lv_draw_task_t * t, const lv_draw_image_dsc_t *dsc,const lv_image_decoder_dsc_t *decoder_dsc, lv_draw_image_sup_t * sup,const lv_area_t *img_coords, const lv_area_t * clipped_img_area)中调用。

（1）参数说明

t：绘制任务相关的指针，来自于上层参数传递；

image_draw_dsc：图片相关的指针，来自于上层参数传递；

draw_area：绘制区域相关信息，由上层相关参数获得；

image_area：图片区域相关信息，由上层相关参数获得。

（2）示例

```
void img_draw_core (lv_draw_task_t *t, const lv_draw_image_dsc_t *dsc, const lv_image_decoder_dsc_t *decoder_dsc, lv_draw_image_sup_t * sup, const lv_area_t * img_coords, const lv_area_t * clipped_img_area)
{
    lv_area_t image_area;
    lv_area_copy (&image_area, img_coords);

    bool flag;
    lv_area_t clip_area;
    flag = lv_area_intersect (&clip_area, &image_area, &t->clip_area);
    if (! flag) return;

    lv_area_move (&image_area, -t->target_layer->buf_area.x1, -t->target_layer->buf_area.y1);
    lv_area_move (&clip_area, -t->target_layer->buf_area.x1, -t->target_layer->buf_area.y1);

    nsgpu_lvgl_image_proc (t, dsc, clip_area, src_area);
}
```

3.4.4. 图层处理

void nsgpu_lvgl_layer_proc(lv_draw_task_t *t,const lv_draw_image_dsc_t * image_draw_dsc,lv_area_t draw_area,lv_area_t image_area)

国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

该函数适用于在 lv_draw_nsgpu_layer.c 的 void nsgpu_lvgl_layer_proc(lv_draw_task_t *t,const lv_draw_image_dsc_t * image_draw_dsc,lv_area_t draw_area,lv_area_t image_area)中调用。

（1）参数说明

t：绘制任务相关的指针，来自于上层参数传递；

image_draw_dsc：图片相关的指针，来自于上层参数传递；

draw_area：绘制区域相关信息，由上层相关参数获得；

image_area：图片区域相关信息，由上层相关参数获得。

（2）示例

```
void lv_draw_nsgpu_layer(lv_draw_task_t * t, const lv_draw_image_dsc_t * draw_dsc, const lv_area_t * coords)
{
    lv_area_t image_area;
    lv_area_copy (&image_area, coords);

    bool flag;
    lv_area_t clip_area;
    flag = lv_area_intersect (&clip_area, &image_area, &t->clip_area);
    if (! flag) return;

    lv_area_move (&image_area, -t->target_layer->buf_area.x1, -t->target_layer->buf_area.y1);
    lv_area_move (&clip_area, -t->target_layer->buf_area.x1, -t->target_layer->buf_area.y1);

    nsgpu_lvgl_layer_proc (t, draw_dsc, clip_area,image_area);
}
```

3.4.5. 线段处理

```
void nsgpu_lvgl_line_proc(lv_draw_task_t *t, const lv_draw_line_dsc_t * draw_dsc, lv_area_t draw_area)
```

该函数适用于在 lv_draw_nsgpu_line.c 的 void lv_draw_nsgpu_line (lv_draw_task_t * t, const lv_draw_line_dsc_t * draw_dsc)中调用。

（1）参数说明

t：绘制任务相关的指针，来自于上层参数传递；

draw_dsc：绘制相关的指针，来自于上层参数传递；

draw_area：绘制区域相关信息，由上层相关参数获得。

（2）示例

```
void lv_draw_nsgpu_line(lv_draw_task_t * t, const lv_draw_line_dsc_t * draw_dsc)
{
    if (draw_dsc->width == 0)
        return;
    if (draw_dsc->opa <= (lv_opa_t) LV_OPA_MIN)
        return;
    if (draw_dsc->p1.x == draw_dsc->p2.x && draw_dsc->p1.y == draw_dsc->p2.y)
        return;

    lv_area_t clip_area;
    bool flag;
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
    clip_area.x1 = LV_MIN (draw_dsc->p1.x, draw_dsc->p2.x) - draw_dsc->width / 2;
    clip_area.x2 = LV_MAX (draw_dsc->p1.x, draw_dsc->p2.x) + draw_dsc->width / 2;
    clip_area.y1 = LV_MIN (draw_dsc->p1.y, draw_dsc->p2.y) - draw_dsc->width / 2;
    clip_area.y2 = LV_MAX (draw_dsc->p1.y, draw_dsc->p2.y) + draw_dsc->width / 2;

    flag = lv_area_intersect (&clip_area, &clip_area, &t->clip_area);
    if (! flag) return;
    lv_area_move (&clip_area, -t->target_layer->buf_area.x1, -t->target_layer->buf_area.y1);
    nsgpu_lvgl_line_proc (t, draw_dsc, clip_area);
}
```

3.4.6. 圆弧处理

```
void nsgpu_lvgl_arc_proc (lv_draw_task_t *t, const lv_draw_arc_dsc_t * draw_dsc, lv_area_t draw_area)
```

该函数适用于在 lv_draw_nsgpu_arc.c 的 void lv_draw_nsgpu_arc(lv_draw_task_t * t, const lv_draw_arc_dsc_t * draw_dsc, const lv_area_t * coords)中调用。

（1）参数说明

t：绘制任务相关的指针，来自于上层参数传递；

draw_dsc：绘制相关的指针，来自于上层参数传递；

draw_area：绘制区域相关信息，由上层相关参数获得。

（2）示例

```
void lv_draw_nsgpu_arc(lv_draw_task_t * t, const lv_draw_arc_dsc_t * draw_dsc, const lv_area_t * coords)
{
    if(draw_dsc->opa <= (lv_opa_t) LV_OPA_MIN)
        return;
    if(draw_dsc->width == 0)
        return;
    if(draw_dsc->start_angle == draw_dsc->end_angle)
        return;

    lv_area_t clip_area;
    bool flag;

    flag = lv_area_intersect (&clip_area, coords, &t->clip_area);
    if (! flag) return;
    lv_area_move (&clip_area, -t->target_layer->buf_area.x1, -t->target_layer->buf_area.y1);
    if(draw_dsc->start_angle == draw_dsc->end_angle)
        return;
    nsgpu_lvgl_arc_proc (t, draw_dsc, clip_area);
}
```

3.4.7. 矩形处理

```
void nsgpu_lvgl_mask_rect_proc (lv_draw_task_t *t, const lv_draw_mask_rect_dsc_t * draw_dsc, lv_area_t draw_area)
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

该函数适用于在 lv_draw_nsgpu_mask_rectangle.c 的 void lv_draw_nsgpu_mask_rect(lv_draw_task_t * t, const lv_draw_mask_rect_dsc_t * draw_dsc, const lv_area_t * coords)中调用。

（1）参数说明

t：绘制任务相关的指针，来自于上层参数传递；

draw_dsc：绘制相关的指针，来自于上层参数传递；

draw_area：绘制区域相关信息，由上层相关参数获得。

（2）示例

```
void lv_draw_nsgpu_mask_rect (lv_draw_task_t * t, const lv_draw_mask_rect_dsc_t * draw_dsc, const lv_area_t * coords)
{
    lv_area_t clip_area;
    bool flag;

    flag = lv_area_intersect (&clip_area, coords, &t->clip_area);
    if (! flag) return;
    lv_area_move (&clip_area, -t->target_layer->buf_area.x1, -t->target_layer->buf_area.y1);
    nsgpu_lvgl_mask_rect_proc (t, draw_dsc, clip_area);
}
```

3.4.8. 边框处理

```
void nsgpu_lvgl_border_proc (lv_draw_task_t *t, const lv_draw_border_dsc_t * draw_dsc, lv_area_t outer_area, lv_area_t inner_area)
```

该函数适用于在 lv_draw_nsgpu_border.c 的 void lv_draw_nsgpu_border(lv_draw_task_t * t, const lv_draw_border_dsc_t * draw_dsc, const lv_area_t * coords)中调用。

（1）参数说明

t：绘制任务相关的指针，来自于上层参数传递；

draw_dsc：绘制相关的指针，来自于上层参数传递；

outer_area：绘制边框外区域相关信息，由上层相关参数获得；

inner_area：绘制边框内区域相关信息，由上层相关参数获得。

（2）示例

```
void lv_draw_nsgpu_border(lv_draw_task_t * t, const lv_draw_border_dsc_t * draw_dsc, const lv_area_t * coords)
{
    if(draw_dsc->opa <= LV_OPA_MIN) return;
    if(draw_dsc->width == 0) return;
    if(draw_dsc->side == LV_BORDER_SIDE_NONE) return;

    lv_area_t inner_area;
    int32_t width = draw_dsc->width;
    inner_area.x1 = coords->x1 + ceil (width / 2.0f);
    inner_area.x2 = coords->x2 – floor (width / 2.0f);
    inner_area.y1 = coords->y1 + ceil (width / 2.0f);
    inner_area.y2 = coords->y2 – floor (width / 2.0f);
    lv_area_move (&inner_area, -t->target_layer->buf_area.x1, -t->target_layer->buf_area.y1);

    lv_area_t clip_area;
    lv_area_copy (&clip_area, &t->clip_area);
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

```
    lv_area_move (&clip_area, -t->target_layer->buf_area.x1, -t->target_layer->buf_area.y1);

    lv_area_t clip_coords;
    bool flag;
    flag = lv_area_intersect (&clip_coords, &inner_area, &clip_area);
    if (! flag) return;

    lv_area_t outer_area;
    lv_area_copy (&outer_area, coords);
    nsgpu_lvgl_border_proc (t, draw_dsc, outer_area, inner_area);
}
```

3.4.9. 三角形处理

void nsgpu_lvgl_triangle_proc (lv_draw_task_t *t, const lv_draw_triangle_dsc_t * draw_dsc, lv_area_t draw_area)

该函数适用于在 lv_draw_nsgpu_triangle.c 的 void lv_draw_nsgpu_triangle (lv_draw_task_t * t, const lv_draw_triangle_dsc_t * draw_dsc)中调用。

（1）参数说明

t：绘制任务相关的指针，来自于上层参数传递；

draw_dsc：绘制相关的指针，来自于上层参数传递；

draw_area：绘制区域相关信息，由上层相关参数获得。

（2）示例

```
void lv_draw_nsgpu_triangle (lv_draw_task_t * t, const lv_draw_triangle_dsc_t * draw_dsc)
{
    if(draw_dsc->opa <= (lv_opa_t) LV_OPA_MIN)
        return;

    lv_area_t clip_area;
    lv_area_t tri_area;

    tri_area.x1 = LV_MIN3(draw_dsc->p[0].x, draw_dsc->p[1].x, draw_dsc->p[2].x);
    tri_area.y1 = LV_MIN3(draw_dsc->p[0].y, draw_dsc->p[1].y, draw_dsc->p[2].y);
    tri_area.x2 = LV_MAX3(draw_dsc->p[0].x, draw_dsc->p[1].x, draw_dsc->p[2].x);
    tri_area.y2 = LV_MAX3(draw_dsc->p[0].y, draw_dsc->p[1].y, draw_dsc->p[2].y);

    bool flag;
    flag = lv_area_intersect(&clip_area, &tri_area, &t->clip_area);
    if (! flag) return;
    lv_area_move (&clip_area, -t->target_layer->buf_area.x1, -t->target_layer->buf_area.y1);
    nsgpu_lvgl_triangle_proc(t, draw_dsc, clip_area);
}
```


国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

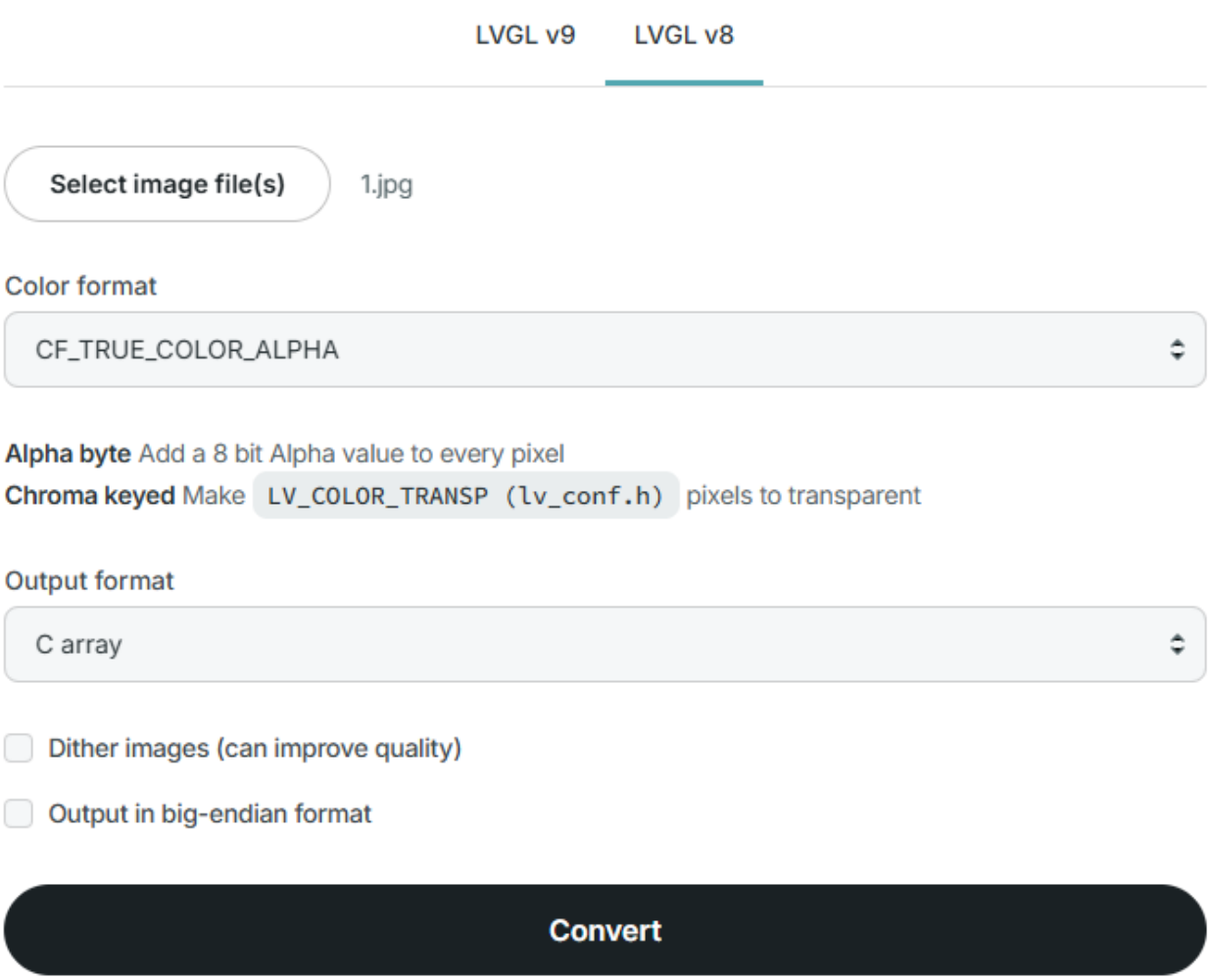
3.5. API 调用

（1）上电后，按模块配置所述，对 GPU 的 PWR、RCC、NVIC 进行配置；
（2）直接或基于 LVGL 等框架调用 GPU 初始化函数 nsgpu_init，对 GPU 进行初始化；
（3）根据需要调用相应的 GPU 功能函数。
注意：以上 API 参数中包含的 u32 *dst 和 u32 *src 等基地址，涉及内存对齐，为保证 API 功能正常、性能更佳，建议基地址至少应 64 位对齐，最好能 256 位对齐（十六进制末尾至少有两个 0）。

3.6. 其他说明

图片格式若为 RGB565A8，转换为数组数据时，需确保数据按 RGB565A8、RGB565A8….顺序排列，不能是 RGB565、RGB565….A8、A8….顺序。

可以参考 https://lvgl.io/tools/imageconverter，选择 LVGL v8 的 CF_TRUE_COLOR_ALPHA 格式对 RGB565A8 图片进行数据转换，得到数组文件，如下图所示：



国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

4. 版本历史

版本	日期	备注
V1.0.0	2025-07-23	创建文档
V1.0.1	2026-01-09	增加 GPU 通用 API 库圆弧和扇形绘制等说明，专用 API 库线段、圆弧、矩形、边框、三角形处理等说明

国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900 传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057

5. 声明

国民技术股份有限公司（下称“国民技术”）对此文档拥有专属产权。依据中华人民共和国的法律、条约以及世界其他法域相适用的管辖，此文档及其中描述的国民技术产品（下称“产品”）为公司所有。

国民技术在此并未授予专利权、著作权、商标权或其他任何知识产权许可。所提到或引用的第三方名称或品牌（如有）仅用作区别之目的。

国民技术保留随时变更、订正、增强、修改和改良此文档的权利，恕不另行通知。请使用人在下单购买前联系国民技术获取此文档的最新版本。

国民技术竭力提供准确可信的资讯，但即便如此，并不推定国民技术对此文档准确性和可靠性承担责任。

使用此文档信息以及生成产品时，使用者应当进行合理的设计、编程并测试其功能性和安全性，国民技术不对任何因使用此文档或本产品而产生的任何直接、间接、意外、特殊、惩罚性或衍生性损害结果承担责任。

国民技术对于产品在系统或设备中的应用效果没有任何故意或保证，如有任何应用在其发生操作不当或故障情况下，有可能致使人员伤亡、人身伤害或严重财产损失，则此类应用被视为“不安全使用”。

不安全使用包括但不限于：外科手术设备、原子能控制仪器、飞机或宇宙飞船仪器、所有类型的安全装置以及其他旨在支持或维持生命的应用。

所有不安全使用的风险应由使用人承担，同时使用人应使国民技术免于因为这类不安全使用而导致被诉、支付费用、发生损害或承担责任时的赔偿。

对于此文档和产品的任何明示、默示之保证，包括但不限于适销性、特定用途适用性和不侵权的保证责任，国民技术可在法律允许范围内进行免责。

未经明确许可，任何人不得以任何理由对此文档的全部或部分进行使用、复制、修改、抄录和传播。

国民技术股份有限公司 NSING TECHNOLOGIES INC.
地址：深圳市南山区高新北区宝深路109号国民技术大厦
电话：+86-755-86309900　　传真：+86-755-86169100
网址：https://www.nsingtech.com 邮编：518057