N32H7xx series GPU to LVGL user guide

Introduction

This document is designed for users to quickly understand the method of porting the 2.5D GPU of the N32H7xx series microcontrollers (MCUs) to LVGL, in order to guide users to easily run the LVGL software in GPU mode.

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

Content

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

1 Overview

Welcome to the N32H7xx series chips. This document introduces the method of porting the 2.5D GPU (abbreviated as NSGPU) of the N32H7xx series microcontrollers (MCUs) to LVGL, aiming to guide users in more easily running LVGL with GPU mode.

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

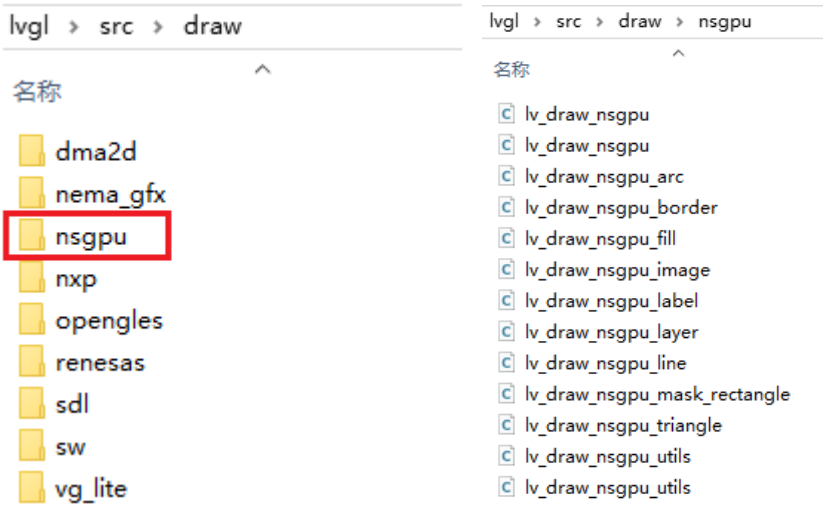
2 Porting guide of LVGL

LVGL runs in pure software mode by default. If it is changed to GPU mode for graphics rendering, data transfer, etc., it will greatly improve the system's operating efficiency. The LVGL porting guide introduces embedding NSGPU into LVGL and running LVGL using GPU mode according to necessary configurations. The LVGL version used is V9.3.0, and other versions may have API incompatibility issues.

2.1 Add files

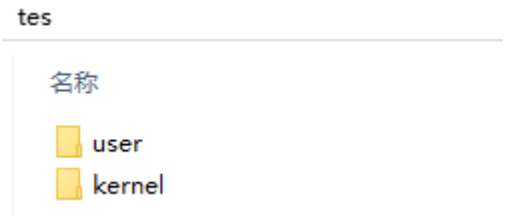
2.1.1 LVGL catalog

As shown in the following figure, add the nsgpu folder in the lvgl/src/raw directory, which contains the support files for the connection between nsgpu and LVGL related controls:



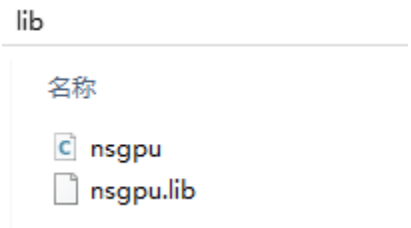
2.1.2 GPU driver

Add the tes folder to the middleware/GPU directory, which contains GPU related drivers..



2.1.3 LIB library

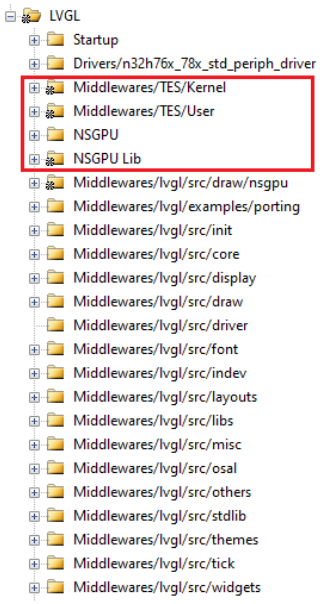
Add the lib folder to the middleware/GPU directory, which contains GPU related API libraries.



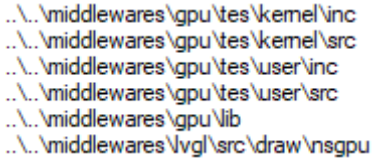
NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

2.2 Join the project

The engineering diagram is shown below, and the files in the nsgpu, tes, lib folders and LVGL related files need to be added separately.



2.3 Include path



Add the header file #include "nsgpu.h" to the project file that needs to be used (such as main.c).

2.4 Code adaptation

2.4.1 lvgl/src/init.c

(1)Add header files to this file

```
#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)Add initialization function lv_draw_nsgpu_init() to void lv_init(void)

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

```
#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)Add deinit function lv_draw_nsgpu_deinit() to void lv_deinit(void)

```
#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
```

2.4.2 **lvgl/src/lv_conf_internal.h**

Add LV_USE_DRAW_NSGPU macro definition:

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

```
#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/src/lv_conf.h**

(1) Add the macro definition LV_USE_DRAW_NSGPU in lv_conf.h or a header file containing the macro definition, such as # include<userconfig.h>. The macro definition is as follows:

#define LV_USE_DRAW_NSGPU 1

(2) When using NSGPU, LV_USE_DRAW_SW needs to be set to 0, which means the default pure software mode is turned off and LVGL is run in GPU mode instead.

```
#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)Add macro definition LV_GPU_NATIONS_CMSIS_INCLUDE

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

```
/*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) If there is not enough code and memory space, users can consider disabling some control functions (changing macro definition 1 to 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
```

2.4.4 lvgl/examples/porting/lv_port_disp_template.c

(1)void lv_port_disp_init(void).

The LVGL cache supports three modes: single cache partial refresh, dual cache partial refresh, and dual cache full screen refresh. Users can choose the cache mode according to their application needs, and the cache size for partial refresh mode is generally recommended to be 1/10 of the screen size. Taking partial refresh mode as an example, the dual cache buf_2_1 [] and buf_2_2 [] of LVGL are assigned to the sdram_area space as follows:

```
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);
```

The sdram_area space is defined in the scatter load file (. sct). Please refer to the relevant documentation for the scatter load file for specific configuration methods.

(2)static void disp_init(void) add LCD initialization function

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

For example: LCDC_DispImage((u32)0xC0000000);

(3) static void disp_flush(lv_display_t * disp_drv, const lv_area_t * area, uint8_t * px_map) add refresh function

For example, for a screen with a resolution of 800 x 480, using partial refresh method, the refresh cache size is 800 x 48, the format is RGB565, and the base address of the screen display cache is 0xC0000000. The reference for copying and calling through nsgpu is as follows:

```
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

If need to use the touch function of the screen, users need to improve the connection function between touch and LVGL. The reference example is as follows:

(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)
        {
            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];
}
```


NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

2.5 Interrupt call

2.5.1 LVGL heartbeat

Call lv_tick_inc (1) once every 1ms using SysTick or timer, which provides the necessary ticking heartbeat for LVGL to function properly.

2.5.2 LCD interrupt

According to application needs, LCD related interrupts can be used, with commonly used interrupts being EV and ER interrupts.

(1)EV interrupt supports generating interrupts after a row refresh is completed, and users can perform some simple application processing during the interrupt.

```
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 interrupt is generated when an error occurs.

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

  NVIC_InitStructure.NVIC_IRQChannel                    = LCD_ER_IRQn;
  NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority  = 0x00;
  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 Touch interrupt

If using the touch function of the screen and scanning with interrupt mode, the corresponding interrupt needs

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

to be initialized first, and then the interrupt handling function needs to be called. The reference examples are as follows:

(1) Initialize the corresponding interrupt (usually called after touch initialization)

```
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) Interrupt handling function

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

2.5.4 GPU interrupt

The GPU interrupt handling function has been implemented in the GPU driver, but in order for the GPU to function properly, the PWR, RCC, and NVIC of the GPU need to be configured in the main function before calling the GPU initialization function for configuration.

2.6 Called in main

2.6.1 Basic Initialization

The system operation requires initializing the basic modules in the main function, including PWR, RCC, GPIO, NVIC, etc. If using a distributed load file and defining SDRAM space within it, it is necessary to initialize PWR, RCC, GPIO, SDRAM, etc. before entering the main function. It is recommended to initialize void SystemInit (void)

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

in system_n32h76x_78x.c by register mode:

```
#ifdef CORE_CM7

  /* Configure the Vector Table location add offset address ------------------*/
#ifdef VECT_TAB_SRAM
  SCB->VTOR = 0x24000000;                              /* Vector Table Relocation in Internal SRAM */
#else
  SCB->VTOR = FLASH_BANK1_BASE | VECT_TAB_OFFSET;      /* Vector Table Relocation in Internal FLASH */
#endif

   /*defaule TCM_SIZE=0x3f,All TCMSRAM are AXI_SRAM,if you want to use ITCM/DTCM, define INIT_TCM_SIZE*/
#ifdef INIT_TCM_SIZE
    ConfigTcmSize(TCM_SIZE_VALUE);
#endif
    /*User can change the way of power supply */
    //PWR_ConfigSupply(PWR_SUPPLY_SELECTION);

    PreInitialize();

#else
#error Please #define CORE_CM4 or CORE_CM7
#endif
#endif
```

The PreInitialize (void) function and its macro definition and function reference code are as follows (the LCDC pixel clock frequency needs to be adjusted according to practical applications):

```
#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;
   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;
   }
```


NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

```
}
//
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
  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;
```


NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

```
__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;
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);
```


NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

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

GPIOE->AFH= 0x00000000;
GPIOE->AFL= 0x0FFFFF00;
GPIOE-> POTYPE= 0x00000000;
GPIOE-> PMODE= 0xAAAABFFA;
```


NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

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


NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

```
    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 Timing initialization

SysTick or timer is commonly used in the system to time, taking SysTick timing of 100us as an example:

```
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 initialization

Before LVGL calls the GPU initialization function to configure, it is necessary to configure the GPU's PWR, RCC, and NVIC (which have already been implemented in the basic initialization and do not need to be called again). The relevant implementations are as follows:

(1) Enable PWR

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

(2) Enable RCC

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

It is recommended to use the highest system frequency by calling RCC_SettysClkToMode0() in the RCC suite.

(3) Enable interrupt

```
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);
}
```

Users can choose to configure interrupt priority grouping according to their needs before calling NVIC_Configuration (void), for example:

```
NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);
```


NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

2.6.4 System initialization

Initialize LVGL system by calling functions lv_init().

2.6.5 Display initialization

Initialize display by calling functions lv_port_disp_init().

2.6.6 Device initialization

Initialize touch and other input devices by calling the function lv_port_indev_init().

2.6.7 Periodically calling

Call the lv_task_handler() function in the main loop, and the running cycle is at least 1ms.

2.7 Application example

After completing the previous porting, LVGL can be run through GPU mode to achieve user interface display and event processing.

2.7.1 Interface display

As shown below, for a simple interface display 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,
};
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);
```


NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

```
        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 Event handling

By calling lv_obj_add_event_cb(screen, click_screen_cb, LV_EVENT_RELEASED, 0), the click_screen_cb event handling function is added to perform corresponding operations, such as switching between different interfaces, when LV_EVENT_RELEASED is activated.

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


NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

3 GPU Usage Guide

GPU usage guide, mainly introducing the use of GPU API libraries, including general APIs and specialized APIs (only applicable to certain platforms). The following dedicated API for LVGL corresponds to the LVGL version V9.3.0, and other versions may have API incompatibility issues.

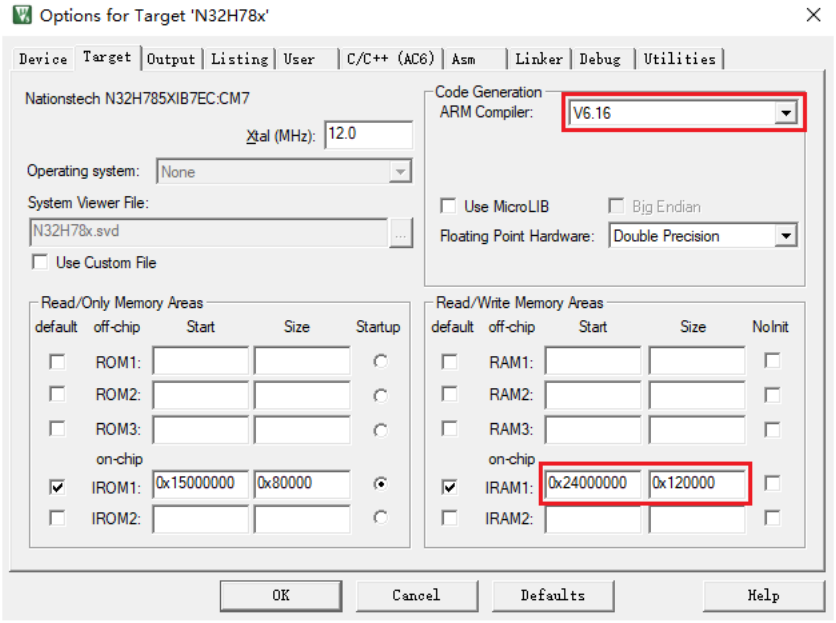
3.1 Basic environment

3.1.1 KEIL configuration

The partial configuration interface of KEIL/Project/Options for target... is as follows:

(1) The compiler recommends using V6 (note that if C++is used in V6, it is not compatible with MicroLIB).

Try to configure RAM as large as possible and ensure the required framebuffer size for GPU initialization configuration, otherwise compilation errors may occur.

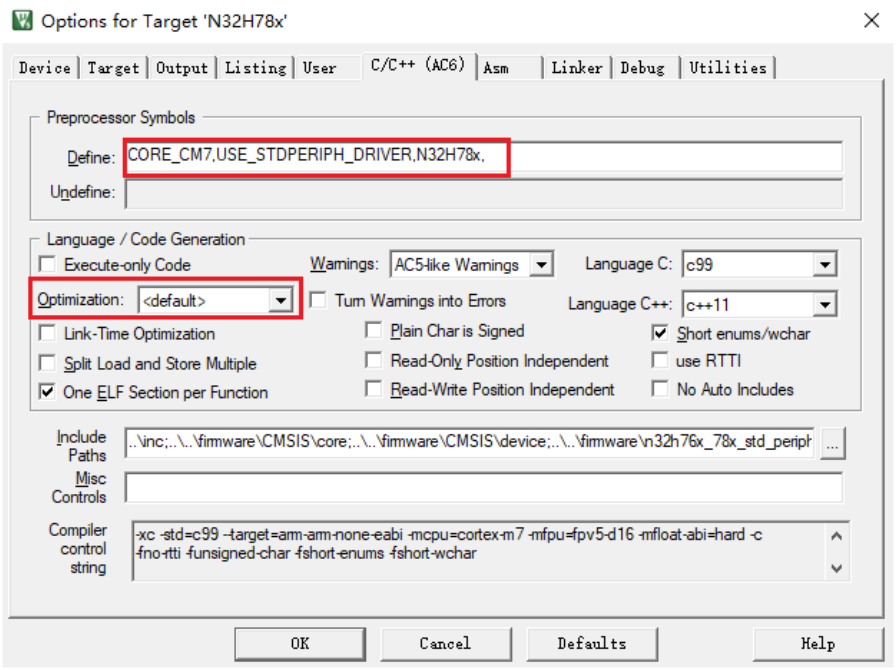


It can also be configured through decentralized file loading. For specific methods, refer to the documentation related to decentralized file loading.

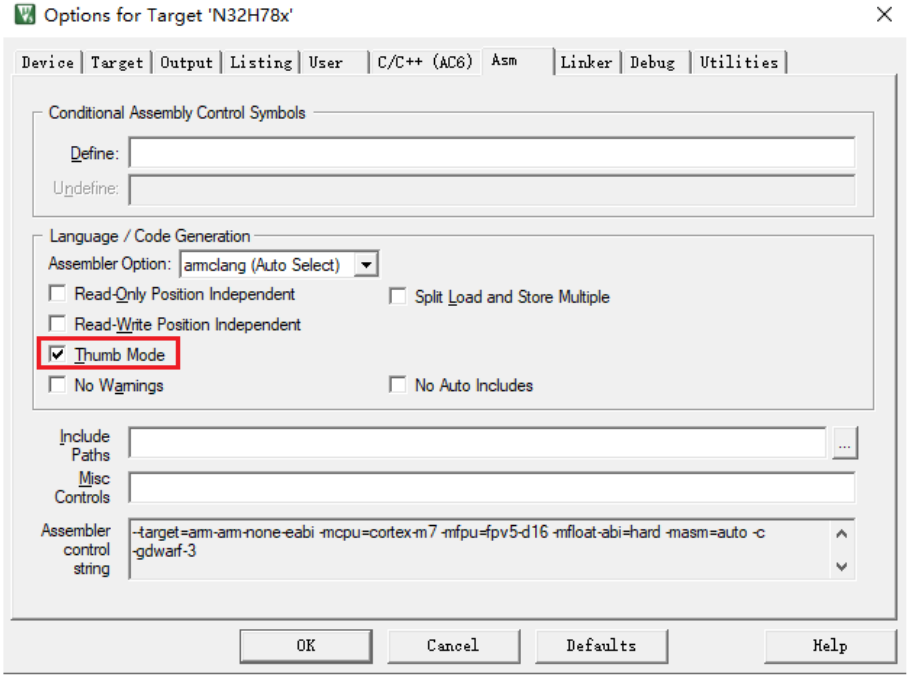
(2) Macro definition and optimization items, etc

Macro definitions are used to support the normal use of basic packages, with optimization options using <default> or -O0.

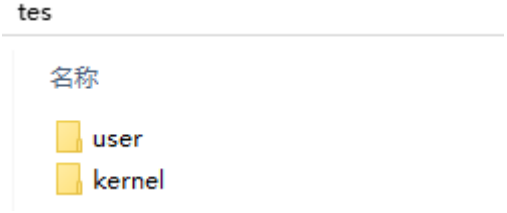
NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg



(3) Instruction set using Thumb Mode



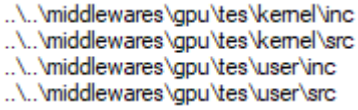
3.1.2 **Bottom driver**



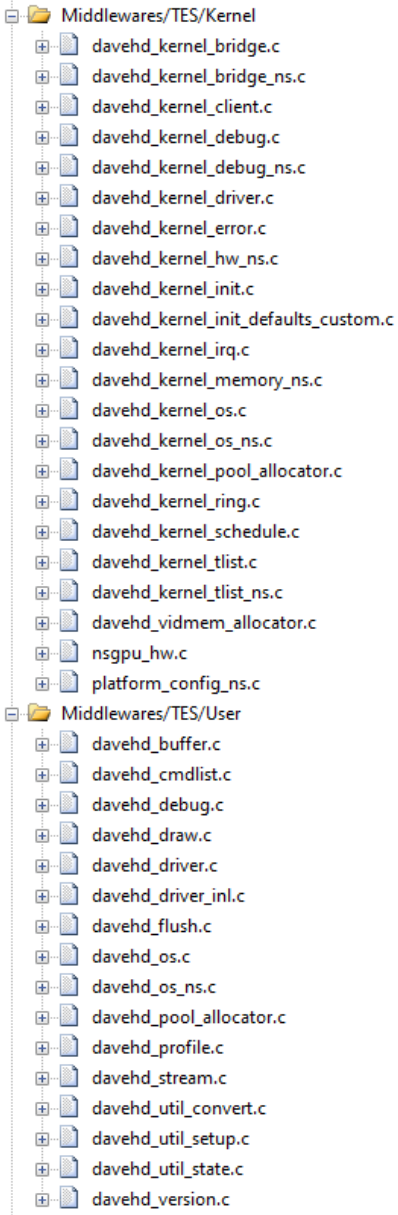
As shown in the following figure, it is necessary to add the path containing the underlying driver files in the

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

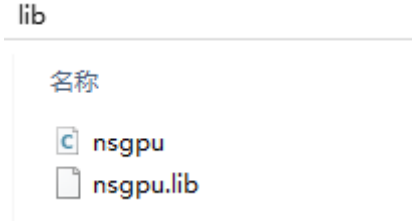
Options of the KEIL project.



And add the corresponding code to the KEIL project:



3.1.3 **API library**



In the Options of the KEIL project, add nsgpu.h to the include path and nsgpu. lib to the code project. Add the header file # include "nsgpu.h" to the project file that needs to be used (such as main.c).

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

3.1.4 Heap and stack configuration

Configure the size of the heap and stack in startup_n32h78x_cm7.s, with Heap_Size requiring at least 0x2000 and preferably 0x10000, and Stack_Size capable of configuring 0x2000.

3.2 Basic module

It is recommended to use the highest system frequency by calling RCC_SetSysClkToMode0() in the RCC suite.

3.2.1 Basic peripherals

To make the GPU work properly, before calling the GPU initialization function to configure, it is necessary to configure the PWR, RCC, and NVIC of the GPU. The relevant implementations are as follows:

(1) Enable PWR

PWR_MoudlePowerEnable(GRAPHIC_GPU_PWRCTRL,ENABLE);

(2) Enable RCC

RCC_EnableAXIPeriphClk3(RCC_AXI_PERIPHEN_M7_GPU|RCC_AXI_PERIPHEN_M7_GPULP,ENABLE);

(3) Enable interrupt (The interrupt handling function GPU_IRQHandler(void) has been implemented in the driver library)

```
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);
}
```

Users can choose to configure interrupt priority grouping according to their needs before calling NVIC_Configuration (void), for example:

```
NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);
```

3.2.2 Extended memory

The memory required for GPU operation is determined by factors such as resolution, pixel bytes, and cache counts. When there is sufficient internal SRAM, unused SRAM addresses such as 0x30000000 or 0x24040000 can be used, otherwise external SDRAM addresses are required.

In nsgpu.h, GPU_MEMORY_BASE_ADDRESS defaults to 0xC0C00000 and can be modified according to actual situations. If the SDRAM space is not defined in the distributed load file, external extended memory can be used after initializing the SDRAM in main. Please refer to the SDRAM demo for details.

3.2.3 Display module

In most cases, the display module LCDC is required in applications, and the use of related functions can refer to the LCDC demo.

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

3.3 General API library description

3.3.1 Initialization

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

(1) Parameter description

gpu_memory_base: The base address allocated to GPU memory. When the result of width * height * pixel-bytes * fb_num is less than 300KB, it is recommended to use unoccupied internal SRAM addresses such as 0x30000000 or 0x24040000. Otherwise, external SDRAM addresses should be used. In nsgpu.h, gpu_memory_base defaults to 0xC0C00000 and can be modified according to actual situations.

fb_num: The number of framebuffers requested, which will occupy memory, is generally set to 1 and cannot exceed FRAME_BUF_MAX.

width: Apply for the width of the framebuffer.

height: Apply for the height of the framebuffer.

pixel_bytes: Set the number of pixel bytes based on the pixel format, with A8, L8 and etc. set to 1, RGB565 and etc. set to 2, RGB556A8 and etc. set to 3, ARGB8888 and etc. set to 4.

(2) Return value description

The initialization status returned is as follows:

-1: Parameter setting exception

-2: Platform error

-3: Device error or program stuck

0: Insufficient memory

1: Normal

(3) Example

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

The SCREEN_WIDTH_SIZE and SCREEN_HEIGT_SIZE refer to the screen width and screen height respectively.

3.3.2 Deinit function

void nsgpu_deinit(void)

GPU deinit to default state, usually only used in case of system exceptions.

3.3.3 Get ready cache

u32 nsgpu_get_ready_framebuffer (u8 index)

After nsgpu_init initialization is completed, use this function to obtain the rendering cache corresponding to the cache number that has been initialized and ready. In general, it is not necessary to call.

(1) Parameter description

index: It is the cache index of the framebuffer, with a value range between 0 and (fb_num-1). The fb_num is configured during the initialization of the nsgpu_init function.

(2) Return value description

The return value is the first address of the initialized rendering cache corresponding to the cache index.

(3) Example

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

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);

3.3.4 Select ready cache

void nsgpu_set_framebuffer (u8 index)

After the initialization of nsgpu_init is completed, use this function to set the cache number corresponding to the initialized ready cache as the rendering cache for the current work. In general, it is not necessary to call.

(1) Parameter description

index: It is the cache index of the framebuffer, with a value range between 0 and (fb_num-1). The fb_num is configured during the initialization of the nsgpu_init function.

(2) Example

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

nsgpu_get_ready_framebuffer (0); or nsgpu_get_ready_framebuffer (1);

3.3.5 Get current cache

u32 nsgpu_get_framebuffer(void)

After the initialization of nsgpu_init is completed, call this function to obtain the current working rendering cache.

(1) Return value description

The return value is the first address of the current rendering cache.

(2) Example

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 Change current cache

void nsgpu_set_framebuffer (u32 addr)

After the initialization of nsgpu_init is completed, this function can be used to change the currently working rendering cache that is ready for initialization to an uninitialized cache. It is generally not recommended to make changes.

(1) Parameter description

addr: It is the first address to be configured as a rendering cache.

(2) Example

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

nsgpu_set_framebuffer (0xC0000000);

3.3.7 Batch data filling

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)

After the initialization of nsgpu_init is completed, this function can be used to change the currently working rendering cache that is ready for initialization to an uninitialized cache. It is generally not recommended to make changes.

(1) Parameter description

dst: Target address for color data filling;

dst_width: The total width of the area, usually the same as the width of the filled area.

dst_ height: The total height of the area, usually the same as the height of the filled area.

dst_startx: The x-direction offset value based on the target address, set as needed, usually 0.

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

dst_starty: The y-direction offset value based on the target address, set as needed, usually 0.

color: The filled color value is always in ARGB8888 format. If nsgpu_init () initializes pixel-bytes=4, the filled value is the same as the color value. If pixel-bytes=2, the filled value is the value converted from color to RGB565, and other pixel-bytes values will also be converted to corresponding format values.

alpha: The transparency of the filling, ranging from 0 to 255, where 0 represents full transparency and 255 represents not transparent.

width: The width of the filled area.

height: The height of the filled area.

(2) Example

Fill 800 x 480 rectangular blocks starting from 0xC0000000, color 0xFFF0000 (red), not transparent.

3.3.8 Batch data copying

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) Parameter description

dst: Destination address for data copying.

dst_width: The width of the target area.

dst_ height: The height of the target area.

dst_startx: The x-direction offset value based on the target address, set as needed, usually 0.

dst_starty: The y-direction offset value based on the target address, set as needed, usually 0.

src: Source address for data copying.

width: The width of the data source area.

height: The height of the data source area.

pitch: The spacing between rows in the data source area, usually equal to the width.

pixel-bytes: Set the number of pixel bytes based on the pixel format, with A8, L8 and etc. set to 1, RGB565 and etc. set to 2, RGB556A8 and etc. set to 3, ARGB8888 and etc. set to 4.

(2) Example

The target area size is 300 x 180, and the rectangular block data of 300 x 180 is copied from the offset coordinates (20,5).

```
int result = nsgpu_init (GPU_MEMORY_BASE_ADDRESS, 1, 800, 480, 2);
memset((void*)0x30000000,0, 0x20000); // Set all dst areas to zero.
memset((void*)0x30020000,0x55, 0x20000); // Write 0x55 in the src area.
nsgpu_data_copy ((u32 *)0x30000000, 300, 180, 20, 5, (u32 *)0x30020000, 300, 180, 300, 2);
```

Debugging and checking the memory, it can be observed that 0x55 is written starting from 0x30000000+(300 * 5+20) * 2=0x30000BE0.

(3) Application

For example, in the LVGL refresh function, for a screen with a resolution of 800 x 480, using partial refresh mode, the refresh cache size is 800 x 48, the format is RGB565, and the base address of the screen display cache is 0xC0000000. Copy call as follows:

```
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);
```


NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

```
    }
    lv_display_flush_ready(disp_drv);
}
```

Note: Due to the GPU data handling mechanism, there may be slight byte differences between the data before and after handling, which is not a problem for image applications, but may not be applicable for communication and other data that require complete fidelity.

3.3.9 Set drawing color

void nsgpu_set_draw_color (u32 color)

After the initialization of nsgpu_init is completed, this function can be used to change the currently working rendering cache that is ready for initialization to an uninitialized cache. It is generally not recommended to make changes.

(1) Parameter description

color: Set the color of the drawing to ARGB8888 format.

(2) Example

nsgpu_set_draw_color (0xFF0000FF); // Set the drawing color to blue.

3.3.10 Set drawing alpha

void nsgpu_set_draw_alpha (u8 alpha)

After the initialization of nsgpu_init is completed, this function can be used to change the currently working rendering cache that is ready for initialization to an uninitialized cache. It is generally not recommended to make changes.

(1) Parameter description

alpha: The transparency of the filling, ranging from 0 to 255, where 0 represents full transparency and 255 represents not transparent.

(2) Example

nsgpu_set_draw_alpha (255); // Set the drawing transparency to 255.

3.3.11 Draw basic graphics

If want to draw points, users can directly fill the pixels through software, or achieve it by drawing lines or rectangles below.

3.3.11.1 Line

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

(1) Parameter description

(x1, y1) and (x2, y2): They are the coordinates of the two endpoints of the solid line.

width: The line width of a solid line, with a minimum value of 0 and a minimum variation value of 0.1.

(2) Example

nsgpu_set_draw_color(0xFFFF0000); // Set color as red.

nsgpu_vgdraw_line (60, 100, 360, 300， 1.0);

3.3.11.2 Rectangle

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

(1) Parameter description

(x, y): It is the coordinate of the top left corner vertex of the rectangle.

width: The horizontal width of a solid rectangle.

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

height: The vertical height of a solid rectangle.

(2) Example

nsgpu_set_draw_color(0xFFFF0000); // Set color as red.

nsgpu_vgdraw_rect (20, 20, 100, 50);

3.3.11.3 Triangle

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

(1) Parameter description

(x1, y1), (x2, y2), (x3, y3): They are the coordinates of the three vertices of a solid triangle. When calling the function, there is no order requirement for configuring the three vertices.

(2) Example

nsgpu_set_draw_color(0xFFFF0000); // Set color as red.

nsgpu_vgdraw_triangle (10, 50, 40, 40, 70, 70);

3.3.11.4 Circle

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

(1) Parameter description

(cx, cy): It is the center coordinate of a solid circle.

r: The radius of a circle.

(2) Example

nsgpu_set_draw_color(0xFFFF0000); // Set color as red.

nsgpu_vgdraw_circle (60, 60, 30);

3.3.11.5 Arc

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

(1) Parameter description

(cx, cy): It is the center coordinate of the arc.

r: The radius of the arc.

width: is the width of the arc.

start_angle: The starting angle of the arc.

src_angle: The angle corresponding to the arc (The horizontal right side is 0 degrees, increasing clockwise).

(2) Example

nsgpu_set_draw_color(0xFFFF0000); // Set color as red.

nsgpu_vgdraw_arc (60, 60, 30, 1, 0, 45);

3.3.11.6 Wedge

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

(1) Parameter description

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

(cx, cy): It is the center coordinate of the wedge.

r: The radius of the wedge.

Start_angle: The starting angle of the wedge.

Arc_angle: The angle corresponding to the wedge (The horizontal right side is 0 degrees, increasing clockwise).

(2) Example

```
nsgpu_set_draw_color(0xFFFF0000); // Set color as red.
nsgpu_draw_wedge (60, 60, 30,0, 45);
```

3.4 LVGL special API library description

3.4.1 Fill processing

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)

This function is applicable to be called by void lv_draw_nsgpu_fill (lv_draw_task_t * t, const lv_draw_fill_dsc_t * draw_dsc, const lv_area_t * coords) of lv_draw_nsgpu_fill.c.

(1) Parameter description

t: Draw task related pointers, derived from upper level parameter passing.

draw_dsc: Fill in relevant pointers, from upper level parameter passing.

draw_area: Draw area related information obtained from upper level related parameters.

coordinates: Coordinate related information obtained from upper level parameters.

(2) Example

```
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 Letter processing

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)

This function is applicable to be called by 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) of lv_draw_nsgpu_label.c.

(1) Parameter description

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

t: Draw task related pointers, derived from upper level parameter passing.

glyph_draw_dsc: Letter related pointers, from upper level parameter passing.

draw_area: Draw area related information obtained from upper level related parameters.

letter_area: Letter related information obtained from upper level parameters.

(2) Example

```
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;
    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;
```


NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

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

3.4.3 Image processing

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)

This function is applicable to be called by 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) of lv_draw_nsgpu_image.c.

(1) Parameter description

t: Draw task related pointers, derived from upper level parameter passing.

image_draw_dsc: Image related pointers, from upper level parameter passing.

draw_area: Draw area related information obtained from upper level related parameters.

image_area: Image related information obtained from upper level parameters.

(2) Example

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);
```


NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

```
    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 Layer processing

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)

This function is applicable to be called by 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) of lv_draw_nsgpu_layer.c.

(1) Parameter description

t: Draw task related pointers, derived from upper level parameter passing.

image_draw_dsc: Image related pointers, from upper level parameter passing.

draw_area: Draw area related information obtained from upper level related parameters.

image_area: Image related information obtained from upper level parameters.

(2) Example

```
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 Line processing

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

This function is applicable to be called by void lv_draw_nsgpu_line (lv_draw_task_t * t, const lv_draw_line_dsc_t * draw_dsc) of lv_draw_nsgpu_line.c.

(1) Parameter description

t: Draw task related pointers, derived from upper level parameter passing.

draw_dsc: Draw related pointers, from upper level parameter passing.

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

draw_area: Draw area related information obtained from upper level related parameters.

(2) Example

```
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;

    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 Arc processing

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

This function is applicable to be called by void lv_draw_nsgpu_arc (lv_draw_task_t * t, const lv_draw_arc_dsc_t * draw_dsc, const lv_area_t * coords) of lv_draw_nsgpu_arc.c.

(1) Parameter description

t: Draw task related pointers, derived from upper level parameter passing.

draw_dsc: Draw related pointers, from upper level parameter passing.

draw_area: Draw area related information obtained from upper level related parameters.

(2) Example

```
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;
```


NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

```
    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 Rectangle processing

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)

This function is applicable to be called by 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) of lv_draw_nsgpu_mask_rectangle.c.

(1) Parameter description

t: Draw task related pointers, derived from upper level parameter passing.

draw_dsc: Draw related pointers, from upper level parameter passing.

draw_area: Draw area related information obtained from upper level related parameters.

(2) Example

```
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 Border processing

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)

This function is applicable to be called by void lv_draw_nsgpu_border (lv_draw_task_t * t, const lv_draw_border_dsc_t * draw_dsc, const lv_area_t * coords) of lv_draw_nsgpu_border.c.

(1) Parameter description

t: Draw task related pointers, derived from upper level parameter passing.

draw_dsc: Draw related pointers, from upper level parameter passing.

outer_area: Draw information related to the outer border, obtained from upper level parameters

inner_area: Draw information related to the inner border, obtained from upper level parameters

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

(2) Example

```
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);
    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 Triangle processing

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

This function is applicable to be called by void lv_draw_nsgpu_triangle (lv_draw_task_t * t, const lv_draw_triangle_dsc_t * draw_dsc) of lv_draw_nsgpu_triangle.c.

(1) Parameter description

t: Draw task related pointers, derived from upper level parameter passing.

draw_dsc: Draw related pointers, from upper level parameter passing.

draw_area: Draw area related information obtained from upper level related parameters.

(2) Example

```
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;
```


NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

```
    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);
}
```

3.5 API call

(1) After powering on, configure the GPU's PWR, RCC, and NVIC according to the module configuration instructions.

(2) Directly or based on frameworks such as LVGL, call the GPU initialization function nsgpu_init to initialize the GPU.

(3) Call the corresponding GPU function as needed.

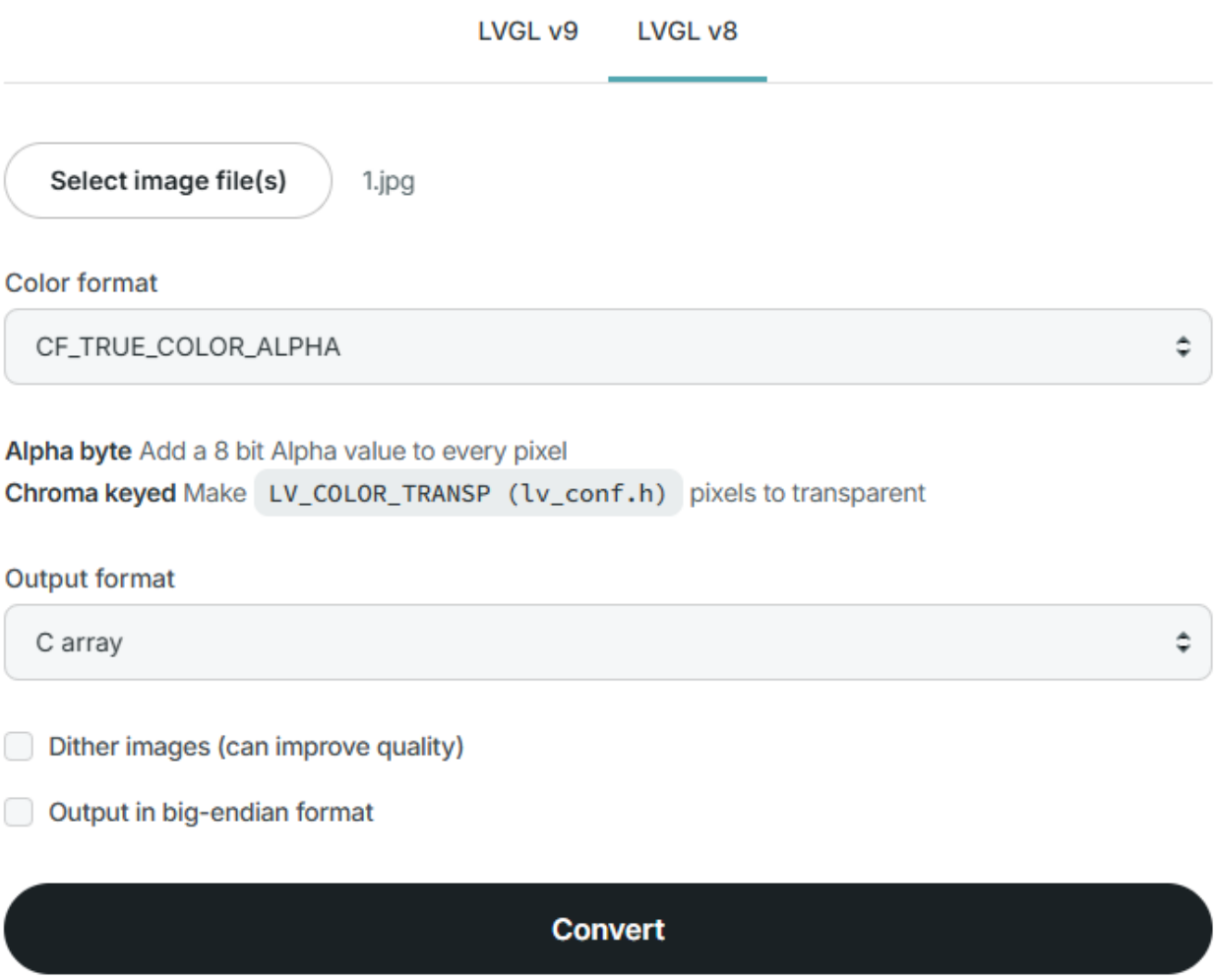
Attention: The base addresses such as u32 *dst and u32 *src included in the above API parameters involve memory alignment. To ensure normal API functionality and better performance, it is recommended that the base address should be aligned at least 64 bits, preferably 256 bits (with at least two zeros at the end of hexadecimal).

3.6 Other notes

If the image format is RGB565A8, ensure the data is arranged in the order of RGB565A8, RGB565A8... when converting to array data. The sequence should not be RGB565, RGB565...A8, A8…

You can refer to https://lvgl.io/tools/imageconverter, select the LVGL v8 CF_TRUE_COLOR_ALPHA format to convert RGB565A8 images into array files, as shown in the figure below:

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg



NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

4 History versions

Version	Date	Notes
V1.0.1	2026.1.9	Initial version

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg

5 Disclaimer

This document is the exclusive property of NSING TECHNOLOGIES PTE. LTD. (Hereinafter referred to as NSING). This document, and the product of NSING described herein (Hereinafter referred to as the Product) are owned by NSING under the laws and treaties of Republic of Singapore and other applicable jurisdictions worldwide. The intellectual properties of the product belong to NSING Technologies Inc. and NSING Technologies Inc. does not grant any third party any license under its patents, copyrights, trademarks, or other intellectual property rights. Names and brands of third party may be mentioned or referred thereto (if any) for identification purposes only. NSING reserves the right to make changes, corrections. enhancements, modifications, and improvements to this document at any time without notice. Please contact NSING and obtain the latest version of this document before placing orders. Although NSING has attempted to provide accurate and reliable information, NSING assumes no responsibility for the accuracy and reliability of this document. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. In no event shall NSING be liable for any direct, indirect, incidental, special, exemplary, or consequential damages arising in any way out of the use of this document or the Product. NSING Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, Insecure Usage'. Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, all types of safety devices, and other applications intended to supporter sustain life. All Insecure Usage shall be made at user's risk. User shall indemnify NSING and hold NSING harmless from and against all claims, costs, damages, and other liabilities, arising from or related to any customer's Insecure Usage Any express or implied warranty with regard to this document or the Product, including, but not limited to. The warranties of merchantability, fitness for a particular purpose and non-infringement are disclaimed to the fullest extent permitted by law. Unless otherwise explicitly permitted by NSING, anyone may not use, duplicate, modify, transcribe or otherwise distribute this document for any purposes, in whole or in part.

NSING Technologies Pte. Ltd.
Address: 20 Science Park, #03-15/16
Teleteck Park, East Wing, Singapore 117674
Email: sales@nsing.com.sg