LVGL移植高通字库

> 字库芯片: GT30L24A3W
>
> MCU: STM32F429
>
> LVGL版本：V8.4

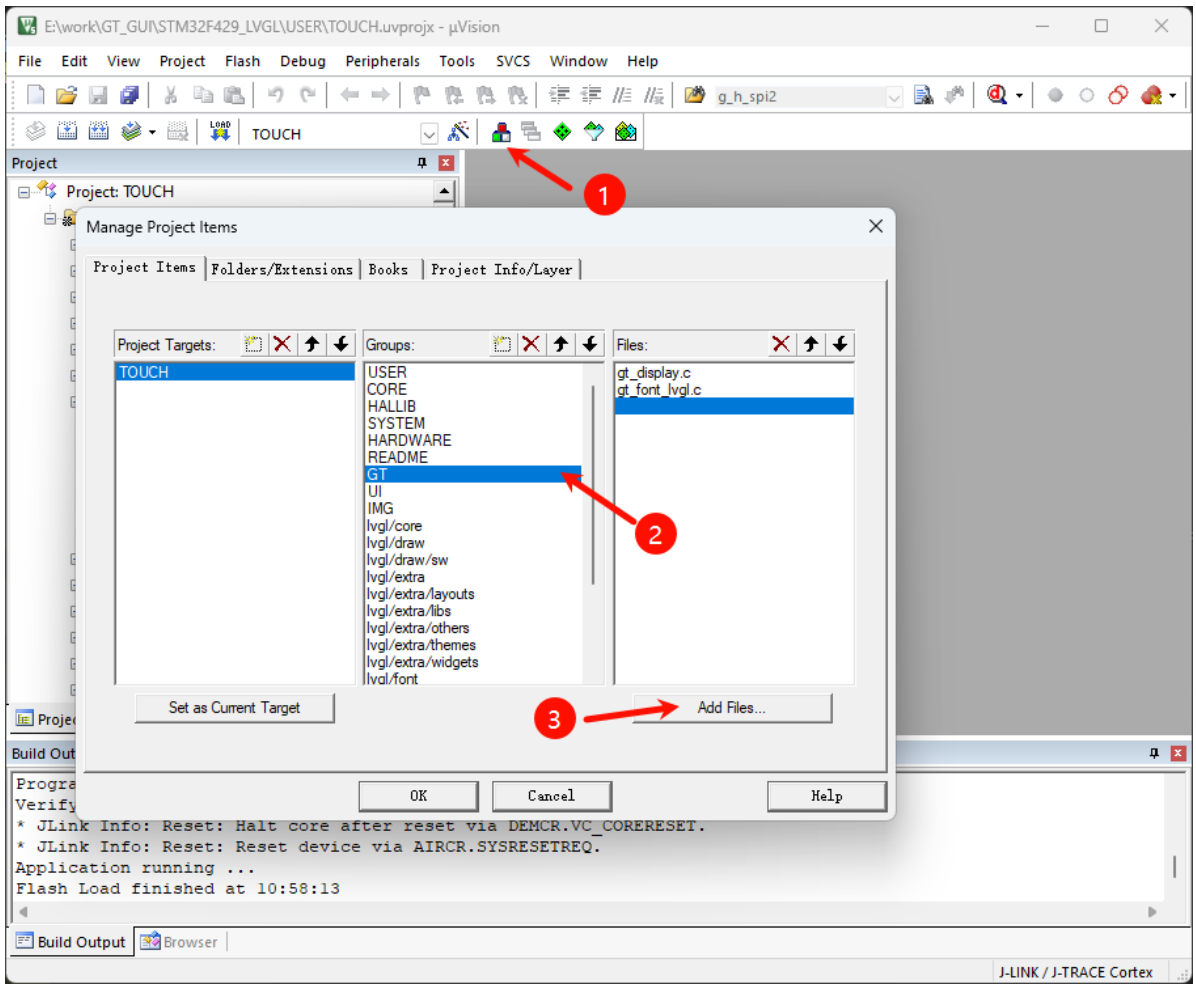
一、实现`gt_read_data()`和`r_dat_bat()`

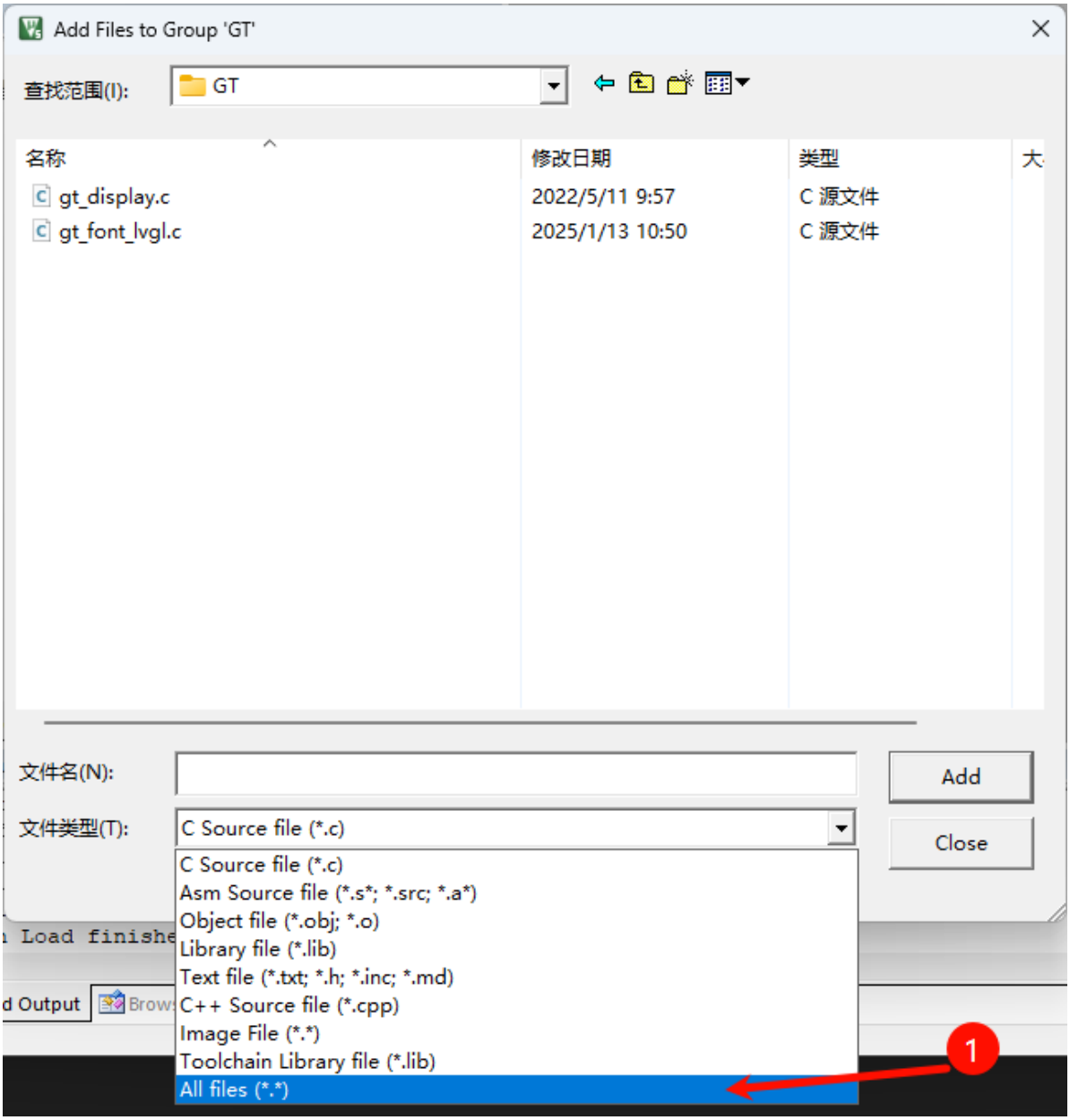
请参考下面视频

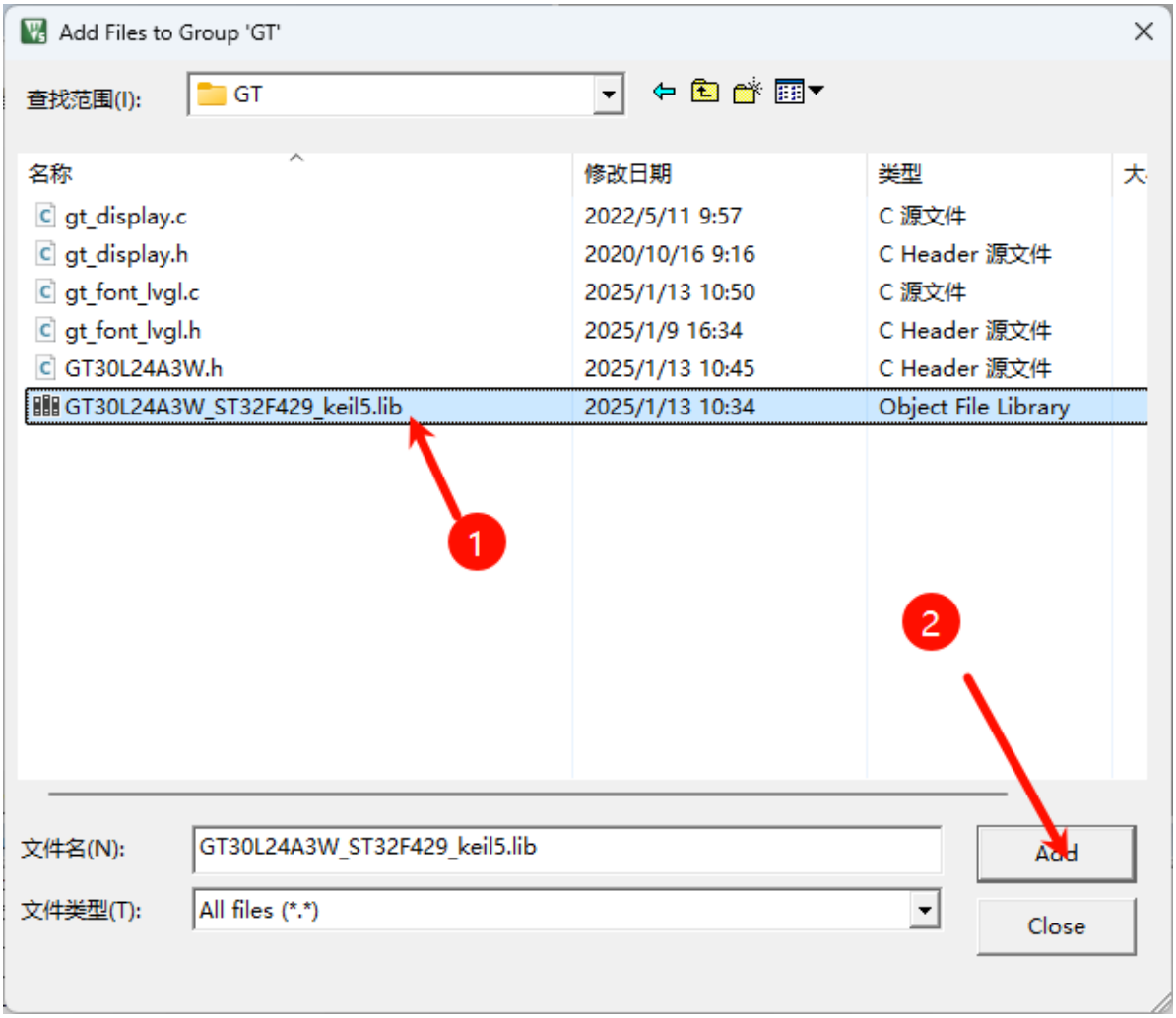
> 如何在 32 位 MCU 上使用高通点阵字库：https://www.bilibili.com/video/BV1aG41117uH/?spm_id_from=333.999.0.0
>
> 高通字库使用教程(1)硬件链接与注意事项部分：https://www.bilibili.com/video/BV1PxcyeZEWW/?vd_source=c084c395e4fcfac58df3ea21ada16b68
>
> 高通字库使用教程(2)SPI底层函数使用：https://www.bilibili.com/video/BV1CxcyeoEpJ/?vd_source=c084c395e4fcfac58df3ea21ada16b68
>
> 高通字库使用教程(3)SPI底层函数验证：https://www.bilibili.com/video/BV1CxcyeoEXM/?vd_source=c084c395e4fcfac58df3ea21ada16b68
>
> 高通字库使用教程(4)关于库函数的讲解：https://www.bilibili.com/video/BV1CxcyeoEnR/?vd_source=c084c395e4fcfac58df3ea21ada16b68

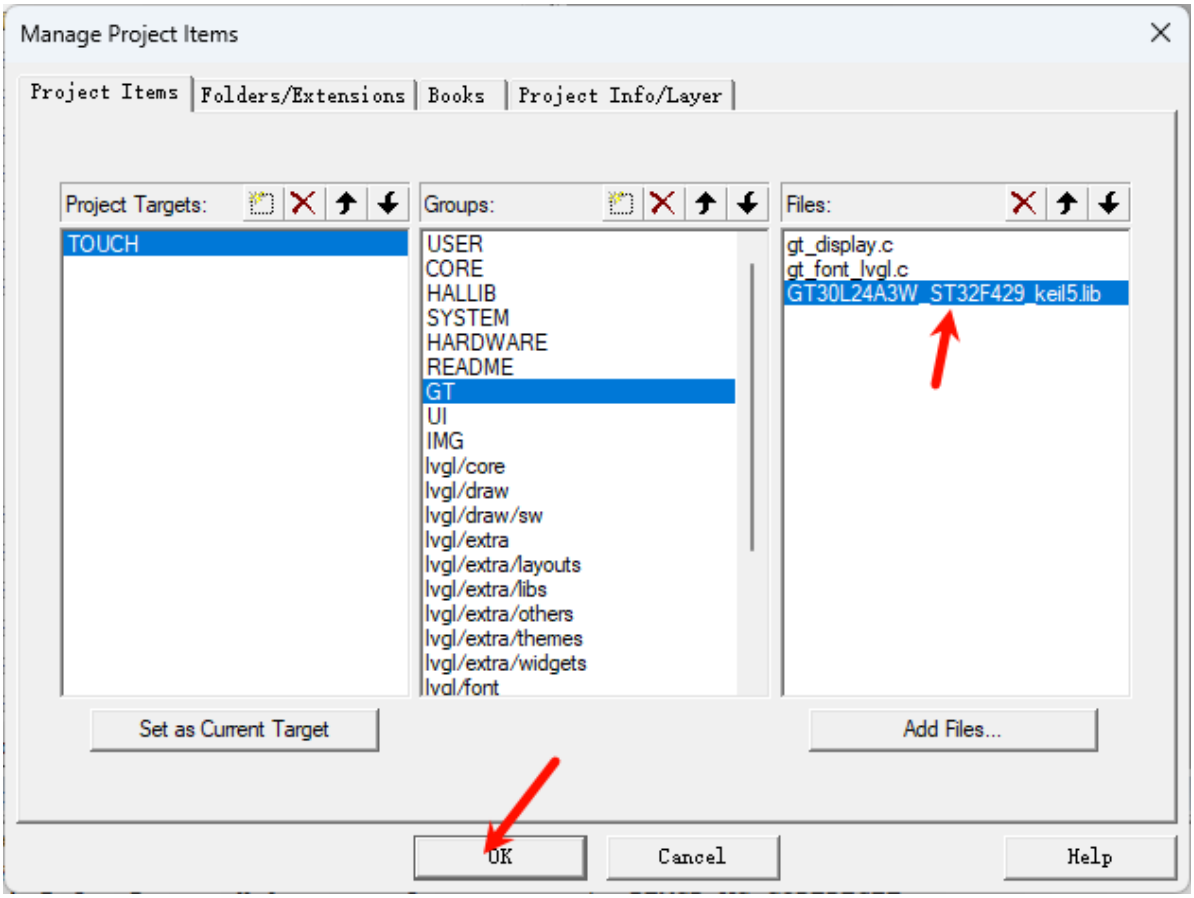
二、移植

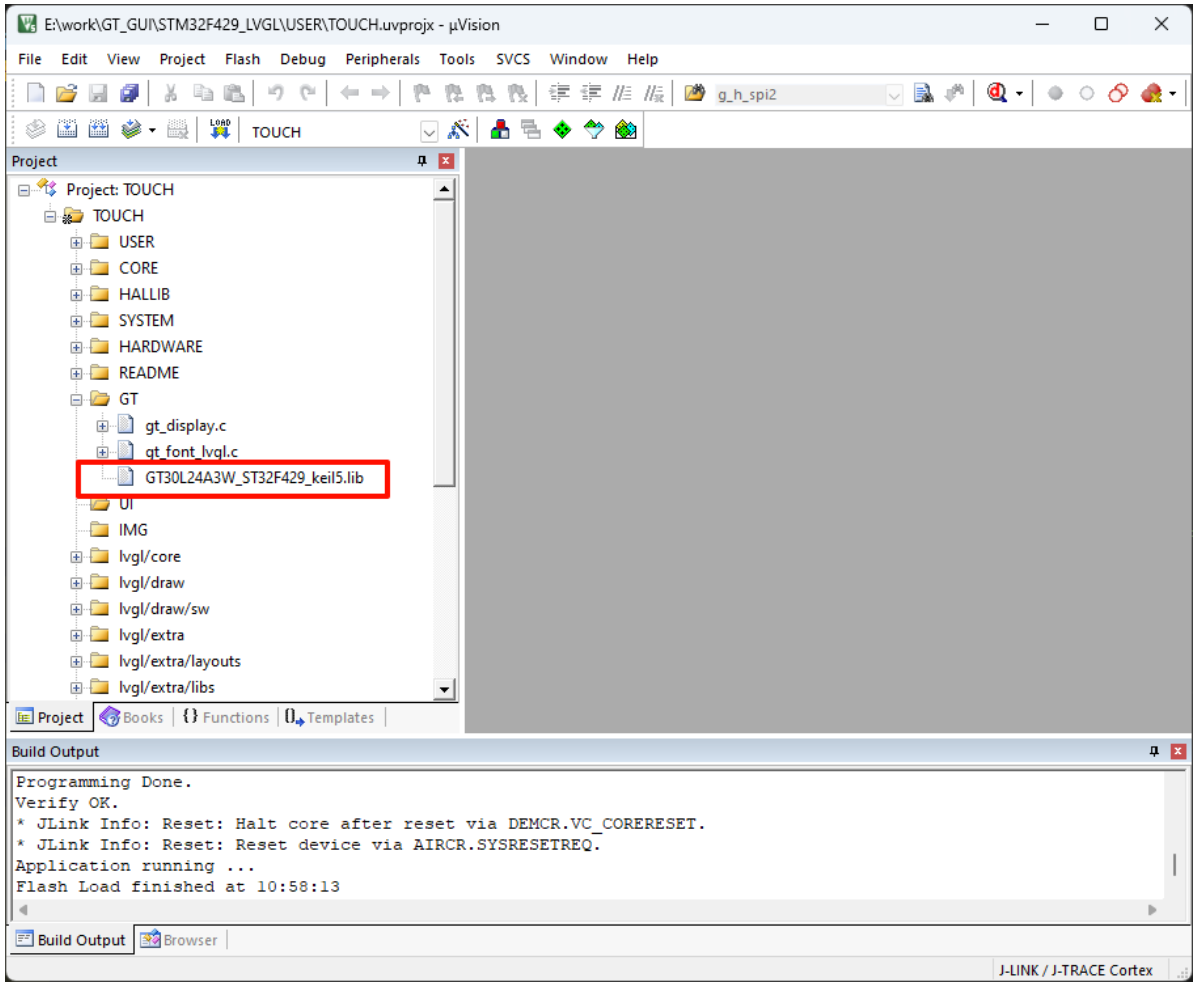
1、添加库文件（`.lib`或`.a`文件）到LVGL项目工程中


E:\work\GT_GUI\STM32F429_LVGL\USER\TOUCH.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
TOUCH
Project
Project: TOUCH
1
Manage Project Items
Project Items
Folders/Extensions
Books
Project Info/Layer
Project Targets:
TOUCH
Groups:
USER
CORE
HALLIB
SYSTEM
HARDWARE
README
GT
UI
IMG
lvgl/core
lvgl/draw
lvgl/draw/sw
lvgl/extra
lvgl/extra/layouts
lvgl/extra/libs
lvgl/extra/others
lvgl/extra/themes
lvgl/extra/widgets
lvgl/font
Files:
gt_display.c
gt_font_lvgl.c
2
Set as Current Target
3
Add Files...
OK
Cancel
Help
Build Out
* JLink Info: Reset: Halt core after reset via DEMCR.VC_CORERESET.
* JLink Info: Reset: Reset device via AIRCR.SYSRESETREQ.
Application running ...
Flash Load finished at 10:58:13
Build Output
Browser
J-LINK / J-TRACE Cortex


Add Files to Group 'GT'
查找范围(I):
GT
名称
修改日期
类型
大
gt_display.c
2022/5/11 9:57
C 源文件
gt_font_lvgl.c
2025/1/13 10:50
C 源文件
文件名(N):
文件类型(T):
C Source file (*.c)
Add
Close
C Source file (*.c)
Asm Source file (*.s*; *.src; *.a*)
Object file (*.obj; *.o)
Library file (*.lib)
Text file (*.txt; *.h; *.inc; *.md)
C++ Source file (*.cpp)
Image File (*.*)
Toolchain Library file (*.lib)
All files (*.*)
Load finishe
d Output
1


Add Files to Group 'GT'
查找范围(I):
GT
名称
修改日期
类型
gt_display.c
2022/5/11 9:57
C 源文件
gt_display.h
2020/10/16 9:16
C Header 源文件
gt_font_lvgl.c
2025/1/13 10:50
C 源文件
gt_font_lvgl.h
2025/1/9 16:34
C Header 源文件
GT30L24A3W.h
2025/1/13 10:45
C Header 源文件
GT30L24A3W_ST32F429_keil5.lib
2025/1/13 10:34
Object File Library
1
2
文件名(N):
GT30L24A3W_ST32F429_keil5.lib
文件类型(T):
All files (*.*)
Add
Close


Manage Project Items
Project Items
Folders/Extensions
Books
Project Info/Layer
Project Targets:
TOUCH
Groups:
USER
CORE
HALLIB
SYSTEM
HARDWARE
README
GT
UI
IMG
lvgl/core
lvgl/draw
lvgl/draw/sw
lvgl/extra
lvgl/extra/layouts
lvgl/extra/libs
lvgl/extra/others
lvgl/extra/themes
lvgl/extra/widgets
lvgl/font
Files:
gt_display.c
gt_font_lvgl.c
GT30L24A3W_ST32F429_keil5.lib
Set as Current Target
Add Files...
OK
Cancel
Help



2、初始化字库

返回值**大于0**即为初始化成功

```
int ret = GT_Font_Init();
printf("font init:%d\r\n",ret);
```

3、定义 `lv_font_t` 变量

```
#if LV_VERSION_CHECK(8, 0, 0)
const lv_font_t gt30l24a3w_montserrat_16 = {
#else
lv_font_t gt30l24a3w_montserrat_16 = {
#endif
    .get_glyph_dsc = _get_gt_font_glyph_dsc_16,    /*Function pointer to get glyph's data*/
    .get_glyph_bitmap = _get_gt_font_bitmap_fmt_txt_16,    /*Function pointer to get glyph's bitmap*/
    .line_height = 16,          /*The maximum line height required by the font*/
    .base_line = 3,             /*Baseline measured from the bottom of the line*/
#if !(LVGL_VERSION_MAJOR == 6 && LVGL_VERSION_MINOR == 0)
    .subpx = LV_FONT_SUBPX_NONE,
#endif
#if LV_VERSION_CHECK(7, 4, 0) || LVGL_VERSION_MAJOR >= 8
```

```
    .underline_position = -1,
    .underline_thickness = 1,
#endif
    .dsc = &font_dsc           /*The custom font data. Will be accessed by
`get_glyph_bitmap/dsc` */
};
```

4、定义 `lv_font_fmt_txt_dsc_t` 变量

```
#if LV_VERSION_CHECK(8, 0, 0)
/*Store all the custom data of the font*/
static  lv_font_fmt_txt_glyph_cache_t cache;
static const lv_font_fmt_txt_dsc_t font_dsc = {
#else
static lv_font_fmt_txt_dsc_t font_dsc = {
#endif
    .glyph_bitmap = glyph_bitmap, // 数据存储数组，最小为一个文字大小 static unsigned
char glyph_bitmap[130] = {0};
    .glyph_dsc = NULL,//glyph_dsc,
    .bpp = 1,
    .cmaps = NULL,
    .kern_dsc = NULL,
    .kern_scale = 0,
    .cmap_num = 0,
    .kern_classes = 0,
    .bitmap_format = 0,
#if LV_VERSION_CHECK(8, 0, 0)
    .cache = &cache
#endif
};
```

5、实现 `_get_gt_font_glyph_dsc_16()` 函数

```
static bool _get_gt_font_glyph_dsc_16(const lv_font_t * font,
lv_font_glyph_dsc_t * dsc_out, uint32_t unicode_letter,
                                      uint32_t unicode_letter_next)
{
    lv_font_fmt_txt_dsc_t * fdsc = (lv_font_fmt_txt_dsc_t *)font->dsc;

    if(unicode_letter >= 0x20 && unicode_letter < 0x80){
        dsc_out->adv_w = 8;      // 文字的实际宽度
        dsc_out->box_w = 8;      // 文字的显示宽度
        dsc_out->ofs_y = -2;     // 文字的显示位置偏移
    }
    else{
        dsc_out->adv_w = 16;     // 文字的实际宽度
        dsc_out->box_w = 16;     // 文字的显示宽度
        dsc_out->ofs_y = 0;      // 文字的显示位置偏移
    }

    dsc_out->box_h = 16;         // 文字的显示高度
    dsc_out->ofs_x = 0;          // 文字的显示位置偏移
    dsc_out->bpp   = 1;          // 文字的位深
```

```
    dsc_out->is_placeholder = false; // 是否是占位符

    return true;
}
```

6、 实现 `_get_gt_font_bitmap_fmt_txt_16` 函数

```
static const uint8_t * _get_gt_font_bitmap_fmt_txt_16(const lv_font_t * font,
uint32_t unicode_letter)
{
    lv_font_fmt_txt_dsc_t * fdsc = (lv_font_fmt_txt_dsc_t *)font->dsc;

    if(fdsc->bitmap_format == LV_FONT_FMT_TXT_PLAIN) {
        if(unicode_letter >= 0x20 && unicode_letter < 0x80){
            // 读取 ASCII 字符
            ASCII_GetData(unicode_letter, ASCII_8X16, &fdsc->glyph_bitmap[0]);
            return &fdsc->glyph_bitmap[0];
        }
        else{
    #if 01
            // Unicode 转换为 GBK 编码
            uint32_t gbk = U2G(unicode_letter);
            if(0 == gbk){
                return NULL;
            }
            // 读取 GBK 字符
            gt_16_GetData((gbk >> 8)&0xFF, gbk & 0xFF, &fdsc->glyph_bitmap[0]);
    #else
            U2G_GetData_16X16(unicode_letter, &fdsc->glyph_bitmap[0]);
    #endif
            return &fdsc->glyph_bitmap[0];
        }
    }

    return NULL;
}
```

7、使用

```
lv_obj_set_style_text_font(label, &gt30l24a3w_montserrat_16, 0);
```