

5. 21vg1 图形化实验

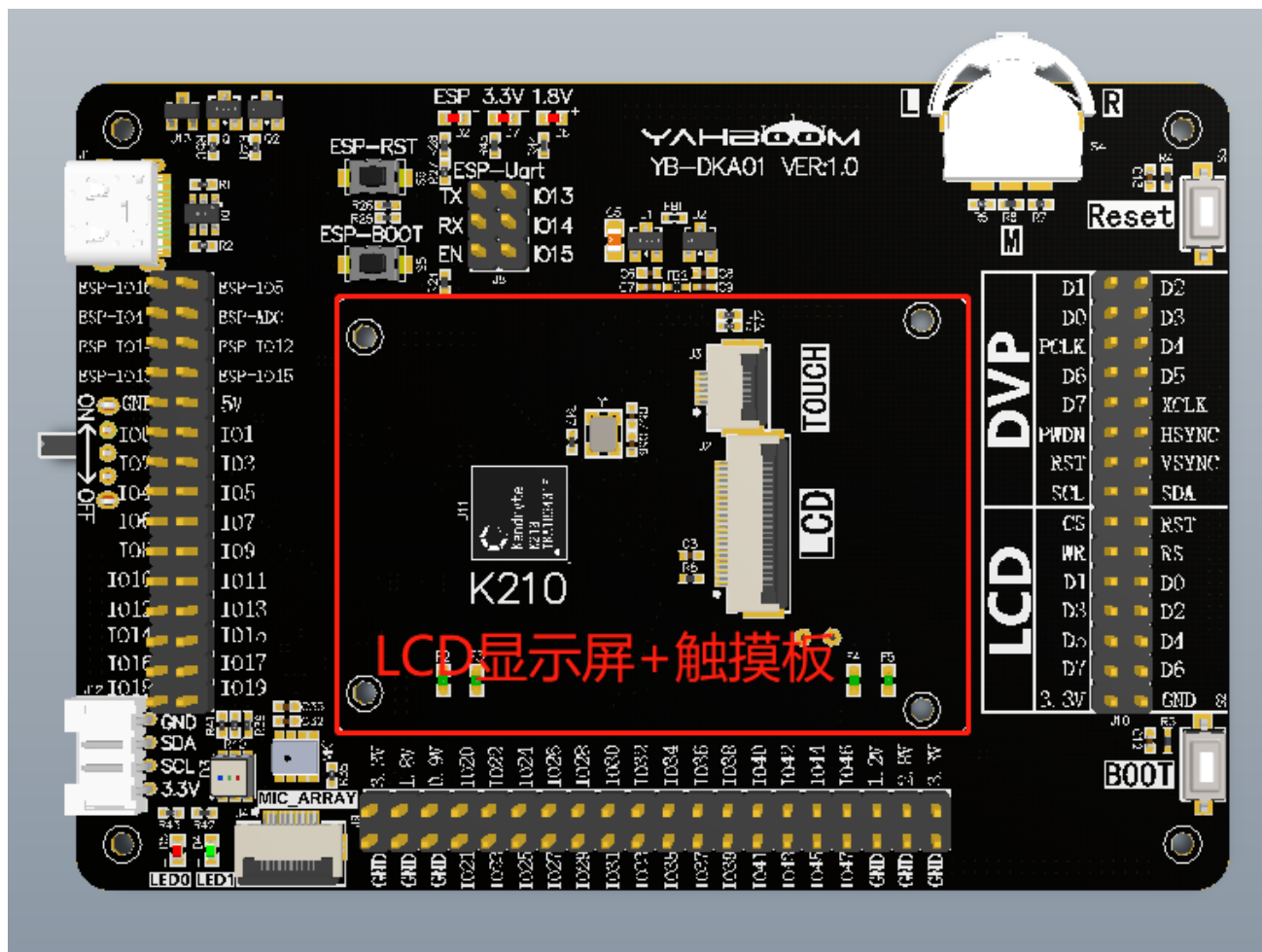
一、实验目的

本节课主要学习 K210 图形化操作界面的功能。

二、实验准备

1. 实验元件

LCD 显示屏、FT6236 触摸板



2. lvgl 图形化库简介

LVGL（轻度综合图形界面库）是一个免费开源图形库，具有使用方便，画面美观，内存占用率低等优点，能够适配大部分嵌入式单片机，库里拥有许多控件，比如图片按钮、滑动杆、对话框等等，都可以轻松使用。

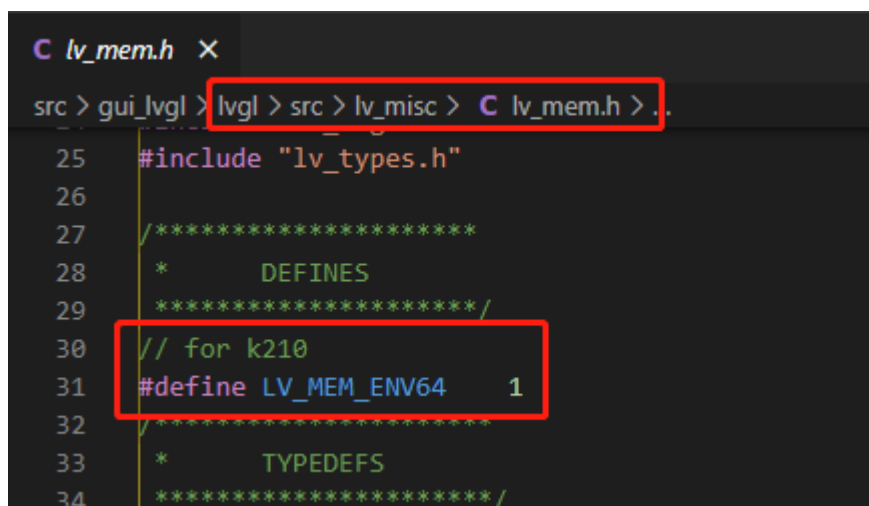
对单片机系统要求：16/32/64 位 MCU，系统时钟速度大于 16MHz，ROM 大于 128K，静态 RAM 大于 16K，栈 RAM 大于 4K，堆 RAM 大于 16K，C99 或更新的编译器。

需要注意一点的是内存使用情况可能因架构、编译器和构建选项的不同而有所不同。

由于 lvgl 图形化库版本之间会有差异，这里以 lvgl-v6.1.1 为实验基础版本。

下载地址：<https://github.com/lvgl/lvgl/releases/tag/v6.1.1>

这里需要注意的是如果从 lvgl 下载的版本，由于 K210 是 64 位芯片，所以需要增加 64 位的支持，找到 lvgl->src->lv_misc->lv_mem.h 文件，在里面增加一个宏定义：`#define LV_MEM_ENV64 1`



三、实验原理

lvgl 图形化库总共可以分为三个大的部分来理解，第一个是 lvgl 的线程，第二个是输入检测，第三个是显示画面。

lvgl 的主线程由外部定时器每几毫秒调用一次，lv_task_handler 主要是处理 lvgl 的任务，lv_tick_inc 则是起到刷新、清除中断和记录时间的作用。

lvgl 的输入检测由 lv_indev_drv_t 这个结构体管理，例如触摸板只需要设置输入参数后，在回调函数 read_cb 中传入触摸板的状态和触摸的 X/Y 坐标值就

可以。

lvgl 的显示画面由 lv_disp_drv_t 结构体管理，同样是设置参数后，调用 flush_cb 函数把回调传回的图像显示出来。

四、实验过程

1. 首先根据上面的硬件连接引脚图，K210 的硬件引脚和软件功能使用的是 FPIOA 映射关系。

```
/******HARDWARE-PIN*****  
// 硬件IO口，与原理图对应  
#define PIN_LCD_CS          (36)  
#define PIN_LCD_RST         (37)  
#define PIN_LCD_RS          (38)  
#define PIN_LCD_WR          (39)  
  
#define PIN_FT_SCL          (9)  
#define PIN_FT_SDA          (10)  
#define PIN_FT_INT          (12)  
#define PIN_FT_RST         (37)  
  
/******SOFTWARE-GPIO*****  
// 软件GPIO口，与程序对应  
#define LCD_RST_GPIONUM      (0)  
#define LCD_RS_GPIONUM      (1)  
  
#define FT_INT_GPIONUM       (2)  
#define FT_RST_GPIONUM      (3)  
  
/******FUNC-GPIO*****  
// GPIO口的功能，绑定到硬件IO口  
#define FUNC_LCD_CS          (FUNC_SPI0_SS3)  
#define FUNC_LCD_RST         (FUNC_GPIOHS0 + LCD_RST_GPIONUM)  
#define FUNC_LCD_RS          (FUNC_GPIOHS0 + LCD_RS_GPIONUM)  
#define FUNC_LCD_WR          (FUNC_SPI0_SCLK)  
  
#define FUNC_FT_SCL          (FUNC_I2C0_SCLK)  
#define FUNC_FT_SDA          (FUNC_I2C0_SDA)  
#define FUNC_FT_INT          (FUNC_GPIOHS0 + FT_INT_GPIONUM)  
#define FUNC_FT_RST         (FUNC_GPIOHS0 + FT_RST_GPIONUM)
```

```
static void hardware_init(void)
{
    /* SPI lcd */
    fpioa_set_function(PIN_LCD_CS,  FUNC_LCD_CS);
    fpioa_set_function(PIN_LCD_RST, FUNC_LCD_RST);
    fpioa_set_function(PIN_LCD_RS,  FUNC_LCD_RS);
    fpioa_set_function(PIN_LCD_WR,  FUNC_LCD_WR);
    sysctl_set_spi0_dvp_data(1);

    /* I2C FT6236 */
    fpioa_set_function(PIN_FT_SCL, FUNC_FT_SCL);
    fpioa_set_function(PIN_FT_SDA, FUNC_FT_SDA);
    fpioa_set_function(PIN_FT_INT, FUNC_FT_INT);
    // fpioa_set_function(PIN_FT_RST, FUNC_FT_RST);
}
```

2. 设置 LCD 的 IO 口电平电压为 1.8V。。

```
static void io_set_power(void)
{
    /* 设置显示器电压为1.8V */
    sysctl_set_power_mode(SYSCTL_POWER_BANK6, SYSCTL_POWER_V18);
}
```

3. 在使用 LCD 触摸屏前需要初始化，先初始化 LCD，在初始化 ft6236，然后显示一秒自定义图片。

```
/* 初始化触摸屏并显示图片 */
lcd_init();
ft6236_init();
lcd_draw_picture_half(0, 0, 320, 240, gImage_logo);
sleep(1);
```

4. 初始化 lvgl，设置 LCD 显示的结构体参数，其中 flush_cb 是 LCD 刷新的回调函数；再设置触摸屏输入的结构体参数，其中 read_cb 是输入设备的回调函数；最后初始化并启动定时器。

```

void lvgl_disp_input_init(void)
{
    lv_init();

    static lv_disp_buf_t disp_buf;
    static lv_color_t buf[LV_HOR_RES_MAX * 10];
    lv_disp_buf_init(&disp_buf, buf, NULL, LV_HOR_RES_MAX * 10);

    lv_disp_drv_t disp_drv;           /*Descriptor of a display*/
    lv_disp_drv_init(&disp_drv);       /*Basic initialization*/
    disp_drv.flush_cb = my_disp_flush; /*Set your driver function*/
    disp_drv.buffer = &disp_buf;      /*Assign the buffer to the display*/
    lv_disp_drv_register(&disp_drv);   /*Finally register the driver*/

    // input
    lv_indev_drv_t indev_drv;
    lv_indev_drv_init(&indev_drv);     /*Descriptor of a input device*/
    indev_drv.type = LV_INDEV_TYPE_POINTER; /*Touch pad is a pointer*/
    indev_drv.read_cb = my_touchpad_read; /*Set your driver function*/
    lv_indev_drv_register(&indev_drv); /*Finally register the driver*/

    /* 初始化并启动定时器 */
    mTimer_init();
}

```

5. LCD 显示刷新回调的主要功能是解析需要显示的位置，然后传给 LCD 显示的函数显示出来，其中 color_p 就是需要显示的数据。显示完成后需要通过 lv_disp_flush_ready 函数通知 lvgl 系统数据更新完成。

```

static void my_disp_flush(lv_disp_drv_t * disp, const lv_area_t * area, lv_color_t * color_p)
{
    uint16_t x1 = area->x1;
    uint16_t x2 = area->x2;
    uint16_t y1 = area->y1;
    uint16_t y2 = area->y2;

    lcd_draw_picture_half((uint16_t)x1, (uint16_t)y1,
        (uint16_t)(x2 - x1 + 1), (uint16_t)(y2 - y1 + 1),
        (uint16_t *)color_p);
    lv_disp_flush_ready(disp); /* Indicate you are ready with the flushing*/
}

```

6. 输入设备回调函数主要的功能读取触摸板的状态和触摸的坐标。

```
static bool my_touchpad_read(lv_indev_drv_t * indev_drv, lv_indev_data_t * data)
{
    static int a_state = 0;
    if (ft6236.touch_state & TP_COORD_UD)
    {
        ft6236.touch_state &= ~TP_COORD_UD;
        ft6236_scan();
        data->point.x = ft6236.touch_x;
        data->point.y = ft6236.touch_y;
        data->state = LV_INDEV_STATE_PR;
        a_state = 1;
        return false;
    }
    else if (ft6236.touch_state & 0xC0)
    {
        if (a_state == 1)
        {
            a_state = 0;
            data->point.x = ft6236.touch_x;
            data->point.y = ft6236.touch_y;
            data->state = LV_INDEV_STATE_REL;
            return false;
        }
    }
    return false;
}
```

7. 定时器定时时间为每毫秒中断调用一次，在定时器中断函数中调用 `lvgl` 的任务管理函数和时钟函数。

```
static void mTimer_init(void)
{
    timer_init(TIMER_DEVICE_0);
    timer_set_interval(TIMER_DEVICE_0, TIMER_CHANNEL_0, 1e6);
    timer_irq_register(TIMER_DEVICE_0, TIMER_CHANNEL_0, 0, 1, timer_irq_cb, NULL);

    timer_set_enable(TIMER_DEVICE_0, TIMER_CHANNEL_0, 1);
}

static int timer_irq_cb(void * ctx)
{
    lv_task_handler();
    lv_tick_inc(1);
    return 0;
}
```

8. 初始化完成后，就可以运行例程程序，这个例程是创建一个有用三个界面

的画面。最后打印 OK，并提示触摸屏幕。

```
/* 运行例程 */  
demo_create();  
  
printf("system start ok\n");  
printf("Please touch the screen\n");  
while (1)  
|  
;
```

9. 编译调试，烧录运行

把本课程资料中的 gui_lvgl 复制到 SDK 中的 src 目录下，然后进入 build 目录，运行以下命令编译。

```
cmake .. -DPROJ=gui_lvgl -G "MinGW Makefiles"
```

```
make
```

```
[100%] Linking C executable gui_lvgl  
Generating .bin file ...  
[100%] Built target gui_lvgl  
PS C:\K210\SDK\kendryte-standalone-sdk-develop\build>
```

编译完成后，在 build 文件夹下会生成 gui_lvgl.bin 文件。

使用 type-C 数据线连接电脑与 K210 开发板，打开 kflash，选择对应的设备，再将程序固件烧录到 K210 开发板上。

五、实验现象

烧录完成固件后，系统会弹出一个终端界面，如果没有弹出终端界面的可以打开串口助手显示调试内容。

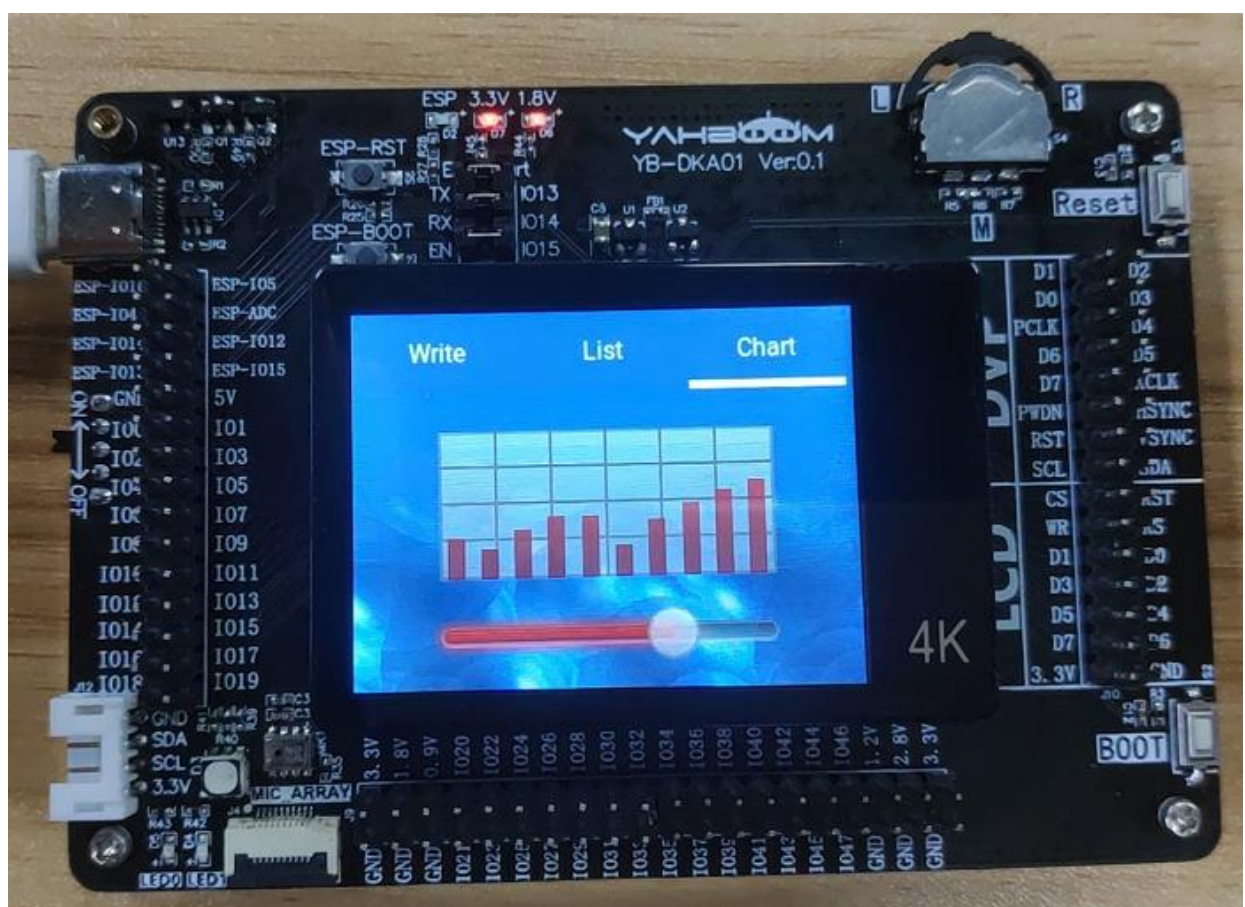
终端会打印 “Please touch the screen” 提示触摸显示屏，显示屏会显示第一个界面，触摸中间输入框，会在底部弹出一个虚拟键盘，在键盘上触摸可以打印字符；上面顶栏的 “List” 切换到第二个界面列表，可以点击列表上的内容，

则会把对应的名称显示到第一个界面的输入框中；点击顶栏的“Chart”，切换到第三个界面，拖动底部的滑动杆，上面的条形图会跟着变化。

C:\Users\Administrator\AppData\Local\Temp\tmp37A2.tmp

```
system start init
system start ok
Please touch the screen
```





六、实验总结

1. lvgl 是一个嵌入式微处理器图形化库，拥有丰富的控件。
2. K210 开发板完成可以跑图形化的 lvgl 库，并且效果很 OK。
3. 由于 lvgl 的不同版本的差异性，跨大版本是存在兼容性问题的。