5.2 lvgl graphical experiment

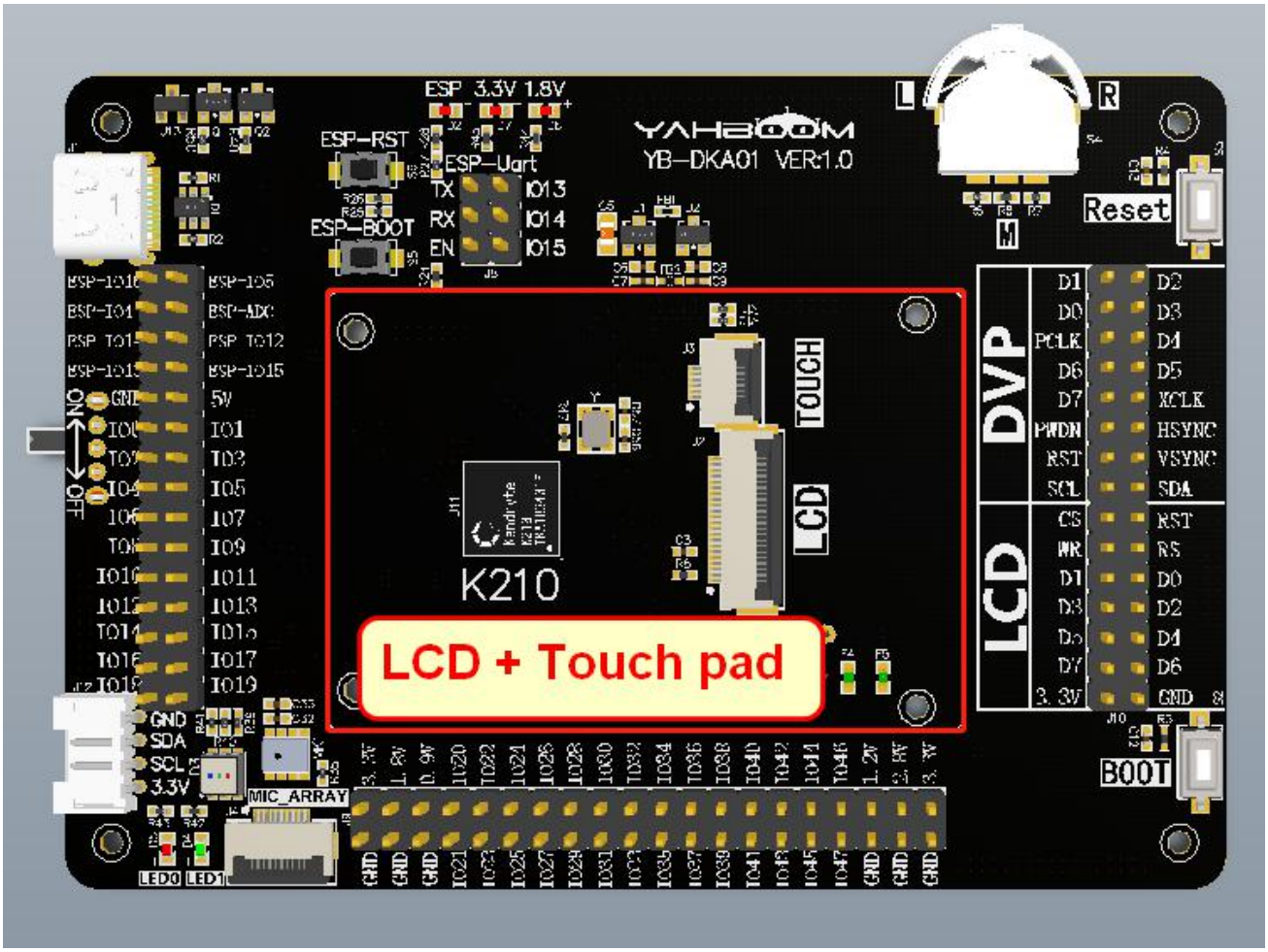
1. Experiment purpose

In this lesson, we mainly learn K210 graphical operation interface function.

2.Experiment preparation

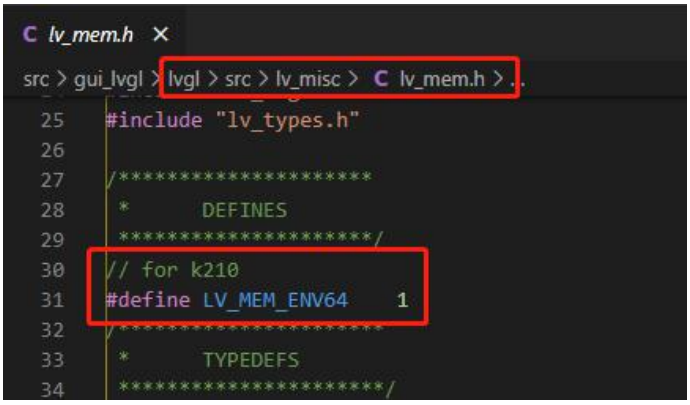
2.1 components

LCD, FT6236 Touch pad



2. 2 Introduction to lvgl graphical library

LVGL (Lightly Integrated Graphical Interface Library) is a free and open source graphic library, which is uploaded as lvgl-v6.1.1 at: https://github.com/lvgl/lvgl/releases/tag/v6.1.1



3. Experiment principle

The lvgl graphical library can be divided into three major parts to understand.
First is the lvgl thread. Second is the input detection. Third is the display screen.

4. Experiment procedure

4.1 K210's hardware pins and software functions use the FPIOA mapping relationship.

```
/*****************************HARDWARE-PIN*********************************/
//Hardware IO port, corresponding Schematic
#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********************************/
//Software GPIO port, corresponding program
#define LCD_RST_GPIONUM         (0)
#define LCD_RS_GPIONUM          (1)

#define FT_INT_GPIONUM          (2)
#define FT_RST_GPIONUM          (3)

/*****************************FUNC-GPIO************************************/
// Function of GPIO port, bound to hardware IO port
#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);
}
```

4.2 Set the IO port level voltage of LCD to 1.8V.

```
static void io_set_power(void)
{
    /* Set the display voltage to 1.8V */
    sysctl_set_power_mode(SYSCTL_POWER_BANK6, SYSCTL_POWER_V18);
}
```

4.3 It needs to be initialized before using the LCD touch screen. Initialize the LCD first, then initialize the ft6236, and then display the custom picture for one second.

```
/* Initialize the touch screen and display the picture */
lcd_init();
ft6236_init();
lcd_draw_picture_half(0, 0, 320, 240, gImage_logo);
sleep(1);
```

4.4 Initialize lvgl, set the LCD display structure parameters, where flush_cb is the callback function of LCD refresh; then set the structure parameters of the touch screen input, where read_cb is the callback function of the input device; finally initialize and start the timer.

```
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 functic
    disp_drv.buffer = &disp_buf;            /*Assign the buffer to th
    lv_disp_drv_register(&disp_drv);        /*Finally register the dr

    // input
    lv_indev_drv_t indev_drv;
    lv_indev_drv_init(&indev_drv);              /*Descriptor of a in
    indev_drv.type = LV_INDEV_TYPE_POINTER;     /*Touch pad is a poi
    indev_drv.read_cb = my_touchpad_read;       /*Set your driver fu
    lv_indev_drv_register(&indev_drv);          /*Finally register t

    mTimer_init();
}
```

4.5 The main function of the LCD display refresh callback is to analyze the position to be displayed, and then the function passed to the LCD display is displayed.

```c
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*/
}
```

4.6 The main function of the input device callback function is to read the state of the touchpad and the coordinates of the touch.

```c
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;
}
```

4.7 The timer time is called once every millisecond interrupt.

```c
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);
}
```

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

4.8 After the initialization is complete, you can run the routine program. This routine is to create a screen with three interfaces. Finally, print OK and prompt to touch the screen.

```c
/* Run routine */
demo_create();

printf("system start ok\n");
printf("Please touch the screen\n");
while (1)
    ;
return 0;
```

4.9 Compile and debug, burn and run

Copy the gui_lvgl to the src directory in the SDK.

Then, enter the build directory and run the following command to compile.

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>
```

After the compilation is complete, the **gui_lvgl.bin** file will be generated in the build folder.

We need to use the type-C data cable to connect the computer and the K210 development board.

Open kflash, select the corresponding device, and then burn the **gui_lvgl.bin** file to the K210 development board.

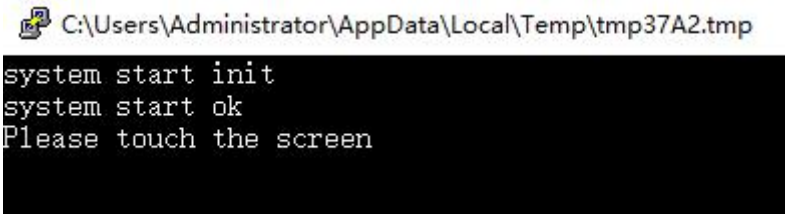
5. Experimental phenomenon

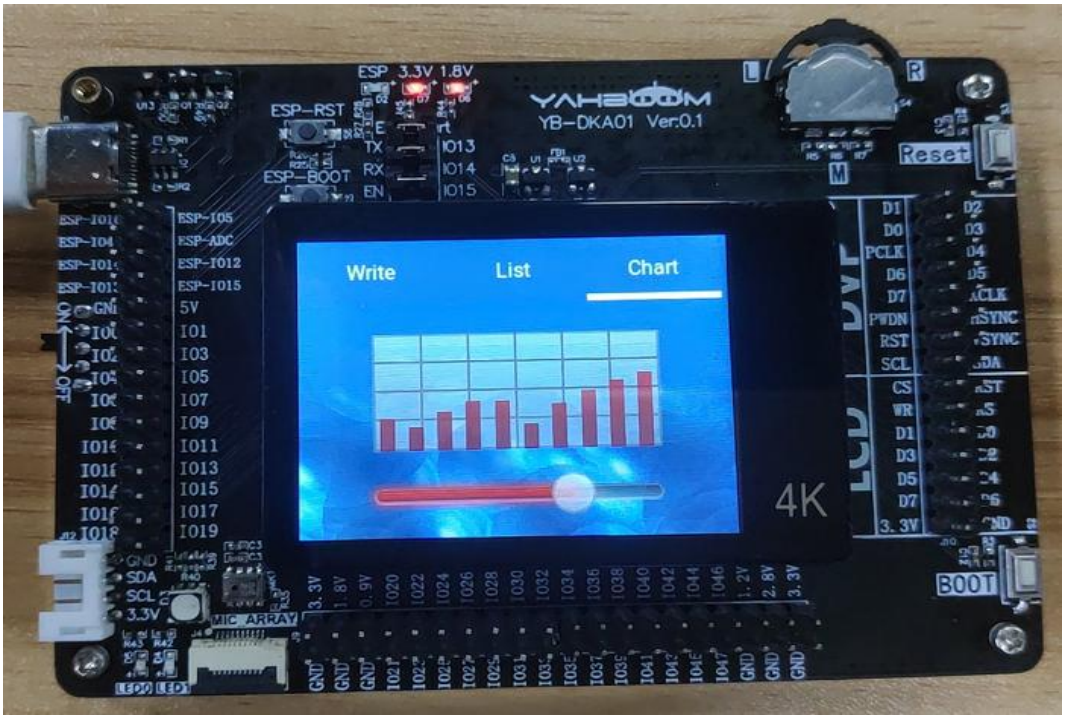
After the firmware is write, a terminal interface will pop up. **If the terminal interface does not pop up, we can open the serial port assistant to display the debugging content.**

We can see that the terminal will print "Please touch the screen".

The screen will show the first interface, touch the middle input box, a virtual keyboard will pop up

at the bottom, touch on the keyboard to print characters.
Click "List" in the top bar switches to the second interface list, you can click on the content on list, the corresponding name will be displayed in the input box of the first interface.
Click "Chart" in the top bar to switch to the third interface, drag the slide bar at the bottom, and the bar chart above will change accordingly.





6. Experiment summary

6.1 lvgl is a graphical library of embedded microprocessors with rich controls.

6.2 K210 development board can run graphical lvgl library, and the effect is very good.

6.3 Due to the differences of different versions of lvgl, there are compatibility issues across major versions.