



厦门大学  
XIAMEN UNIVERSITY

# 嵌入式系统开发

## 实验五 串口触摸屏控制实验

姓名	
学号	
学院	信息学院
专业	计算机科学与技术专业
年级	2022
日期	2024/12/6

## 目录

1 实验目的.....	1
2 实验方案设计.....	1
2.1 基础实验 .....	1
2.2 提高实验 .....	3
2.3 拓展实验 .....	4
3 实验过程与结果.....	6
3.1 基础实验 .....	6
3.2 提高实验 .....	8
3.3 拓展实验 .....	9
4 实验分析.....	10
5 实验总结.....	11

# 1 实验目的

掌握 DGUS Tool V7.624.exe 软件使用，通过串口触摸屏控制的基本操作，并扩展到显示和控制环境传感器数据，以及实现数据显示的动态曲线图。

## 2 实验方案设计

### 2.1 基础实验

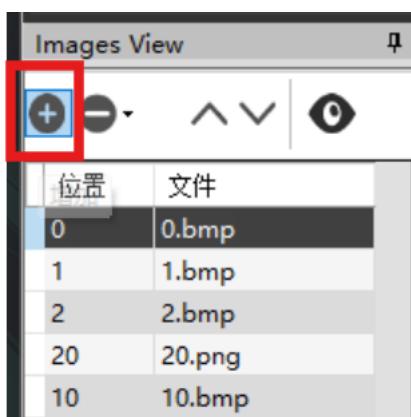
#### (1) 实验要求

用触摸屏控制 LED 灯。

#### (2) 实验步骤

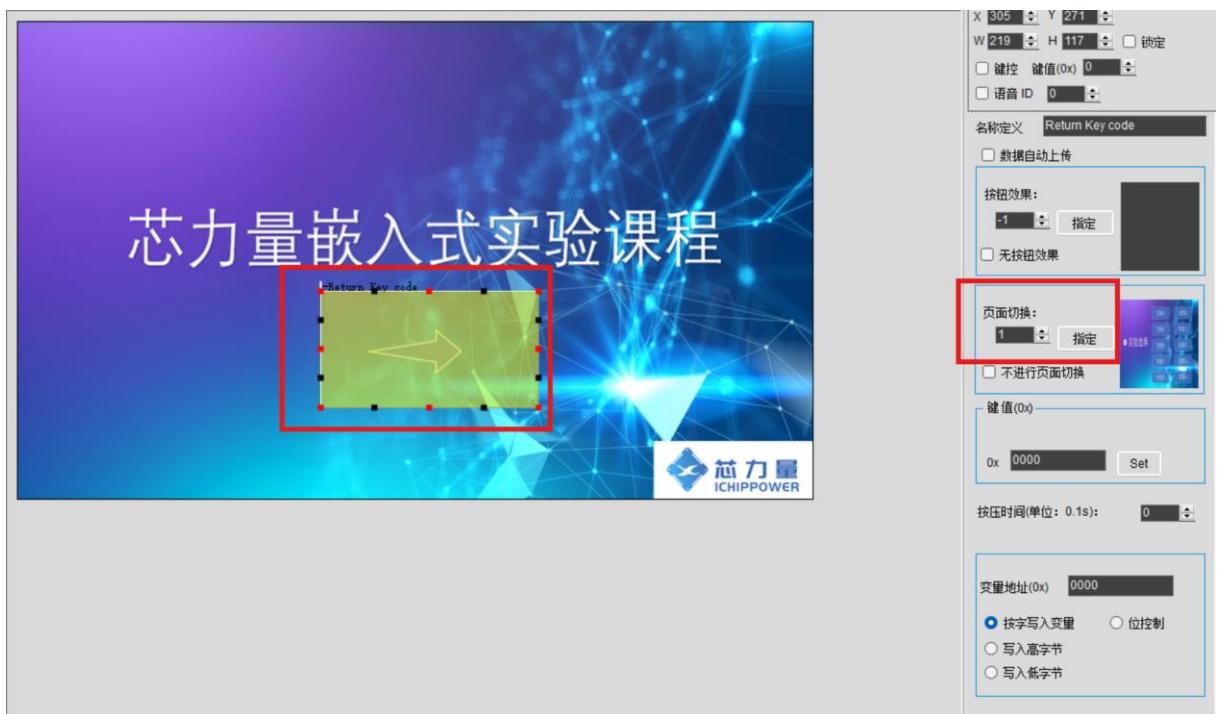
步骤一：先进行触摸屏界面设计。建立触摸屏工程，用 ps 或画图修改图像分辨率，生成背景库文件和字库文件。

步骤二：添加背景图片

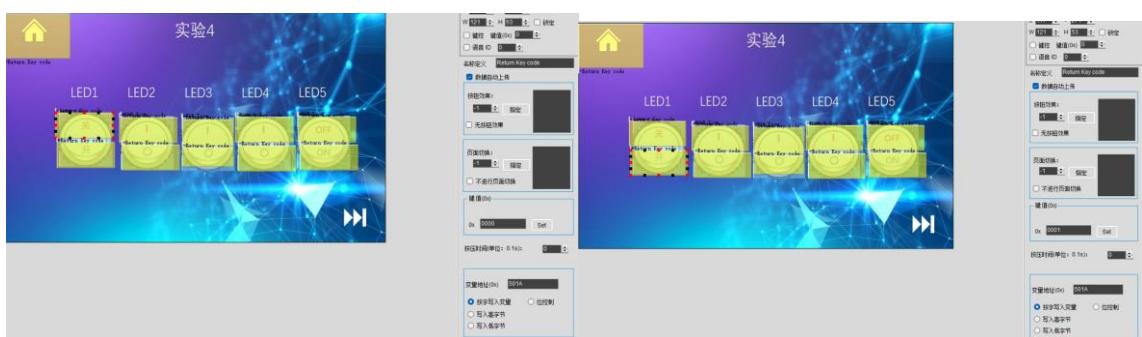


步骤三：添加按键返回，修改页面切换



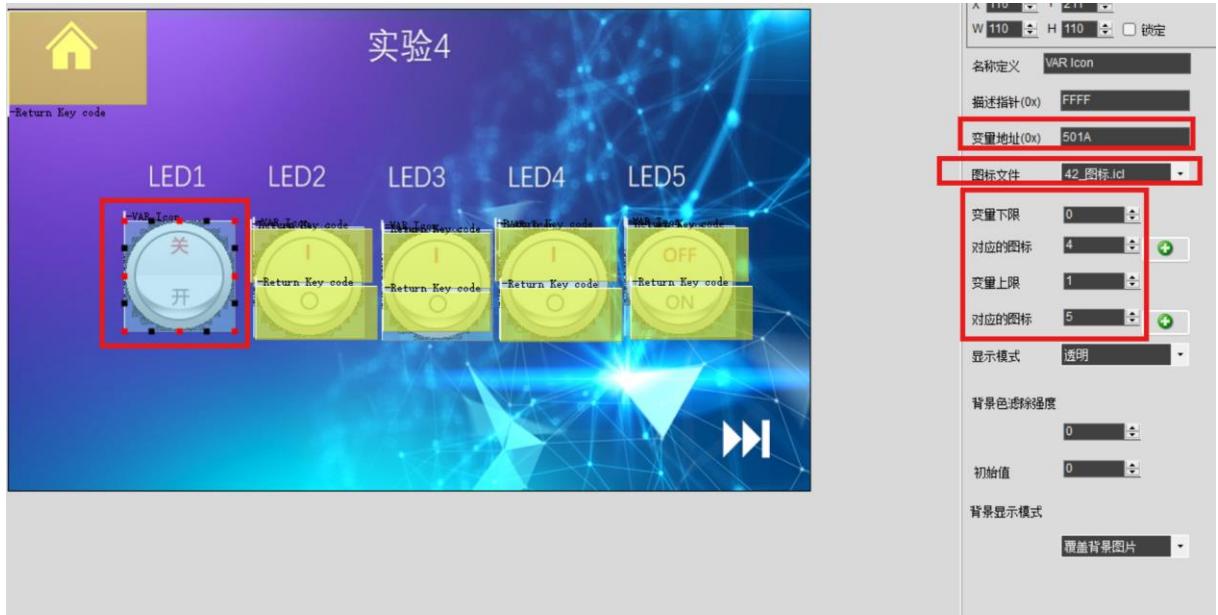


步骤四：继续添加按键返回，并设置对应的变量地址、键值。



步骤五：添加变量图标显示，并设置对应的变量地址、图标、键值。





步骤六：最后，使用 SD 卡将包含以下文件的 DWIN\_SET 文件夹下载到 dwen 屏即可：

0 号字库文件

12 输入法文件(可选，可以没有)

13 触控文件(.bin)

14 显示文件(.bin)

22 变量初始化文件(.bin)

23 其他字库文件(可选，可以没有)

32 背景图片

48 图标库(可多个)

步骤七：编程。

## 2.2 提高实验

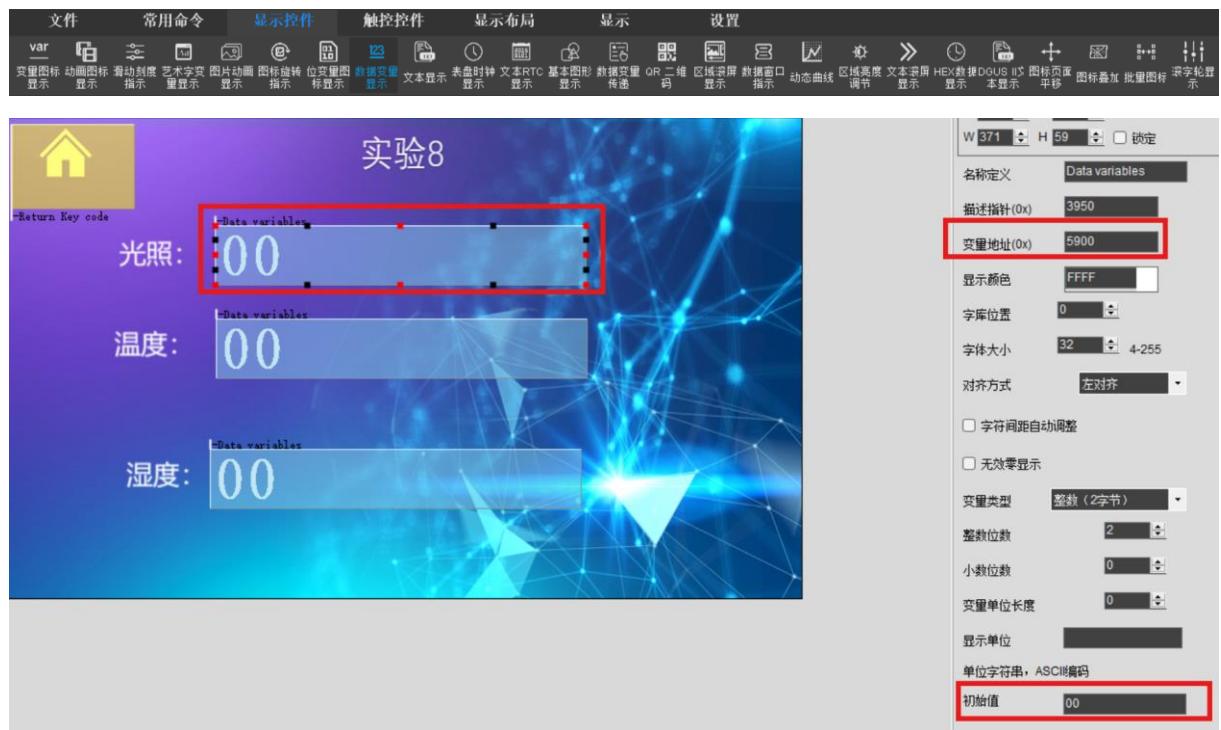
### (1) 实验要求

用触摸屏显示光照、温度、湿度信息。

### (2) 实验步骤

步骤一：前期准备同基础实验。

步骤二：添加数据变量显示，修改变量地址和初始值。



步骤三：保存，生成，导出，拷入 sd 卡，拷入迪文屏，编程。

## 2.3 拓展实验

### (1) 实验要求

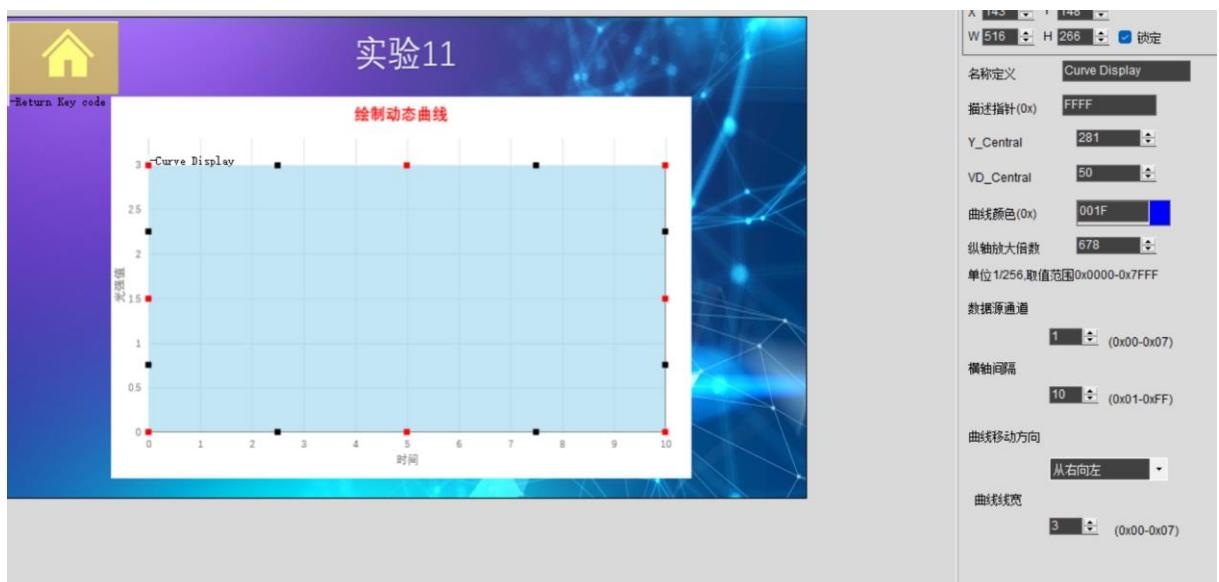
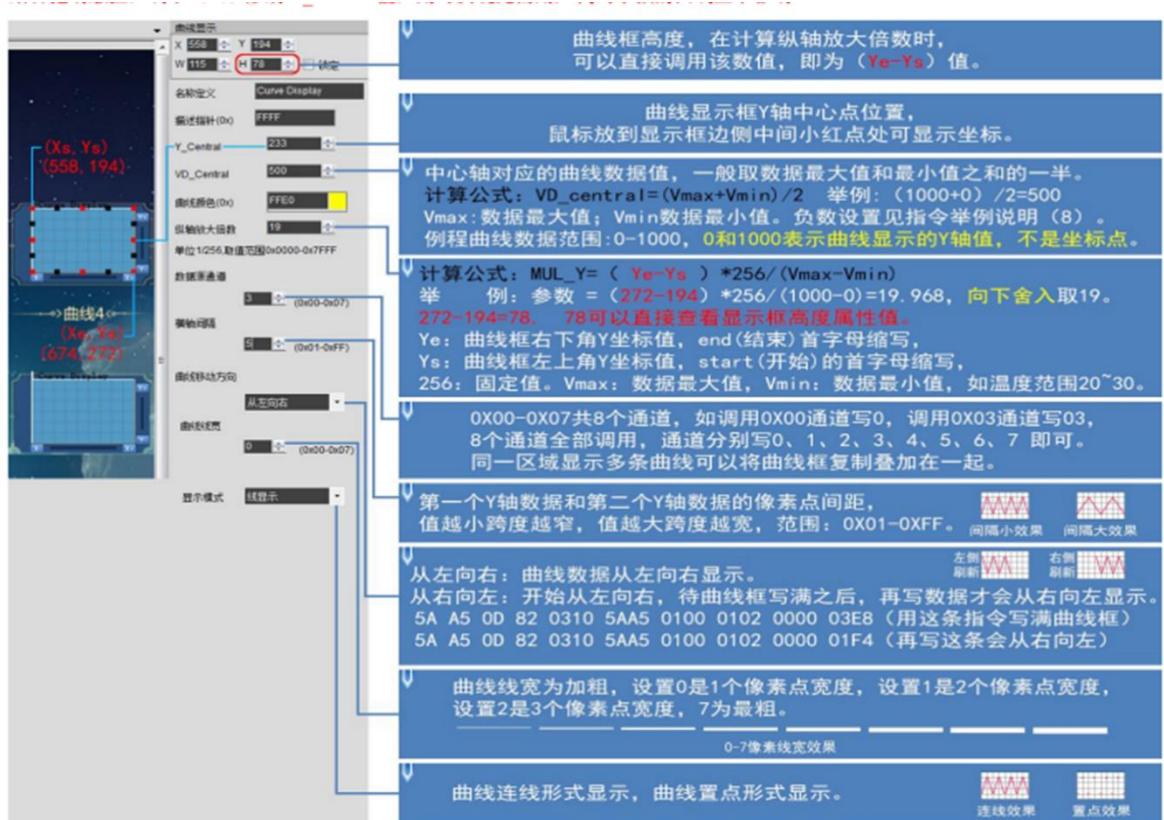
在提高实验的基础上，用触摸屏显示亮度变化曲线。

### (2) 实验步骤

步骤一：前期准备同提高实验。

步骤二：添加数据变量显示，修改变量地址和各种参数，参考 ppt 的教程。





步骤三：保存，生成，导出，拷入 sd 卡，拷入迪文屏，编程。

### 3 实验过程与结果

#### 3.1 基础实验

软件设置的过程在实验步骤里已写。下同。

先初始化

```
//初始化 系统时钟、XLL\拨码开关、OLED、智慧农业模块等按需编写
void system_init(void) {
    systick_config();          //系统时钟
    gd_XII_systeminit();      //x11 实验箱初始化

    usart0Init(EVAL_COM0, 115200); //串口初始化
    Init_E53_IA1();           //情景模块初始化 若调用情景模块 实物一定要连接
否则无法正常初始化
    dipinit();                //拨码开关的初始化

    diwin_init(115200U);       //串口屏初始化
    //OLED 初始化(可添加进来调试用，参考以前的 OLED 使用)
}
```

控制五个灯的函数：

```
void test1(void) {
    if(0x1A==uart5_rx_buffer[5])
    {
        delay_1ms(100);
        if(0x01==uart5_rx_buffer[8]){
            gd_eval_led_on(LED1);
        }

        else if(0x00==uart5_rx_buffer[8]){
            gd_eval_led_off(LED1);
        }
    }

    if(0x1B==uart5_rx_buffer[5])
    {
        delay_1ms(100);
        while(0x01==uart5_rx_buffer[8]){
            gd_eval_led_on(LED2);
        }

        if(0x00==uart5_rx_buffer[8]){
            gd_eval_led_off(LED2);
        }
    }
}
```

```

        }
    }
    if(0x1C==uart5_rx_buffer[5])
    {
        delay_1ms(100);
        while(0x01==uart5_rx_buffer[8]){
            gd_eval_led_on(LED3);
        }

        if(0x00==uart5_rx_buffer[8]){
            gd_eval_led_off(LED3);
        }
    }
    if(0x1D==uart5_rx_buffer[5])
    {
        delay_1ms(100);
        while(0x01==uart5_rx_buffer[8]){
            gd_eval_led_on(LED4);
        }

        if(0x00==uart5_rx_buffer[8]){
            gd_eval_led_off(LED4);
        }
    }
    if(0x1E==uart5_rx_buffer[5])
    {
        delay_1ms(100);
        while(0x01==uart5_rx_buffer[8]){
            gd_eval_led_on(LED5);
        }

        if(0x00==uart5_rx_buffer[8]){
            gd_eval_led_off(LED5);
        }
    }
}

```

可以观察到，可以用屏幕上的按钮开关控制相应 Led 灯的亮灭。

## 3.2 提高实验

情景模块读取传感器数据，温度、湿度、光照数据分别输出。

```
void test2(void)
{
    E53_IA1_Read_Data(); //情景模块读取传感器数据，温度、湿度、光照，数据分别保存至
    (E53_IA1_Data.Temperature, E53_IA1_Data.Humidity,E53_IA1_Data.Lux)

    //调试时可以借助 OLED 屏幕先观察传感器值的变化
    //  OLED_ShowString(0,4,"lux:");
    //  OLED_ShowNum(80,2,(int)E53_IA1_Data.Lux,4,17);

    message1=(uint16_t)E53_IA1_Data.Lux;
    WRITE_Lux(message1); //往地址里写光照数据
    delay_1ms(200);
    WRITE_Tem((uint16_t)E53_IA1_Data.Temperature); //往地址里写光照数据
    delay_1ms(200);
    WRITE_Hum((uint16_t)E53_IA1_Data.Humidity);
    delay_1ms(200);
}
```

其中仿照样例写函数：

```
void WRITE_Lux(uint16_t addr)
{
    usart5_txBuf[0] = 0x5a;
    usart5_txBuf[1] = 0xa5;
    usart5_txBuf[2] = 0x05;
    usart5_txBuf[3] = 0x82;
    usart5_txBuf[4] = 0x59;
    usart5_txBuf[5] = 0x00;
    usart5_txBuf[6] = (uint8_t)((addr >> 8) & 0xff);
    usart5_txBuf[7] = (uint8_t)(addr & 0xff);
    usart5_send(usart5_txBuf, 9);
    // color_change(0x5910,0xffff);
}

void WRITE_Tem(uint16_t addr)
{
    usart5_txBuf[0] = 0x5a;
    usart5_txBuf[1] = 0xa5;
    usart5_txBuf[2] = 0x05;
    usart5_txBuf[3] = 0x82;
```

```

        usart5_txBuf[4] = 0x59;
        usart5_txBuf[5] = 0x10;
        usart5_txBuf[6] = (uint8_t)((addr >> 8) & 0xff);
        usart5_txBuf[7] = (uint8_t)(addr & 0xff);
        usart5_send(usart5_txBuf, 9);
    }

void WRITE_Hum(uint16_t addr)
{
    usart5_txBuf[0] = 0x5a;
    usart5_txBuf[1] = 0xa5;
    usart5_txBuf[2] = 0x05;
    usart5_txBuf[3] = 0x82;
    usart5_txBuf[4] = 0x59;
    usart5_txBuf[5] = 0x20;
    usart5_txBuf[6] = (uint8_t)((addr >> 8) & 0xff);
    usart5_txBuf[7] = (uint8_t)(addr & 0xff);
    usart5_send(usart5_txBuf, 9);
}

```

可以看到光照、温度和湿度会修改。注意：要每一次 `WRITE` 之后都要 `delay_1ms(200);`，否则它反应不过来，会卡死。

### 3.3 拓展实验

修改函数

```

void test3(){
    E53_IA1_Read_Data();
    message1=(uint16_t)E53_IA1_Data.Lux;
    sys_writ(message1/10);
    delay_1ms(500);
}

```

画曲线用的函数

```

// 画曲线示例 5AA5 0C 82 0310 5AA5 0110 0000 XX XX 参考说明书 P196
void sys_writ(uint16_t y)
{
    usart5_txBuf[0] = 0x5a;
    usart5_txBuf[1] = 0xa5; // 帧头 5AA5

```

```

uart5_txBuf[2] = 0x0C; /*数据字节长度
uart5_txBuf[3] = 0x82; // 82 为写命令
uart5_txBuf[4] = 0x03;
uart5_txBuf[5] = 0x10; // 0310 启动曲线缓冲区

uart5_txBuf[6] = 0x5a;
uart5_txBuf[7] = 0xa5; // 5AA5 启动一次曲线写操作

uart5_txBuf[8] = 0x01;
uart5_txBuf[9] = 0x00; // 0x0110, *高字节 01 定义占用通道数为 1, 低字节 00 未定义写 00

uart5_txBuf[10] = 0x01;
uart5_txBuf[11] = 0x01; /*0x0000, *高字节 00 表示 0 通道, *低字节 01 表示 1 个字长度数据“y”

uart5_txBuf[12] = (uint8_t)((y >> 8) & 0xff);
uart5_txBuf[13] = (uint8_t)(y & 0xff); // 所要显示的数据

uart5_send(uart5_txBuf, 14);
}

```

可以看到光照曲线被画出来。

视频还是传到 qq 群的群作业里。

## 4 实验分析

在基础实验中，我通过触摸屏界面设计和编程，实现了对 LED 灯的控制。这一步骤涉及到了触摸屏工程的建立、背景库文件和字库文件的生成、背景图片的添加、按键返回的设置以及变量图标的显示。通过这些步骤，我能够理解触摸屏的基本工作原理和如何通过串口通信来控制硬件设备。

在提高实验中，我扩展了触摸屏的功能，使其能够显示光照、温度和湿度信息。这一步骤要求我添加数据变量显示，并修改变量地址和初始值。通过这一实验，我学会了如何从传感器读取数据，并通过触摸屏显示这些数据，进一步理解了数据采集和显示的过程。

在拓展实验中，我进一步增加了触摸屏显示亮度变化曲线的功能。这要求我添加数据变量显示，并修改变量地址和各种参数。通过这一步骤，我掌握了如何将静态数据显示转变为动态曲线显示，这对于实时监控和数据分析具有重要意义。

在实验 2 里，要每一次 `WRITE` 之后都要 `delay_1ms(200);`，否则它反应不过来，会卡死。在实验 3 里，注意参数的计算，而且方向似乎是从右到左才是正确的。

## 5 实验总结

通过本次串口触摸屏控制实验，我成功实现了：

通过触摸屏控制 LED 灯的开关。

显示并更新环境传感器数据，如光照、温度和湿度。

实现了亮度变化曲线的动态显示。

在实验过程中，我遇到了触摸屏响应延迟和数据更新不及时的问题。通过增加适当的延时，我解决了这些问题，确保了触摸屏的稳定运行。

本次实验加深了我对嵌入式系统开发中触摸屏应用的理解，特别是在人机交互和数据可视化方面。我学会了如何设计用户界面，并通过编程实现界面与硬件的交互。

后面曲线可能要做到多个曲线一起出现。还没有做颜色变化。