***2022***



**嵌入式系统 课程实验报告**

|  |  |
| --- | --- |
| 题 目： | 香橙派相关实验 |
| 专 业： | 计算机科学与技术 |
| 班 级： | CS2008 |
| 学 号： | U202015533 |
| 姓 名： | 徐瑞达 |
| 电 话： | 17837353795 |
| 邮 件： | xuruida.huster@qq.com |

目 录

[1 实验一 系统烧录 3](#_Toc8310)

[1.1 实验要求 3](#_Toc21533)

[1.2 实验过程 3](#_Toc1792)

[1.3 实验结果 3](#_Toc7100)

[2 实验二 图形界面基础 4](#_Toc20618)

[2.1 实验要求 4](#_Toc10126)

[2.2 实验过程 4](#_Toc5865)

[2.3 实验结果 4](#_Toc11664)

[3 实验三 图片文字显示 5](#_Toc13730)

[3.1 实验要求 5](#_Toc2764)

[3.2 实验过程 5](#_Toc3807)

[3.3 实验结果 5](#_Toc21726)

[4 实验四 多点触摸开发 6](#_Toc13672)

[4.1 实验要求 6](#_Toc16551)

[4.2 实验过程 6](#_Toc2819)

[4.3 实验结果 6](#_Toc20373)

[5 实验五 蓝牙通讯 7](#_Toc7968)

[5.1 实验要求 7](#_Toc39)

[5.2 实验过程 7](#_Toc14286)

[5.3 实验结果 7](#_Toc25494)

[6 实验六 综合实验 8](#_Toc32569)

[6.1 实验要求 8](#_Toc18894)

[6.2 实验过程 8](#_Toc25161)

[6.3 实验结果 8](#_Toc32278)

[7 实验总结与建议 9](#_Toc20769)

[7.1 实验总结 9](#_Toc8732)

[7.2 实验建议 9](#_Toc22657)

# 实验一 系统烧录

## 实验要求

1. 连接开发板，并烧录香橙派系统镜像到SD卡中；
2. 启动开发板，并使用SSH登陆配置开发板；
3. 在开发主机中的虚拟机内配置交叉编译环境，编译基本应用程序并在开发板上运行。

## 实验过程

1. **烧录香橙派映像**

使用BalenaEtcher软件烧录香橙派映像到SD卡中并插入开发板的TF槽内。

1. **连接开发板与触摸屏**

依次连接开发板的电源接插口、开发板的HDMI接口与显示屏的HDMI接口、开发板的USB接口与显示屏的TOUCH接口。

1. **SSH远程登陆开发板**

将开发板与键盘连接后，连接手机热点，并使用ifconfig指令获取开发板IP地址；然后使用MobaXterm软件建立SSH连接，输入用户名root和密码orangepi后实现远程登陆。

1. **禁用开发板桌面**

在MobaXterm软件中，使用orangepi-config指令禁用开发板图形化界面，重启后即可进入命令行模式。

1. **配置开发主机交叉编译环境**

在Windows系统中部署WSL子系统，在/home文件夹中创建orangepi目录，拷贝交叉编译器和实验源代码并解压；然后修改/lab-2022-st/common/rules.mk文件内编译器路径。

1. **运行Hello embedded linux程序**

进入/lab-2022-st/lab1文件夹，执行make命令，输出可执行程序；使用scp指令将可执行程序发送至开发板；然后在MobaXterm软件中执行该程序。

## 实验结果

最终输出为Hello embedded linux!

# 实验二 图形界面基础

## 实验要求

1. 理解Linux下的LCD显示驱动接口framebuffer的使用原理；

2. 理解framebuffer的双缓冲机制；

3. 理解基本图形的显示，如点、线、边框、矩形。

## 实验过程

1. **实现基本函数**
2. fb\_draw\_rect(int x, int y, int w, int h, int color)

该函数用于绘制矩形。绘制时，首先需要检查参数是否超出屏幕范围，然后通过循环将对应位置的像素使用fb\_draw\_pixel函数绘制即可。

1. fb\_draw\_line(int x1, int y1, int x2, int y2, int color)

该函数用于绘制线条。绘制时，为使得所画的直线连续，需要根据线条的斜率采用不同的画线方式。当直线的斜率的绝对值小于1或大于1时，分别以x坐标或y坐标作为循环变量循环绘制线条上的像素点。

1. **编译运行**

函数实现完成后，依次使用make、scp指令编译输出lab2可执行程序并发送到开发板上；然后在MobaXterm软件中进入out目录执行lab2程序。

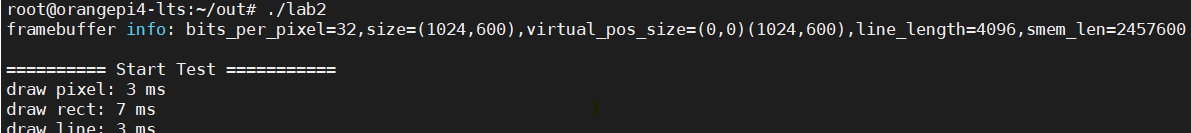
## 实验结果

运行lab2程序后，开发板最终效果如图2.1所示。



**图2.1 lab2运行效果图**

运行lab2程序时的时间输出如图2.2所示。



**图2.2 lab2输出结果**

关键函数实现如下方代码块所示。



# 实验三 图片文字显示

## 实验要求

1. 显示jpg不透明图片；
2. 显示png半透明图片；
3. 显示矢量字体，包括字模的提取和字模的显示。

## 实验过程

1. **实现基本函数**

实现fb\_draw\_image函数时，需要根据不同的图片格式（jpg不透明图片、png 半透明图片、矢量字体）的图片进行处理。

当图片的颜色类型为FB\_COLOR\_RGB\_8880时，表明图片为jpg 类型。由于不需要处理透明度alpha，直接将存储的图片复制到framebuffer缓冲区即可。

当图片的颜色类型为FB\_COLOR\_RGBA\_8888时，表明图片为png类型，此时需要考虑透明度alpha。根据图片存储字节时，从高位到低位分别代表alpha、R1、G1、B1值，需要对复制到缓冲区的目标地址的内容进行修正，处理如下：

dst[j] += (((B1 - dst[j]) \* alpha) >> 8);

dst[j + 1] += (((G1 - dst[j + 1]) \* alpha) >> 8);

dst[j + 2] += (((R1 - dst[j + 2]) \* alpha) >> 8);

当图片的颜色类型为FB\_COLOR\_ALPHA\_8时，表明图片为矢量字体。矢量字体与半透明图片的不同之处是，颜色根据函数参数统一设置，因此处理时，将上述修正公式中的B1、G1、R1统一替换为函数参数指定颜色对应的RGB值即可。

1. **编译运行**

编译运行过程与实验二类似。

## 实验结果

运行lab3程序后，开发板最终效果如图3.1所示。



**图3.1 lab3运行效果图**

关键函数实现如下方代码块所示。



# 实验四 多点触摸开发

## 实验要求

1. 了解Linux下的触摸屏驱动接口，学习Input event的使用和多点触摸协议的工作流程；
2. 掌握获取多点触摸的坐标及其参数，并在LCD上显示多点触摸轨迹；
3. 完成基于多点触摸协议的应用程序，选择绘制手指移动轨迹或者绘制手指触点进行应用开发。

## 实验过程

1. **补充函数touch\_event\_cb(int fd)**

这里我选择了绘制手指移动轨迹的开发模式。在函数touch\_event\_cb中，需要存储每个手指的状态（是否按下）和坐标（供下次触摸事件发生时绘制直线）。

当触摸事件为TOUCH\_PRESS时，将对应手指标记为按下状态并存储坐标，同时判断按下坐标是否在绘制的清空屏幕按钮范围内，如果是则调用函数init清空屏幕；当触摸事件为TOUCH\_MOVE时，根据上次坐标和当前坐标绘制线条（这里绘制的线条需要调用fb\_draw\_line\_weight函数以增加线宽，并为不同手指定义不同的线条颜色），并更新上次坐标为当前坐标；当触摸事件为TOUCH\_RELEASE时，将对应手指标记为松开状态。

1. **编译运行**

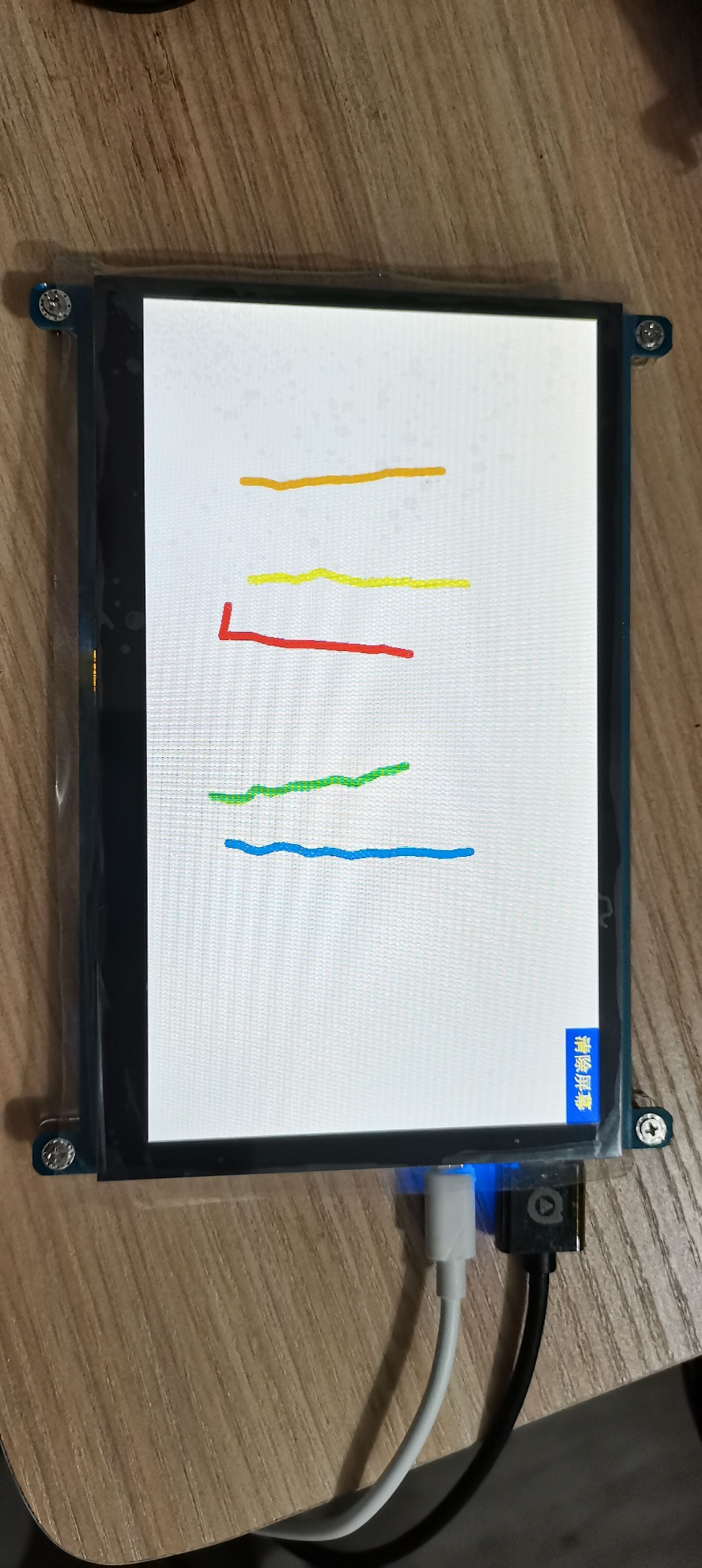
编译运行过程与实验二类似。

1. **测试效果**

通过测试多根手指同时触摸时的轨迹颜色与平滑度、点击清除屏幕按钮是否清空轨迹来验证功能的正确性。

## 实验结果

运行lab4程序后，开发板最终效果如图4.1所示。



**图4.1 lab4运行效果图**

关键函数实现如下方代码块所示。



# 实验五 蓝牙通讯

## 实验要求

1. 正确配置并启动蓝牙服务；

2. 熟练使用蓝牙的常用命令工具；

3. 通过RFCOMM串口（蓝牙串行通讯）实现；

4. 无线通讯，测试通过lab5实验代码。

## 实验过程

1. **配置启动蓝牙服务并配对**

首先修改开发板的配置文件并重启；然后使用hciconfig查看蓝牙是否正确初始化；最后使用bluetoothctl扫描蓝牙设备，选择手机蓝牙后进行配对，双方确认后即配对成功。

1. **蓝牙串行通讯**

开发板运行rfcomm -r watch 0 1指令建立蓝牙Socket并监听；手机端通过使用SPP蓝牙串口软件连接开发板。连接成功后开发板端将会有反馈。

1. **编译运行**

编译运行过程与实验二类似。

1. **发送消息**

手机端选择UTF-8格式发送消息，正在运行lab5程序的开发板即可将消息绘制在屏幕上。

## 实验结果

运行lab5程序并在手机端发送数条消息后，开发板最终效果如图5.1所示。



**图5.1 lab5运行效果图**

# 实验六 综合实验

## 实验要求

综合运用所学内容，设计并实现一个在多个开发板之间进行蓝牙互联协作的程序或一个功能比较复杂的单机程序。例如：

1. 共享白板：两个开发板之间共享屏幕手绘内容；

2. 共享文件：显示远程开发板上的目录文件列表,选中指定文件，显示指定文件内容(文本和png/jpg图片) ；

3. 联网游戏：比如五子棋等。

## 实验过程

1. **确定实验方案并编写代码**

本次实验我们选择开发共享白板，通过蓝牙在两个开发板间共享手绘内容。

本次实验基于实验四和实验五开发，但是我们又在实验四的基础上添加了自定义颜色和线宽的功能，提高了共享白板的实用性和用户体验。

本实验主要涉及到多点触摸事件touch\_event\_cb 和蓝牙串行通讯事件bluetooth\_tty\_event\_cb的处理。

在本机绘制内容时，需要将触摸事件参数（类型，坐标，线宽，颜色，手指）通过蓝牙缓冲写入函数myWrite\_nonblock传输给另一个开发板。

在本机接收到另外一个开发板发送的触摸事件参数时，通过蓝牙缓冲读取函数myRead\_nonblock读取缓冲区，并使用sscanf依次读取各个触摸事件参数，并以与处理触摸事件类似的方式绘制出另一个开发板绘制的内容。

1. **蓝牙串行通讯**

与实验五不同，本实验需要一个开发板作为服务端，另一个开发板作为客户端。两个开发板进行蓝牙配对后，服务端使用rfcomm -r watch 0 1进行监听，客户端使用rfcomm -r connect 0 SERVER-MAC 1连接到服务端。

1. **编译运行**

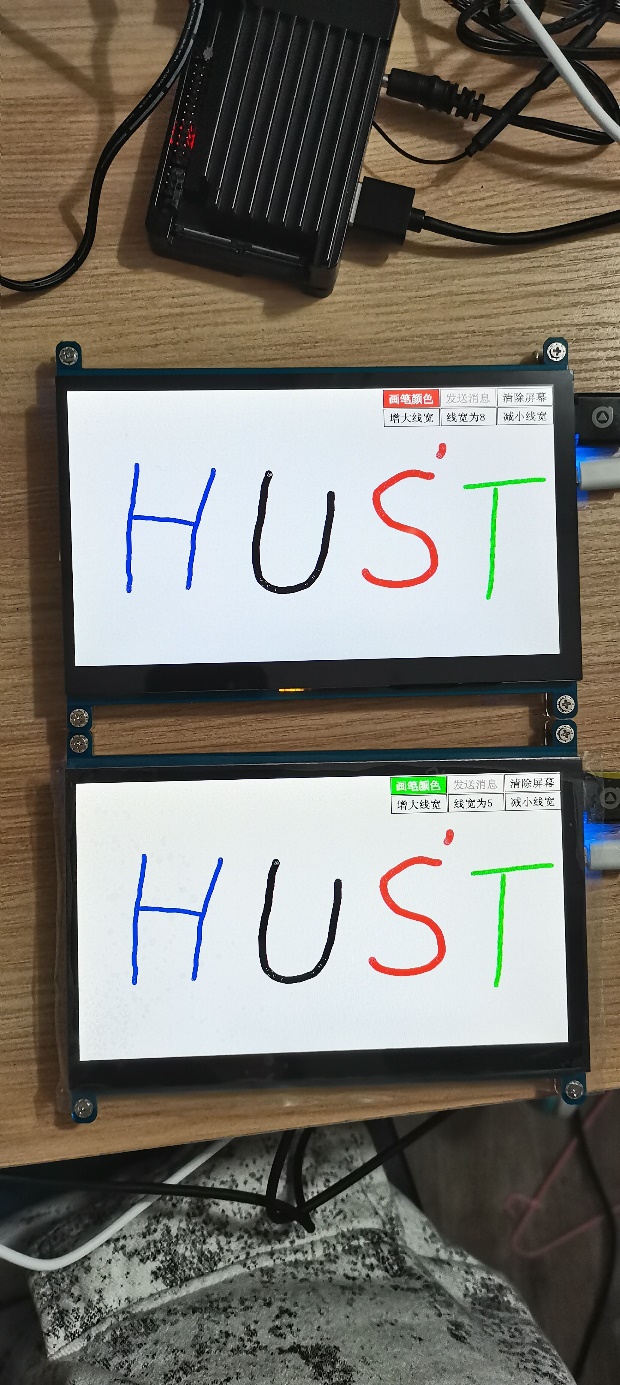
编译运行过程与实验二类似。

1. **测试效果**

服务端和客户端均运行team程序。通过双方使用不同的线宽、颜色、手指数量以及各自清空屏幕来验证程序的正确性。

## 实验结果

运行team程序并在双方使用不同颜色、线宽手绘内容后，开发板最终效果如图6.1所示。



**图6.1 team运行效果图**

关键函数init实现如下代码块所示。



关键函数touch\_event\_cb实现如下代码块所示。



# 实验总结与建议

## 实验总结

实验1是整个实验的基础——烧录系统并搭建工作环境。

实验2通过编写画线函数和画矩形函数，让我了解了framebuffer的工作原理与其双缓冲机制的工作流程，并初步掌握了framebuffer应用的开发过程。

实验3通过编写画图函数，使我对于framebuffer开发更加得心应手，同时也掌握了不同类型图片的区别所在。

在实验4中，我学习了触摸屏驱动接口的使用，掌握了多点触摸协议的工作原理，并在此基础上完成了绘制手指移动轨迹并可清除屏幕的功能。

在实验5中，我掌握了如何在不同设备之间进行蓝牙互联，认识到蓝牙连接的方式与网络通信中的Socket工作原理很相似。

在实验6中，我和我的小伙伴选择了基于蓝牙互联的共享白板这一选题，在实验4和实验5的基础上，通过同时使用触摸屏驱动接口和蓝牙驱动接口，实现了在两个白板间共享手绘内容的功能。

通过以上实验的开发，我初步掌握了嵌入式系统的开发流程，也真正在开发过程中体会到了乐趣，当看到自己开发出的程序可以在不同白板之间共享内容时，成就感油然而生。同时，我也认识到Linux开发的重要性，只有不断地自主创新，才能避免别国科技封锁造成的影响。

## 实验建议

本课程的实验大多围绕framebuffer进行开发，较为基础。希望老师能够开发其他难度相对提高并体现嵌入式系统特点的实验。