运行前，查找触摸屏的设备号，并更改程序

在开发板使用 ls /dev/input，因为触摸屏被当做输入设备

我的是 /dev/input/event1

main

1.关于debug\_manager:

功能：实现了终端和网络调试功能

其中，manager结构为初始化管理器，注册管理器，注册了2个调试器。定义了调试器的struct结构，规定了调试的等级。

终端调试器：没啥好说的

网络调试器，将本程序作为服务端，创建了发送线程和接收线程，

其中，接收线程主要用来接收外部客户端的指令：包括设置客户端地址，更改调试等级，更改打印通道等。使用时，需要实现定义外部客户端，一般在本机Linux设备上，然后向开发板通信，发送数据为“setclient”时，开发板会获得发送方的socket地址，从而记录下来，发送线程的时候通过该地址发送到客户端。

发送线程，则为将调试信息通过udp发送到客户端。其中，发送的过程并非一次性发送，因为如果由于网络问题导致数据并未发送成功，并不希望这个数据就此丢失。因此，设置了环形数据缓冲区。当调试信息通过接口时，会写入环形数据区，然后唤醒发送函数，再进行发送。如果发送失败，缓冲区数据也在。

其中，关于条件变量和互斥锁的结合可以了解一下。

keys:

debug\_manager:c中日志文件的写法。不同于c++，利用了可变参数列表。

2.关于input\_manager

InputInit() 注册stdin和touchscreen结构体到链表

AllInputDeviceInit，初始化stdin和touchsrceen，跟设备相关。然后创建两个子线程，子线程while(1)，不断调用结构体实例的getInputEvent函数，非阻塞方式获得输入（多线程），得到值后把获得的值（临时变量）赋给全局变量，因此该线程主要作用是不断获得输入。

之后要设置外部接口获得该输入事件，因此在manager函数设置GetInputEvent,主要是给外部获得输入事件，把全局变量赋给输入。因为全局变量涉及到线程安全，因此需要互斥锁，同时需要条件变量唤醒和阻塞全局变量的获得，需要在实例中获得输入后唤醒manager中外部接口，此时外部才能拿到数据（实例拿到后，才能唤醒外部接口去拿）

1. disp\_manager 除了电子书的常规，更多使用了显存的管理VideoMem，构建了显存链表，提前从内存中根据需要生成多张显存，并标注ID，随后可以根据ID和状态提取显存数据。这样，完成了主线程显示和子线程准备prepare图片的功能。

视频链接：<https://www.bilibili.com/video/BV1Sh4y1r7cR?p=36&vd_source=b5bc3f93b784d3162c93bae07108c1db> 显存管理 小节

**GetVideoMem分析**

*// 主要情况有，1.输入ID存在，则提取该显存数据，确定是主线程需要还是子线程需要 2.输入ID不存在，则为新数据，找到空闲显存（此时数据也为空，方便注入）*

*// 3.如果找不到空闲的且为空的数据，那么没办法，找到一个空闲的显存，但是已有数据，只能选择将原有数据覆盖（也就是预先分配的内存有限，只能供优先的使用）*

**AllocVideoMem**

**分配显存，首先分配一块供framebuffer的显存，也就是可以直接Lcd显示的，然后根据需要设置num，即提前分配显存的个数，isDevFm为0，也就是仅内存分配。**

**然后用内存VideoMemHead链表串起来，接下来无论是子线程还是主线程都可以从这个链表中找到显存数据以供调用。**

1. page

page中需要区分pt\_layout 和pt\_page\_layout

page\_layout是页面布局

layout是图标布局 --单个图标

main.page纯粹是图片的显示，比较简单，只要先获取对应ID的内存，然后计算图标的位置，再一一放进去，合并即可，

browse\_page 文件浏览器有点意思，最基础的计算功能图标的位置并显示，即向上，选择等，还有文件和目录的显示，及其名字显示。

Dirclosed和DirOpened分别代表？与选择按钮相关，在browsepage中

1.选择按钮的使用还是没搞明白

2.很奇怪，为什么在点击文件或者目录的地方写debug，显示不出来

额外：

1.

char strTmp[256];

struct stat tStat;

if ((stat(strTmp, *&tStat*) == 0) && S\_ISDIR(tStat.st\_mode))

stat(strTmp,&tStat)函数 获取文件的状态信息，包括权限、作者、文件类型等，并给tStat,获取成功返回0，S\_ISDIR(tStat.st\_mode))j检查st\_mode来判断是否是目录

2.

struct dirent \*\*aptNameList; //目录指针

number = scandir(strDirName, &aptNameList, 0, alphasort);

scandir() 函数用于读取目录的内容并返回一个指向 struct dirent 结构体数组的指针列表。

&aptNameList 是一个指向 struct dirent \*\* 的指针，scandir() 将填充这个指针，使其指向一个动态分配的数组，数组中的每个元素都是一个指向 struct dirent 的指针，代表目录中的一个条目

<https://lxblog.com/qianwen/share?shareId=4544e751-34c2-41db-b6b1-469437e831a8>

3.

**snprintf**(strTmp, 256, "%s/%s", s\_strCurDir, s\_ptDirContents[DirFileContentIndex]->strName);

1. 关于mmap，内存

<https://lxblog.com/qianwen/share?shareId=9761d540-9756-4f42-bd70-a0f070579209>

ptFileMap->pFileMem = (unsigned char \*)mmap(NULL, tStat.st\_size, PROT\_READ | PROT\_WRITE, MAP\_SHARED, fd, 0);

NULL：映射的地址，NULL表示让内核挑选一个合适的地址，PROT\_READ | PROT\_WRITE权限，映射地址和原文件共享，fd表示原文件地址

附加功能：

提示： micicom连接时终端显示有延迟，不利于进行调试

本身功能障碍

1. 上一页和下一页有Bug,即使没有超过1页的内容，依然可以下一页，但是是空白

解决：browse\_page.cpp 735行 下一页时的判断条件改为<=s\_DirContentsNumber

1. interval的加图标和减图标没有，网上能找到的是png格式，写png格式支持

1. 连播模式黑屏

附加：

1. 单击各个文件或者目录，右下角显示基本信息，双击才是进入
2. 文件与目录的双击不同，目录是进去，txt等可显示文件进行文件内容的显示。这里类似电子书
3. 对各个文件进行区分，最基本的txt 、代码文件等，图标ubuntu应该有，仿照即可。

扩展

1. 接入鼠标和键盘
2. 使用QT组件实现部分功能，涉及Qt库的使用
3. 相册 是否可以加入图片类型转换功能，各个类型图片的相互转换

附加：

1. 解决板子时不时黑屏的bug

**write**(s\_FBFD,"\033[9;0]",8);*//加入解决黑屏问题，原理还不清楚*

未解决

eg.1.头插法和尾插法的应用场景，为什么要用头插法。