**LCD显示图像**

市面上主流的图片格式有哪些？

jpg jpeg png gif bmp

bmp图片组成：54字节头数据（图片的长和宽）+ 颜色数据

bmp图片的像素点的组成：BGR--> 3字节

分辨率800\*480的bmp图片的大小：800\*480\*3+54=1152054字节

获取bmp图片

注意： 不能直接修改图片文件的后缀名

 任意获取一张图片，格式任意

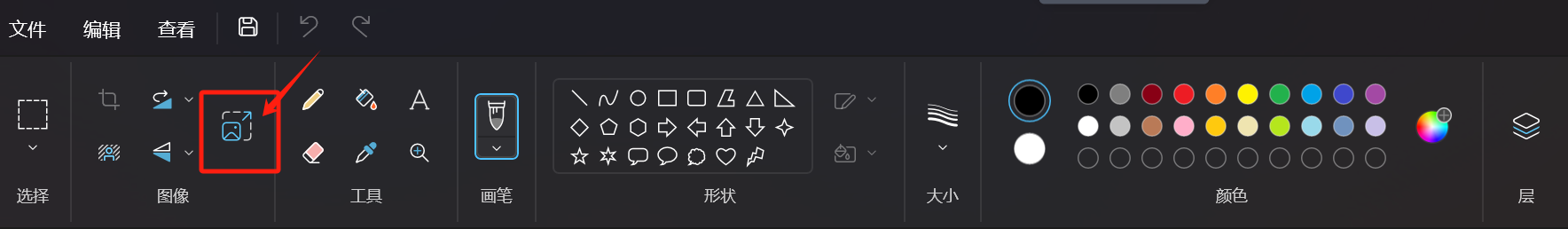
 打开方式选择画图

 菜单栏--》重设大小和倾斜--》修改分辨率：800\*480

win10的画图如下：



win11的画图如下：





 菜单栏--》文件--》另存为--》bmp图片--》1.bmp--》确定



注意：如果修改过后的bmp图片大小不是1152054字节，那么就是有问题的

映射空间

头文件：

#include <sys/mman.h>

参数：

void \*mmap(void \*addr, size\_t length,

int prot, int flags,

int fd, off\_t offset);

addr： NULL，系统会默认自动寻到一个合适的位置

length：映射空间的大小：800\*480\*4

prot：PROT\_READ

|PROT\_WRITE

flags：MAP\_SHARED // 共享

fd：lcd\_fd

offset： 0

返回值：

成功： 映射空间的起始地址

失败：(void \*)-1 等价于 MAP\_FAILED

int munmap(void \*addr, size\_t length);

addr： 映射空间的起始地址（mmap函数的返回值）

length： 映射空间的大小：800\*480\*4

返回值：

成功：0

失败：-1

代码（注意：bmp图片也要传到开发板上）

#include <stdio.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include <sys/mman.h>

int main(int argc, char const \*argv[])

{

// 1. 打开两个文件（lcd屏幕、bmp图片）

int lcd\_fd = open("/dev/fb0", O\_RDWR);

int bmp\_fd = open("GGbond.bmp", O\_RDWR);

if (lcd\_fd==-1 || bmp\_fd==-1)

{

printf("fail to open file\n");

return -1;

}

// 2. 先获取bmp图片的像素数据

lseek(bmp\_fd, 54, SEEK\_SET);

char buf[800\*480\*3];

bzero(buf, 800\*480\*3);

// bzero是用来清空数组的

read(bmp\_fd, buf, 800\*480\*3);

// 3. 写入获取到的像素数据到lcd屏幕文件中

int \*mmap\_start = (int \*)mmap(NULL, 800\*480\*4,

PROT\_READ|PROT\_WRITE,

MAP\_SHARED, lcd\_fd, 0);

if (mmap\_start == (void \*)-1)

{

printf("mmap error\n");

return -2;

}

int x,y,n;

for(y=0,n=0; y<480; y++)

{

for(x=0; x<800; x++,n+=3)

{

\*(mmap\_start+800\*(479-y)+x) = buf[n+0]<<0|

buf[n+1]<<8|

buf[n+2]<<16|

0<<24;

}

}

munmap(mmap\_start, 800\*480\*4);

// 4. 关闭两个文件（lcd屏幕、bmp图片）

close(lcd\_fd);

close(bmp\_fd);

return 0;

}

**触摸屏的使用**

需要了解两个东西

事件

输入子系统

举例：我向左滑动屏幕，之后又点击了一次屏幕

5次事件

触摸屏设备文件：/dev/input/event0

存储数据要的是结构体，结构体在：/usr/include/linux/input.h

struct input\_event {

struct timeval time;// 事件发生的时间（不做讨论）

\_\_u16 type;// 事件的类型，压力事件、坐标事件

\_\_u16 code;// 事件的编码，对事件的再次细分

\_\_s32 value;// 事件的值

};

分类

type：

#define EV\_KEY 0x01// 压力事件

#define EV\_ABS 0x03// 坐标事件

code:

#define BTN\_TOUCH 0x14a// 压力

#define ABS\_X 0x00// 坐标轴x

#define ABS\_Y 0x01// 坐标轴y

value:

value==0// 松开手

value==1// 没有松开手

value就是对应的坐标

坐标轴的原点（类似第四象限（x、y轴都>0），原点在左上角）

黑色框（新）的坐标范围： 1024\*600

蓝色框（旧）的坐标范围： 800\*480

注意： 为了一致性，黑色框的同学就要把坐标范围按比例缩小成800\*480

代码1：研究压力事件

如果输出的汉字是乱码，记得修改编码格式为UTF-8

#include <stdio.h>

#include <linux/input.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

int main(int argc, char const \*argv[])

{

// 1. 打开文件

int fd = open("/dev/input/event0", O\_RDWR);

if (fd == -1)

{

printf("open input error\n");

return -1;

}

// 2. 读取文件内的数据

struct input\_event buf;

// 压力事件

while(1)

{

read(fd, &buf, sizeof(buf));

if (buf.type==EV\_KEY && buf.code==BTN\_TOUCH && buf.value==0)

{

printf("松开手\n");

}

if (buf.type==EV\_KEY && buf.code==BTN\_TOUCH && buf.value==1)

{

printf("没有松开手\n");

}

}

// 3. 关闭文件

close(fd);

return 0;

}

代码2：研究坐标事件

#include <stdio.h>

#include <linux/input.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

int main(int argc, char const \*argv[])

{

// 1. 打开文件

int fd = open("/dev/input/event0", O\_RDWR);

if (fd == -1)

{

printf("open input error\n");

return -1;

}

// 2. 读取文件内的数据

struct input\_event buf;

// 坐标事件

// 蓝色框的范围：800\*480

// 黑色框的范围：1024\*600

// 黑色框的同学要记得等比例缩小到800\*480

int x,y;

while(1)

{

read(fd, &buf, sizeof(buf));

if (buf.type==EV\_ABS && buf.code==ABS\_X)

{

x = buf.value;

x = x\*800/1024;// 黑色框才需要这行代码

}

if (buf.type==EV\_ABS && buf.code==ABS\_Y)

{

y = buf.value;

y = y\*480/600;// 黑色框才需要这行代码

}

if (buf.type==EV\_KEY && buf.code==BTN\_TOUCH && buf.value==0)

{

printf("松开手:%d, %d\n", x, y);

}

}

// 3. 关闭文件

close(fd);

return 0;

}

**整合--电子相册**

分析项目需要用到哪些功能？

默认会显示一张图片

点击左边显示上一张图片

点击右边显示下一张图片

点击中间退出程序

封装需要用到的功能（写成一个函数）

压力事件和坐标事件

显示800\*480分辨率的图片

编译的命令：

arm-linux-gcc src/\*.c -o bin/main -I inc

作业：

程序启动时，两个播放供用户选择

自动播放（sleep(1); // 休眠一秒的意思）

手动播放（直接拿来用就行）