
实验五 电子相机实验

一、实验目的：

1. 了解 Linux 中摄像头的使用方法
2. 了解不同颜色空间的图像表示方法，并学会不同颜色空间图像相互转换的方法
3. 学会使用 QT 进行基本的界面设计，绑定按钮函数
4. 学会使用 QT 定时器触发任务
5. 学会使用 QT 中显示、保存图像的功能

二、实验器材：

iTop-4412 实验开发板一套，配套摄像头 OV5640 模块一个，PC 机一台（Windows 操作系统），U 盘一个（FAT 格式）

三、实验原理：

1. Linux 摄像头操作框架——Video for Linux 2(V4L2)

V4L2 是 Linux 内核中关于视频设备的内核驱动框架，为上层访问底层的视频设备提供了统一的接口。操作摄像头的步骤较为复杂，因此只讲解用户使用时可以设定参数的部分。可以设定的参数包括：具体摄像头的设备地址，图像分辨率，图像像素颜色格式。以例程代码 camera.cpp 为范例讲解。

a) 打开设备文件

例程中这部分代码在

```
bool Camera::open_device(void);
```

函数中。该函数主要使用了两个底层函数：

```
stat(dev_name, &st);
```

```
open(dev_name, O_RDWR /* required */ | O_NONBLOCK, 0);
```

在第二章串口和文件 IO 中已经学过，Linux 中一切皆文件，不论是普通的文件，还是硬件设备。其中，stat 函数用于判断文件是否存在，open 用于打开特定文件。本实验使用的摄像头模块设备地址为/dev/video0。

b) 设置视频格式

例程中这部分代码在

```
bool Camera::init_device(void);
```

函数中。首先要创建一个 v4l2_format 结构体变量，这个变量用于设定视频格式。

```
struct v4l2_format fmt;
```

然后对结构体变量的元素赋值，主要的元素有图像宽、高、像素颜色格式等。本实验默认使用 640*480 分辨率，像素颜色格式为 YUYV 也称 YUV2 格式。

```
fmt.fmt.pix.width = width;//设置图像宽度
```

```
fmt.fmt.pix.height = height;//设置图像高度
```

```
fmt.fmt.pix.pixelformat = V4L2_PIX_FMT_YUYV;//设定图像像素颜色格式
```

最后使用 xioctl 函数，使设定生效。

```
xioctl(fd, VIDIOC_S_FMT, &fmt)
```

2. 颜色空间转换

实验中，默认配置下，获取到的原图像（未经编码的原始图像）是 YUYV 格式的。这种

格式是 YUV 格式的一种。而在 QImage 中显示图像，要求是 RGB 格式的原图像，所以需要进行颜色空间的转换。

a) YUV 颜色空间

YUV(亦称 YCrCb)是被欧洲电视系统所采用的一种颜色编码方法。在现代彩色电视系统中，通常采用三管彩色摄像机或彩色 CCD 摄影机进行取景，然后把取得的彩色图像信号经分色、分别放大校正后得到 RGB，再经过矩阵变换电路得到亮度信号 Y 和两个色差信号 R-Y(即 U)、B-Y(即 V)，最后发送端将亮度和两个色差总共三个信号分别进行编码，用同一信道发送出去。这种色彩的表示方法就是所谓的 YUV 色彩空间表示。采用 YUV 色彩空间的重要性是它的亮度信号 Y 和色度信号 U、V 是分离的。如果只有 Y 信号分量而没有 U、V 信号分量，那么这样表示的图像就是黑白灰度图像。彩色电视采用 YUV 空间正是为了用亮度信号 Y 解决彩色电视机与黑白电视机的兼容问题，使黑白电视机也能接收彩色电视信号。

YUV 格式图像在存储时还分为打包格式和平面格式。打包格式存储时，每个像素点的色彩连续存储；平面格式存储时，图像像素点的相同颜色分量连续存储，如第一段全为 Y，第二段全为 U，第三段全为 V。

b) YUYV 打包格式到 RGB 格式的转换

本实验中，从摄像头获取到的图像都是**未经编码的原始图像**。也就是图像中每个像素点都使用固定长度的数值来记录颜色。YUYV 打包格式中，每两个像素的颜色值用四个色彩分量表示，这两个像素共用 UV 值，也即四个色彩分量分别为：第一个像素的 Y 值，共有的 U 值，第二个像素的 Y 值，共有的 V 值。

RGB 图像格式使用三个颜色分量，分别代表红绿蓝三原色的分量。两者间的转换公式为（假设两者颜色分量的最大值都相同）：

$$R = 1.164*(Y-16) + 1.159*(V-128);$$

$$G = 1.164*(Y-16) - 0.380*(U-128) + 0.813*(V-128);$$

$$B = 1.164*(Y-16) + 2.018*(U-128);$$

知晓这两个原理，我们就可以编程实现 YUYV 格式原始图像转换到 RGB 格式原始图像。

转换时需要注意，YUYV 四个分量对应两个像素值。

3. 图像的显示和存储

显示图像使用的函数为：

```
[QLabel Object]->setPixmap([QPixmap Object]);
```

方括号内部表示，这是一个某类型的变量。该函数的意义是，用 QT 的这个部件显示一个 [QPixmap Object] 内部的图像。

但是 QPixmap 只能从图像文件中读取图像。QImage 才可以直接从原始图像数据中读取每个像素颜色值。两者的转换函数为：

```
QPixmap::fromImage ([Qimage Object]);
```

QImage 是 QT 中较为基础的图像类，它可以从编码好的图像文件中读取图像并显示到 QT 界面中，或者从原始图像数据中读取每个像素颜色值并显示到 QT 界面中。从原始图像数据中读取图像的函数为：

```
QImage(frameBufRGB, width, height, QImage::Format_RGB888)
```

四个参数含义分别为，原始图像数据，宽，高，图像格式。这里使用最简单的 RGB888 格式，即每个像素使用 3 个 8 位无符号整数分别存储 R，G，B 三个分量，整个数组的大小为长*宽*3 个字节。按照默认的配置，为 640*480*3=921,600 字节。

图像的存储也采用 QImage 类来实现。使用的函数为

```
[Qimage Object]->save(filename, "JPG", -1);
```

函数的前两个参数分别为保存的文件名，编码格式。编码格式可以有 PNG，JPG，BMP

等。

综上所述，实验中图像的显示需要：使用 QImage 读取原始图像数组，转换为 QPixmap 对象，显示在界面的一个控件里。

实验中保存图像只需要使用 QImage 的 save 接口，并指定文件名和格式即可。

4.QT 绘制界面过程和其它可能用到的 QT 函数

常见函数包括：connect, timer 的用法等

详见实验压缩包中附带的 PPT

四、实验内容

1. QT 界面绘制和交互逻辑的编写

原程序只有打开图片文件并显示的功能，并且窗口分辨率也不符合实验中使用的屏幕（480*272）。需要每位同学调整窗口大小和按钮位置，并且加入两个新按钮：打开摄像头与摄像保存。

填写按下“打开摄像头”和“摄像保存”两个按钮触发的函数，并且要绑定函数与按钮。需要达到的效果为：按下“打开摄像头”按钮，按钮名称变为“关闭摄像头”；若此时按下“关闭摄像头”按钮，则“摄像保存”为灰。

2. 电子相机功能的实现

填写按下“打开摄像头”和“摄像保存”两个按钮触发的函数。需要达到的效果为：按下“打开摄像头”按钮，界面中会显示实时的摄像头图像；此时按下“摄像保存”按钮，可以保存当前显示的图像到 U 盘内。若此时按下“关闭摄像头”按钮，摄像头关闭，原显示图像的区域清空。

3. 多张图像拍摄

按时间戳命名图像文件名，使得拍出的图像不会覆盖上一次的图像。

4. 提高部分

加入 QT 组件（如 comboBox 或 spinBox 等），使图像刷新闻隔从 0.1s~5s 可调。

五、实验步骤

1. 仔细阅读实验原理，以及 PPT 中的基础教学，尤其是使用 camera 测试摄像头；
2. 完成 QT 界面绘制，使用 QT Creator 打开 qt1.ui 文件，调整窗口大小和按钮位置，并且加入两个新按钮：打开摄像头与摄像保存。该版本 QT 对中文支持不太好，尽量在界面上显示英文。下拉菜单功能的使用可以参考例程中的演示代码，例程可以实现在界面上显示下拉菜单中选中的字符串。
3. 填写 qt1.cpp 中的 void Qt1::fun_cap_open()完成打开摄像头的触发函数（提示：使用 QT 的定时器触发摄像头图像的刷新）；填写 qt1.cpp 中的 void Qt1::fun_take_photo()完成拍摄照片并保存的触发函数。注意要绑定槽函数，把按钮和函数关联起来。填写 camera.cpp 中的 bool Camera::process_image(unsigned char *imageSrc,unsigned char * imageDst)，完成 YUYV 颜色空间到 RGB 颜色空间的转换。注意函数参数的顺序。注意色彩空间转换函数的 size 为像素点的个数（长*宽）
4. 按时间戳命名图像，注意图像名称中不能有空格和冒号。
5. 提高加入 QT 组件（如 comboBox 或 spinBox 等）和相应的逻辑，使图像刷新闻隔从 0.1s~5s 可调。注意范围达到即可，如果是 comboBox 这类离散的选择控件，需要覆盖 0.1~5s 内至少 4 个时间间隔（且包括 0.1 和 5s）

六、实验注意事项：

摄像头插入时，要注意板子和摄像头之间的防呆箭头需要对在一起。

如果摄像头第一次打开成功，但是由于强制退出等原因导致第二次无法打开，推荐敲入

`reboot` 指令重启。如果重启还不行，推荐 `poweroff` 关机后断电，等待十来秒再插电。