

## 2.3 jpg 格式及 libjpeg 库\_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 2.3 jpg 格式及 libjpeg 库涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

JPEG（Joint Photographic Experts Group）即联合图像专家组，是用于连续色调静态图像压缩的一种标准。遵照此标准产生的图像文件就是 jpg 格式文件，文件后缀名为 .jpg 或 .jpeg。它的特点如下

1) 在进行压缩时，允许用户自由调整图像质量（1-100）； 2) 该格式的文件尺寸相对较小，有利于传输和存储； 3) 因为压缩算法的存在，图像会一定程度的失真，不适宜用于高清图像；

注意：

图片的格式不是后缀这么简单，它是指图片像素数据如何组织起来。

比如：图片文件中的第 14 字节记录图片宽度，第 58 字节记录图像高度，第 9 字节开始逐行记录所有的像素 RGB 数据，那么这就是一种人为约定的简易格式；

这也就是说：只要把 RGB 数据按照 jpg 的格式进行存储，就能得到了 jpg 图像；

方式一：可以直接到官网下载后使用，地址如下：

方式二：以在虚拟机里面直接下载，我这里采用在虚拟机中下载 1.5.3 版本：

1) 在 /opt/arm/ 目录下新建 libjpeg 目录存放相关文件

2) 进入 /opt/arm/libjpeg 目录下载：

1) 解压 `sudo tar -zxvf libjpeg-turbo-1.5.3.tar.gz`

2) 新建 libjpegMakeResult 存放编译后的文件

3) 进入源码目录 `cd libjpeg-turbo-1.5.3/`

4) 配置编译参数：`sudo ./configure CC="/opt/arm/RV1126/toolchain/gcc-arm-8.3-2019.03-x86_64-arm-linux-gnueabi/bin/arm-linux-gnueabi-gcc" --host="arm-linux-gnueabi" --prefix="/opt/arm/libjpeg/libjpegMakeResult" --enable-static`

5) 执行 make 编译

`make install`

安装完成后，在 libjpegMakeResult 有 include,lib,share,bin 四个文件夹，include 为头文件，lib 为库文件，lib 下面同时有动态库和静态库，这里使用静态库。

使用时，把 include 下的头文件和 lib 下的 libturbojpeg.a, libjpeg.a 两个文件拷贝到工程目录即可；或者修改环境变量将路径添加到系统环境变量下，推荐使用后者。

## 6.3 使用 libjpeg 将 YUV 数据转为 jpeg 图片

这里提供一个固定的流程，至于为什么是这个样子的，需要查阅其源码分析压缩算法，鉴于目标不是研究算法，所以这里并不深究 jpeg 的压缩算法。

```
/*
 * @img: yuyv422格式图像原始数据
 * @fp: yuyv压缩为jpeg格式后的数据存放文件的文件描述符
 * @width: 图像宽度
 * @height: 图像高度
 * @quality: 压缩质量(1-100)
 */
int yuyv_to_jpeg(const char *yuv, FILE *fp, int width, int height, int quality)
{
    struct jpeg_compress_struct cinfo; //定义一个压缩对象.
    struct jpeg_error_mgr jerr; //用于存放错误信息.
    JSAMPROW row_pointer[1]; //一行位图.
    cinfo.err = jpeg_std_error (&jerr); //错误信息输出绑定到压缩对象.
    jpeg_create_compress (&cinfo); //初始化压缩对象.
    jpeg_stdio_dest(&cinfo, fp); //将保存输出数据的文件的文件描述符与压缩对象绑定.
    cinfo.image_width = width;
    cinfo.image_height = height; //图像的宽和高, 单位为像素.
    cinfo.input_components = 3; //3表示彩色位图, 如果是灰度图则为1.
    cinfo.in_color_space = JCS_RGB; //JCS_RGB表示彩色图像.
    jpeg_set_defaults(&cinfo); //采用默认设置对图像进行压缩.
    jpeg_set_quality(&cinfo, quality, TRUE); //设置图像压缩质量.
    jpeg_start_compress(&cinfo, TRUE); //开始压缩.
    //申请buf空间, 大小为yuyv数据转换为rgb格式后每一行的字节数.
    unsigned char *line_buf = calloc(width, sizeof(rgb_t)); //开辟行缓冲内存
    rgb_t *px = (rgb_t *)line_buf;
    int r, g, b;
    int y, u, v; //对每行yuv数据进行rgb转换
    while (cinfo.next_scanline < height) { //逐行进行图像压缩
        for (int col = 0; col < width; col += 2) {
            ... //此处将获得的YUV数据转换为RGB数据后写入到行缓冲当中
        }
        row_pointer[0] = line_buf;
        jpeg_write_scanlines(&cinfo, row_pointer, 1); //将行数据写入压缩对象
    }
    jpeg_finish_compress(&cinfo); //压缩完成
    jpeg_destroy_compress(&cinfo); //释放申请的资源
    free (line_buf);
    return 0;
}
```

使用 eog 命令查看图片文件,

\$eog filename.jpg

---

全文完

本文由 简悦 SimpRead 优化, 用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta, 点击查看详细说明

