

# 嵌入式多媒体智能终端系统

班级：22 物联网工程 2    姓名：                      学号：

## 1. 实验简介（不少于 300 字）

嵌入式多媒体智能终端系统实验中，我们使用了 arm-linux-gcc 工具链、Ubuntu 16 虚拟机和 Windows 11 宿主机，结合 VScode 开发环境，以及粤嵌 Linux 开发板 GEC6818，搭建了一套嵌入式多媒体智能终端系统。该系统旨在提供一系列多媒体功能，主要包括单张图片播放、多张图片轮询播放、简易电子相册、音频播放和音乐播放器等功能。通过这些功能的实现，用户能够在嵌入式平台上体验基本的多媒体展示和播放效果。

开发环境选择了 Ubuntu 16 虚拟机，宿主机为 Windows 11 X86 架构系统，以便于在虚拟机中进行交叉编译和调试。VScode 作为主要的开发工具，配合 arm-linux-gcc 交叉编译工具链，实现对 GEC6818 开发板的程序编译和调试。为了实现嵌入式开发板与主机的通信，我们使用了 SecureCRT 串口通讯工具，确保了与开发板的稳定连接。

该多媒体智能终端系统包含了多种展示形式和互动方式。例如，单张图片播放功能可通过简单的操作进行图片的显示；多张图片轮询播放功能则可以在预设的时间间隔内自动切换不同的图片，呈现简易电子相册的效果；音频和音乐播放器功能则支持音频文件的播放，用户可以根据需求播放背景音乐或音频内容。这些功能为终端系统的应用提供了基础，同时也为后续的系统优化和扩展奠定了基础。

## 2. 实验内容

- (1) 使用嵌入式开发板 GEC6818 显示自己喜欢的颜色。
- (2) 使用嵌入式开发板 GEC6818 显示一张和屏幕大小匹配的图片。
- (3) 使用嵌入式开发板 GEC6818 显示任意大小的图片。
- (4) 使用嵌入式开发板 GEC6818 按一定时间播放图片。
- (5) 使用嵌入式开发板 GEC6818 制作简易电子相册。
- (6) 使用嵌入式开发板 GEC6818 制作音乐播放器。

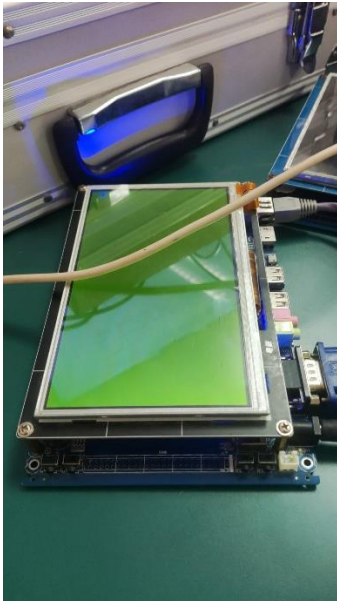
### 2.1 图片显示

### (1) 显示自己喜欢的颜色

实验原理：颜色可以使用 RGB 来表示，把 RGB 转换成 16 进制赋值给也屏幕的每一个像素点。

实验步骤：VScode 编写代码，上传到 Ubuntu 16 虚拟机使用 arm-linux-gcc 交叉编译，使用 SecureCRT 把编译后的可执行文件上传到开发板，给可执行文件赋予执行权限，运行可执行文件。

实验结果如下：

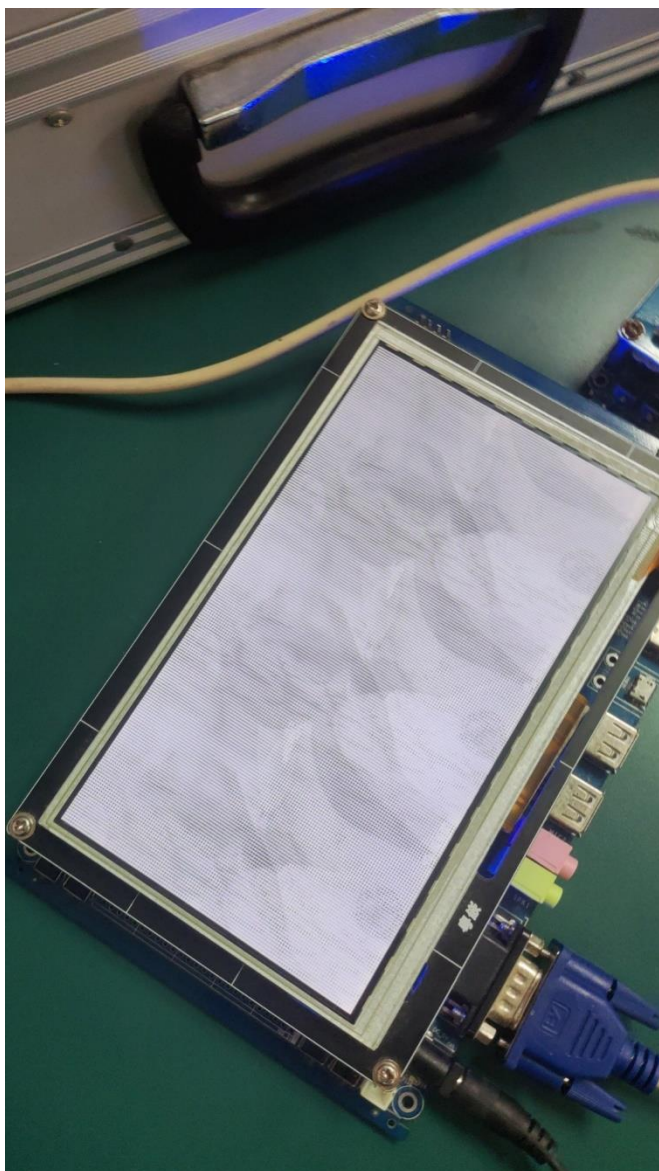


### (2) 显示一张图片

实验原理：读取图片的每一个像素点，把他赋值给屏幕对应的像素点

实验步骤：选取一张与屏幕规格一样的图片上传到开发板，VScode 编写代码，代码里面的图片名称要与上传的一致，上传到 Ubuntu 16 虚拟机使用 arm-linux-gcc 交叉编译，使用 SecureCRT 把编译后的可执行文件上传到开发板，给可执行文件赋予执行权限，运行可执行文件。

实验结果如下：

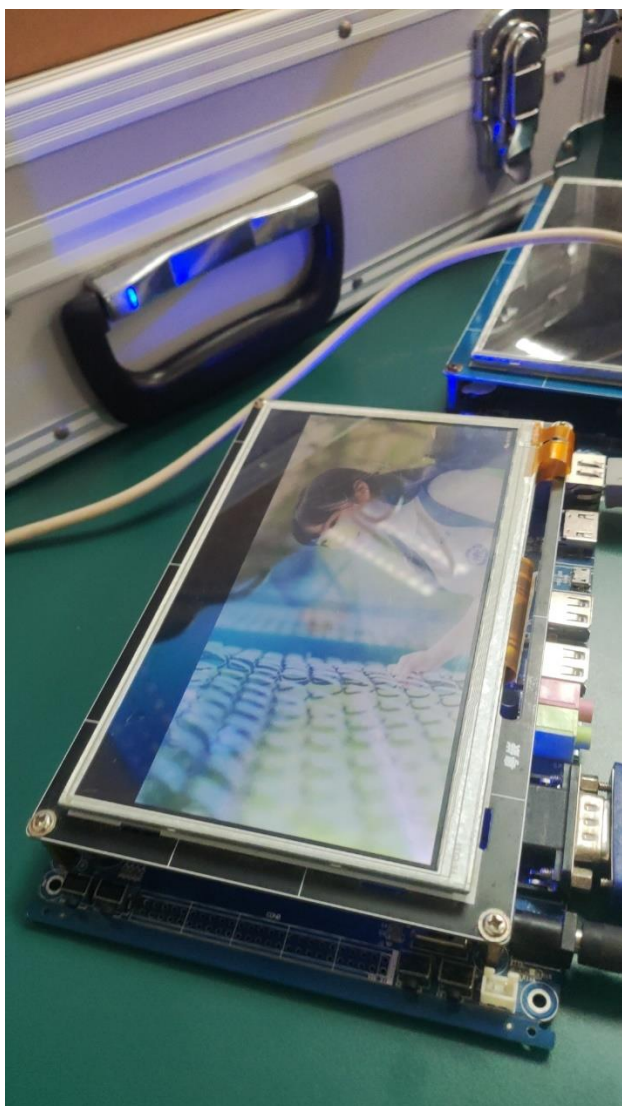


### (3) 显示一张任意大小的图片

实验原理：原理：读取图片的每一个像素点，把多余的像素点抛弃，如果像素点不够全屏，那就空出来，把他赋值给屏幕对应的像素点上。

实验步骤：选取一张任意大小的图片上传到开发板，VScode 编写代码，代码里面的图片名称要与上传的一致，上传到 Ubuntu 16 虚拟机使用 arm-linux-gcc 交叉编译，使用 SecureCRT 把编译后的可执行文件上传到开发板，给可执行文件赋予执行权限，运行可执行文件。

实验结果如下：

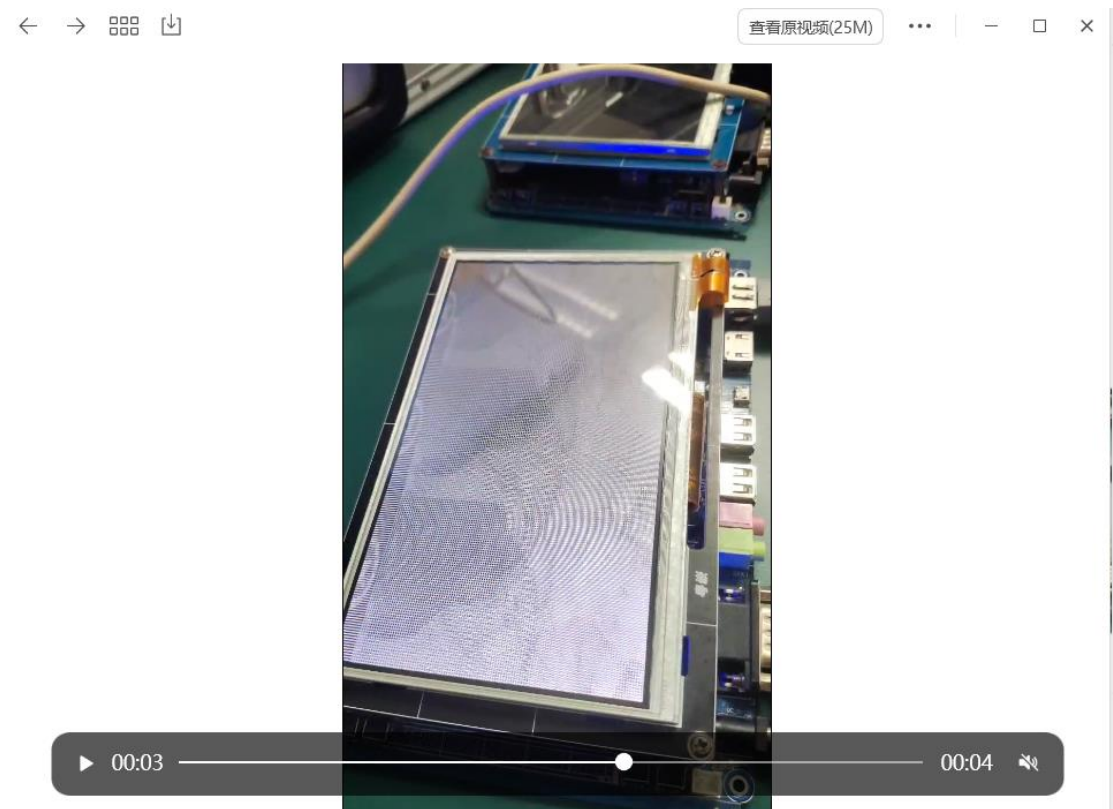
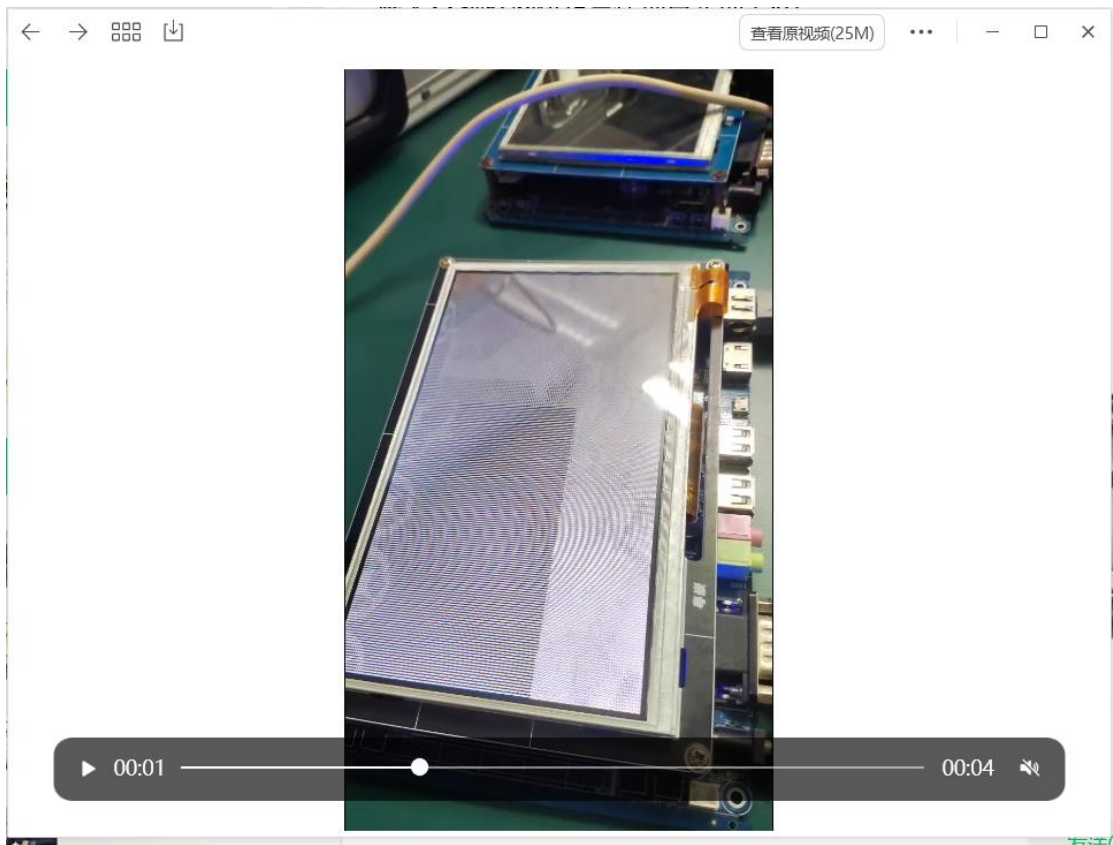


#### (4) 图片轮询播放

实验原理：使用线程睡眠技术，让主线程睡眠几秒再轮询读取图片进行显示。

实验步骤：选取 4 张任意大小的图片上传到开发板，VScode 编写代码，代码里面的图片名称存储在一个数组里面，我们对数组进行遍历，把代码上传到 Ubuntu 16 虚拟机使用 arm-linux-gcc 交叉编译，使用 SecureCRT 把编译后的可执行文件上传到开发板，给可执行文件赋予执行权限，运行可执行文件。

实验结果如下：



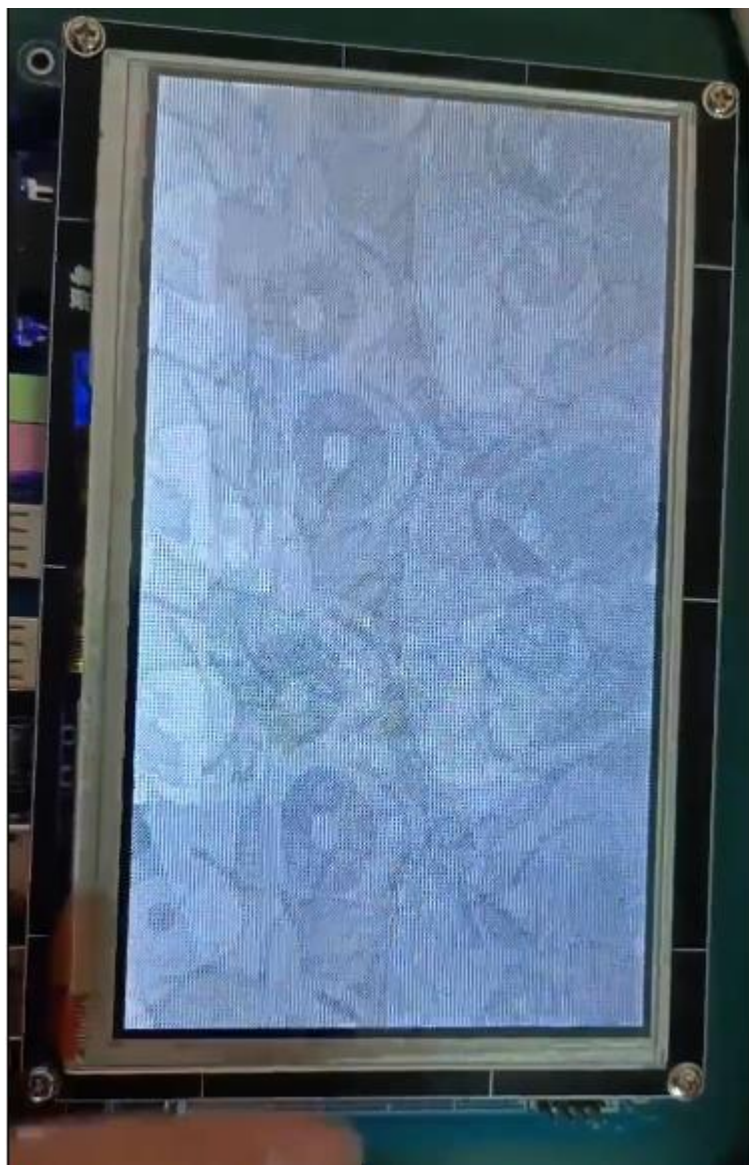
(5) 简易电子相册



实验原理：监听屏幕点击事件，屏幕点击触发数组遍历，进行相册浏览

实验步骤：选取多张任意大小的图片上传到开发板，VScode 编写代码，代码里面的图片名称存储在一个数组里面，当触发屏幕点击事件时，对数组进行下标+1 显示图片，把代码上传到 Ubuntu 16 虚拟机使用 arm-linux-gcc 交叉编译，使用 SecureCRT 把编译后的可执行文件上传到开发板，给可执行文件赋予执行权限，运行可执行文件。

实验结果如下：



## 2.2 音频播放器

### （1）音频播放器

实验原理：读取音频 MP3 文件，对 MP3 文件进行解码，使用音频驱动将 PCM 数据写入音频设备缓冲区，使用 ALSA 函数和屏幕点击事件实现暂停，播放功能。

实验步骤：选取喜欢的 MP3 上传到 GEC6818 开发板，使用 VScode 编写代码，代码里面的音频名称存储在一个数组里面，当触发屏幕点击暂停事件时，调用暂停函数，点击播放时调用播放函数，把代码上传到 Ubuntu 16 虚拟机使用 arm-linux-gcc 交叉编译，使用 SecureCRT 把编译后的可执行文件上传到开发板，给可执行文件赋予执行权限，运行可执行文件。

实验结果：



### 3. 实验总结

本次实验成功搭建了一个嵌入式多媒体播放器系统，基于粤嵌 GEC6818 开发板，通过使用 Ubuntu16 虚拟机和 Vscode 进行开发。系统实现了图片单张播放、多张图片轮询播放、简易电子相册、音频播放及音乐播放器等功能。通过音频解码器、图片解析算法以及 ALSA 音频框架的结合，播放器能够流畅地处理多种格式的音视频数据，达到了实验预期效果。

实验结果表明，系统能够稳定运行，界面响应较快，功能切换流畅。但也发现了一些不足之处，例如音频解码部分对于高压缩比文件可能会出现短暂延迟，图片轮询播放过程中偶尔存在错位，颜色失真的现象。此外，界面功能设计尚可优化，用户体验还需进一步提升。

为了改进，可以尝试优化音频和图片处理的算法，减少资源占用，提高解码效率。同时，可以引入硬件加速功能（如 GPU 解码），提升系统性能。进一步研究可以围绕支持更多多媒体格式的解码功能、丰富 UI 设计以及加入更多智能交互功能

展开，例如通过语音控制播放器，实现更加便捷的用户体验。实验为嵌入式多媒体开发奠定了坚实基础，同时也为进一步优化提供了方向

#### 4. 心得体会

通过本次嵌入式多媒体智能终端系统的开发实践，我深刻体会到嵌入式开发是一项软硬件结合的综合性工程。整个过程中，工具链的配置、代码编写和调试等环节都需要严谨的态度和扎实的技术基础。在开发图片轮询播放和音频播放功能时，我学会了计算机处理图片数据的原理，通过错误的示范我更加深刻了解了图片的“位”。同时，通过 SecureCRT 与开发板的串口通讯实践，我更加熟悉了嵌入式设备与主机交互的原理和实际操作。

在课程学习中，我建议可以增加更多关于屏幕驱动原理的讲解，特别是对初学者来说，这些内容能够帮助快速入门。此外，增加一些实际案例分析，例如快速刷新屏幕，也能让课程内容更加丰富且贴近实际需求。

下一步的学习计划中，我希望进一步深入学习嵌入式操作系统的内核机制，特别是与多媒体相关的驱动开发。同时，计划学习更多关于音视频编解码技术的知识，以便未来能够在资源受限的嵌入式环境中实现更高效的多媒体功能。此外，我还希望加强对于嵌入式系统安全性的研究，完善整体系统的稳定性和可靠性。