**实 验 报 告**

|  |  |  |  |
| --- | --- | --- | --- |
| **课程名称：** |  | | |
| **学 院：** |  | | |
| **专 业：** |  | **班级：** | 2019级 班 |
| **姓 名：** |  | **学号：** |  |

**2022年 6 月 20日**

**山东科技大学教务处制**

**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **组 别** |  | **姓 名** |  | **同组实验者** | 无 |
| **实验项目名称** | 触摸图片展台 | | | **实验日期** | 6月 20 日 |
| **教 师 评 语** |  | | | | |
| **实验成绩：** | | | **指导教师（签名）：**  2022年 月 日 | | |
| 1. **实验目的**   编写程序，实现SD卡存储图片的交替显示，把试验台上改造成图片展台。   1. **实验要求**   从Fat32格式的SD卡的pictures目录读取图片文件，并交替显示，支持jpg和bmp格式的图片。实现暂停和继续功能，暂停时可以通过触摸屏实现下一张。当程序出现故障或SD卡读写时通过LED不同闪烁频率指示   1. **实验内容** 2. 移植液晶显示驱动 3. 使用CubeMX实现Fat32格式读取框架的建立 4. 实现文件的循环读取 5. 实现jpg和bmp格式图片的打开和显示 6. 移植触摸板驱动 7. 实现暂停继续和下一张 8. LED状态指示 9. **实验原理** 10. **移植液晶显示驱动**   在示例项目中可以找到BSP\_TFTLCD和BSP\_Graphics两个驱动文件，这两个驱动文件是使用固件库编写的，而我们需要使用CubeMX的Hal库编写，它们的头文件和API显然是不兼容的，需要进行移植。移植的过程并不复杂，固件库中任何一个API都有对应的Hal库实现，我们只需要将它们替换过去即可；固件库中的一些变量类型定义，如u32和vu32，则从固件库中找到定义搬运过来即可；引脚和模块功能的初始化部分则全部依靠CubeMX自动生成代码，不需要再单独编写和调用函数，只保留外设的初始化，如显示白屏。  图示, 示意图  描述已自动生成  阅读RA8875的手册可知，其中有一个SRAM供控制的MCU操作，然后RA8875以SRAM中的数据为依据控制液晶显示图像，且均为16位RGB565。在操作其他制式的图像时需要进行转换。   1. **使用CubeMX实现Fat32格式读取框架的建立**   FATFS是为MCU编写的一个Fat文件系统读写库，在CubeMX的“中间件”选项中选中FATFS，并在模式中选中SD Card即可实现对Fat32格式的SD卡的读写。     1. **实现文件的循环读取**   由f\_mount、f\_opendir、f\_readdir、f\_read、f\_close配合，即可实现对目录的循环读取。首先由f\_mount对文件系统进行挂载，然后由f\_opendir打开指定的目录，如果没有这个目录则可以调用f\_mkdir建立这个目录；f\_readdir将从头读取该目录下的所有文件，如果是支持的图片格式则由f\_read读取，读取操作完成后由f\_close关闭。   1. **实现jpg和bmp格式图片的打开和显示**   由于不涉及压缩算法，bmp的格式比较简单的读取比较简单，只需要按照格式的要求获取全部的像素并显示即可。注意C结构体默认是以最长元素的长度对齐的，这可能会导致读取出现错误，故需要#pragma pack(1)和#pragma pack()指定这部分以字节对齐。  不同于bmp，jpg涉及到了一些压缩算法，但是有现成的TJpgDec库来解码的，且和FATFS库配套，可以一起使用。   1. **移植触摸板驱动和实现下一张**   触摸板使用了I2C总线，但是FSMC和硬件I2C在硬件上有冲突而不能同时使用。可以注意到示例代码的所有I2C实现都使用了模拟I2C，这样一来移植示例代码就成了最好的选择。  触摸板我们只需要实现下一张的功能，所以只需要在暂停的时候读取触摸屏状态（判断是否被触摸），而触发下一张的标志就是触摸点出现了变化。     1. **实现暂停继续和LED状态指示**   暂停和继续由按键中断实现，由一个flag标记暂停状态即可，这和之前的按键中断实验是一样的原理。LED状态指示则由和LED实验的原理一样，读写指示只要在读写过程中翻转LED阴极电平。     1. **实验步骤** 2. **移植液晶显示驱动**   涉及到需要自行定义的数据类型如下：   |  |  | | --- | --- | | **数据类型** | **别名** | | uint\_32t | u32 | | uint16\_t | u16 | | uint8\_t | u8 | | \_\_IO uint\_32t | vu32 | | \_\_IO uint16\_t | vu16 | | \_\_IO uint8\_t | vu8 |   由于使用的引脚常量不同，也需要加以修改，如GPIO\_Pin\_10改为GPIO\_PIN\_10。  CubeMX配置如下：  表格  描述已自动生成  图形用户界面, 文本, 应用程序, 电子邮件  描述已自动生成  图形用户界面, 文本, 应用程序, 电子邮件  描述已自动生成   1. **使用CubeMX实现Fat32格式读取框架的建立**   CubeMX会自动生成FATFS库和SD卡相关代码，我们只需要了解FATFS的API使用。配置如下：       1. **实现文件的循环读取**   根据实验原理解释的算法，编写得到的核心代码如下：     1. **实现jpg和bmp格式图片的打开和显示**   Bmp格式编码非常简单易懂，一个14字节的文件头表示了文件类型和数据起始位置，随后的40字节包含我们需要的图片位深度和大小，根据位深度和数据起始位置，就可以读到所有像素并显示。不过需要注意的是，像素的排布方式是从下到上、从左到右的。  结构体定义如下，注意是字节对齐的：    对于不同位深度需要对像素进行不同的处理，如RGB233和24位需要统一更为RGB565显示。转换如下：    Jpeg格式图片则可以通过调用TJpgDec实现，只需要实现两个输入输出函数即可，代码中分别为in\_jpg\_func和out\_jpg\_func，但是要注意jpg图片解码后是一个一个像素块，要根据其位置和大小显示在液晶的对应位置上。  核心输出代码如下：    由于TJpgDec为我们实现了图像的缩放功能，对于大于液晶屏像素大小的图片，可以计算其缩放倍率：     1. **移植触摸板驱动和实现下一张**   触摸屏驱动使用实验箱例程中的模拟I2C驱动，依赖BP6和BP7两个引脚，在CubeMX中配置如下：    调用GSL2681\_IRQ\_Callback()后，当前触摸位置会被存放在PENX和PENY中，如果没有被触摸，则数据不变。  根据是否有触碰而实现下一张的核心代码如下：     1. **实现暂停继续和LED状态指示**   由外部中断简单触发，使用软件防抖的方式防止单次按下多次触发，核心代码如下：      在程序出错时进入预定义的错误处理函数，并LED闪烁，实现错误状态指示。读写状态指示只需要在读写的前后加入LED翻转的语句即可。错误处理函数如下：     1. **实验结果**   成功实现了从SD卡的pictures目录循环读取图片，对256色、16位和24位真彩的bmp以及普通的jpg格式都可以很好地支持，色彩上几乎没有色差地显示在显示屏上。在按键按下时可以实现暂停和继续，在暂停的状态下可以通过触摸电容屏实现下一张的切换。  由于图片显示需要时间，暂停和继续的操作并不能立即反馈出来，所有所有操作都以信息提示的方式显示在了液晶屏上，实现了很好的人机交互。   1. **实验总结**   实验本身并不是很难，但是对于实验课上并没有做过的FATFS、SD卡读写、电容屏驱动，以及资料匮乏的TJpgDec库，我需要花费大量的时间进行调试，对代码进行微调以实现功能和交互上的统一。  由于系统是单线程的，而图片的解码和显示需要大量的CPU时间进行处理，导致系统的所有交互并不能立即反馈，如果加入FREERTOS则可以在交互上再上一个台阶。 | | | | | |