

# 嵌入式系统与应用课程设计

电子信息 14-01，信息工程 14-01

## 一、课程设计的目的

嵌入式系统应用课程设计是配合理论课程而设立的设计性实践环节。

其目的是：

- 1、提高综合运用所学的专业基础理论、基本技能和专业知识分析问题和解决工程设计问题的能力；
- 2、培养调查研究，查阅技术文献、资料、手册，进行程序设计及编写技术文件的基本能力；
- 3、培养团队协作能力。

## 二、课程设计的任务和要求

任务：利用 UP-CUP4412 实验平台设计一款简单的电子相框。

要求：

- 1、实现电子相框的基本功能
  - (1) 在 UP-CUP4412 实验平台的 LCD 上显示一幅 BMP 图片
  - (2) 调整图片的显示位置使其显示在屏幕的中间位置
  - (3) 制作幻灯显示，在 LCD 上循环播放多幅 BMP 图片
- 2、给电子相框添加特效
  - (1) 从中间向上下展开
  - (2) 从中间向两边展开
  - (3) 淡入淡出
  - (4) 全屏显示
  - (5) 放大或缩小
- 3、组合功能设计
  - (1) 用 UP-CUP4412 实验平台的按键控制图片的顺序播放
  - (2) 用 UP-CUP4412 实验平台的按键控制特效

## 三、成绩评定

小组成绩：

- 1、基本完成设计题目所要求的内容，程序有一定的实用性，可扩充性，创

新性。( 40% )

2、课程设计报告。( 40% )

3、答辩及演示。( 20% )

个人成绩：按组长分工及报告分工评定。

#### 四 . 参考资料

1、UP-CUP4412 实验指导书

2、本文档

3、网络资源

#### 五 . 课程设计报告撰写要求

课程设计报告必须包括以下内容：

(1) 目录

(2) 设计目的 - 根据本组的实际情况写出各自的设计目的。

(3) 设计要求 - 根据设计任务的要求写出设计要求。

(4) 设计内容 - 给出原理分析、设计思路、程序结构流程图、各功能函数介绍 ( 包括自己编写和系统调用 ) 源程序代码及注释。

(5) 总结 - 本设计的心得体会、不足之处及未实现功能的设计思路。

要求：各小组组长认真划分本组成员的分工，各成员根据自己的分工完成上述报告的相关部分，并在目录中注明各部分的实际完成者和撰写者。

#### 附录 1：一般工作流程

##### 1、查找资料

(1) BMP 文件的格式

(2) Linux 平台下 C 程序打开文件及读取文件的系统函数调用规范

(3) UP-CUP4412 实验平台提供的 LCD 及键盘驱动函数

##### 2、研究并制定方案

(1) 如何读取 BMP 文件

(2) 如何将 BMP 文件显示在 LCD 上 ( 参考实验指导书 P124 - LCD 驱动实验 )

(3) 各特效的数学建模

(4) 如何实现按键控制 (参考实验指导书 P116 - 键盘驱动实验)

3、设计软件流程

4、编写代码

5、调试

## 附录 2：可能用到的系统函数

### 1. 打开文件函数 fopen

FILE \* fopen(const char \* path,const char \* mode);

**参数 path** 字符串包含欲打开的文件路径及文件名；

**参数 mode** 字符串则代表着流形态。mode 有下列几种形态字符串：

r 打开只读文件，该文件必须存在。

r+ 打开可读写的文件，该文件必须存在。

rb+ 读写打开一个二进制文件，只允许读写数据。

rt+ 读写打开一个文本文件，允许读和写。

w 打开只写文件，若文件存在则文件长度清为 0，即该文件内容会消失。若文件不存在则建立该文件。

w+ 打开可读写文件，若文件存在则文件长度清为零，即该文件内容会消失。若文件不存在则建立该文件。

a 以附加的方式打开只写文件。若文件不存在，则会建立该文件，如果文件存在，写入的数据会被加到文件尾，即文件原先的内容会被保留。(EOF 符保留)

a+ 以附加方式打开可读写的文件。若文件不存在，则会建立该文件，如果文件存在，写入的数据会被加到文件尾后，即文件原先的内容会被保留。

wb 只写打开或新建一个二进制文件；只允许写数据。

wb+ 读写打开或建立一个二进制文件，允许读和写。

wt+ 读写打开或建立一个文本文件；允许读写。

at+ 读写打开一个文本文件，允许读或在文本末追加数据。

ab+ 读写打开一个二进制文件，允许读或在文件末追加数据。

上述的形态字符串都可以再加一个 b 字符，如 rb、w+b 或 ab+ 等组合，加入

b 字符用来告诉函数库打开的文件为二进制文件，而非纯文字文件。

**返回值：**文件顺利打开后，指向该文件的指针就会被返回，文件打开失败则返回 NULL，并把错误代码存在 errno 中。

## 2. 读文件函数 fread

```
size_t fread(void * ptr,size_t size,size_t nmemb,FILE * stream);
```

**参数 \*ptr：**表示保存数据的地址；

**参数 size：**表示一次读取的数据单元大小；

**参数 nmemb：**表示读取的次数；

**参数 \*stream：**表示文件指针。

**返回值：**成功时 fread 返回的值与 nmemb 相等；若小于 nmemb 但是大于 0，则可能是到了文件末尾，不够次数；若返回 0，则是文件读取错误，不满一个 size 的大小。

## 3、移动文件指针位置函数 fseek

```
int fseek(FILE *stream, long offset, int fromwhere);
```

用于二进制方式打开的文件，移动文件读写指针位置。

**参数 stream：**为文件指针；

**参数 offset：**偏移量，整数表示正向偏移，负数表示负向偏移；

**参数 origin：**设定从文件的哪里开始偏移,可能取值为 :SEEK\_SET( 文件头 ) \ SEEK\_CUR ( 当前位置 ) \ SEEK\_END ( 文件尾 ), 也可分别用为 0 , 1 和 2 表示；

**返回值：**成功时则返回 0，若有错误则返回-1，errno 会存放错误代码。

例如：

```
unsigned int OffSet = 0; //定义整形变量 OffSet
```

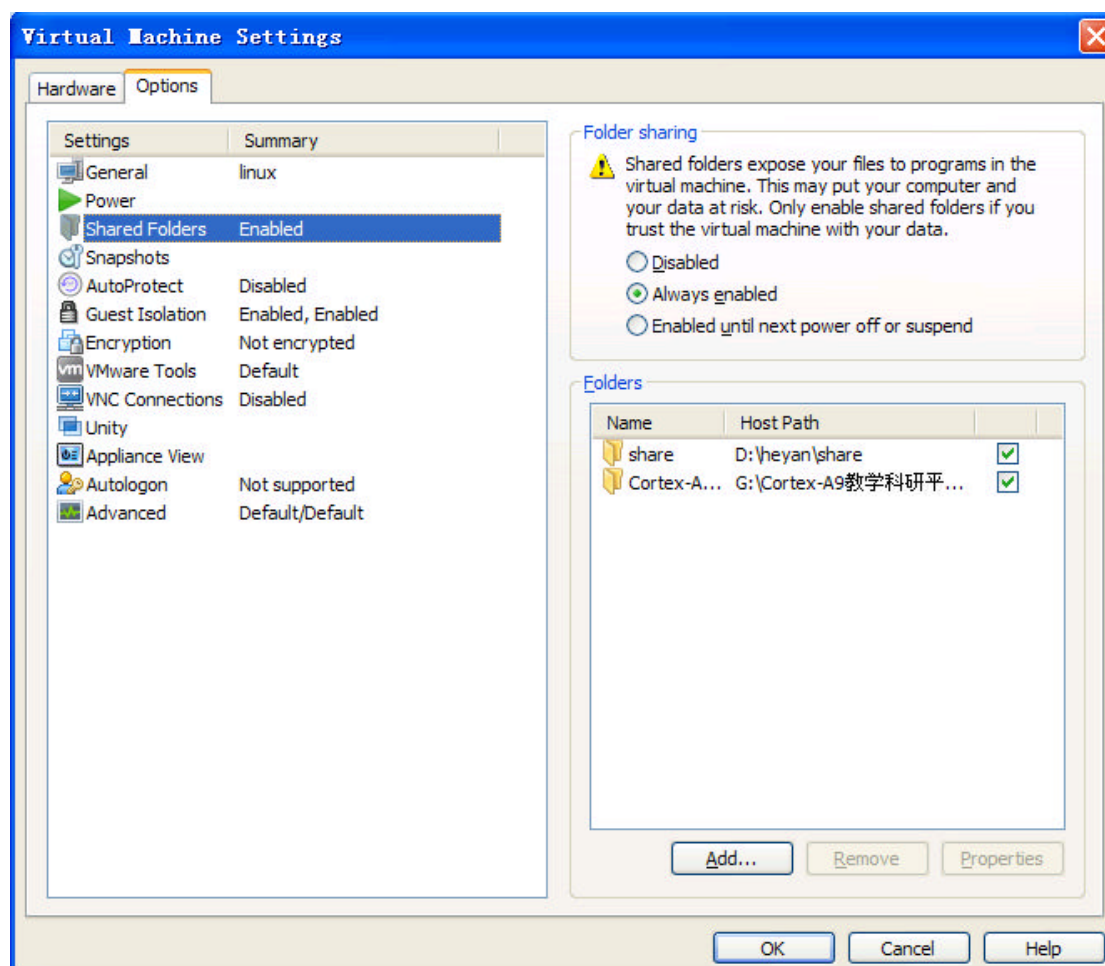
```
FILE *fpbmp = fopen("/mnt/hgfs/1.bmp", "rb");//以二进制方式打开只读文件返回指向文件的指针 fpbmp
```

```
fseek(fpbmp, 10L, SEEK_SET);//将文件指针 fpbmp 移到距离文件头 10 字节处
```

```
fread(&OffSet, sizeof(char), 4, fpbmp);//从上述打开的文件指定位置读取四次，每次读一个字节，读回的数据存入 OffSet
```

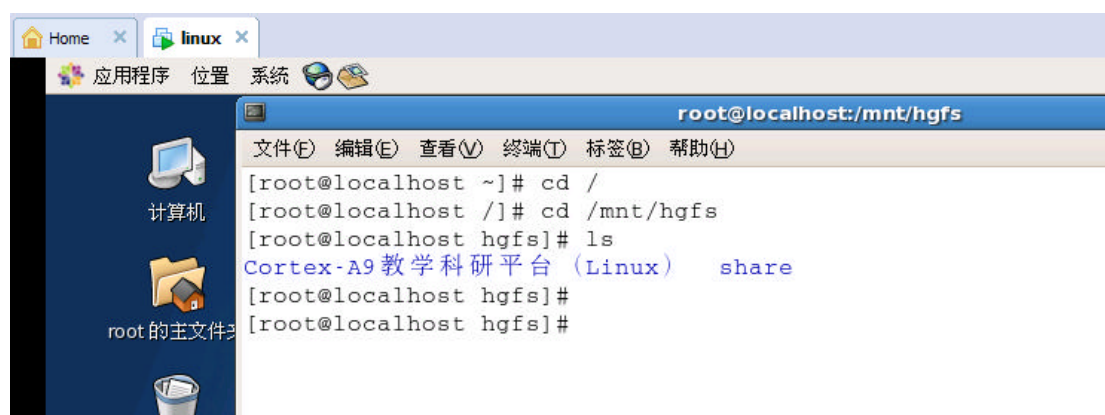
### 附录 3：Windows 与 Linux 共享文件（仅适用于 111 实验室的开发环境）

1. 打开 VMware Workstation
2. 运行虚拟机 Fedora14
3. 点击 Vm 菜单的 Settings 项，在弹出的对话框中选择 Options 选项卡的 Shared Folders，如下图所示：



- (1) 在右上侧的 Folder sharing 栏内选择 Always enabled 单选框；
- (2) 在右下侧的 Folders 栏内查看已共享的 Windows 文件夹，如果没有可用的，点击下方的 Add 按钮添加一下 Windows 下的共享文件夹。  
例如：本例选择 G:\Cortex-A9 教学科研平台（Linux）为 Windows 与 Linux 的共享文件夹。
4. 在 Windows 下，将要共享的文件复制到上述设置好的共享文件夹，例如，共享文件为 17.bmp,我们将 17.bmp 复制到 G:\Cortex-A9 教学科研平台（Linux）文件夹下。

5. 在 Linux 虚拟机上打开一个终端，进入/mnt/hgfs/目录  
在终端上键入如下命令：



```
root@localhost: /mnt/hgfs
[root@localhost ~]# cd /
[root@localhost /]# cd /mnt/hgfs
[root@localhost hgfs]# ls
Cortex-A9教学科研平台 (Linux)  share
[root@localhost hgfs]#
```

可以看到 Windows 下的共享文件夹正是刚才设置的

**Cortex-A9教学科研平台 (Linux) share**

进入该文件夹：

```
[root@localhost hgfs]# cd Cortex-A9教学科研平台 (Linux) /
[root@localhost Cortex-A9教学科研平台 (Linux)]# ls
17.bmp  CrossTools  install.sh  SRC  up-tech.ico  硬件文档
1.bmp   DOC          key_bmp.c  test_bmp.c  使用GPS跳线方式.jpg
2.bmp   IMG          readme.txt  TOOLS      启动参数.txt
[root@localhost Cortex-A9教学科研平台 (Linux)]#
```

可以看到要共享的文件 17.bmp。

6. 找到共享文件后，可以用“cp”命令将共享文件复制到目标文件夹。  
该命令的使用规范为 # cp 源路径及文件名 （空格） 目标路径