

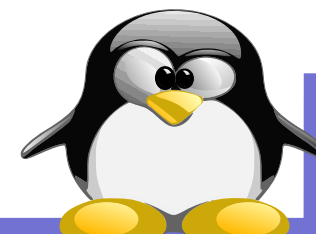


BeagleBone Black 实验系统

嵌入式系统实验

yfang@nju.edu.cn

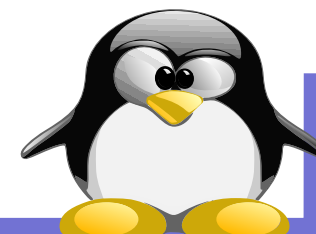
School of Electronics Science and Engineering
Nanjing University
Nanjing 210046



1. 注意事项



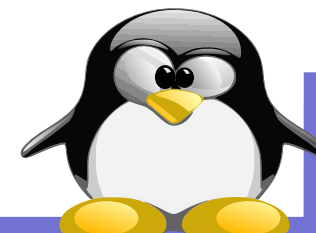
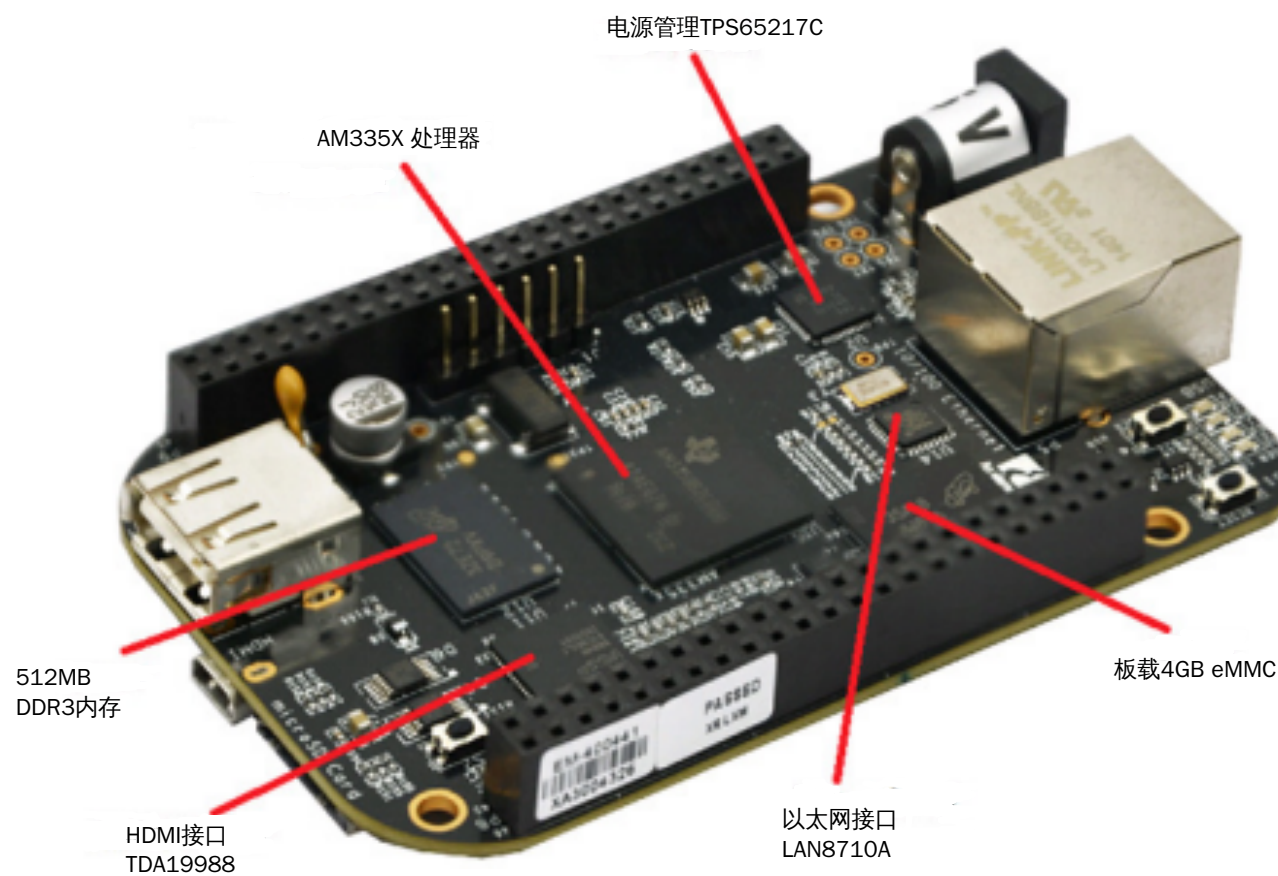
- 安全 (电源线、个人物品等)
- 习惯 (饮食、环境等)
- 把握时间, 控制节奏
- 学习方法 (笔记、资料、目的、过程)
- 实验要求 (过程、结果、报告)



2. 实验环境介绍



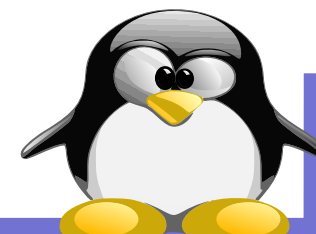
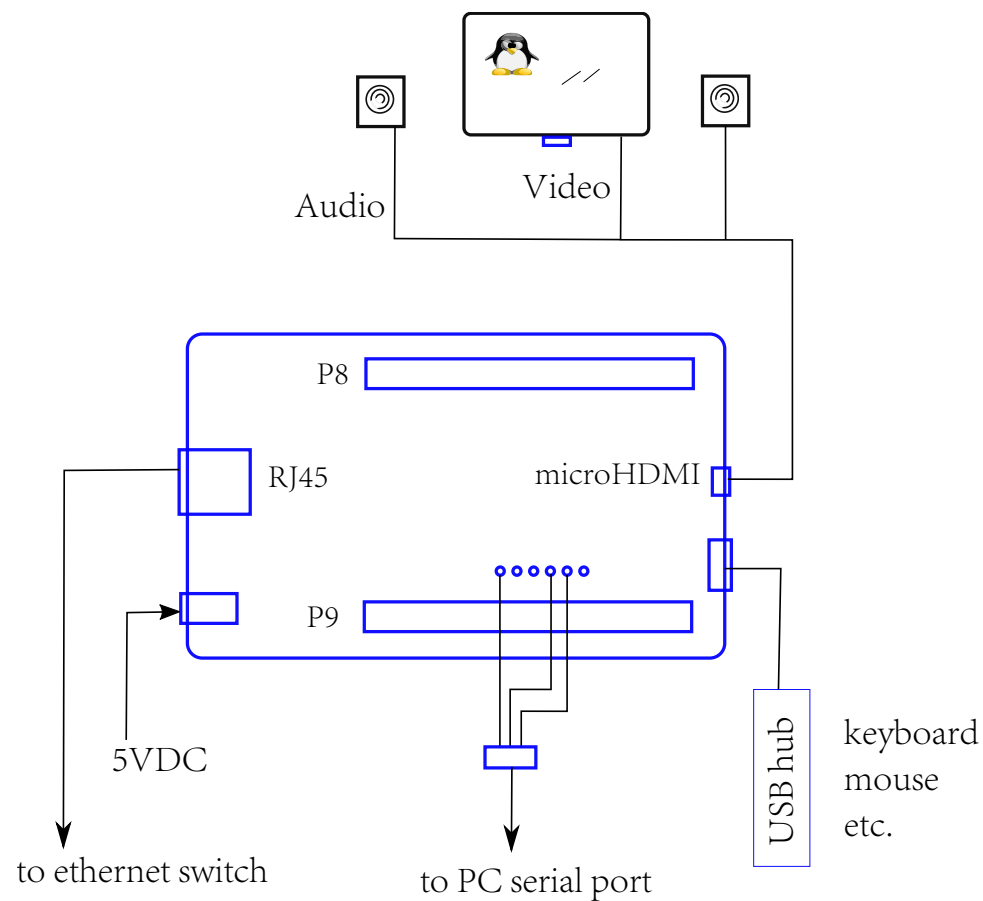
实验板布局



2. 实验环境介绍



BB Black 实用系统

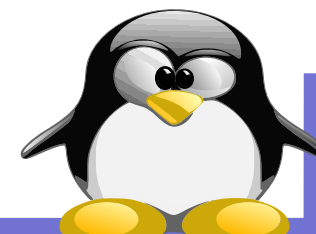


2. 实验环境介绍



实验环境

- 实验由个人计算机 (主机, 或称 host)、实验机 (目标机, 或称 target) 及目标机编译软件 arm-linux-gcc 组成
- 每个实验台上提供两个 IP:
个人计算机使用 192.168.208.xx (普通用户不可更改)
实验板的 IP 由实验人员自行设置, 应与实验室个人计算机处于同一网段, 建议使用 192.168.208.1xx 或 192.168.208.2xx
所有计算机连至实验室的总交换机
- 编译器路径:
/opt/armhf-linux-2018.08
请将编译器路径添加到环境变量 PATH:
`export PATH=/opt/armhf-linux-2018.08/bin:$PATH`



3. 实验方法



开发模式

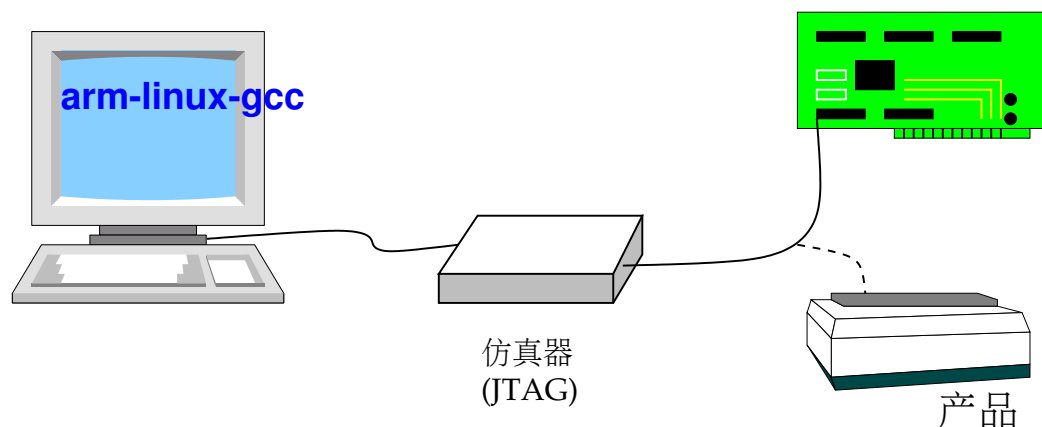
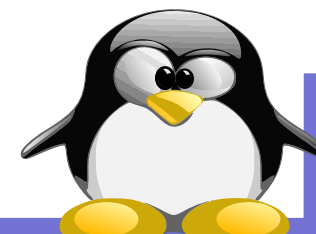


图 1: 宿主机/目标机的开发模式

实验板上电后自动启动 TF 卡或 eMMC 的 u-boot (bootloader 实验不做要求)。
利用宿主机编辑工具、交叉编译器开发软件,
通过下载工具或其它联接方式传到目标机使之运行。
u-boot 支持 RS232 和 tftp 两个协议。



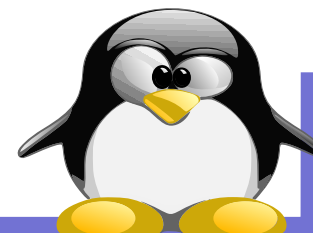
主机串口监控

串口监控软件可以使用 minicom 或 screen。

- **minicom -s**表示对串口进行设置 (只需在第一次使用 minicom 时设置)
- 串口设置 /dev/ttyUSB0、bps=115200、8 位数据、无校验、无流控制

u-boot 会向终端送出显示信息、接收键盘的命令并回显。在 u-boot 提示符下面可设定本机 IP、宿主机 IP、引导方式、启动参数等

如果引导参数设置正确，启动 Linux 后仍可继续通过串口监控。





Boot 的必要设置

- 配置 IP

配置目标机的 IP、tftp 服务器的 IP (服务器存放待下载的镜像文件)

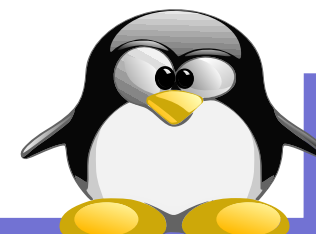
U-Boot # set ipaddr xxx.xxx.xxx.xxx (target IP)

U-Boot # set serverip xxx.xxx.xxx.xxx (tftp server IP)

- 检查网络连接状态 ping

- 配置启动选项 (内核地址、文件系统、监控终端等等)

U-Boot # set bootargs ... (串口、根文件系统。。。)

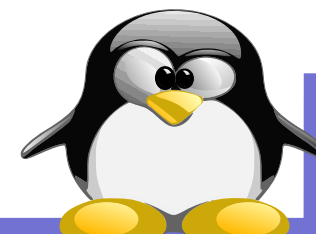




tftp 服务

- tftp 使用 UDP 协议
- tftp 的目的是为了通过网络加载 Linux 操作系统内核及根文件系统镜像
- tftp 的设置脚本文件在 /etc/xinet.d/tftp, 它设置了 tftp 的服务器目录
- PC 端重启 tftp 服务 (由管理员执行, 包括其它由 xinet 管理的服务) 可以执行

```
# /etc/rc.d/init.d/xinetd restart
```



3. 实验方法



加载 Linux 内核

基本 Linux 操作系统由两个文件构成:

- **zImage** — kernel

```
U-Boot # tftp 0x82000000 zImage
```

- **ramdisk_img.gz** — file system(ext2fs)

```
U-Boot # tftp 0x88080000 ramdisk_img.gz
```

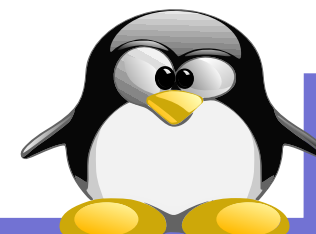
设备树文件 `am335x-boneblack.dtb` 来自内核, 加载方式:

```
U-Boot # tftp 0x88000000 am335x-boneblack.dtb
```

加载成功后可通过 `u-boot` 的命令引导启动:

```
U-Boot # bootz 0x82000000 - 0x88000000
```

或者将 `zImage` 和 `ramdisk_img.gz` 复制到 TF/eMMC 引导分区,
使用 `fatload` 读入内存。



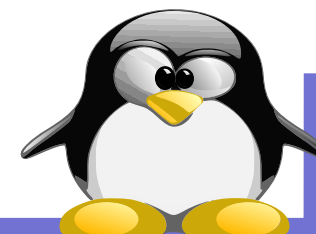
3. 实验方法



熟悉 Linux 环境

目标系统启动后，请尝试进行一些简单的操作：

- 配置目标机的 IP: `ifconfig`
- 主机和目标机之间互相 `ping`
- 尝试用另外的窗口或终端从主机 `telnet` 到目标机
- 熟悉目标机的 Linux 环境
- 使用 NFS 服务传输文件

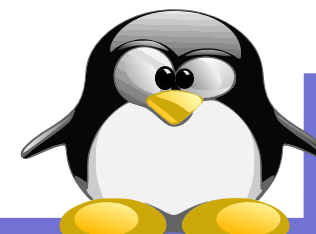




NFS 服务

NFS 服务的目的是将主机上的一个目录用来和目标机共享, 作为目标机的存储设备。

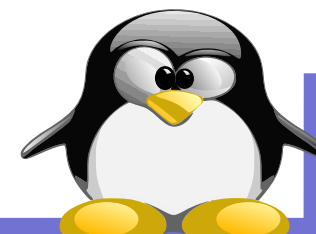
- NFS 服务器的目录及权限设置在文件 `/etc/exports` 中
- 主机作为 NFS 服务器, 启动 `nfs-server` (管理员负责):
`# service nfs-kernel-server`
- 目标机上执行 `mount` 命令, 将指定 IP 下的 NFS 共享目录挂载到 `/mnt` 目录
- 在主机和目标机上分别观察、操作共享目录下的文件



编写应用程序

- 在主机上编写一个简单的 C 程序并编译、运行
- 将可执行程序复制到 NFS 共享目录下, 尝试在目标机上运行该程序
- ???
- 在主机上用 arm-linux-gcc 编译器编译, 再尝试用两边的机器分别运行
- !!!
- 在目标机上编译.....@#\$\$%& * ()

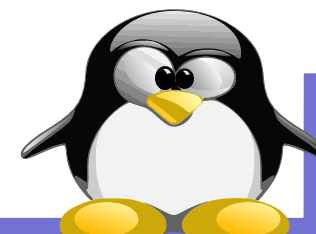
养成使用 “Makefile” 的习惯 (事半功倍)





源码获取

- 内核源码: <https://github.com/beagleboard/linux>。内核源码用于编译生成内核镜像文件 zImage
- BusyBox 源码: <https://busybox.net/busybox>。busybox 编译后用于构成根文件系统的核心部分。
- 其他移植软件可从互联网下载
- 因为同一计算机会有多人使用相同账号 student, 建议在 /home/student 目录下建立自己的主目录。
- 一些频繁的操作, 建议使用便签记录, 复制 → 粘帖可以提高效率。

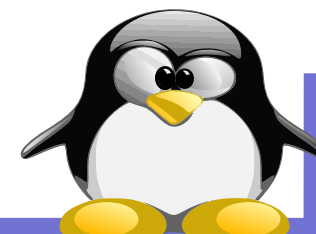


4. 实验要求



- 实验环境+操作系统移植 (必做)
- 图形用户接口 (必做)
- 设备驱动 (GPIO、PWM 等) (必做)
- 音频接口及数据采集处理
- 嵌入式系统应用软件移植
- 扩展实验、综合实验

除必做实验以外，必须再完成至少一个选做实验。

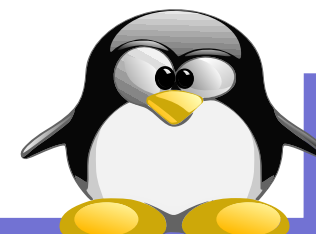


4. 实验要求



实验要求

1. 预习, 制定实验方案, 充分利用实验室时间
2. 实验操作规范
3. 实验记录完整详尽
4. 善于发现问题



4. 实验要求



实验一

使用教师预制的内核镜像、ramdisk 根文件系统启动 Linux:

```
U-Boot # tftp 0x82000000 zImage
```

```
U-Boot # tftp 0x88080000 ramdisk_img.gz
```

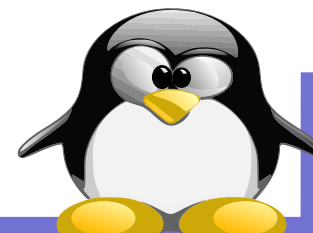
```
U-Boot # tftp 0x88000000 am335x-boneblack.dtb
```

设置环境 `setenv bootargs console=ttyO0 root=/dev/ram
initrd=0x88080000,8M`

引导启动:

```
U-Boot # bootz 0x82000000 - 0x88000000
```

进入 Linux 环境后，配置网络，挂载 NFS，编写任意一个 C 语言程序，交叉编译、运行。

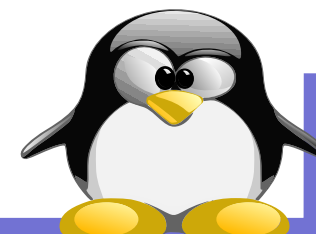


4. 实验要求



实验一基本要求

1. 实现串口监控
2. 加载内核、根文件系统镜像 (tftp)
3. 设置启动参数
4. 启动目标系统的 Linux, 完成初始化设置
5. 编写目标系统的应用程序 (交叉编译)、运行 (NFS)

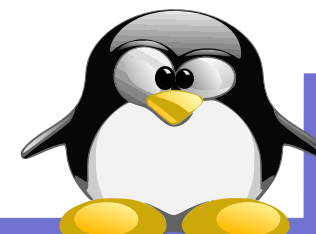


4. 实验要求



实验二内核裁剪、编译

下载 beagleboard Linux 内核，配置、编译，生成内核镜像 zImage 和设备树文件代替实验一中由教师制作的文件，正确引导系统启动



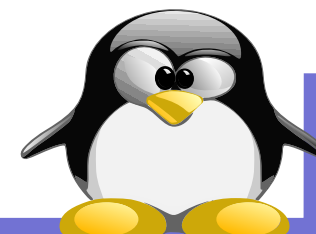


实验三制作根文件系统

1. 下载 BusyBox, 制作 ramdisk 根文件系统镜像, 代替实验一中由教师制作的 ramdisk_img.gz, 实现正常工作的系统
2. 学习其他根文件系统的制作

BusyBox 编译完成后, 如果是动态链接, 注意最后的提示, 将它需要的动态库复制到 ramdisk 中。

由于动态库较大, 建议首选使用 arm-linux-strip 瘦身, 而不是增大 ramdisk。



4. 实验要求

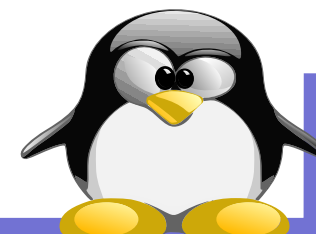


实验三基本要求

Linux 根文件系统有下面几种形式, 本实验只要求完成一种:

- ramdisk (难度系数 8)
- TF/eMMC (难度系数 8.5)
- NFS (难度系统 8.8)
- initial RAM FS (难度系数 9.5)

此后的实验, 除 bootloader 以外, 整个系统应以实验者自行移植的操作系统为基础。





实验四图形用户接口/帧缓冲

内核应支持 framebuffer。如果没出现 /dev/fb0 设备，应重新配置内核。

本实验有三种做法：

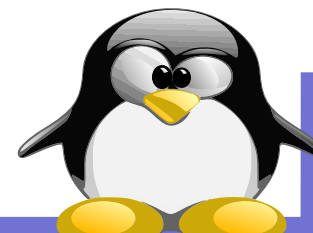
1. 实验室使用的触摸屏只在 3.8.13 内核版本支持。可使用该版本内核，外接触摸屏完成本实验。

2. 目标系统使用 VNC 服务器 x11vnc (由教师提供) 启动服务：

```
# x11vnc -rawfb /dev/fb0 -forever &
```

主机通过 vncviewer 登录目标系统的图形界面，借用主机显示器的一部分作为目标机的显示设备。

3. 使用 micro-HDMI 外接显示器。



4. 实验要求

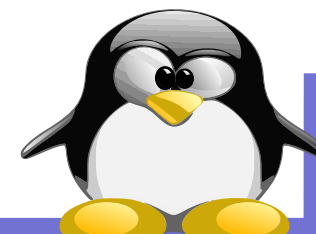


实验四实验要求

1. 完成基本画图功能 (点、线、框、圆等等)
2. 研究软件层次结构: 以画点函数为基础, 制作图形库 libgraphics.a 或 libgraphics.so, 编写基于图形库的应用程序。

扩展要求:

- 研究如何提高绘图效率 (Bresenham 算法)
- 设计动态图形, 改善显示效果
- 显示 bmp 图形文件
- 显示字符 (英文字符、汉字)



4. 实验要求



实验五音频接口

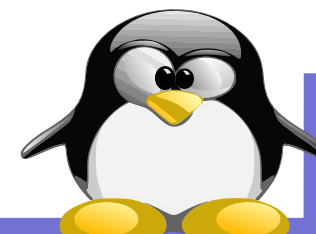
耳机、话筒通过外接 USB 声卡与 BB Black 连接。

将编译内核时生成的 USB 声卡驱动 `snd-usb-audio.ko` 加载进内核 (或编译进内核镜像)。注意 USB 声卡对应的设备文件 (`/dev/snd` 目录下, 或 `/dev/dsp1`), 通常 HDMI 对应 0 号声卡。

实验要求:

使用 OSS 或 ALSA 架构, 实现录音和放音功能。

ALSA 架构需要移植 `alsa-lib`。alsa 主页 <https://www.alsa-project.org/>



4. 实验要求

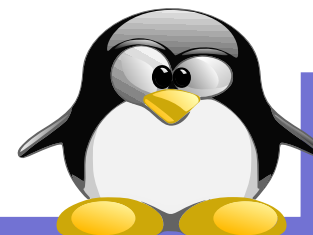


实验六设备驱动要求

BB Black 的扩展连接器 P8、P9 引出了大量的 GPIO 以及其他可编程功能模块。查阅 AM335X 数据手册，以及 BB Black 电路原理图，实现任意一个设备的基本控制功能 (GPIO 或 PWM)。

1. 编写设备驱动
2. 编写应用程序，实现与设备的交互控制
3. 研究设备驱动的结构，如何使设备驱动更加通用？

实验提示：GPIO 可以控制 LED 和开关按键，PWM 可以控制 LED 的亮度或双色 LED 的颜色。



4. 实验要求



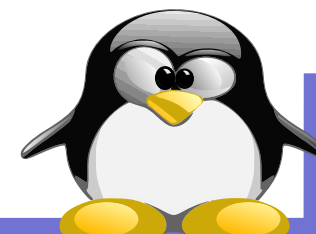
实验七 应用软件的移植

了解软件层次结构，学习解决依赖关系。

本实验不限制实验内容，鼓励学生在嵌入式平台实现自己的设想。

现有知识基础可完成下面的实验：

- MPlayer 移植 (编译 zlib、libmad 和 mplayer)
- 打造 Qt 桌面环境 (编译 Qt)
- 搭建无线路由器 (编译 libmnl、libnftnl 和 iptables, 需要 USB 无线网卡)
- 移植 GTK+ 库



4. 实验要求



实验报告要求

1. 原理性的内容简述
2. 实验过程详细, 有根据, 与原理结合, 避免流水帐式的记录
3. 对实验结果的分析讨论
4. 附上经实验指导教师签字的原始实验记录
5. 列出参考文献 (参考来源)
6. 避免成段照抄, 已有的结论标明出处可直引用
7. 格式规范, 篇幅适当 (A4 篇幅打印稿 12 页以内)

