

# 嵌入式系统实验环境的建立

## 嵌入式Linux开发流程

嵌入式 linux 开发, 根据应用需求的不同有不同的配置开发方法, 但是一般都要经过以下过程: 建立开发环境。操作系统一般使用 REDHAT-LINUX, 版本 7 到 9 都可以, 选择定制安装或全部安装, 通过网络下载相应的 GCC 交叉编译器进行安装 (比如 armv4l-unknown-linux-gcc), 或者安装产品厂家提供的交叉编译器。

配置开发主机。配置超级终端, 一般参数为波特率115200, 数据位8位, 停止位1, 无奇偶校验, 软硬件控制流设为无。在WINDOWS下的超级终端的配置也是这样。超级终端的作用是作为调试嵌入式开发板信息输出的监视器和键盘输入的工具; 配置网络, 主要是配置NFS网络文件系统, 需要关闭防火墙, 简化嵌入式网络调试环境设置过程。建立引导装载程序BOOTLOADER, 从网络上下载一些公开源代码的BOOTLOADER, 如U-BOOT、BLOB、VIVI、LILO、ARM-BOOT、RED-BOOT等, 根据自己具体芯片进行移植修改。有些芯片没有内置引导装载程序, 比如三星的ARM7、ARM9系列芯片, 这样就需要编写烧写开发板上flash的烧写程序, 网络上有免费下载的WINDOWS下通过JTAG并口简易仿真器烧写ARM外围flash芯片的程序。也有LINUX下公开源代码的J-FLASH程序。如果不能烧写自己的开发板, 就需要根据自己的具体电路进行源代码修改。这是让系统可以正常运行的第一步。如果你购买了厂家的仿真器当然比较容易烧写flash了, 但是其中的核心技术是无法了解的。这对于需要迅速开发应用的人来说可以极大地提高开发速度。下载别人已经移植好的LINUX操作系统, 如UCLINUX、ARM-LINUX、PPC-LINUX等, 如果有专门针对你所使用的CPU移植好的LINUX操作系统那是再好不过, 下载后再添加自己的特定硬件的驱动程序, 进行调试修改, 对于带MMU的CPU可以使用模块方式调试驱动, 对于UCLINUX这样的系统好像只能编译进内核进行调试。建立根文件系统, 从www.busybox.net下载使用BUSYBOX软件进行功能裁减, 产生一个最基本的根文件系统, 再根据自己的应用需要添加其他的程序。默认的启动脚本一般都不会符合应用的需要, 所以就要修改根文件系统中的启动脚本, 它的存放位置位于/etc目录下, 包括: /etc/init.d/rc.S、/etc/profile、/etc/.profile等, 自动挂装文件系统的配置文件/etc/fstab, 具体情况会随系统不同而不同。根文件系统在嵌入式系统中一般设为只读, 需要使用mkcramfs、genromfs等工具产生烧写映象文件。建立应用程序的flash磁盘分区, 一般使用JFFS2或YAFFS文件系统, 这需要在内核中提供这些文件系统的驱动, 有的系统使用一个线性flash (NOR型) 512K-32M, 有的系统使用非线性flash (NAND型) 8-512M, 有的两个同时使用, 需要根据应用规划flash的分区方案。开发应用程序, 可以下载到根文件系统中, 也可以放入YAFFS、JFFS2文件系统中, 有的应用程序不使用根文件系统, 而是直接将应用程序和内核设计在一起, 这有点类似于UCOS-II的方式。烧写内核、根文件系统、应用程序。发布产品。

### 1.4 嵌入式Linux开发环境的建立

嵌入式LINUX开发环境有几个方案: 1、基于PC机WINDOWS操作系统下的CYGWIN; 2、在WINDOWS下安装虚拟机后, 再在虚拟机中安装LINUX操作系统; 3、直接安装LINUX操作系统。

基于 WINDOWS 的环境要么有兼容性问题, 要么速度有影响, 所以我推荐大家使用纯 LINUX 操作系统开发环境。我们实际的开发环境为 REDHAT9, 它已经支持中文,

并且包含了绝大部分的开发工具，不用担心装了 LINUX 就不能使用 WINDOWS 的问题。一般的情况都是用户已经有了 WINDOWS 操作系统，再安装 LINUX，LINUX 会自动安装一个叫作 GRUB 的启动引导软件，可以选择引导多个操作系统。

### 三、开发工具软件的安装与配置

#### 1. REDHAT LINUX 9.0的安装

在一台 PC 上安装 RedHat LINUX9.0, 选择 Custom 定制安装, 在选择软件 Package 时最好将所有包都安装, 需要空间约 2.7G, 如果选择最后一项: everything, 即完全安装, 将安装 3 张光盘的全部软件, 需要磁盘空间大约 5G。因此建议提前为 REDHAT LINUX 的安装预留大约 5—15G 的空间, 具体视用户的硬盘空间大小来确定, 在安装完 Redhat 后还要

安装 Linux 的编译器和开发库以及 ARM-Linux 的所有源代码, 这些包安装后的总共需要空间大约为 800M。 2. 开发工具软件的安装 配置好网络和 smb 服务器, 在“开始”中点击“运行”输入 \\192.168.1.234 (注意 IP 为您虚拟机的 IP)。关于网络和 smb 的配置在后面会详细介绍。 输入 IP 后回车会出现如图 1.4.1 界面所示。



图 1.4.1 连接界面 输入用户名 bc ， 密码 123456 然后确定， 如图 1.4.2 所示。

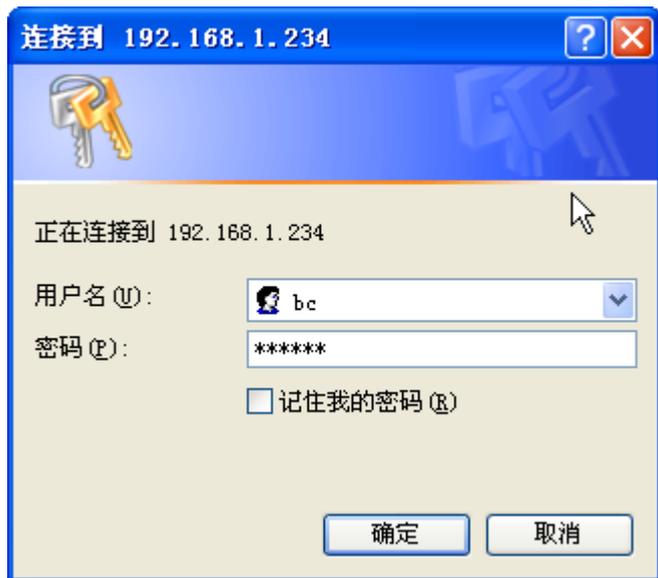


图 1.4.2 输入用户名和密码

回车后会出现就可以访问虚拟机的文件了，如图 1.4.3 所示。



图 1.4.3 samba 共享服务

将光盘中的 Linux 文件夹拷贝到所建的共享文件夹，进入文件夹内执行 ./install 指令安装环境,如下图 1.4.4 和 1.4.5 所示。

```
[root@BC bc]# cd 2410-dvp-Linux/
[root@BC 2410-dvp-Linux]# ls
2410经典版快速开始手册2.0.pdf  exp      kernel   实验补充说明
armv4l-tools                  gdb      readme.txt 经典2410demo
develop                       gui      rootfs    经典2410演示视频
doc                           ing      sh        经典开发平台硬件文档
editor                        install.sh uninstall.sh
```

图 1.4.4

```
[root@BC 2410-dvp-Linux]# ./install.sh
```

图 1.4.5

安装脚本程序将自动建立/arm2410cl 目录，并将所有开发软件包安装到/arm2410cl 目录下，同时自动配置编译环境，建立合适的符号连接。

注意：安装完成后看一下主编译器 armv4l-unknown-linux-gcc 是否在 /2410-dvp-Linux/armv4l-tools/opt/host/armv4l/bin，如果不是这个路径，请使用 vi 修改 /root/.bash\_profile 文件中 PATH 变量为 PATH=\$PATH:\$HOME/bin:/2410-dvp-Linux/armv4l-tools/opt/host/armv4l/bin，存盘后执行：source /root/.bash\_profile，则以后 armv4l-unknown-linux- 会自动搜索到，可以在终端上输入 arm-，然后按 tab 键，会自动显示 armv4l-unknown-linux-。

### 3. 开发环境配置

配置网络，包括配置 IP 地址、NFS 服务、防火墙。网络配置主要是要安装好以太网卡，对于一般常见的 RTL8139 网卡，REDHAT9.0 可以自动识别并自动安装好，完全不要用户参与，因此建议使用该网卡。然后配置宿主机 IP 为 192.168.1.28（IP 地址完全跟根据需求配置，无需跟实验指导书中配置相同）。如果是在有多台计算机使用的局域网环境使用此开发设备，IP 地址可以根据具体情况设置。如图 1.4.6 所示。

图 1.4.6

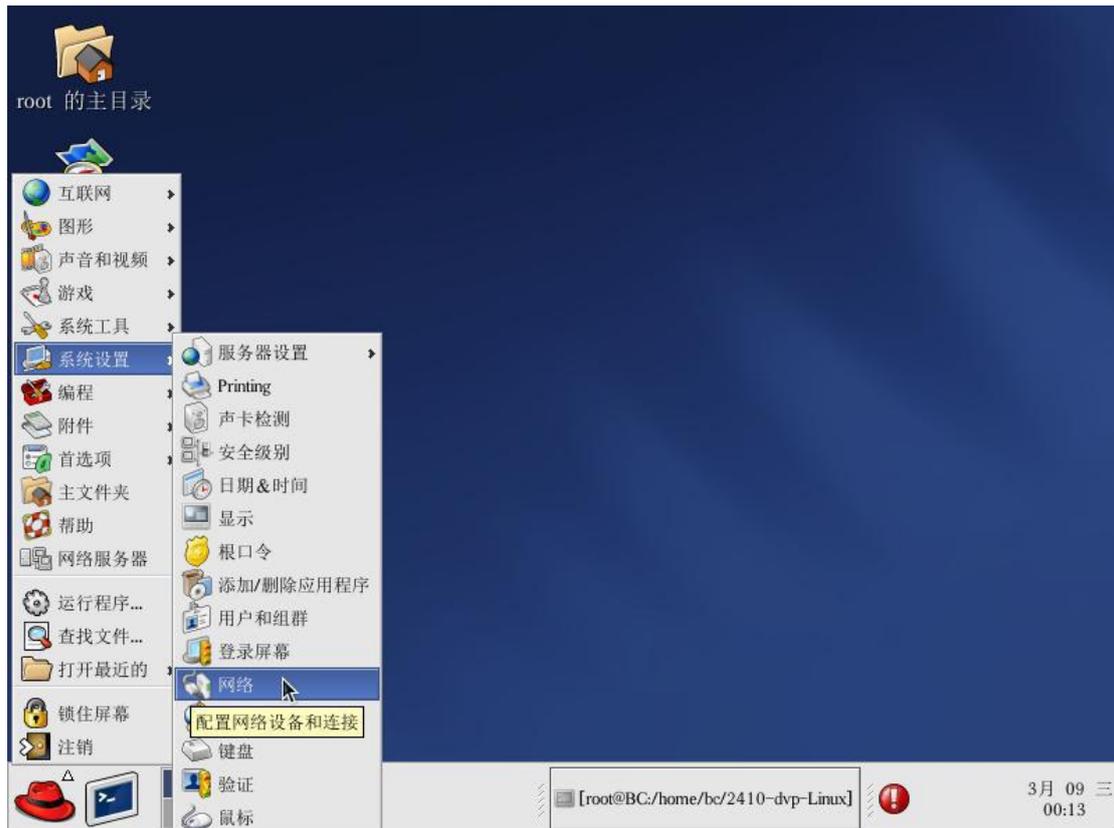


图 1.4.6 网络设置

双击设备 eth0 的蓝色区域，进入以太网设置界面，如图 1.4.7 和图 1.4.8。其中的 IP 根据自己的需要设置，但要保证是跟 PC 主机处于同一网段。

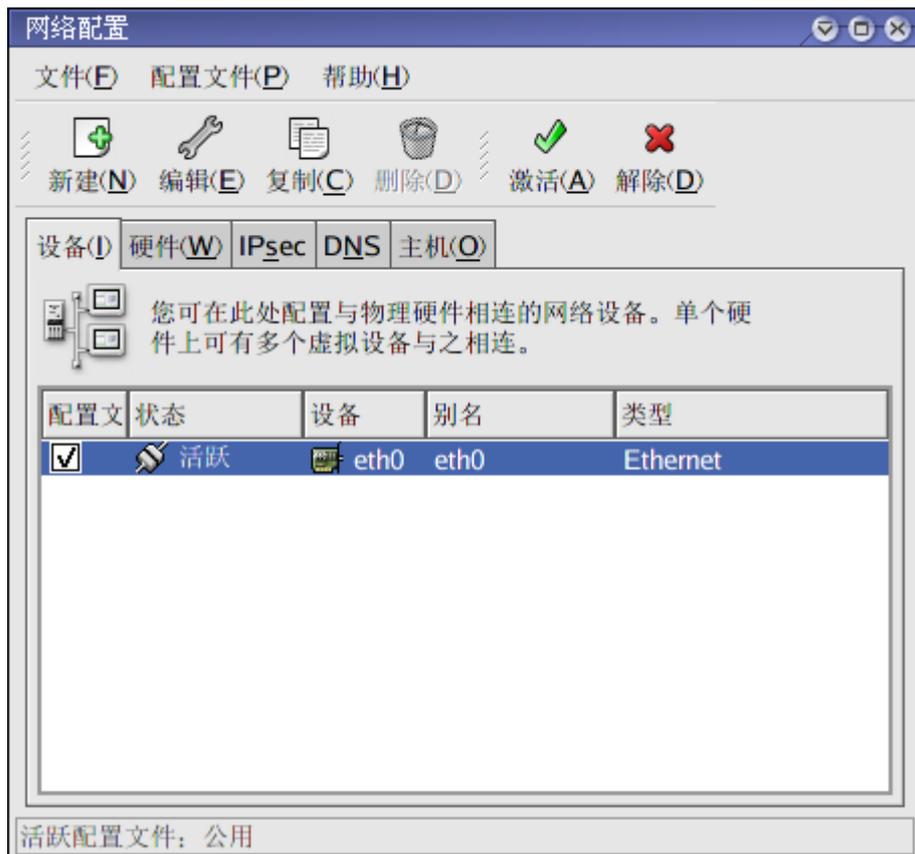


图 1.4.7 网络配置

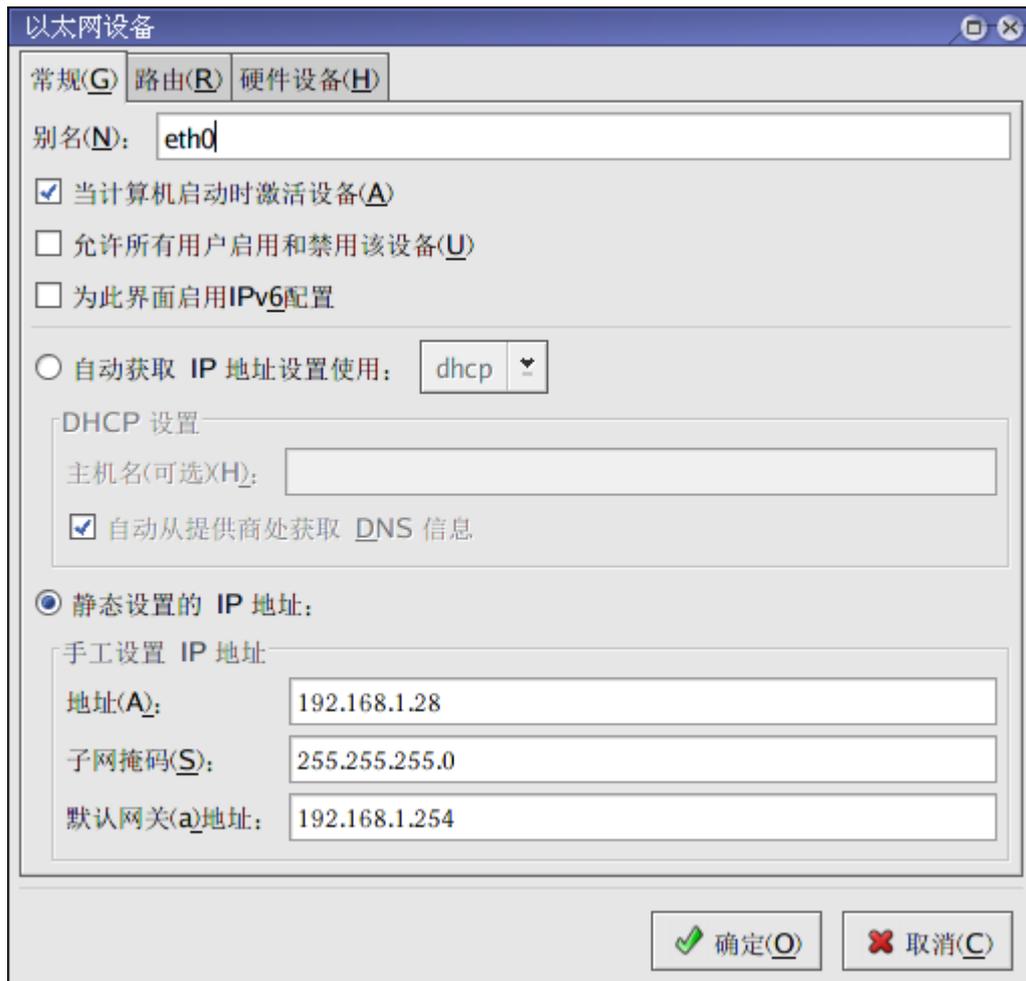


图 1.4.8 以太网设备

对于 REDHAT9.0, 它默认的是打开了防火墙, 因此对于外来的 IP 访问它全部拒绝, 这样其它网络设备根本无法访问它, 即无法用 NFS mount 它, 许多网络功能都将无法使用。因此网络安装完毕后, 应立即关闭防火墙。操作如下: 点击红帽子开始菜单, 选择安全级别设置, 选中无防火墙, 如图 1.4.9。

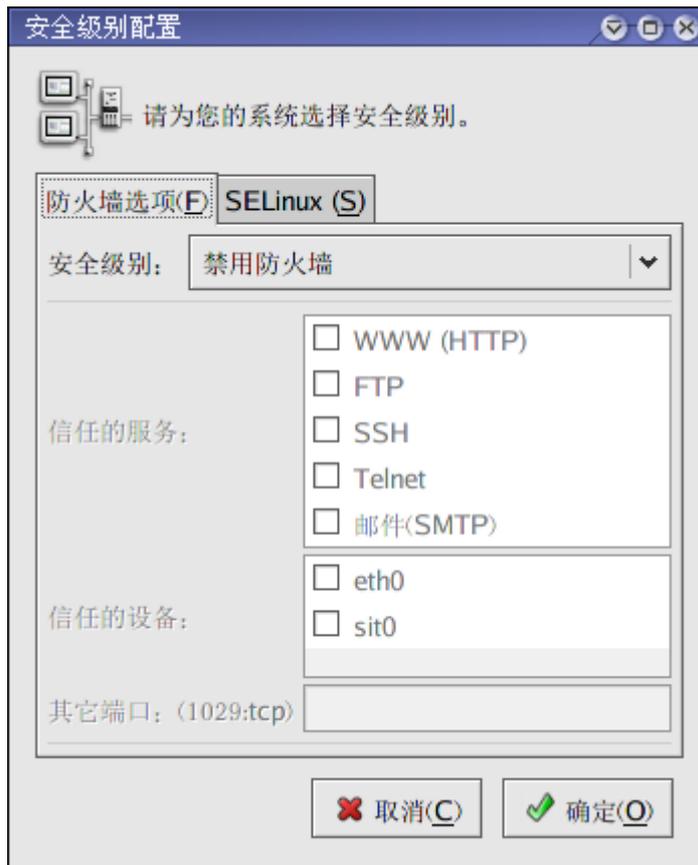


图 1.4.9 安全级别设置

在系统设置菜单中选择服务器设置菜单，再选中服务菜单，将 iptables 服务的勾去掉,并确保 nfs 选项选中。

#### **Sabam** 服务器配置

如图选择 Samba，打开。如图 1.4.10。

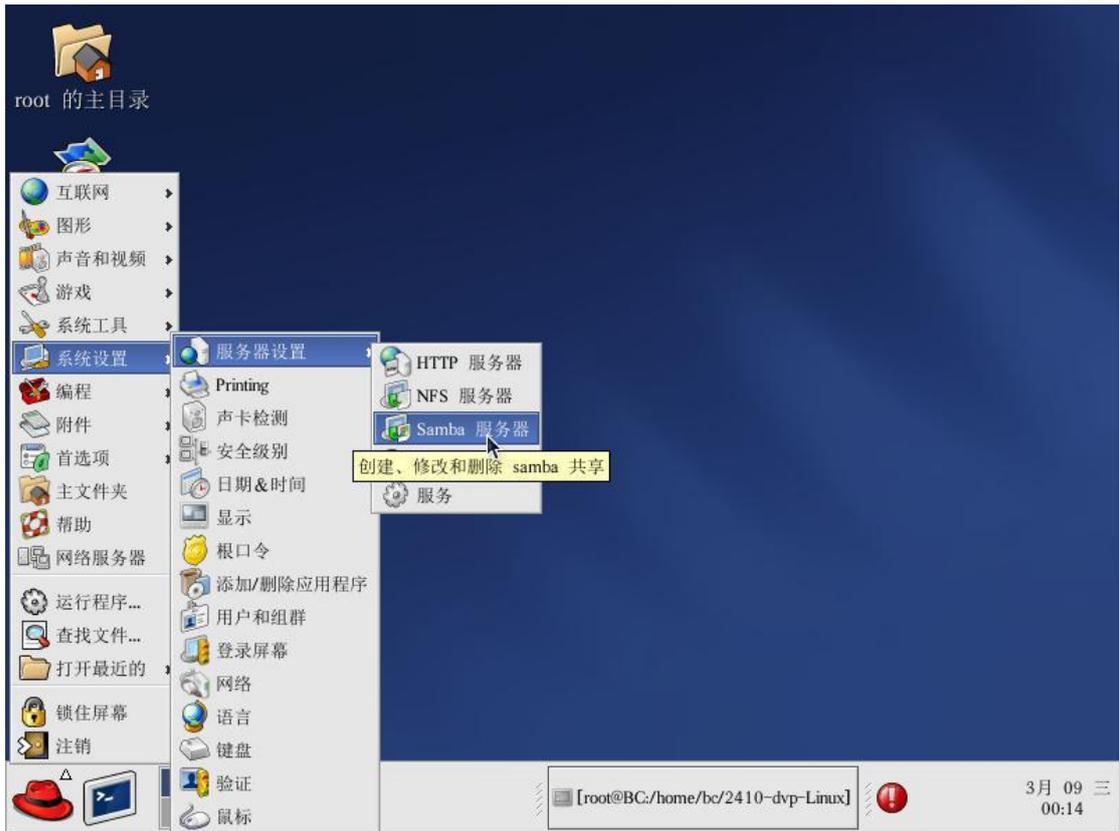


图 1.4.10 samba 服务器配置

点击添加，按照下图配置，注意点击“浏览”选择要共享的目录。“描述”要填写与 PC 主机同一网段的 IP。如图 1.4.11 和 1.4.12 所示。最后点击“确定”按钮可看到如图 1.4.13 所示内容。

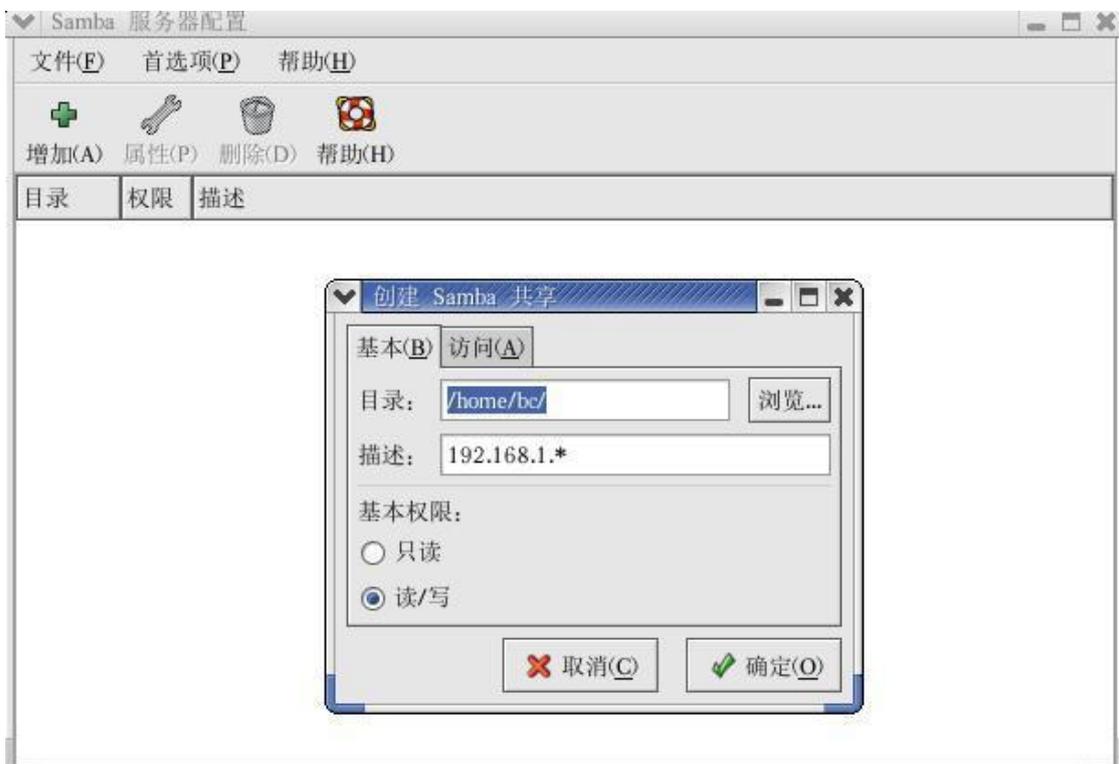


图 1.4.11 创建 samba 共享

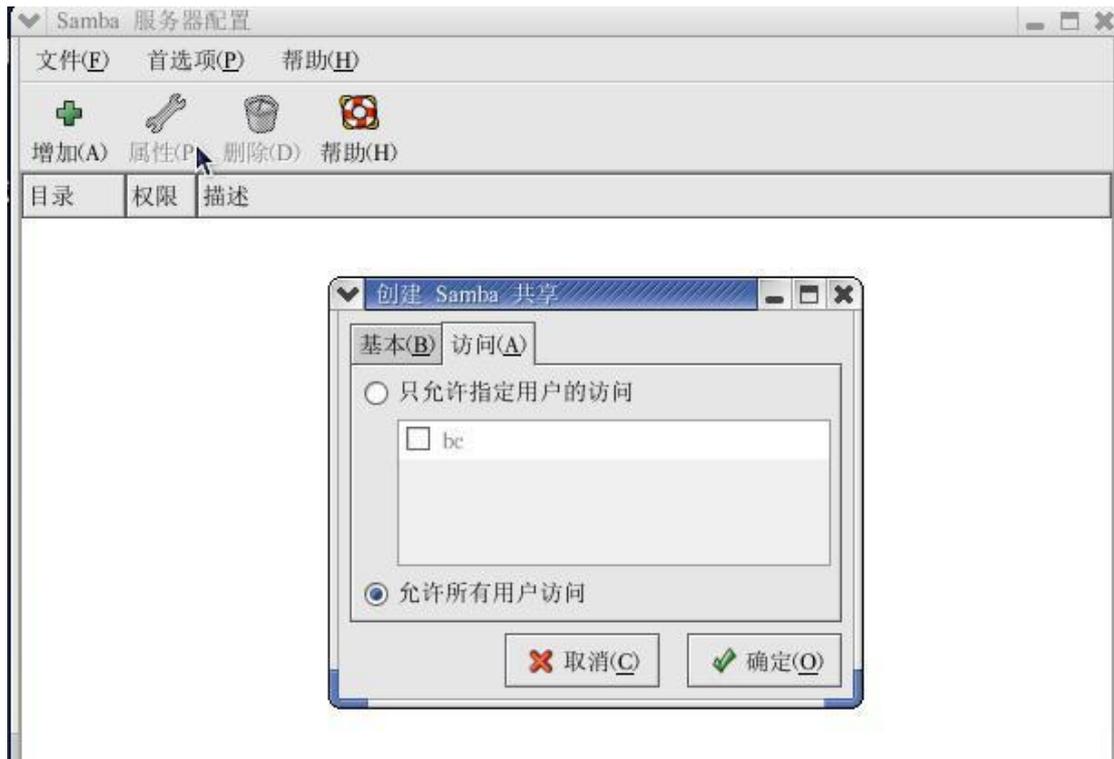


图 1.4.12 访问权限的设置

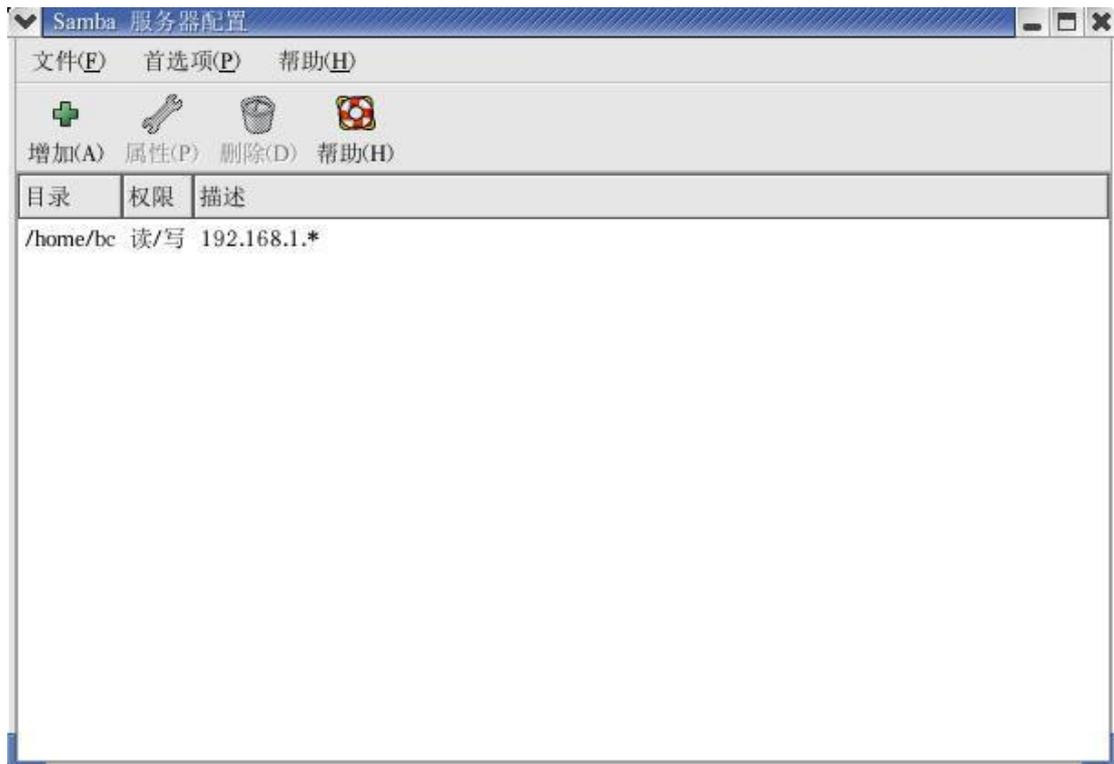


图 1.4.13 samba 服务器配置

**nfs** 服务器配置 打开 **nfs** 服务，选项位于刚才 **samba** 选项下，如图 1.4.14。

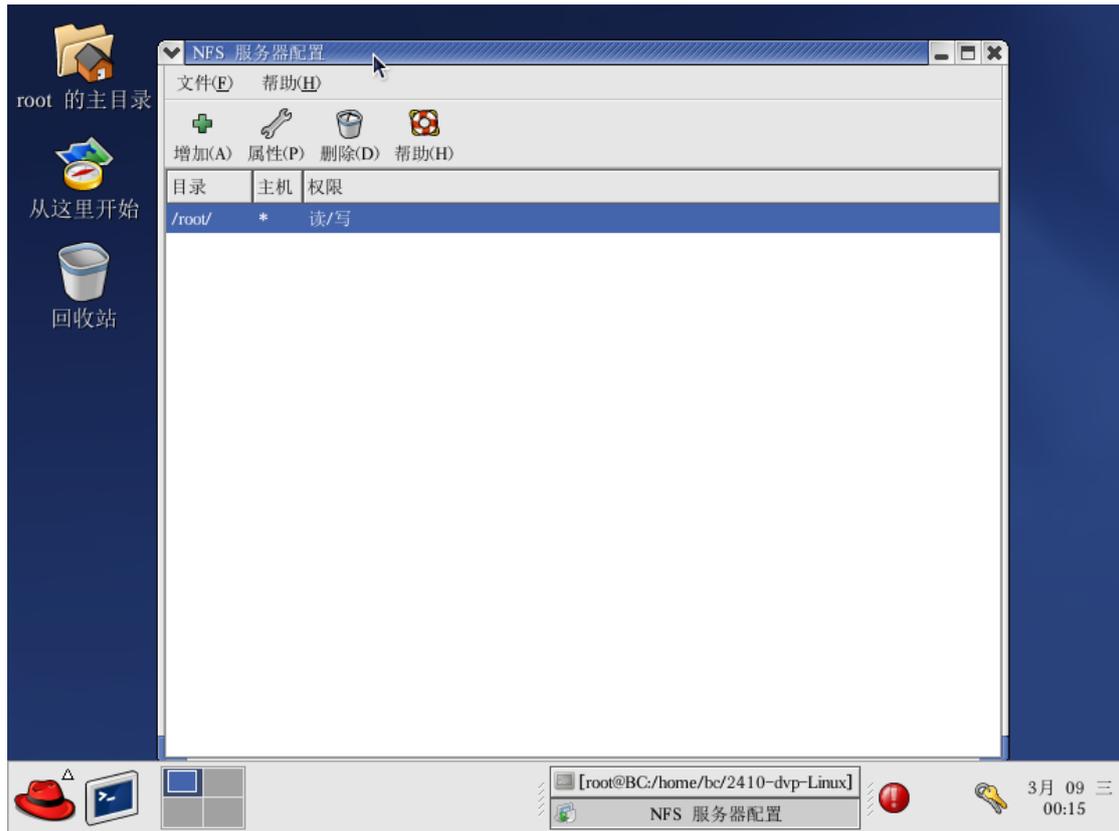


图 1.4.14 nfs 服务器配置

点击“浏览”选择共享的目录，然后按照如下配置，如图 1.4.15、1.4.16 和 1.4.17。

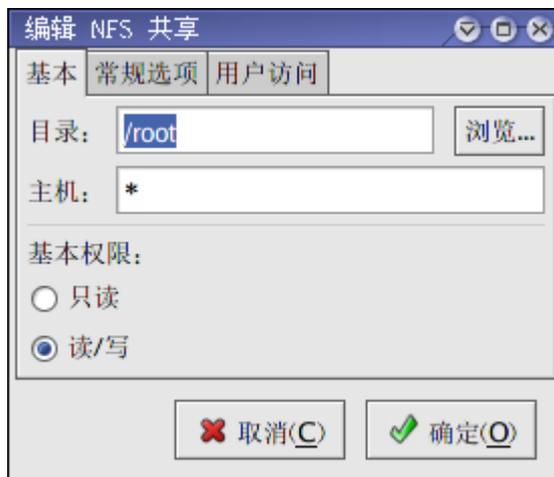


图 1.4.15 编辑 nfs 共享基本设置



图 1.4.16 编辑 nfs 共享常规设置



图 1.4.17 编辑 nfs 共享用户访问设置

配置完网络、smb 服务器和 nfs 服务器后需要对各项服务重新启动指令如下：

网络：`/etc/init.d/network restart`

smb 服务：`/etc/init.d/smb restart`

nfs 服务：`/etc/init.d/nfs restart`

4. 配置超级终端 用串口线将 PC 机与 ARM 开发板连接好后，将 UP-TECH S2410、P270 DVP 开发板开机，然后点击 PC 机上的开始“菜单”然后找到“附件”中“通讯”选项中的“超级终端”，如下图 1.4.18 所示。

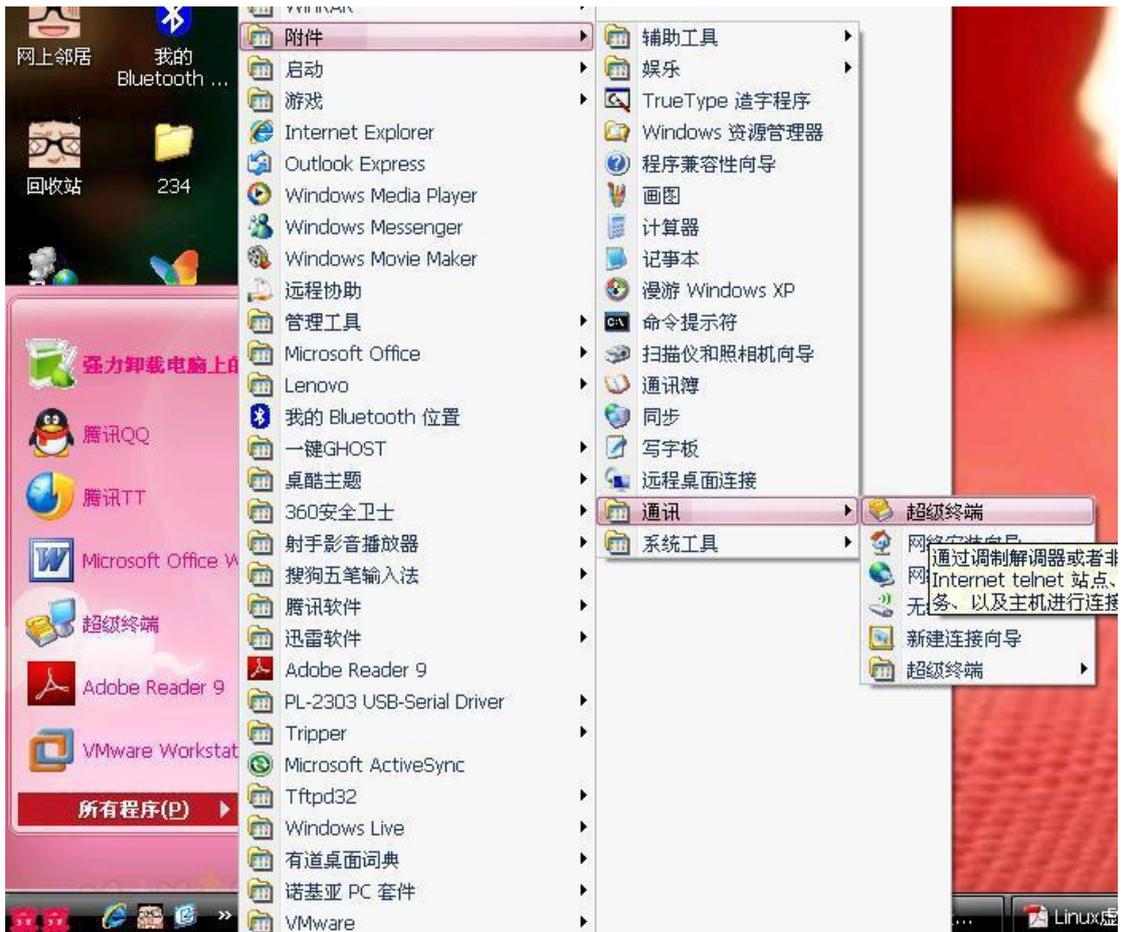


图 1.4.18 超级终端位置

然后在超级终端里进行配置，配置过程如下图 1.4.19、1.4.20、1.4.21 和 1.4.22 所示。在“名称”中输入要建立的超级终端的名字。



图 1.4.19 超级终端连接设置

在“连接时使用”一栏选择可用的串口号，（这里根据自己的实际情况进行选择）。



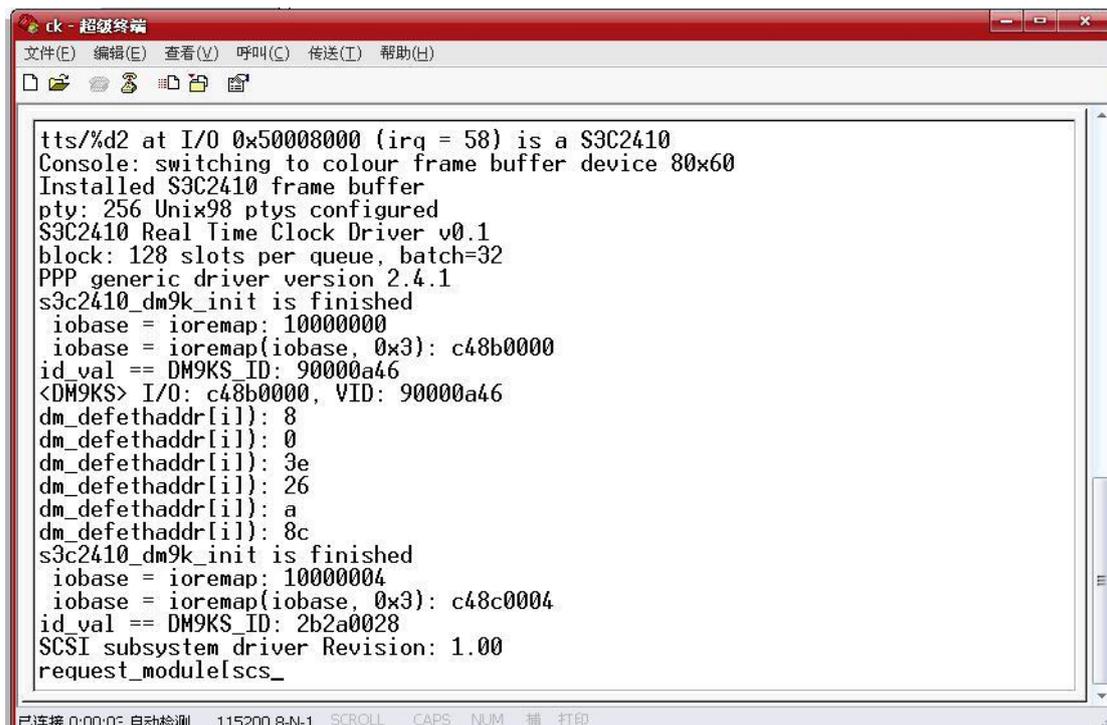
图 1.4.20 超级终端端口设置

在“每秒位数”中选择“115200”，“数据流控制”选择“无”，然后点击“确定”按钮，设置完毕。



图 1.4.21 端口的属性设置

这时重新启动开发板，可看到终端输出如下图所示信息。



```
ck - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

tts/%d2 at I/O 0x50008000 (irq = 58) is a S3C2410
Console: switching to colour frame buffer device 80x60
Installed S3C2410 frame buffer
pty: 256 Unix98 ptys configured
S3C2410 Real Time Clock Driver v0.1
block: 128 slots per queue, batch=32
PPP generic driver version 2.4.1
s3c2410_dm9k_init is finished
 iobase = ioremap: 10000000
 iobase = ioremap(iobase, 0x3): c48b0000
id_val == DM9KS_ID: 90000a46
<DM9KS> I/O: c48b0000, VID: 90000a46
dm_defethaddr[il]: 8
dm_defethaddr[il]: 0
dm_defethaddr[il]: 3e
dm_defethaddr[il]: 26
dm_defethaddr[il]: a
dm_defethaddr[il]: 8c
s3c2410_dm9k_init is finished
 iobase = ioremap: 10000004
 iobase = ioremap(iobase, 0x3): c48c0004
id_val == DM9KS_ID: 2b2a0028
SCSI subsystem driver Revision: 1.00
request_module[scs_

已连接 0:00:03 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印
```

图 1.4.22 超级终端的显示

在安装完成 LINUX 后，就是对 LINUX 的网络环境的配置，最主要是配置 NFS 服务器。