

S3C6400/6410 移植 Android 内核

主要过程:

- ✓ 安装 linux 环境
- ✓ 安装编译工具
- ✓ 下载 Linux kernel
- ✓ 安装 Android SDK
- ✓ 获得 root file system
- ✓ 修改 Linux kernel 源码
- ✓ 配置 Linux kernel
- ✓ 修改 root file system
- ✓ 编译 Linux kernel
- ✓ 下载 kernel Image

1. 安装 linux 环境

安装 Ubuntu Linux 系统，从网站下载操作系统安装光盘映像，地址：

<http://mirror.lupaworld.com/ubuntu/releases/8.04/>

下载 ubuntu-8.04.2-desktop-i386.iso, 刻录成光盘安装，安装可以在 windows 系统下进行，选取有 15G 空间的硬盘安装，输入用户密码开始安装



这一步结束退出光盘重启进入 Ubuntu 系统，完成余下系统安装。

Linux 安装完成，进入 Ubuntu 系统，确保电脑连接 Internet，安装一些必要软件。

打开终端输入命令框，进行下面的操作：

```
$ sudo apt-get install ssh
```

```
$ sudo apt-get install flex bison gperf libstdc++-dev libstdc++0-dev libxgtk2.6-dev
```

build-essential zip curl

```
$ sudo apt-get install valgrind
$ sudo apt-get install sun-java6-jdk
$ sudo apt-get install libncurses5-dev
```

所有软件系统会自动从网络下载安装，完成后可以开始 Android 的移植。

2. 安装编译工具

下载:

```
$ wget
http://www.codesourcery.com/public/gnu_toolchain/arm-none-linux-gnueabi/arm-2008q1-126-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2
```

安装:

```
$ tar -xjvf arm-2008q1-126-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2
```

3. 下载 Linux kernel

从下面地址获得:

<http://code.google.com/p/android/downloads/list?can=1&q=&colspec=Filename+Summary+Uploaded+Size+DownloadCount>

Kernel: linux-2.6.23-android-m5-rc14.tar.gz

解压文件

```
$ tar -xf linux-2.6.23-android-m5-rc14.tar.gz
```

4. 安装 Android SDK

从网上下载 linux 版本的 Android SDK，如下

http://dl.google.com/android/android-sdk_m5-rc15_linux-x86.zip

下载后解压，获得 android-sdk_m5-rc15_linux-x86 文件

5. 获得 root file system

(1) 下载 busybox 工具从下面地址

<http://benno.id.au/blog/2007/11/14/android-busybox> (linux 环境中下载)

(2) 运行 emulator 获取 root file system

```
$ cd <目录>/ android-sdk_m5-rc15_linux-x86/tools
$ ./emulator&
```

等待 emulator 启动，看到出现 Android 系统画面，进入菜单 ALL/Dev Tools/Development Settings 下，将 Wait for debugger, Show running processes, Show screen updates 这三项打勾，回到命令输入终端。

```
$ ./adb push <local-path>/busybox /data
$ ./adb shell
```

```
# /data/busybox tar -czf system.tar.gz /system
# /data/busybox tar -czf data.tar.gz /data
# /data/busybox tar -czf etc.tar.gz /etc
# /data/busybox tar -czf sbin.tar.gz /sbin
# exit
$ ./adb pull /system.tar.gz ./
$ ./adb pull /data.tar.gz ./
$ ./adb pull /etc.tar.gz ./
$ ./adb pull /sbin.tar.gz ./
$ ./adb pull /init ./
```

通过 busybox 工具获得了 emulator 的文件并打包，下面建立 root 文件目录

```
$ mkdir initial_ram_disk
$ cd initial_ram_disk
$ mkdir cache
$ mkdir d
$ mkdir dev
$ mkdir proc
$ mkdir root
$ mkdir sdcard
$ mkdir sys
$ mkdir tmp
$ mkdir var
```

```
$ tar -xf <local-path>/system.tar.gz ./
$ tar -xf <local-path>/data.tar.gz ./
$ tar -xf <local-path>/etc.tar.gz ./
$ tar -xf <local-path>/sbin.tar.gz ./
$ cp <local-path>/init ./
```

包含 Android root 文件系统的 initial_ram_disk 做好。

6. 修改 Linux kernel 源码

下载的 kernel 文件是不支持 smdk6400 开发板的，必须加入 smdk6400 的配置文件到 kernel 中，同时修改 Kconfig 和 Makefile 文件。

修改如下：

1> 选择 /arch/arm/mach-s3c2410/Makefile.boot 改为下面内容，

```
zreladdr-$(CONFIG_PLAT_S3C24XX)      := 0x30008000
params_phys-$(CONFIG_PLAT_S3C24XX)   := 0x30000100
zreladdr-$(CONFIG_PLAT_S3C64XX)      := 0x50008000
params_phys-$(CONFIG_PLAT_S3C64XX)   := 0x50000100
```

2> 从成熟的 smdk6400 平台文件中拷贝下面文件到 kernel 目录

```
/arch/arm/mach-s3c6400  
/drivers/serial/s3c6400.c  
/drivers/input/touchscreen/s3c-ts.c  
/drivers/usb/gadget
```

3> 拷贝/drivers/android/ashmem.c 文件到相应目录下从另外的 android kernel (The Android source code for 2.6.23 不包含此文件)

4> 修改相应的 Kconfig 和 Makefile 文件, 使最终能编译到加载的 smdk6400 文件。

7. 配置 Linux kernel

修改 kernel 后进行配置, 用到第 2 步的编译工具

```
$ cd <kernel-src>
```

```
$ make CROSS_COMPILE=<path-to-compiler>/bin/arm-none-linux-gnueabi- ARCH=arm  
menuconfig
```

这时进入 kernel 配置画面, 注意配置 Inittarfs source file 选项指定到 initial_ram_disk 文件。

8. 修改 root file system

配置 Touchscreen 使之能正常工作, initial_ram_disk 文件下修改/etc/init.rc, 加入下面内容

```
TSLIB_CONSOLEDEVICE none  
TSLIB_FBDEVICE /dev/graphics/fb0  
TSLIB_TSDEVICE /dev/input/ts0  
TSLIB_CALIBFILE /etc/pointercal  
TSLIB_CONFFILE /etc/ts.conf  
TSLIB_PLUGINDIR /lib/ts
```

9. 编译 Linux kernel

```
$ make CROSS_COMPILE=<path-to-compiler>/bin/arm-none-linux-gnueabi- ARCH=arm  
zImage
```

编译成功, 在 arch/arm/boot 下面是生成的 zImage 文件。

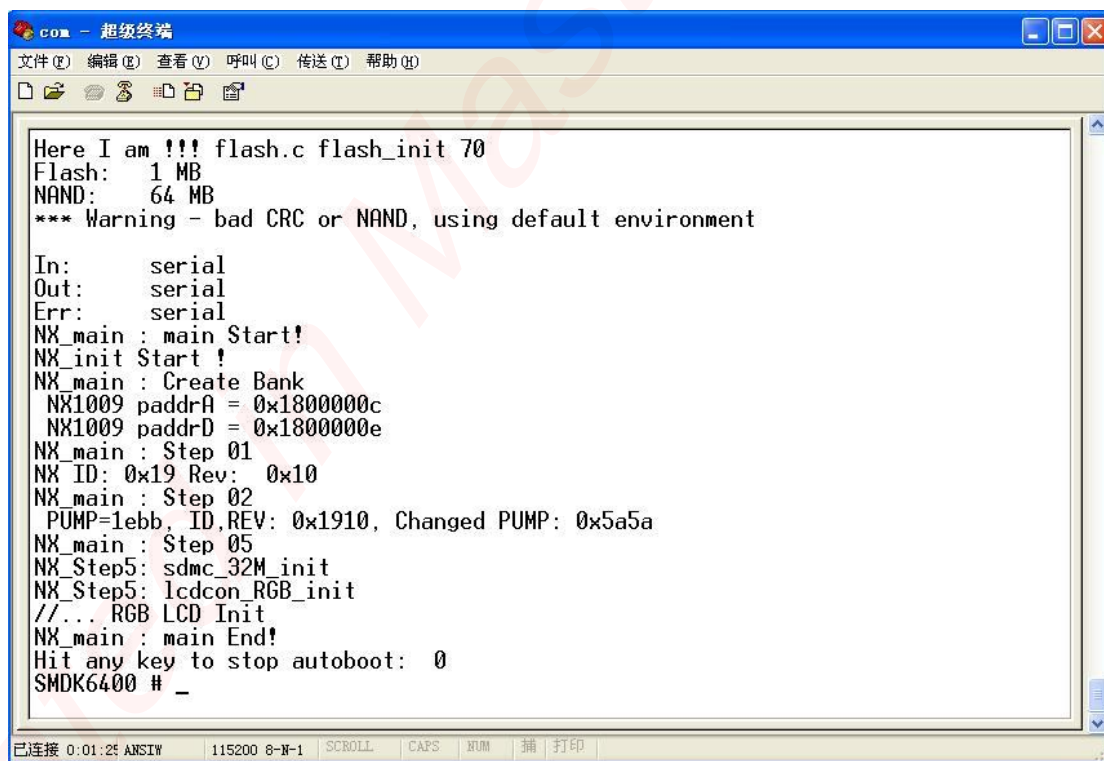
10. 下载 kernel image

进入 Windows 系统, 下载过程在 Windows 系统中完成。

① 打开: 程序\附件\通讯\超级终端, 配置 COM 口, 115200, 8N1

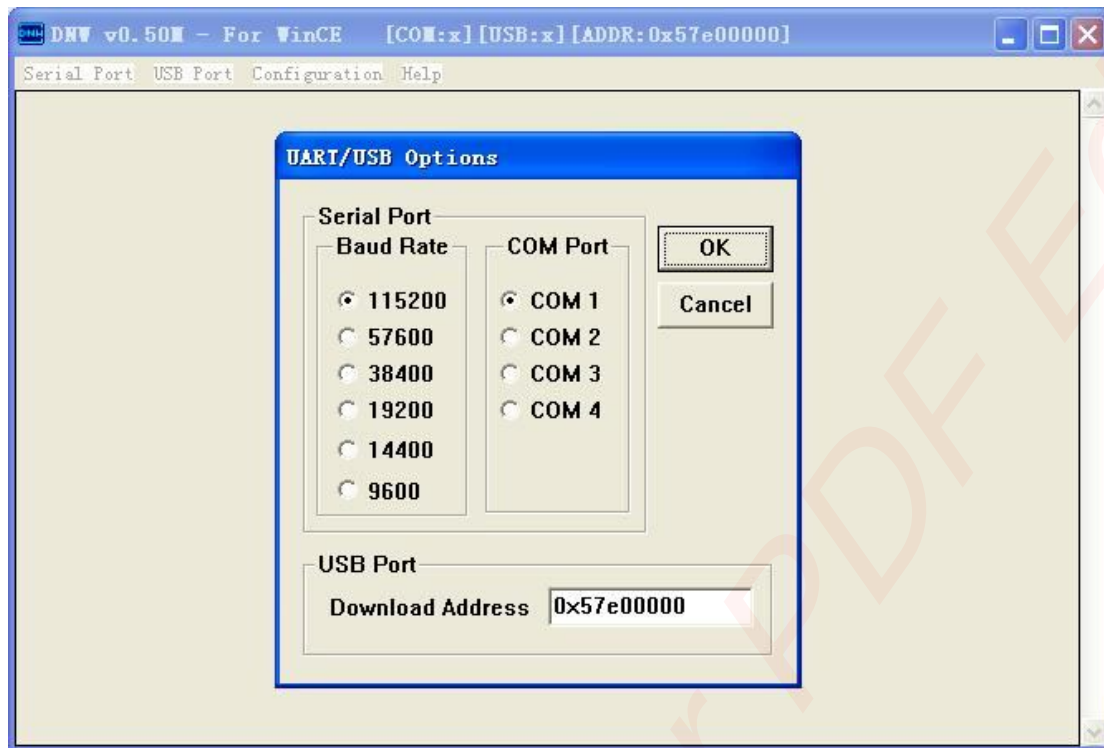


- ② 连接 smdk6400 开发板和电脑通过串口线，打开开发板电源，可以在超级终端中看到输出信息，bootloader 启动，按空格键进入命令输入状态



- ③ 板上连接 USB 线到电脑，安装 USB 驱动程序

- ④ 运行 dnw.exe 程序，配置 Configuration/Options，地址 0x57e00000，确定



⑤ 在超级终端中输入相应命令，开始下载

```
$ run bootup      (下载 bootloader)
$ run kernelup    (下载 kernel Image)
$ run rootup      (下载 root file system)
```

⑥ 这里我们输入

```
$ run kernelup
```

等待 dnw 发送文件，dnw 显示[COM1,115200bps][USB:OK][ADDR:0x57e00000]说明连接成功，选择 USB Port->Transmit，选取编译成功的 zImage 文件，开始下载。

下载完成，重启开发板，等待，LCD 上出现一些 Android 字符，一段时间白屏，出现一个左右滚动的红色滚动条，进入主界面，Android 移植成功。

嵌入式资源免费下载

总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)
11. [USB30 电路保护](#)
12. [USB30 协议分析与框架设计](#)
13. [USB 30 中的 CRC 校验原理及实现](#)
14. [基于 CPLD 的 UART 设计](#)
15. [IPMI 在 VPX 系统中的应用与设计](#)
16. [基于 CPCI 总线的 PMC 载板设计](#)
17. [基于 VPX 总线的工件台运动控制系统研究与开发](#)

VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)

12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)
14. [基于 MPC8270 的 VxWorks BSP 的移植](#)
15. [Bootrom 功能改进经验谈](#)
16. [基于 VxWorks 嵌入式系统的中文平台研究与实现](#)

Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)
7. [Linux 串口编程实例](#)
8. [基于 Yocto Project 的嵌入式应用设计](#)
9. [Android 应用的反编译](#)
10. [基于 Android 行为的加密应用系统研究](#)

Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)
9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)
16. [基于 WinCE 的 BootLoader 研究](#)

PowerPC:

1. [Freescale MPC8536 开发板原理图](#)
2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)
4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)
7. [用 PowerPC860 实现 FPGA 配置](#)

ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)
7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)
10. [ARM 经典 300 问](#)
11. [基于 S5PV210 的频谱监测设备嵌入式系统设计与实现](#)

Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)

RT Embedded <http://www.kontronn.com>

6. [DM642 高速图像采集系统的电磁干扰设计](#)
7. [使用 COMExpress Nano 工控板实现 IP 调度设备](#)