

DEV3530 用户手册

公司: *EMA*

發行版本: **v2.0**

发布时间: **03/24/2010**



广州英码信息科技有限公司（EMA）成立于 2006 年 12 月，专注于研发、生产、销售嵌入式产品和解决方案，包括嵌入式系统模块（SOMs）、单板计算机（SBC）、工控板（IPC），以及支持用户进行快速二次开发的配套工具与软件包。公司作为德州仪器（TI）、飞思卡尔（Freescale）等芯片公司的合作伙伴，按 OEM/ODM 方式为用户提供全面解决方案，在售前售后技术支持、用户培训和系统维护等各个环节为用户提供完善的服务，使客户能以低成本、低风险的方式运作，并加快产品上市时间，实现双赢。

公司秉承“客户之上、服务至上、为客户利益而努力创新”的经营理念，对客户承诺：

➤ **更低的研究成本**

针对具有新技术、新产品研发需求的合作伙伴，EMA 将致力于新技术、新产品解决方案的研发、服务和技术升级，让合作伙伴更专注于其自身优势领域，从而帮助合作伙伴在节省大量研发成本、降低研发风险的前提下，实现产业升级和产品创新。

➤ **更高的品质保证**

EMA 的生产部门和完善的生产控制体系为合作伙伴提供专业级的、高标准的产品品质保证。



➤ **更优质的服务**

EMA 技术服务部门为客户提供全方位的技术支持与产品服务，包括产品售前售后支持、产品技术培训、24 小时热线服务等。

➤ **更快的市场响应能力**

EMA 针对新的市场需求为客户提供新产品决策支持，并帮助客户进行快速的技术和产品实现，以更快的速度引领市场。

联系方式：

地址：广州市新港西路 152 号广东轻院工业实训中心 B704

电话：020-61230220

传真：020-61230221

手机：18924191634

电邮：sales@ema-tech.com(销售)

support@ema-tech.com (技术支持)

网址：<http://www.ema-tech.com>



目录

1. DEV3530 单板机介绍	7
1.1. DEV3530 简介	7
1.2. DEV3530 的型号	8
2. DEV3530 的硬件介绍	9
2.1. DEV3530 的硬件规格	9
2.1.1. 框图	9
2.1.2. DEV3530 的硬件参数	10
2.1.3. DEV3530 工作环境	11
2.1.4. DEV3530 机构资料	11
2.2. DEV3530 的接口说明	12
2.2.1. 接口一览表	12
2.2.2. 按键/开关一览表	12
2.2.3. LED 一览表	12
2.3. 接口详述	13
2.4. DEV3530 接口引脚定义	17
3. DEV3530 的基本使用	19
3.1. 以 DEV3530 为核心的计算机系统	19
3.2. DEV3530 连接设置	20
3.2.1. 外部接口连接	20
3.2.2. DEV3530 系统上电步骤	22
3.2.3. DEV3530 系统关闭步骤	23
4. DEV3530 的 LINUX 系统使用与设置	24
4.1. 系统启动方法和过程	24
4.1.1. 系统启动过程:	24
4.1.2. SD 卡启动过程	25
4.1.3. Nandflash 启动过程	26
4.2. 显示设置	26
4.2.1. DVI 接口显示器显示:	27



4.2.2.	4.3 寸触摸屏显示:	27
4.2.3.	7 寸触摸屏显示:	27
4.3.	DEMO 软件演示	28
4.3.1.	3D Demo 演示	28
4.3.2.	Quake3 游戏演示	29
4.3.3.	DVSDK 演示	30
5.	构建嵌入式 LINUX 的软件开发环境	32
5.1.	LINUX 软件开发环境概述	32
5.2.	构建软件开发环	33
5.2.1.	VMware-workstation-6.5.0 的安装	33
5.2.2.	Ubuntu 的安装	36
5.2.3.	虚拟机跟主机的文件共享设置	40
5.2.4.	构建交叉编译器	42
5.3.	串口终端软件的使用	43
5.4.	挂载网络文件系统 NFS	44
6.	构建 DEV3530 的软件开发环境	46
6.1.	系统编译	46
6.1.1.	一级启动代码 x-loader 编译	46
6.1.2.	二级启动代码 u-boot 编译	47
6.1.3.	内核编译	48
6.1.4.	制作文件系统镜像	49
6.2.	LINUX 系统镜像更新	50
6.2.1.	制作 SD 系统启动卡	50
6.2.2.	SD 卡的系统镜像更新	52
6.2.3.	NAND Flash 系统映像更新	52
6.3.	LINUX 系统操作	54
6.3.1.	设置开机自动运行程序	54
6.3.2.	屏蔽显示器的登陆界面	55
6.3.3.	下载安装软件	55
6.3.4.	ALSA 声音设置	56
6.3.5.	Linux 应用程序开发一般流程	57
6.4.	DEMO 系统运行效果	58



7. 附录	61
7.1. 原理图及核心板的接口功能	61
7.2. 相关服务	61
7.3. 相关链接	61

广州英码信息科技有限公司



第一章

1. DEV3530 单板机介绍



1.1. DEV3530 简介

DEV3530 是广州英码信息科技有限公司（EMA）自主研发的一款基于 TI Omap35x 处理器的低功耗、高性能单板机，在设计上兼容 EPIC 标准，并使用可扩展的结构。

DEV3530 帮助用户充分利用 Omap35x 处理器的优点，包括超标量的 ARM Cortex-A8 RISC core、C64x+ DSP、PowerVR SGX 2D/3D 图形加速引擎。

目前基于 DEV3530 实现的产品有移动互联网设备 MID、全球定位系统（GPS）、2D/3D 游戏机、2D/3D 游戏机平台、医疗仪器、图像采集设备、人机界面等。DEV3530 支持分辨率可达 1920*1080；支持多种标准的 30fps 的 D1(720*480)视频编解码、720P 的高清视频播放；支持 Linux、Android、Wince 等多种嵌入式操作系统。



1.2. DEV3530 的型号

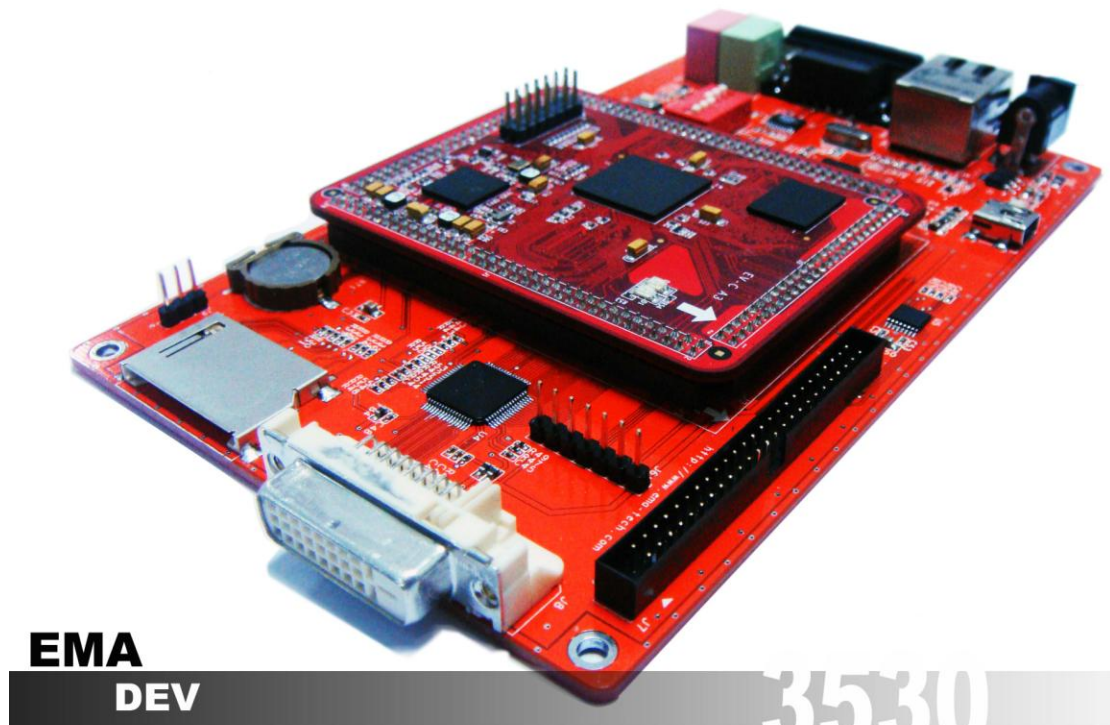


图 1.1 DEV3530

产品型号	产品规格
SBC3530-256M	TI OMAP3530 CortexA8+C642+ DSP+PowerVR /RAM 256MB/ Flash 256M/ 10/100 Mbps LAN/ 4xUSB Host 2.0/g /1xUSB OTG/ 3x UART/1x VGA/AV video output
SBC3530-256MV	TI OMAP3530 CortexA8+C642+ DSP+PowerVR /RAM 256MB/ Flash 256M/ 10/100 Mbps LAN/ 4xUSB Host 2.0/g /1xUSB OTG/ 3x UART/1x VGA/AV video output/ 4x video input/...
DEV3530-128M	TI OMAP3530 CortexA8+C642+ DSP+PowerVR /RAM 128MB/ Flash 128M/ 10/100 Mbps LAN/ /1xUSB OTG/ 2x UART/1x DVI/AV video output/



第二章

2. DEV3530 的硬件介绍



2.1. DEV3530 的硬件规格

2.1.1. 框图

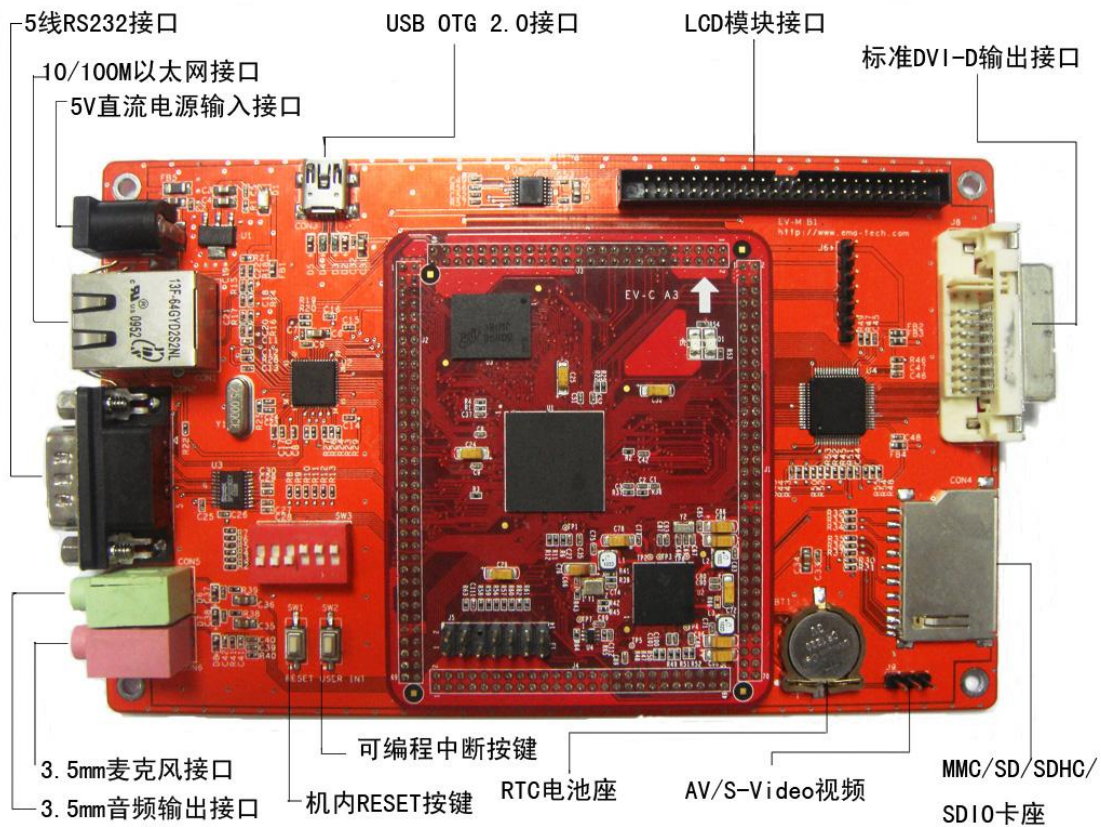


图 2.1 DEV3530 框图



2.1.2. DEV3530 的硬件参数

处理器	TI OMAP3530	600MHz ARM Cortex™-A8 内核 NEON™ SIMD 协处理器 430MHz TMS320C64x+ DSP POWERVR SG™ 2D/3D 图形加速处理引擎
RAM	128MByte/256MByte/512MByte DDR	
Flash	128MByte/256MByte/512MByte/1GByte Nand Flash	
以太网	1x 10M/100M 高性能以太网 RJ45 接口	
串口	1x 5 线 RS232 接口 (TX,RX,CTS,RTS,GND)	
	1x 1.8V 5 线 UART (TX,RX,CTS,RTS,GND)	
USB OTG	1x USB 2.0 高速 OTG (可用作 Device)	
音频	1x 3.5mm 音频输出接口	
	1x 3.5mm 麦克风输入接口	
DVI-D	标准 DVI-D 输出接口，支持所有 VESA 标准分辨率	
AV/S-Video 视频输出	支持下列标准： NTSC-J, M PAL-B, D, G, H, I PAL-M CGMS-A	
SD 卡槽	MMC/SD/SDIO/SDHC 卡槽，最高支持 32GByte 容量	
RTC	内置 RTC，使用 CR1220 电池供电	
按钮	1x 可编程用户中断按钮	
	1x 机内 RESET 按钮	
LED 指示灯	2x 电源指示灯	
	1x 可编程电源管理模块指示灯	
启动模式选择开关	1x 6 位启动模式选择拨码开关	
电源	1x 5V 1A DC 插口	
扩展接口	1x 1.8V LCD 模块接口 (支持触摸)	



2.1.3. DEV3530 工作环境

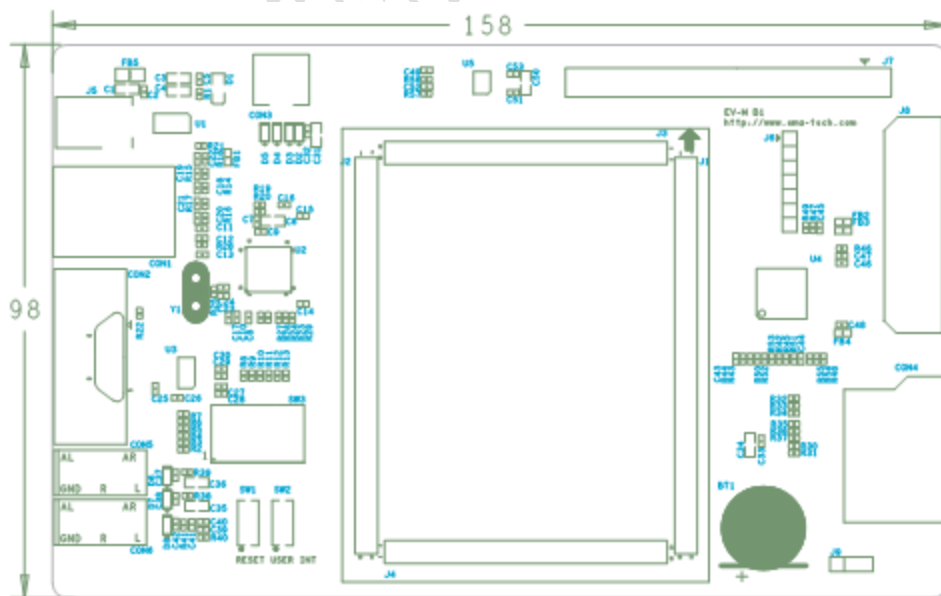
环境	最小值	典型值	最大值
供电	5V 0.3A	5V 0.5A (备注 2)	5V 1A (备注 1)
工作功耗	1.5W	2.5W (备注 2)	5W (备注 1)
工作温度 (商业级版本)	0 ℃	/	70 ℃
工作温度 (工业级版本)	-40 ℃	/	80 ℃

备注 1：接上 LCD 模块情况下的平均值。

备注 2：该值为 CPU 处于 600MHz 全速工作状态，板上其他电路处于工作状态，并且不接入其他模块，USB 接口不对外供电的情况下的平均值。

2.1.4. DEV3530 机构资料

PCB 尺寸	158.00 x 98.00 mm
安装孔数目	4 个





2.2. DEV3530 的接口说明

2.2.1. 接口一览表

接口序号	接口描述	备注
CON1	10/100M 以太网接口	高性能以太网
CON2	5 线 RS232 接口	
CON3	USB OTG 2.0 接口	
CON4	MMC/SD/SDHC/SDIO 卡座	3V/1.8V 兼容
CON5	3.5mm 音频输出接口	GREEN
CON6	3.5mm 麦克风接口	PINK
BT1	RTC 电池座	使用 CR1220 电池
J1	核心板接口	
J2	核心板接口	
J3	核心板接口	
J4	核心板接口	
J5	5V 直流电源输入接口	DC-208 接口，内正外负，最大电流 1A
J6	5 线 UART 接口	1.8V I/O
J7	LCD 模块接口	2x25Pin 简易牛角座
J8	标准 DVI-D 输出接口	
J9	AV/S-Video 视频输出接口	

2.2.2. 按键/开关一览表

按键序号	按键描述	备注
SW1	机内 RESET 按键	
SW2	可编程中断按键	GPIO18

2.2.3. LED 一览表

LED 序号	LED 描述	备注
D1	3.3V 供电指示	



2.3. 接口详述

1) +5V 直流电源输入插座 J5

CON1 为+5V 直流电源输入座，接口为 DC-208，内正外负，最大电流 4A。

2) 网络接口插座 CON1

此接口为标准 RJ45 10/100M 自适应以太网接口，带有链路灯、数据灯。

3) 拨码开关 SW3 (备注：1：开关向上拨 0：开关向下拨)

通过拨码开关设置 DEV3530 启动设备的顺序。常用的拨码顺序，可查看 DEV3530 系统连接说明。

Booting Sequence When SYS.BOOT[5] = 0 Memory Booting Preferred Order					
sys_boot [4:0]	First	Second	Third	Fourth	Fifth
0b00000			Reserved(1)		
0b00001					
0b00010					
0b00011					
0b00100	OneNAND	USB			
0b00101	MMC2	USB			
0b00110	MMC1	USB			
0b00111			Reserved(1)		
0b01000					
0b01001					
0b01010					
0b01011					
0b01100					
0b01101	XIP	USB	UART3	MMC1	
0b01110	XIPwait	DOC	USB	UART3	MMC1



0b01111	NAND	USB	UART3	MMC1	
0b10000	OneNAND	USB	UART3	MMC1	
0b10001	MMC2	USB	UART3	MMC1	
0b10010	MMC1	USB	UART3		
0b10011	XIP	UART3			
0b10100	XIPwait	DOC	UART3		
0b10101	NAND	UART3			
0b10110	OneNAND	UART3			
0b10111	MMC2	UART3			
0b11000	MMC1	UART3			
0b11001	XIP	USB			
0b11010	XIPwait	DOC	USB		
0b11011	NAND	USB			
0b11100			Reserved(1)		
0b11101					
0b11110					
0b11111	Fast XIP booting.devices)Wait monitoring OFF (only for GPdevices)	USB (only on GP devices)	UART3 (only on GP devices)		
Bootling Sequence When SYS.BOOT[5] = 1 Peripheral Bootling Preferred Order					
sys_boot [4:0]	First	Second	Third	Fourth	Fifth
0b00000			Reserved(1)		
0b00001					
0b00010					
0b00011					
0b00100	USB	OneNAND			
0b00101	USB	MMC2			
0b00110	USB	MMC1			
0b00111			Reserved(1)		
0b01000					
0b01001					
0b01010					
0b01011					
0b01100					



0b01101	USB	UART3	MMC1	XIP	
0b01110	USB	UART3	MMC1	XIPwait	DOC
0b01111	USB	UART3	MMC1	NAND	
0b10000	USB	UART3	MMC1	OneNAND	
0b10001	USB	UART3	MMC1	MMC2	
0b10010	USB	UART3	MMC1		
0b10011	UART3	XIP			
0b10100	UART3	XIPwait	DOC		
0b10101	UART3	NAND			
0b10110	UART3	OneNAND			
0b10111	UART3	MMC2			
0b11000	UART3	MMC1			
0b11001	USB	XIP			
0b11010	USB	XIPwait	DOC		
0b11011	USB	NAND			
0b11100					
0b11101			Reserved(1)		
0b11110					
0b11111	Fast XIP booting.Wait monitoring ON (only for GP devices)	SB (only on GP devices)	UART3 (only on GP devices)		

4) 音频输出插座 CON5

提供立体声（双声道）音频信号输出，请在此端口连接有源音箱的音频输入插头

或者立体声耳机/耳塞。

5) 麦克风输入插座 CON6

提供麦克风输入（单声道），请在此端口连接标准的单声道麦克风音频输入插

头。



6) 复位开关 SW1

当按下并释放时，会使 DEV3530 进入上电复位状态，可以使开发板重新启动。

7) SD 卡插座 CON4

可接标准 MMC/SD/SDHC/SDIO 卡,与 3V/1.8V 兼容。系统可以从此插座插入 SD/SDIO 卡启动。

8) 串口通信插座 J6

串口插座提 5 线 RS232 接口通讯信号到其他设备。使用标准串口电缆连接开发板和其他标准串口接口的设备（如通用的 PC 机）。

9) 数字视频 DVI 插座 J8

DEV3530 可通过此端口连接一个 DVI-D 输入接口液晶显示器。通过此接口，标准 TMDS 数字彩色显示信号。此接口提供的 I2C 通讯了实现对显示器的识别与通信。

在连接评估板和显示器的 DVI-D 接口时，要选用 DVI-D 型视频连接线。在选配液晶显示器时，除了要兼容 DVI-D 接口，还建议选用能够支持 1280*720 或以上分辨率的显示器。



10) LCD 接口座 J7

该接口为 50Pin 扁平排线接口，包含 24 位真彩色 CMOS 显示信号与触摸屏接口，所接液晶屏显示的分辨率可达到 1920*1080。

11) AV 端子 (CVBS) / SVIDEO OUT 插座 J9

插座支持 CVBS 或者 S-VIDEO 输出，即可 2 选 1 输出。

12) USB OTG 2.0 接口 CON3

此接口为 Mini 型 USB 主从复用插座，可连接标准的 USB 从设备。支持 OTG、HOST 两种模式（配用不同标准的接线）。

13) D1 电源指示灯

当 DEV3530 正确输入 DC+5V 电源后，指示灯会亮起来，提示供电正常。如果供电后指示灯不亮，请立即断开电源并查找故障原因。

2.4. DEV3530 接口引脚定义

J6 UART3 接口，该接口用于接入 UART3

引脚	功能描述	备注
1	3.3V 电压输出	转接板供电
2	1.8 V I/O 参考电压输出	I/O 参考电压
3	地	
4	CTS	1.8V I/O



5	RTS	1.8V I/O
6	TX	1.8V I/O
7	RX	1.8V I/O

J7 LCD 模块接口，该接口用于接入液晶屏设备。

引脚	功能描述	备注
1,2	5V 电源输出	用于 LCD 设备供电
4,18,31,50	地	
6-17	数据线 D0-D11	1.8V I/O
19-30	数据线 D12-D23	1.8V I/O
36	DSS_VDEN	1.8V I/O
37	DSS_VSYNC	1.8V I/O
38	DSS_HSYNC	1.8V I/O
39	DSS_PCLK	1.8V I/O
43	X+	触摸屏输入信号
45	X-	触摸屏输入信号
47	Y+	触摸屏输入信号
49	Y-	触摸屏输入信号
3,5,32-35,40-42,44,46,48	悬空	

J9 AV/S-Video 视频输出接口

引脚	功能描述	备注
1	AV/SVIDEO_Y	CVBS 输出/S-Video Y 输出
2	SVIDEO_C	S-Video C 输出
3	地	



第三章

3. DEV3530 的基本使用



3.1. 以 DEV3530 为核心的计算机系统

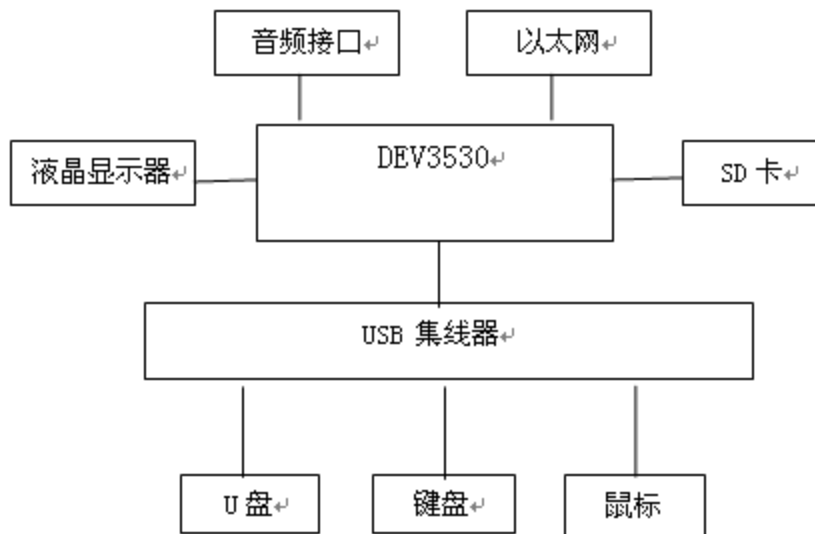


图 3.1

标准配件

- 1) 直流+5v的电源一只
- 2) 标准交叉串口线1条，两端均为母头。
- 3) 软件光盘1张 DVD
- 4) 网线
- 5) USB A 公头转MINI B 公头线



扩展配件

- 1) USB 键盘
- 2) USB鼠标
- 3) 音频连接线（双声道）1条
- 4) 4口USB集成器

3.2. DEV3530 连接设置

3.2.1. 外部接口连接

1) 断电

各个连接设备均断开电源、连接时不带电操作，并且需要防止静电伤害。

建议：选取一个单独的、有总电源开关的电源接线板，所有DEV3530相关电源(除PC)都连接 在此，这样系统通电和断电比较统一开关此接线板上总电源，在操作上也比较安全。

2) 准备DEV3530

取出DEV3530，去除包装后，观察是否各部分完整，有无明显的损毁和异常情况。



3) 连接串口通信电缆

连接串口电缆一端到DEV3530插座CON7，另一端连接到PC机串口接口（通常是COM1口）。

4) 连接液晶显示器

准备好一台VESA标准分辨率、DVI接口的液晶显示器。取出DVI视频电缆，将一端连接到DEV3530上CON11接口。再将另一端插接到液晶显示器的DVI输入接口上。

5) 连接LCD触摸屏

取出LCD触摸屏，用屏线连接DEV3530 CON10接口，屏线的金手指对应LCD触摸屏的驱动板接口和DEV3530 CON10接口都是向下。

注：3.3 V LCD CON10 接口不要误错接 5V GPMC接口，有烧板的危险。

6) 连接键盘跟鼠标

取出USB接口的键盘和鼠标，将他们分别连到USB Host上。

7) 插上SD卡

将可以启动的SD卡插入到DEV3530 CON4插座，注意卡的正面向上，缺口对应插座缺口的方向。



8) 设置拨码开关J12 (备注：1：开关向上拨 0：开关向下拨)

拨码顺序	第一设备	第二设备	第三设备	第四设备
111100	NAND	USB	UART3	MMC1
111101	USB	UART3	MMC1	NAND
000111	UART3	MMC1		

将拨码开关设置为 (1-6)：111100，

检测顺序是NANDFLASH->USB->UART3->MMC1

将拨码开关设置为 (1-6)：111101，

检测顺序是USB->UART3->MMC1-> NANDFLASH

将拨码开关设置为 (1-6)：000111，

检测顺序是UART3->MMC1

3.2.2. DEV3530 系统上电步骤

1) 接通外设电源

请逐一开启个外设电源的电源，如液晶显示、USB集成器等。

2) 连接并接通DEV3530电源

取出DEV3530配套的5V电源，先将其接到220V接线板上通电，再将5V的电源输

出接到DEV3530的电源插座CON1。



3) 同步可移动存储设备

如果使用USB接口热插拔存储设备并向其中写过文件，可以再Linux控制台命令行中输入sync命令，使存放在缓存中的数据全部写入这些设备。

3.2.3. DEV3530 系统关闭步骤

- 1) 卸载可移动的存储设备
- 2) 关闭系统电源
- 3) 切断DEV3530系统的供电电源
- 4) 关闭外设电源



第四章

4. DEV3530 的 Linux 系统使用与设置



4.1. 系统启动方法和过程

DEV3530 的常用启动途径分为两个：

- ✧ 一为从板载 NandFlash 启动；
- ✧ 一为从 SD 卡启动；

从 SD 卡启动可以获得更灵活的存储容量配置，另外可以灵活地更换系统软件。

两种启动途径都使用到 SD 卡。SD 卡启动是一个完整的系统都从 SD 里启动，而 Nandflash 启动是从 SD 卡启动进入到 U-Boot，通过命令将 SD 卡里的四个系统镜像文件下载到内存，再烧写到 Nandflash 上，然后重上电后就不用 SD，一个完整的系统就可以从 Nandflash 里启动。所以 SD 卡的格式化分区对启动系统是很重要的，具体实现可查看 **6.2.1 制作 SD 系统启动卡**。

4.1.1. 系统启动过程：



- 检测 Nandflash/SD 卡。从 Nandflash/SD 卡的 Fat32 分区的起始地址装置 X-Loader 程序到存储器，并开始执行 X-Loader。
- X-Loader 首先做一些初始化工作，然后从 Nandflash/SD 卡 Fat32 分区读取 U-Boot 程序到存储器，并开始执行 U-Boot。
- U-Boot 从 Nandflash/SD 卡 Fat32 分区读入启动参数，然后读入 Linux 内核 uImage 到 DEV3530 存储器，解压缩 uImage 并开始运行内核。
- Linux 内核启动后，重新初始化 DEV3530，接着加载 NandFlash 上存放的文件系统或者是 SD 卡 EXT3 分区的文件系统，执行文件系统程序并启动控制台。

4.1.2. SD 卡启动过程

- 1) 准备好启动系统的镜像文件 (MLO , u-boot.bin,uImage,rootfs) ,并将三个镜像文件放到 SD 卡的 FAT32 分区里，将 rootfs 放在 SD 卡的 EXT3 分区里 (将光盘提供的文件系统压缩包，在 linux 系统下解压到 SD 卡的 EXT3 分区)。镜像文件的生成，可查看 **6.1 系统编译**。
- 2) 连接硬件，请参考 **3.2.1 外部接口连接**。
- 3) 在宿主机 (PC 机) 上启动超级终端程序，其配置可查看 **5.3 串口终端软件的使用**。



- 4) 连接并接通DEV3530电源，请参考**3.2.2 系统上电步骤**。

正常启动系统后，就可以在超级终端看到目标板的启动信息，当超级终端里显示

login 字符时，输入用户名“root”，直接按回车就可以进入 Linux 系统。

4.1.3. Nandflash 启动过程

- 1) 准备好启动系统的镜像文件（MLO，u-boot.bin, uImage, ubi.img），并将四个镜像文件放到 SD 卡的 FAT32 分区里。镜像文件的生成，可查看 **6.1 系统编译**。
- 2) 将启动系统的镜像文件烧写到 Nandflash 上，具体步骤可查看 **6.2.3 NAND Flash 系统映像更新**。
- 3) 连接硬件（不用将 SD 插到 DEV3530 板上），请参考 **3.2.1 外部接口连接**。
- 4) 在宿主机（PC 机）上启动超级终端程序，其配置可查看 **5.3 串口终端软件的使用**。
- 5) 连接并接通DEV3530电源，请参考**3.2.2 系统上电步骤**。
- 6) 正常启动系统后，就可以在超级终端看到目标板的启动信息，当超级终端里显示 login 字符时，输入用户名“root”，直接按回车就可以进入 Linux 系统。

4.2. 显示设置

DEV 3530 支持多种显示输出，系统默认是从 DVI 输出。如果用户想使用 LCD 输出或

者想修改分辨率，则用户需要在 U-Boot 里修改对应的参数。



通用的分辨率对应的刷新率，可以参考下表：

分辨率	刷新率
480x272	60
800x480	60
800x600	60
1024x768	60
1280x720	60
1366x768	60
1280x1024	50
1440x900	50
1920x1080	30

4.2.1. DVI 接口显示器显示：

```
# setenv dvmode 1024x768MR-16@60
#setenv defaultdisplay dvi
```

setenv dvmode 1024x768MR-16@60 这个表示使用模式是 DVI、1024x768 的分辨率、16 色位和刷新率是 60。

4.2.2. 4.3 寸触摸屏显示：

```
# setenv dvmode
# setenv defaultdisplay lcd043
```

参数的意义是指不用设置 DVI 显示模式的分辨率、色位和刷新率，再把修改默认的显示模式改成 4.3 寸触摸屏,其分辨率是 480x272。

4.2.3. 7 寸触摸屏显示：

```
# setenv dvmode
# setenv defaultdisplay lcd070
```

参数的意义是指不用设置 DVI 显示模式的分辨率、色位和刷新率，再把修改默认的显示模式改成 7 寸触摸屏,其分辨率是 800x480。



➤ 复制光盘 Demo 文件夹到/home/ema/目录下(文件夹是在/mnt/hgfs/share 共享目录)

➤ 将 Demo 的镜像和系统复制到已经格式化好的 SD 卡里面

- 从虚拟机里断开 USB 连接，取出 SD 卡插到开发板，上电启动开发板

➤ 进入 U-BOOT 设置参数。(默认是从 DVI 接口输出, 要想 LCD 输出显示, 可查看

OMAP3 Stalker #setenv vram 12M omapfb.vram=0:12M

OMAP3 Stalker # boot

```

      .-----  

      |       |  

      |   |   |-----|   |   |-----|  

      |   |   |___|---|'-'|   |   |  

      |   |   |   |---||---|'-'|   |   |||||  

      '-----'-----"-----"  

           2 |

```



```
'...'  
The Angstrom Distribution stalker ttyS2  
Angstrom 2009.X-stable stalker ttyS2
```

stalker login: root

- 用户名是：root，密码是空，直接按回车就可以登陆

```
root@beagleboard:~# cd /opt/gfxsdkdemos/ogles  
  
root@beagleboard:/opt/gfxsdkdemos/ogles# ls  
Balloon_fixed.pod      OGLESOptimizeMesh      SphereOpt_fixed.pod  
Balloon_float.pod      OGLESParticles          SphereOpt_float.pod  
ChameleonScene.pod     OGLESPhantomMask       Sphere_fixed.pod  
ChameleonScene_Fixed.pod OGLESPolyBump           Sphere_float.pod  
Mesh_fixed.pod         OGLESShadowTechniques  balloon.pvr  
Mesh_float.pod         OGLESSkybox             o_model_fixed.pod  
OGLESChameleonMan      OGLESTrilinear          o_model_float.pod  
OGLESCoverflow         OGLESUserClipPlanes    skybox1.pvr  
OGLESEvilSkull          OGLESVase               skybox2.pvr  
OGLESFiveSpheres       PhantomMask_fixed.pod   skybox3.pvr  
OGLESFur                PhantomMask_float.pod   skybox4.pvr  
OGLESLighting          Scene_fixed.pod         skybox5.pvr  
OGLESMouse              Scene_float.pod         skybox6.pvr  
  
root@beagleboard:/opt/gfxsdkdemos/ogles# ./OGLESVase
```

4.3.2. Quake3 游戏演示

- 进入 U-BOOT 设置参数。(默认是从 DVI 接口输出，要想 LCD 输出显示，可查看 4.2 节的设置)

```
OMAP3 Stalker # setenv vram 4M omapfb.vram=0:4M
```

```
OMAP3 Stalker # saveenv  
Saving Environment to NAND...  
Erasing Nand...  
Erasing at 0x260000 -- 100% complete.  
Writing to Nand... done
```

```
OMAP3 Stalker # boot
```

```
Starting Connection Manager
```

```
.-----.  
| | | | |  
| | | | |  
| | | | |
```



➤ 用户名是：root，密码是空，直接按回车就可以登陆。

➤ 执行运行脚本后就可以在 DVI 显示器上看到游戏界面了，选择 SINGLE PLAYER 进入选择游戏地图，然后选择 CHOOSE LEVEL FIGHT→DIFFICULTY FIGHT，就可以开始玩游戏了。退出按连接开发板键盘的“Esc”键之后选择 EXIT GAME→YES，由于将 console 屏蔽了，所以只能通过串口终端来控制，所以游戏也可以在 PC 机的键盘上直接用“Ctrl+c”来结束游戏的。

➤ 进入 U-BOOT 设置参数。(默认是从 DVI 接口输出, 要想 LCD 输出显示, 可查看 4.2 节的设置)

OMAP3 Stalker # boot



stalker login: root

```
root@beagleboard:/opt/dvSDK/omap3530# ./loadmodules.sh
root@beagleboard:/opt/dvSDK/omap3530# ./decode -v ./data/videos/davincieffect_ntsc_1.264
```



第五章

5. 构建嵌入式 Linux 的开发环境



5.1. Linux 软件开发环境概述

在开始开发使用之前，您需要构建DEV3530系统的开发环境。在这一章中概述了装载DEV3530系统相关的软件到开发主机所需要的步骤。你将需要一张DEV3530系统附带的光盘，安装操作需要用户对windows和Linux操作系统有一定的了解和应用的知识。能够在这样的环境下简单的操作。

作为一款开放的Linux系统，DEV3530所用到的软件一般均为免费和开源的，能在互联网上自由下载，这些软件稍经配置和修改后，就可以在DEV3530上运行。系统附带的光盘上已经包含了这些内容，可以免去下载过程。同时在使用到这些资源时，在本文中会给出下载网址、网页。访问<http://code.google.com/p/ema3530/>的网址可以获得及时更新软件和文档。

DEV3530系统提供一套较为完整的软件开发环境。当按照系统安装的步骤完成后，用户可以无需另行安装下载多余的软件或组件，就可以立即开始进行应用程序的开发工作。

开发环境以一台操作系统的Windows XP主机为基础的开发平台，进行开发操作和测试。



5.2. 构建软件开发环

5.2.1. VMware-workstation-6.5.0 的安装

1. 双击 VMware-workstation-6.5.0-118166.exe 进入如下所示的安装界面



图 4.1

2. 单击 next , 进入图 4.2 所示的界面。选择 Typical 安装



图 4.2



3. 单击 next 按钮，进入下图所示的界面。单击 change..按钮，选择需要安装路径，单击 next 按钮，界面如图 4.3 所示。单击 next 如图 4 所示

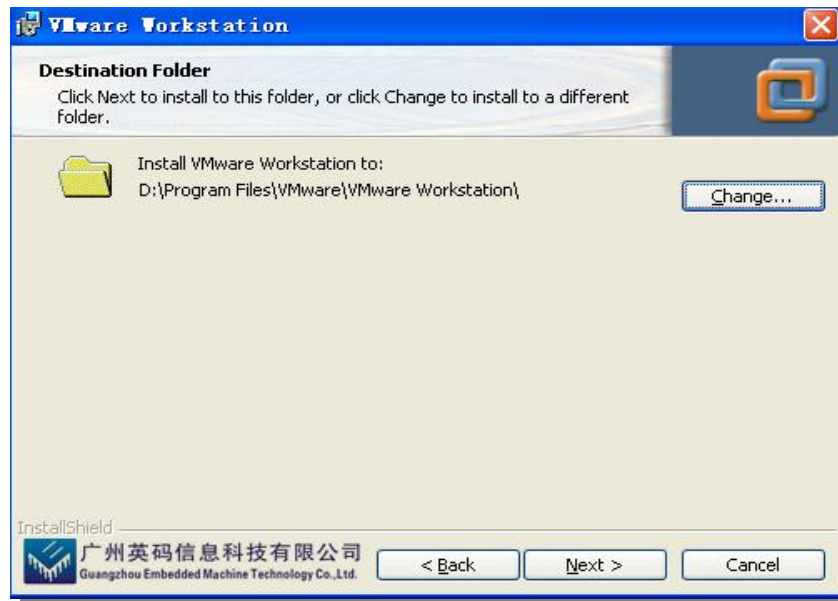


图 4.3

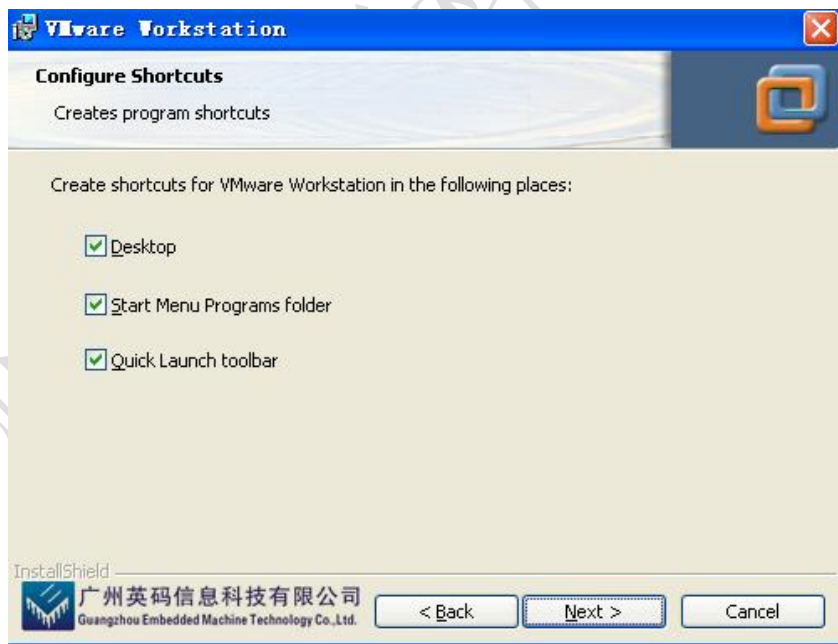


图 4.4

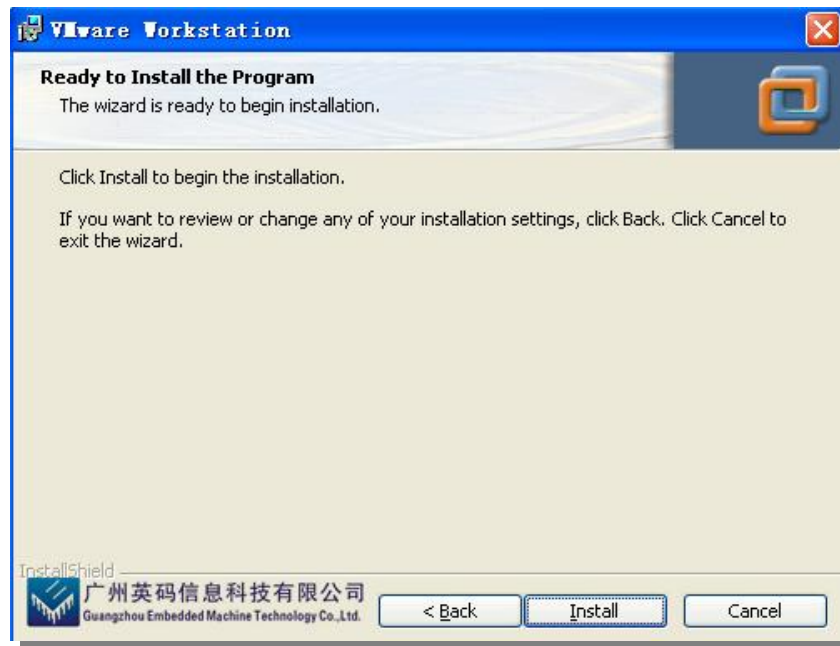


图 4.5

4. 单击 install 后，安装开始启动如图 4.6 所示

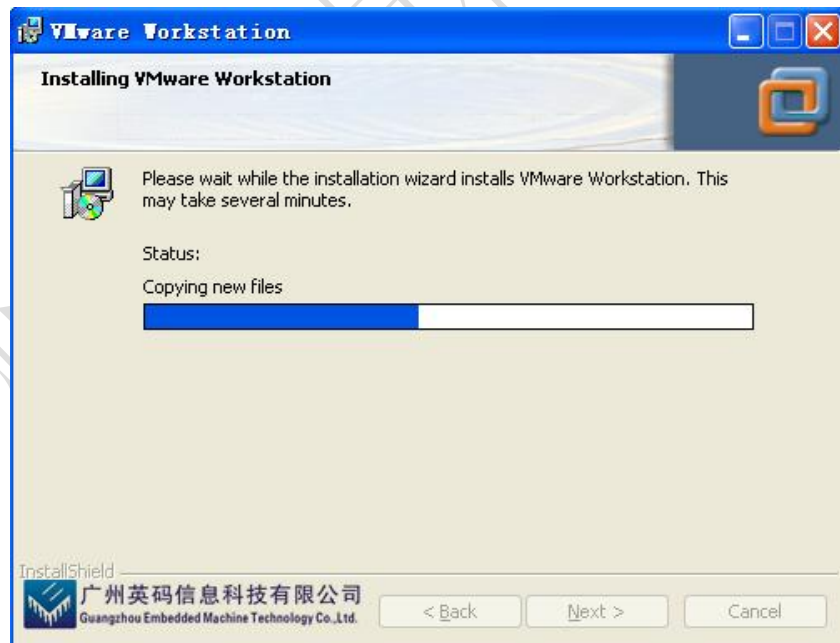


图 4.6



5. 安装完成后系统会弹出如图 4.7 所示的界面，点击 Yes 按钮重启计算机

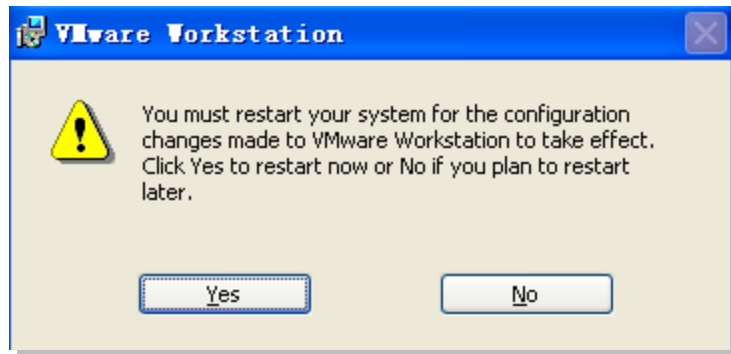


图 4.7

6. 最终在桌面生成 VMware-workstation 图标



图 4.8

5.2.2. Ubuntu 的安装

为了方便客户使用，在配套的光盘中附带了一个 Ubuntu 系统。用户只需要将其复制到自己的电脑中，解压后用虚拟机打开。这样非常方便用户直接使用 Ubuntu 系统，节省安装时间。具体步骤如下：

- 1) 打开光盘、复制并解压 vm.7z 文件如下图所示 4.9

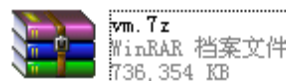


图 4.9



2) 打开 VMware-workstation 图标 (如 4.10)

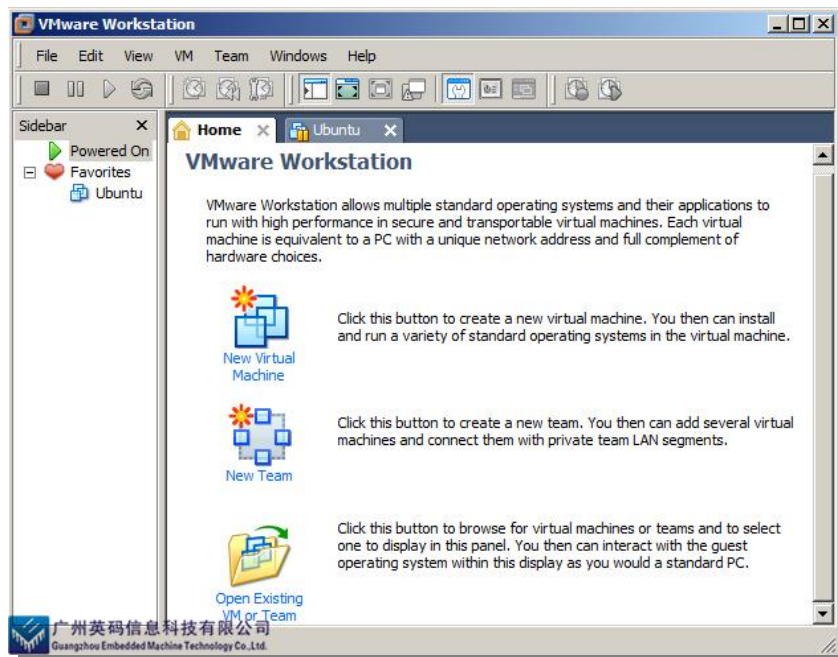


图 4.10

3) 打开 File 下的 Open 如图 4.11 所示

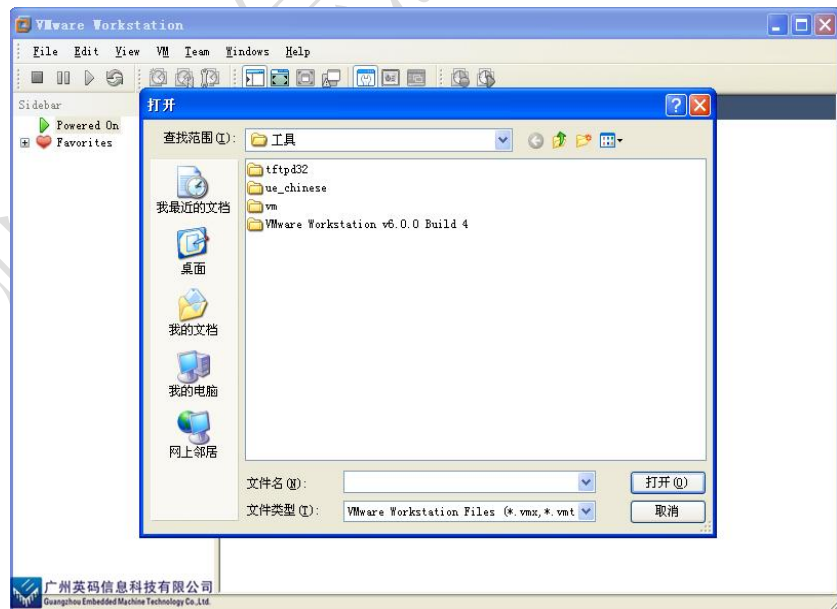


图 4.11



4) 选择刚刚解压的 vm 文件如图 4.12 所示

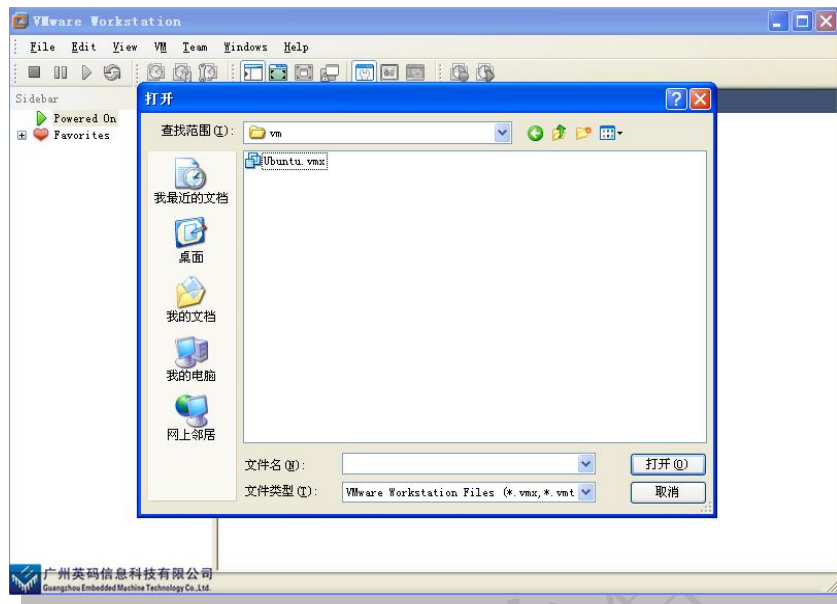


图 4.12

5) 选择左上角的绿色三角，启动 ubuntu 如图 4.13 所示

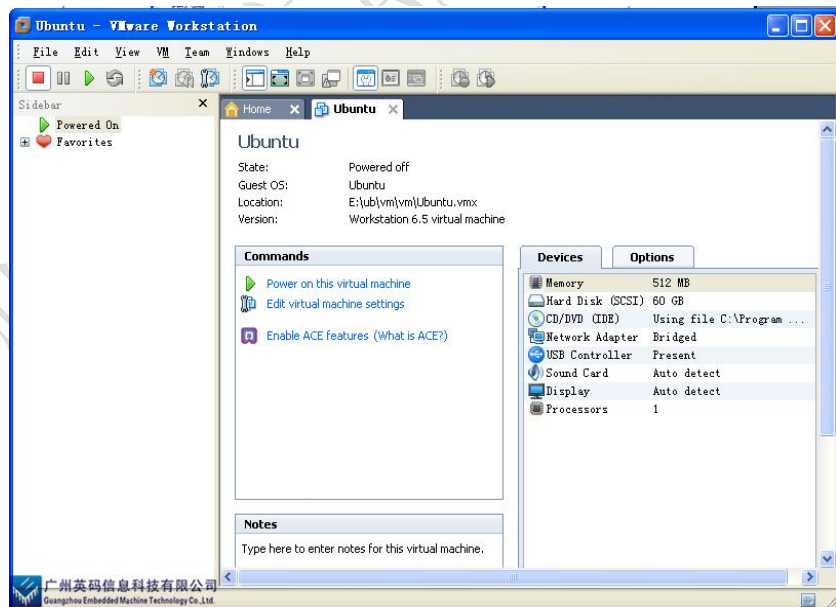


图 4.13



6) 登陆 ubuntu 如图 4.14 所示

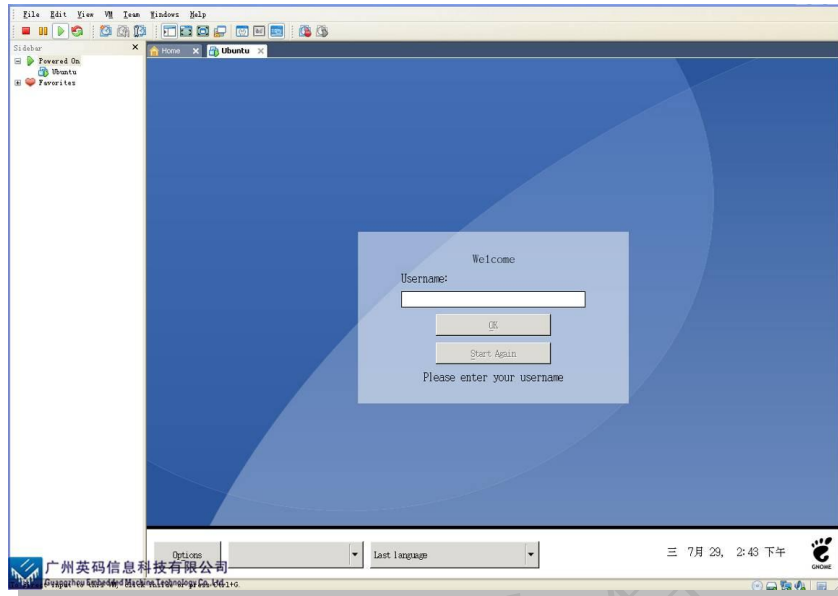


图 4.14

在出现的对话框中 Username 输入用户名：ema Password 中输入密码 ema

7) 登陆成功如图 4.15 所示

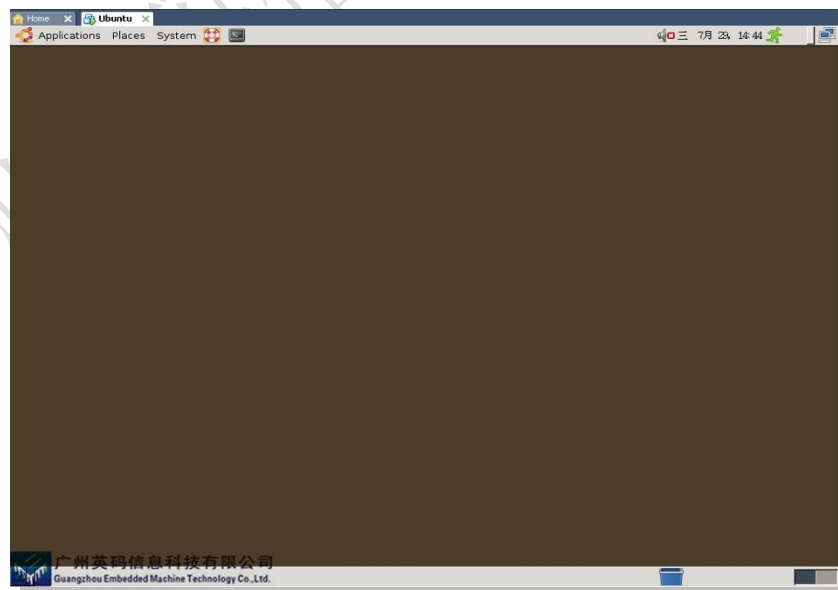


图 4.15



5.2.3. 虚拟机跟主机的文件共享设置

- 1) 设置共享文件夹。选择 VM/Settings 或者 Ctrl+ D 系统弹出如图 4.16 所示的对话框

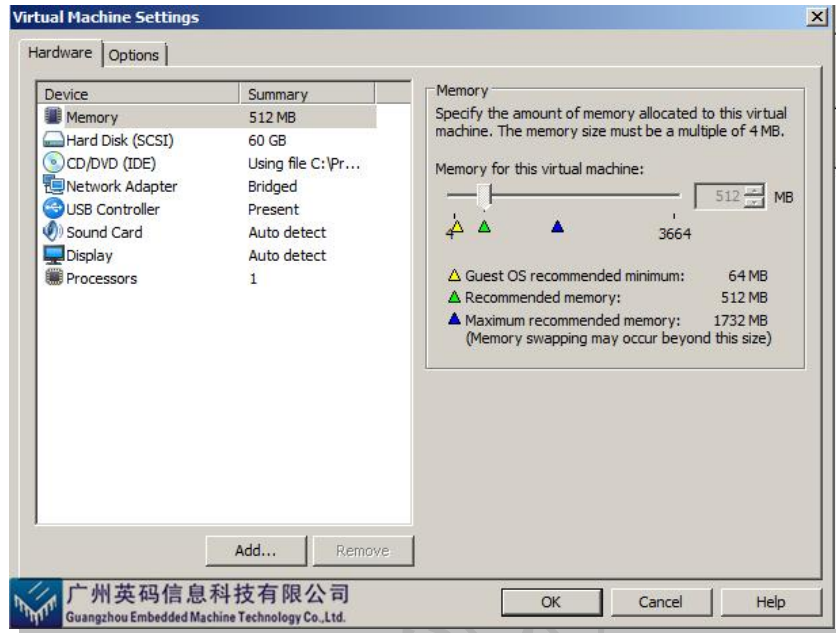


图 4.16

- 2) 选择 options 下的 Shared Folders 。如图 4.17 所示：

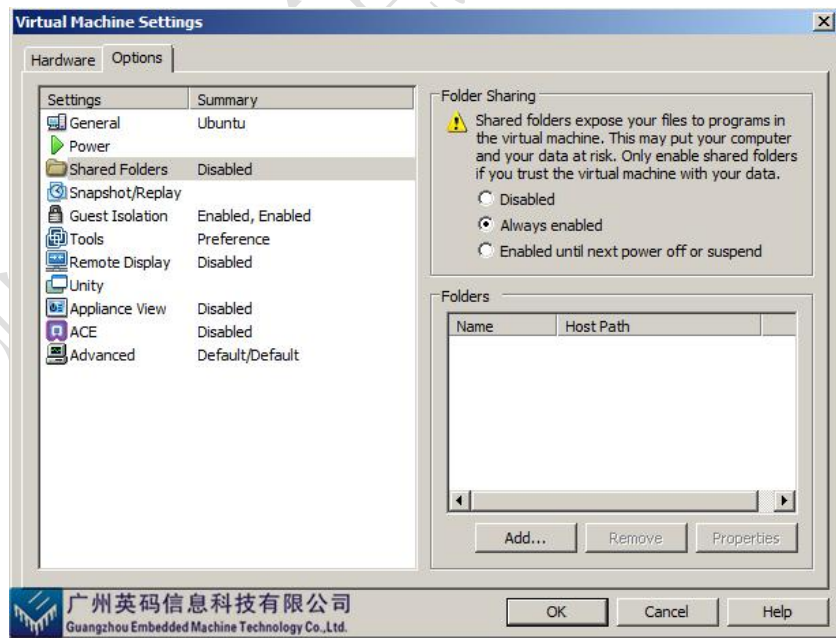


图 4.17



- 3) 通过添加按钮，添加共享的文件夹。也就是选择让虚拟机的 linux 系统可以访问的文件夹。Linux 访问这些文件夹的路径就是“/mnt/hgfs/”在弹出的向导对话框中，设置共享的文件的路径及名字

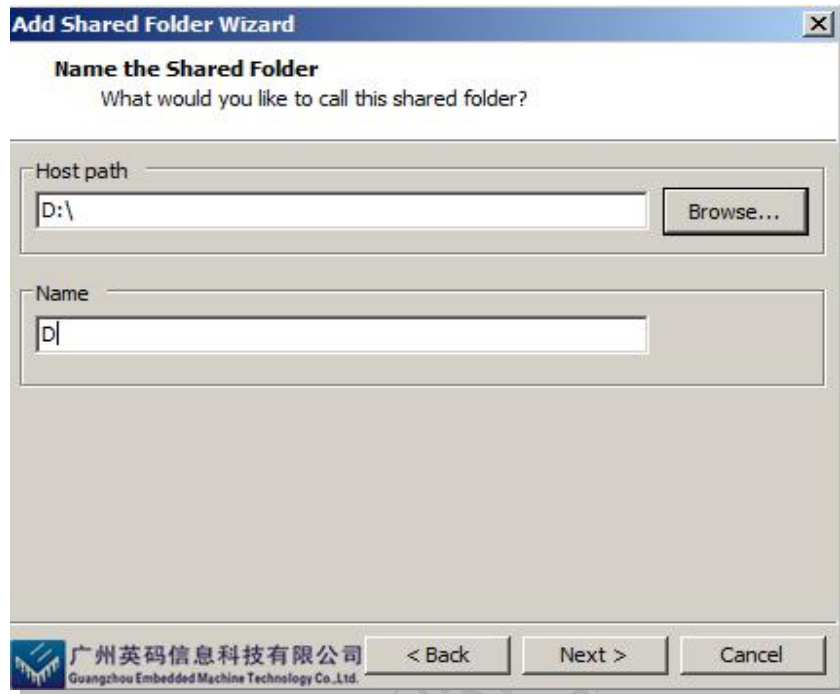


图 4.18

- 4) 设置完后，在 Shared Folders 中自动添加了共享文件路径。如下所示：

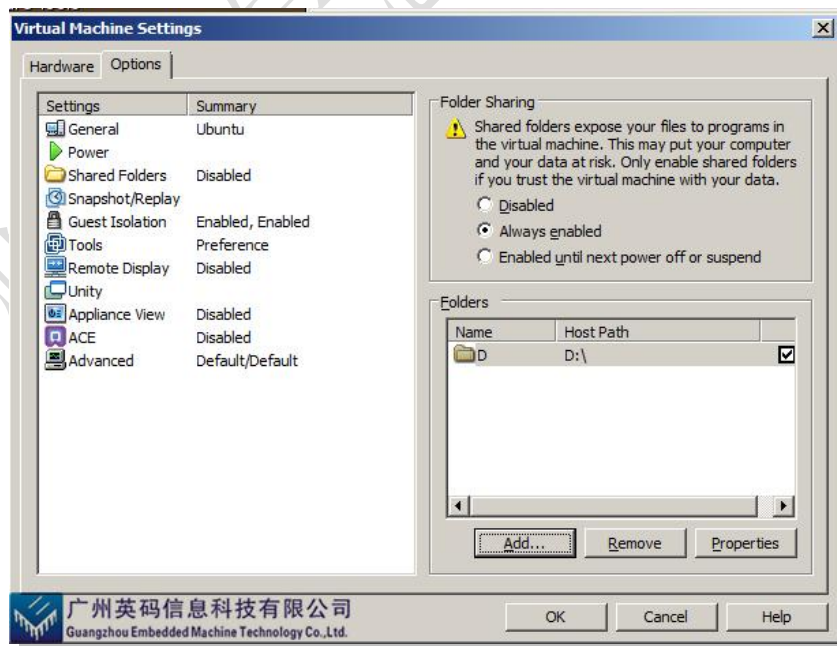


图 4.19



- 5) 安装完成后，在/mnt/目录下面多了一个 hgfs 子文件夹，这个文件夹就是用于主机与虚拟机通信的文件夹。

命令为：

```
#cd /mnt/hgfs  
#ls
```

就可以查看到共享的目录。

5.2.4. 构建交叉编译器

- 1) 在/usr/local/目录下面创建文件夹 arm

命令为：

```
#su  
#ema  
#mkdir /usr/local/arm
```

- 2) 复制光盘

arm-2009q1-203-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 到/home/ema/tool(压

缩文件是在/mnt/hgfs/share 盘目录)

命令为：

```
#cd /mnt/hgfs/share  
#cp arm-2009q1-203-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 /home/ema/tool
```

- 3) 解压到/usr/local/arm 目录下。

命令为：

```
#cd /home/ema/tool  
#tar jxvf arm-2009q1-203-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 -C /usr/local/arm
```



4) 设置环境变量

命令为：

```
#export PATH= /usr/local/arm/arm-2009q1/bin:$PATH
```

5) 查看环境变量

命令为：

```
#echo $PATH
```

5.3. 串口终端软件的使用

- 在 WindowsXP 桌面上选择菜单：开始->所有程序->附件->通讯->超级终端
- 在“连接描述”窗口中输入名称：ema，选择“确定”
- 在“连接到”窗口中确认“连接时使用”项为“COM1”，选择“确认”
- 在“COM1 属性”窗口中设定：每秒位数=115200，数据位=8，奇偶校验=无，停止位=1，数据流控制=无，选择“确定”
- 在桌面上会自动打开“ema-超级终端”窗口
- 在“ema-超级终端”窗口中选择菜单：文件->退出，在系统询问是否存储连接时回答“是”
- 在 Window XP 桌面上展开并右键单击单项：开始->所有程序->附件->通讯->超级终端->ema,选择“发送到桌面快捷方式”，回到桌面将新建立的连接改名为“连接 ema”启动超级终端



- 在 Window XP 桌面上双击“连接 ema”图标启动超级终端

建议使用 ZOC Terminal，软件在光盘 Linux\Tools\ZOC 文件夹里，里面有配置说明。

5.4. 挂载网络文件系统 NFS

NFS 服务就是将宿主机的一个目录通过网络可以被挂载到其他计算机上，并且作为其他计算机的一个目录。在嵌入式开发中，通过 NFS 可以很方便的将修改的文件通过 NFS 传输到目标板上。

Ubuntu 默认是没有 NFS 服务的，所以需要自己安装。

1. 安装 NFS 服务版

```
ema@ema3530:~$ sudo apt-get install nfs-kernel-server
```

2. 修改 NFS 配置文件，vim /etc/exports

```
ema@ema3530:~$ vim /etc/exports
```

在文件中添加 NFS 的目录格式如下，并存盘退出。

```
/nfs_share *(rw,sync,no_root_squash)
```

3. 在根目录下建立共享的目录，并创建 test 文件放在目录里

```
ema@ema3530:~$ mkdir /nfs_share  
ema@ema3530:~$ touch /nfs_share/test
```

4. 重新启动 NFS

```
ema@ema3530:~$ sudo /etc/init.d/nfs-kernel-server restart
```



5. 查看宿主机 IP

```
ema@ema3530:~$ ifconfig
```

6. 挂载本地目录，测试 NFS

```
ema@ema3530:/$ sudo mount 192.168.1.162:/nfs_share/ /tmp/
[sudo] password for ema:
ema@ema3530:/$ ls /tmp/
aaa
ema@ema3530:/$ sudo umount /tmp/
ema@ema3530:/$ ls /tmp/
gconfd-ema keyring-QhwEr7 orbit-ema ssh-oHsGDk5300 VMwareDnD vmware-root
```

7. 目标板需要安装 nfs-utils-client

```
root@beagleboard:~# opkg install nfs-utils-client
Installing nfs-utils-client (1.1.2-2.1) to root..
Downloading http://www.angstrom-distribution.org/feeds/2008/ipk/glibc/armv7a/base/nfs-utils-
client_1.1.2-2.1_armv7a.ipk
Configuring nfs-utils-client
```

8. 目标板通过 NFS 挂载宿主机目录

```
root@beagleboard:~# mount 192.168.1.162:/nfs_share /tmp/
Can't set permissions on mtab: Operation not permitted
root@beagleboard:~# ls /tmp/
test
```



第六章

6. 构建 DEV3530 的软件开发环境



6.1. 系统编译

6.1.1. 一级启动代码 x-loader 编译

首先设置好环境变量（步骤查看 5.2.4），命令为：

```
ema@ema3530:~$ export PATH= /usr/local/arm/arm-2009q1/bin:$PATH
```

- a) 复制光盘 x-loader.tar.bz2 到/home/ema/source 目录下(压缩文件是在 share 盘目录)

```
ema@ema3530:~$ cd /mnt/hgfs/share  
ema@ema3530:/mnt/hgfs/share$ cp x-loader.tar.bz2 /home/ema/Source
```

- b) 解压 x-loader.tar.bz2 压缩到当前文件夹

```
ema@ema3530:~$ cd /home/ema/Source  
ema@ema3530:~/Source$ sudo tar jxvf x-loader.tar.bz2  
ema@ema3530:~/Source$ cd xloader
```

- c) make distclean 命令除依赖关系并清除以前编译产生的文件

```
ema@ema3530:~/Source/xloader$ make CROSS_COMPILE=arm-none-linux-gnueabi-  
ARCH=arm distclean  
find . -type f \  
    \( -name 'core' -o -name '*.bak' -o -name '*~' \  
    -o -name '*.o' -o -name '*.a' \) -print \  
    | xargs rm -f  
find . -type f \  
    \( -name .depend -o -name '*.srec' -o -name '*.bin' \) \  
    -print \  
    | xargs rm -f  
rm -f *.bak tags TAGS  
rm -fr *~  
rm -f x-load x-load.map
```



```
rm -f include/asm/proc include/asm/arch  
rm -f include/config.h include/config.mk
```

d) 接着配置和编译

```
ema@ema3530:~/Source/xloader$ make CROSS_COMPILE=arm-none-linux-gnueabi-  
ARCH=arm sbc3530_config  
rm -f include/config.h include/config.mk  
Configuring for omap3530beagle board...  
ema@ema3530:~/Source/xloader$ make CROSS_COMPILE=arm-none-linux-gnueabi-  
ARCH=arm
```

编译完成后，xloader 目录下会生成新的 x-load.bin。

```
ema@ema3530:~/Source/xloader$ ls  
arm_config.mk config.mk drivers lib README System.map x-load.map  
board cpu fs Makefile scripts x-load  
common disk include mkconfig signGP x-load.bin
```

e) 转换成 SD 卡启动用程序：MLO

```
ema@ema3530:~/Source/xloader$ ./signGP  
ema@ema3530:~/Source/xloader$ ls  
arm_config.mk cpu include README x-load  
board disk lib scripts x-load.bin  
common drivers Makefile signGP x-load.bin.ift  
config.mk fs mkconfig System.map x-load.map  
ema@ema3530:~/Source/xloader$ cp x-load.bin.ift MLO
```

6.1.2. 二级启动代码 u-boot 编译

首先设置好环境变量（步骤查看 5.2.4），命令为：

```
ema@ema3530:~$ export PATH= /usr/local/arm/arm-2009q1/bin:$PATH
```

a) 复制光盘 u-boot-release_20100301.tar.bz2 到/home/ema/source 目录下(压缩文件是在

share 盘目录)

```
ema@ema3530:~$ cd /mnt/hgfs/share  
ema@ema3530:/mnt/hgfs/share$ cp u-boot-release_20100301.tar.bz2 /home/ema/Source
```



b) 解压 u-boot-release_20100301.tar.bz2 压缩到当前文件夹

```
ema@ema3530:~$ cd /home/ema/Source
ema@ema3530:~/Source$ sudo tar jxvf u-boot-release_20100301.tar.bz2
ema@ema3530:~/Source$ cd u-boot-release-20100301
```

c) 配置编译

```
ema@ema3530:~/Source/u-boot-release-20100301$ make sbc3530_rev_a_config
ema@ema3530:~/Source/u-boot-release-20100301$ make
```

编译成功后就可以在当前目录下看到生成的 u-boot.bin

d) 复制 mkimage 到/usr/bin,这是为了编译内核生成 uImage

```
ema@ema3530:~/Source/u-boot-release-20100301$ cd tools/
ema@ema3530:~/Source/u-boot-release-20100301/tools$ cp mkimage /usr/bin
```

6.1.3. 内核编译

首先设置好环境变量（步骤查看 5.2.4），命令为：

```
ema@ema3530:~$ export PATH= /usr/local/arm/arm-2009q1/bin:$PATH
```

a) 复制光盘

ema_psp- 03.00.00.03 _kernel.tar.bz2 到/home/ema/source 目录下(压缩文件是在/share 盘目录)

```
ema@ema3530:~$ cd /mnt/hgfs/share
ema@ema3530:/mnt/hgfs/share$ cp ema_psp- 03.00.00.03 _kernel.tar.bz2 /home/ema/Source
```

b) 解压 ema_psp- 03.00.00.03 _kernel.tar.bz2 压缩到当前文件夹

```
ema@ema3530:/mnt/hgfs/share$ cd /home/ema/Source
ema@ema3530:~/Source$ sudo tar jxvf ema_psp- 03.00.00.03 _kernel.tar.bz2
ema@ema3530:~/Source$ cd linux-03.00.00.03
```



c) 配置编译

```
ema@ema3530:~/Source/linux-03.00.00.03$ make CROSS_COMPILE=arm-none-linux-gnueabi-  
ARCH=arm ema3530_lks_defconfig  
ema@ema3530:~/Source/linux-03.00.00.03$ make CROSS_COMPILE=arm-none-linux-  
gnueabi- ARCH=arm  
ema@ema3530:~/Source/linux-03.00.00.03$ make CROSS_COMPILE=arm-none-linux-  
gnueabi- ARCH=arm uImage
```

编译成功后就可以在 arch/arm/boot 目录下看到生成的 uImage

6.1.4. 制作文件系统镜像

- a) 复制光盘 UBIFS_tools 文件夹到/home/ema/ 目录下(文件夹是在/mnt/hgfs/share 共享目录)

```
ema@ema3530:~$ cd /mnt/hgfs/share  
ema@ema3530:~$ sudo cp -a UBIFS_tools /home/ema/
```

- b) 将 mkfs.ubifs 和 ubinize 复制到/usr/bin 目录下。

```
ema@ema3530:~$ cd /home/ema/UBIFS_tools  
ema@ema3530:~/UBIFS_tools$ cp mkfs.ubifs ubinize /usr/bin
```

- c) 解压文件系统到新建的 rootfs 的目录下。

```
ema@ema3530:~/UBIFS_tools$ cd /home/ema/  
ema@ema3530:~$ mkdir rootfs  
ema@ema3530:~$ sudo tar jxvf Angstrom-console-image-glibc-ipk-2009.X-stable-  
beagleboard.rootfs.tar.bz2 -C rootfs/
```

- d) 通过 mkfs.ubifs 和 ubinize 工具将文件系统源码生成镜像文件。

```
ema@ema3530:~$ sudo mkfs.ubifs -r rootfs -m 2048 -e 129024 -c 812 -o ubifs.img  
ema@ema3530:~$ sudo ubinize -o ubi.img -m 2048 -p 128KiB -s 512  
/home/ema/UBIFS_tools/ubinize.cfg
```



6.2. Linux 系统镜像更新

6.2.1. 制作 SD 系统启动卡

为了实现以后的各步骤操作，需要准备一张空白SD 卡(以下操作各步骤中涉及的均为此卡)，并且需要在它上面建立Fat32 分区和Ext3 分区。如果使用的是我们提供的SD卡，那么可能SD卡已经有了这两个分区，可省去分区及格式化部分操作，只需要删除已有文件即可。

- 下面命令都是在 root 权限执行的，首先下载安装 bc。(注：确保虚拟机能连接外网)

```
root@ema3530:/home/ema# apt-get install bc
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
Reading state information... 完成
下列【新】软件包将被安装：
  bc
共升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 5 个软件未被升级。
需要下载 73.1kB 的软件包。
After this operation, 201kB of additional disk space will be used.
获取：1 http://cn.archive.ubuntu.com hardy/main bc 1.06.94-3ubuntu1 [73.1kB]
下载 73.1kB，耗时 55s (1310B/s)
选中了曾被取消选择的软件包 bc。
(正在读取数据库 ... 系统当前总共安装有 53291 个文件和目录。)
正在解压缩 bc (从 .../bc_1.06.94-3ubuntu1_i386.deb) ...
正在设置 bc (1.06.94-3ubuntu1) ...
```

```
root@ema3530:/home/ema# sudo ./mkcard.sh /dev/sdb
1024+0 records in
1024+0 records out
1048576 bytes (1.0 MB) copied, 0.652779 s, 1.6 MB/s
Disk /dev/sdb doesn't contain a valid partition table
DISK SIZE - 1995440128 bytes
CYLINDERS - 242
Checking that no-one is using this disk right now ...
```



```

OK
Disk /dev/sdb: 242 cylinders, 255 heads, 63 sectors/track
sfdisk: ERROR: sector 0 does not have an msdos signature
/dev/sdb: unrecognized partition table type
Old situation:
No partitions found
New situation:
Units = cylinders of 8225280 bytes, blocks of 1024 bytes, counting from 0
   Device Boot Start   End  #cyls  #blocks  Id System
/dev/sdb1 *    0+    8    9-   72261   c  W95 FAT32 (LBA)
/dev/sdb2      9   241   233  1871572+ 83  Linux
/dev/sdb3      0    -    0     0 0 Empty
/dev/sdb4      0    -    0     0 0 Empty
Successfully wrote the new partition table
Re-reading the partition table ...
If you created or changed a DOS partition, /dev/foo7, say, then use dd(1)
to zero the first 512 bytes: dd if=/dev/zero of=/dev/foo7 bs=512 count=1
(See fdisk(8).)
mkfs.vfat 2.11 (12 Mar 2005)
mke2fs 1.40.8 (13-Mar-2008)
Filesystem label=rootfs
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
117120 inodes, 467893 blocks
23394 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=482344960
15 block groups
32768 blocks per group, 32768 fragments per group
7808 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 35 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.

```

- 虚拟机断开 SD 卡连接随即又重新连接上，用“df”命令就可以看到已经分好两个分区。

```

root@ema3530:/home/ema# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1        60924160  7115212  50738528  13% /
varrun          257720      88   257632    1% /var/run
varlock         257720       0   257720    0% /var/lock
udev            257720      56   257664    1% /dev
devshm          257720       0   257720    0% /dev/shm
.host:/         62468720 62430096   38624 100% /mnt/hgfs
/dev/sdb1         71133       1    71133    1% /media/boot
/dev/sdb2       1856764    35568  1727620    3% /media/rootfs

```



6.2.2. SD 卡的系统镜像更新

- 复制启动系统文件（具体 Demo 操作步骤可查看 4.3）

```
# sudo cp MLO /media/boot
# sync
# sudo cp u-boot.bin /media/boot
# sudo uImage /media/boot
# sudo cp -a rootfs/* /media/rootfs
```

- 卸载 SD 卡并等待“boot”和“rootfs”图标消失并且读卡器上指示灯不再闪烁。

6.2.3. NAND Flash 系统映像更新

- **NAND FLASH 分区：**

```
* 0x00000000-0x00080000 : "X-Loader"
* 0x00080000-0x00260000 : "U-Boot"
* 0x00260000-0x00280000 : "U-Boot environment data"
* 0x00280000-0x00680000 : "Kernel"
```

- **对于 128MB 的核心板：**

```
* 0x00680000-0x08000000 : "File System"
```

- **对于 256MB 的核心板：**

```
* 0x00680000-0x10000000 : "File System"
```

- **下面的指南是使用 SD 卡引导将镜像写入到 NAND FLASH**

准备：

- (1) 准备一个可启动的 SD 卡。
- (2) 请确保以下文件在 FAT32 分区的 SD 卡里面：



MLO (X-Loader)

u-boot.bin (U-Boot)

uImage (Linux kernel image)

ubi.img (UBIFS file system image)

上面列出的文件可以从这里下载 <http://code.google.com/p/ema3530/downloads/list>

- X-Loader 是第一阶段引导加载程序，使用下面的命令烧写 X-Loader 到 NAND

FLASH：

```
OMAP3 Stalker # mmc init
OMAP3 Stalker # fatload mmc 0:1 80000000 MLO
OMAP3 Stalker # nandeccl hw
OMAP3 Stalker # nand erase 0 80000
OMAP3 Stalker # nand write.i 80000000 0 80000
```

- U-Boot 是第二阶段引导加载程序，使用下面的命令烧写 U-Boot 到 NAND

FLASH：

```
OMAP3 Stalker # mmc init
OMAP3 Stalker # fatload mmc 0:1 80000000 u-boot.bin
OMAP3 Stalker # nandeccl sw
OMAP3 Stalker # nand erase 80000 160000
OMAP3 Stalker # nand write.i 80000000 80000 160000
```

- 使用下面的命令烧写内核镜像到 NAND FLASH：

```
OMAP3 Stalker # mmc init
OMAP3 Stalker # fatload mmc 0:1 80000000 uImage
OMAP3 Stalker # nandeccl sw
OMAP3 Stalker # nand erase 280000 400000
OMAP3 Stalker # nand write.i 80000000 280000 400000
```

- 我们使用的文件系统 UBIFS。使用下面的命令加载文件系统镜像到 RAM。这里举

例的文件系统镜像名是：ubi.img

```
OMAP3 Stalker # mmc init
OMAP3 Stalker # fatload mmc 0:1 84000000 ubi.img
OMAP3 Stalker # nandeccl sw
```



- 对于 128MB 的核心板，使用下面的命令擦除文件系统分区：

```
OMAP3 Stalker # nand erase 680000 8000000
```

- 对于 256MB 的核心板，使用下面的命令擦除文件系统分区：

```
OMAP3 Stalker # nand erase 680000 10000000
```

- 烧写文件系统镜像到 NAND FLASH。这里举例的文件系统镜像的大小是 0xD40000 (Bytes)，具体大小可根据实际镜像大小来调整:

```
OMAP3 Stalker # nand write.i 84000000 680000 D40000
```

当所有上述工作完成后就可以关掉电源，把 SD 卡拔出来，设置好拨码开关（111100），重新上电启动开发板，在 U-BOOT 里设置好传给内核的参数就可以从 NAND FLASH 启动。

6.3. Linux 系统操作

6.3.1. 设置开机自动运行程序

- ◆ 在/etc/init.d/rcS 文件里添加一个的 shell 脚本，开机就会执行这个脚本，例如 set_fb。

```
#!/bin/sh
#
# rcS      Call all S??* scripts in /etc/rcS.d in
#          numerical/alphabetical order.
#
# Version:  @(#) /etc/init.d/rcS 2.76 19-Apr-1999 miquels@cistron.nl
#
PATH=/sbin:/bin:/usr/sbin:/usr/bin
runlevel=S
prevlevel=N
umask 022
export PATH runlevel prevlevel
/etc/init.d/set_fb &
```



- ◆ 在/etc/init.d/set_fb 文件里加入执行 3D Demo 的演示程序。

```
#!/bin/sh
sleep 5
echo "fbset success"
VYRES="$(cat /sys/devices/platform/omapfb/graphics/fb0/virtual_size |awk -F, '{print$2*3}')"
echo $VYRES
fbset -vyres $VYRES
/etc/init.d/rc.pvr start
cd /opt/gfxsddemos/ogles
./OGLES Vase
```

6.3.2. 屏蔽显示器的登陆界面

启动开发板，进入系统后，在串口终端执行下面的命令。

```
root@beagleboard:~# opkg remove psplash-support -force-depends
root@beagleboard:~# cd /etc/rc5.d/
root@beagleboard:/etc/rc5.d# ls
S10dropbear S20syslog S30ntpd S99rmnologin
S20apmd S21avahi-daemon S50usb-gadget
S20dbus-1 S22connman S99gpe-dm
root@beagleboard:/etc/rc5.d# mv S99gpe-dm bakS99gpe-dm
```

6.3.3. 下载安装软件

首先确认开发板的网络能正常上网，输入“route”命令查看默认网关有没有设置好。

```
root@beagleboard:~# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.1.0 * 255.255.255.0 U 0 0 0 eth0
default 192.168.1.1 0.0.0.0 UG 0 0 0 eth0
```

如果没有设置，就用下面的命令重新获取。

```
root@beagleboard:~# udhcpc
```

下载安装软件，这里举例安装 mplayer。

```
root@beagleboard:~# opkg update
root@beagleboard:~# opkg install mplayer
```



6.3.4. ALSA 声音设置

ALSA 播放声音设置，命令：alsamixer。

功能键：Tab 、 space 空格、左右方向、上下键调值、m 设是否静音（静音显示 MM)

用 TAB , 在 VIEW 项内部选择：PLAYBACK , CAPTURE , ALL (ALL 项，其实是 PLAYBACK 和 CAPTURE 两个图的综合)

1. 音频输出设置

用 TAB , 在 VIEW 项内部选择：PLAYBACK

启用下面的选项（按键盘“m”，显示“00”就表示启用了该选项“MM”是表示静音）。

- ✧ DAC2 Analog (上下键调值)
- ✧ DAC2 Digital coarse (上下键调值)
- ✧ DAC2 Digital Fine (上下键调值)
- ✧ PreDriv
- ✧ PredriveL Mixer AudioL2
- ✧ PredriveR Mixer AudioL2



2. 音频输入设置（是在音频输出设置上增加选项，测试是用麦克风录音之后通过耳机输出）

1) 用 TAB，在 VIEW 项内部选择：CAPTURE

用 space 来选择某项的是否加 CAPTURE，出现红色的 L 和 R,就是加上了，这里选择

Analog Right Sub Mic 就可以了。

2) 用 TAB，在 VIEW 项内部选择：PLAYBACK

都启用下面的选项（按键盘“m”，显示“00”就表示启用了该选项“MM”是表示静音）。

✧ DAC Voice Analog Downlink（上下键调值）

✧ PredriveL Mixer Voice

✧ PredriveR Mixer Voice

✧ Voice Analog Loopback

6.3.5. Linux 应用程序开发一般流程

1) 编写 Hello.c 的程序

```
#include<stdio.h>

main()

{

    printf("Helloworld!\n");

}
```



2) 交叉编译

```
# arm-none-linux-gnueabi-gcc Hello.c -o Hello
```

3) 下载运行

可以将可执行文件放在 SD 卡 EXT3 分区的文件系统；

可以将可执行文件直接放在 U 盘；

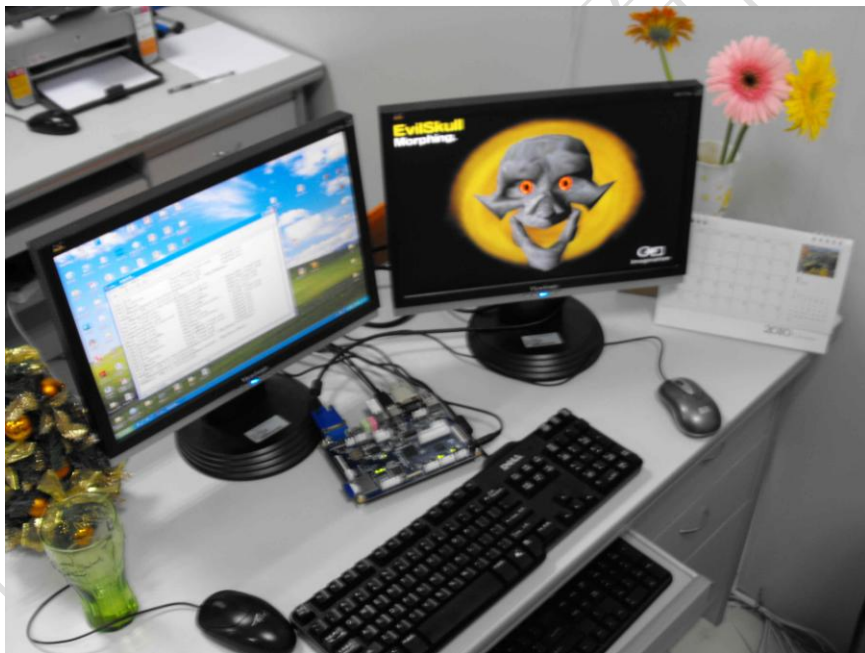
可以将可执行文件通过网络下载到开发板上，[请查看 5.4 挂载网络文件系统 NFS](#)；

可以将可执行文件通过软件下载到开发板上，例如 SSH, WinSCP3。

```
# ./Hello
```

6.4. Demo 系统运行效果

本系统提供的Demo程序为基于Angstrom的Linux桌面系统。透过DVI信号输出的界面(分辨可达1080p)。可使用Angstrom系统中的各种软件，包括文档编辑，上网浏览，音频视频播放及图形编辑等功能，并可用USB键盘鼠标对此系统进行操作控制。







第七章

7. 附录



7.1. 原理图及核心板的接口功能

- 请参看 DEV3530 系统附带光盘中
- DEV3530 原理图.pdf。

7.2. 相关服务

现在登陆广州英码信息科技有限公司（EMA）官方网站 www.ema-tech.com 进行注册

登记的用户，即可享受 EMA 产品软件升级服务。

7.3. 相关链接

✧ 广州英码信息科技有限公司

- <http://www.ema-tech.com>

✧ 美国德州仪器公司

- <http://www.ti.com>

✧ WIKI

- <http://code.google.com/p/ema3530/>