

SBC35x 用户手册

公司: *EMA*

發行版本: *v3.0*

发布时间: *08/20/2010*

作者: *Vincent*



公司简介

广州英码信息科技有限公司（EMA）是一家提供专业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的嵌入式解决方案，在售前售后技术支持，用户培训和系统维护等各种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。产品应用领域已广泛覆盖消费电子，医疗仪器，无线通讯，仪器仪表，节能电子，汽车电子等行业，成为中国广大电子企业客户最满意的合作伙伴。

产品定制服务

- 1) 提供基于 SOM35x 的产品定制服务；
- 2) 提供基于 OMAP3 的产品定制服务；
- 3) 提供 Linux、Android 和 WinCE 应用开发服务；
- 4) 提供芯片“一站式采购”服务；
- 5) 提供 OMAP3 应用培训服务。



帮助客户快速完成产品设计，实现产品的快速上市，快速抢占市场。如果您需要了解更多的关于产品定制服务信息，请跟英码市场部联系，电话：+86-20-61230220 或者 +86-18922754967，或者 email：omap@ema-tech.com

产品订购

如果您想订购英码 SOM35x 系统核心模块或者配套 EVM35x 主板，请跟市场部联系：+86-20-61230220/61230221，或者手机：18924191364、18924191634、18922754967，电子邮件：sales@ema-tech.com

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

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. SBC35X 单板机介绍	7
1.1. SBC35x 简介	7
1.2. SBC35x 的型号	8
2. SBC35X 的硬件介绍	9
2.1. SBC35x 的硬件规格	9
2.1.1. 框图	9
2.1.2. SBC35x 的硬件参数	10
2.1.3. SBC35x 工作环境	12
2.1.4. SBC35x 机构资料	12
2.2. SBC35x 的接口说明	13
2.2.1. 接口一览表	13
2.2.2. 按键/开关一览表	14
2.2.3. LED 一览表	14
2.3. 接口详述	14
2.4. SBC35x 接口引脚定义	19
3. SBC35X 的基本使用	25
3.1. 以 SBC35x 为核心的计算机系统	25
3.2. SBC35x 连接设置	26
3.2.1. 外部接口连接	26
3.2.2. SBC35x 系统上电步骤	28
3.2.3. SBC35x 系统关闭步骤	29
4. SBC35X 的 LINUX 系统使用与设置	30
4.1. 系统启动方法和过程	30
4.1.1. 系统启动过程:	30
4.1.2. SD 卡启动过程	31
4.1.3. Nandflash 启动过程	32
4.2. 显示设置	32
4.2.1. VGA 接口显示器显示:	33



4.2.2.	4.3 寸触摸屏显示:	33
4.2.3.	7 寸触摸屏显示:	33
4.3.	DEMO 软件演示	34
4.3.1.	3D Demo 演示	34
4.3.2.	Quake3 游戏演示	35
4.3.3.	DVSDK 演示	36
4.3.4.	视频输入演示	37
4.3.5.	USB 摄像头演示	40
4.3.6.	播放视频演示	43
5.	构建嵌入式 LINUX 的软件开发环境	44
5.1.	LINUX 软件开发环境概述	44
5.2.	构建软件开发环	45
5.2.1.	VMware-workstation-6.5.0 的安装	45
5.2.2.	Ubuntu 的安装	48
5.2.3.	虚拟机跟主机的文件共享设置	52
5.2.4.	构建交叉编译器	54
5.3.	串口终端软件的使用	55
5.4.	挂载网络文件系统 NFS	56
6.	构建 SBC35X 的软件开发环境	58
6.1.	系统编译	58
6.1.1.	一级启动代码 x-loader 编译	58
6.1.2.	二级启动代码 u-boot 编译	59
6.1.3.	内核编译	60
6.1.4.	制作文件系统镜像	61
6.2.	LINUX 系统镜像更新	62
6.2.1.	制作 SD 系统启动卡	62
6.2.2.	SD 卡的系统镜像更新	64
6.2.3.	NAND Flash 系统映像更新	64
6.3.	LINUX 系统操作	66
6.3.1.	设置开机自动运行程序	66
6.3.2.	屏蔽显示器的登陆界面	67
6.3.3.	下载安装软件	67



6.3.4.	ALSA 声音设置.....	68
6.3.5.	Linux 应用程序开发一般流程.....	69
6.4.	DEMO 系统运行效果	70
7.	附录	73
7.1.	原理图及核心板的接口功能.....	73
7.2.	相关服务	73
7.3.	相关链接	73



第一章

1. SBC35x 单板机介绍



1.1. SBC35x 简介

SBC35x 是广州英码信息科技有限公司 (EMA) 自主研发的一款基于 SOM35x 的评估板, 在设计上兼容 EPIC 标准, 并使用可扩展的结构, 帮助客户充分利用 OMAP35x 处理器的优点, 包括超标量的 ARM Cortex-A8 RISC core、C64x+ DSP、SGX 2D/3D 图形加速引擎。

SBC35x 帮助用户围绕 OMAP35x 构建自己的产品, 是用户理想的产品软件开发平台。分辨率可达 1280×720, 支持多种显示输出, 例如 VGA、LCD、S-Video 和 AV 复合视频输出。支持多种标准的 30FPS 的 D4(1280×720p) 视频编解码。支持 Linux、Android、WinCE 等多种嵌入式操作系统。

SBC35x 针对 Linux 平台提供 X-Loader、U-boot、Linux Kernel (V2.6.32) 和 UBIFS 文件系统。对 TI 发布的 PSP 软件套件进行优化改进, 并提供开发工具等, 是用户开发自己应用产品的理想平台。针对 Android 操作系统进行优化处理, 提供完善的系统和常见的应用软件, 是实现便携式的移动互联网设备的理想平台。目前基于 SBC35x 实现的产品有移动互联网设备 MID、全球定位系统 (GPS)、2D/3D 游戏机产品、医疗设备、人机界面等。



1.2. SBC35x 的型号

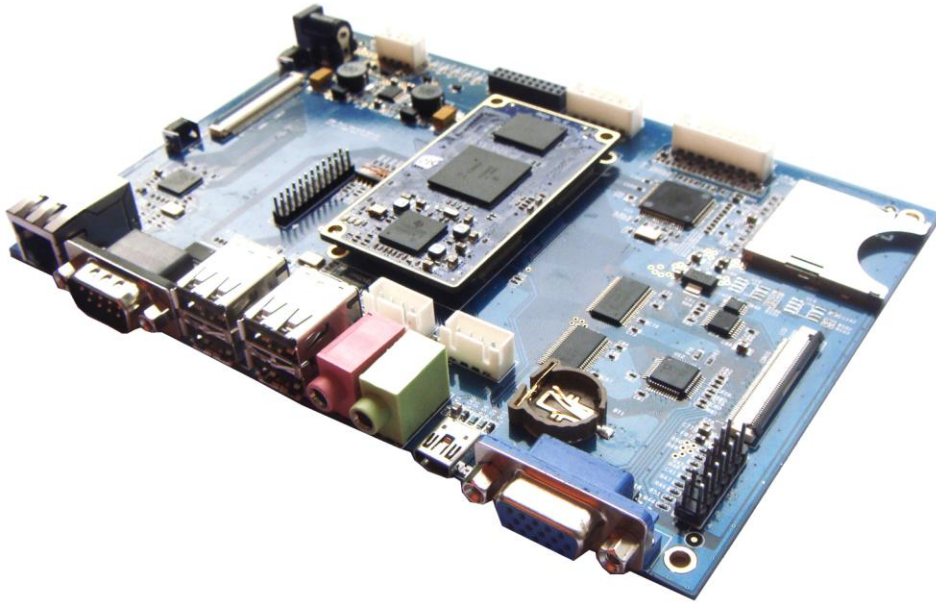


图 1.1 SBC35x

评估板（开发板）型号列表：

型号	处理器	频率 (MHz)	SDRAM (MB)	NAND Flash (MB)	温度	视频 输入	WIFI/ BT
SBC3530-B1- 1880-LUAC0	OMAP 3530	600	256	256	0℃- 70℃	Y	N
SBC3530-B1- 1880-LUEC0	OMAP 3530	600	256	256	0℃- 70℃	Y	Y



第二章

2. SBC35x 的硬件介绍



2.1. SBC35x 的硬件规格

2.1.1. 框图

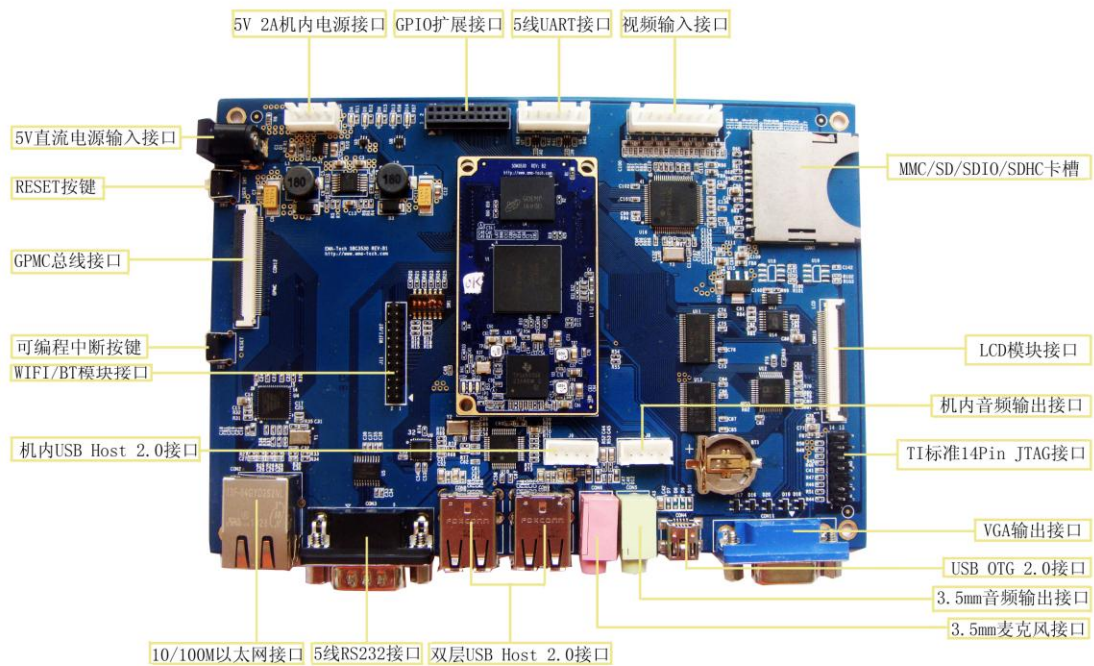


图 2.1 SBC35x 框图



2.1.2. SBC35x 的硬件参数

SOM3530	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 3.3V 5 线 UART (TX,RX,CTS,RTS,GND)	
USB Host	4x USB 2.0 高速 Host	
	1x 机内 USB 2.0 高速 Host	
USB OTG	1x USB 2.0 高速 OTG (可用作 Device)	
JTAG	1x 14Pin TI 标准 JTAG	
音频	1x 3.5mm 音频输出接口	
	1x 机内音频输出接口	
	1x 3.5mm 麦克风输入接口	
VGA	标准 VGA 输出接口	
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	



视频输入	4 通道视频输入 最大支持： 4 路 CVBS 或 1 路 CVBS、1 路 YPbPr 或 1 路 CVBS、1 路 RGB（内置复合同步信号） 或 2 路 CVBS、1 路 S-Video 或 2 路 S-Video CVBS 可支持 NTSC/PAL/SECAM 制式
SD 卡槽	MMC/SD/SDIO/SDHC 卡槽，最高支持 32GByte 容量
铁电存储器（可选）	64Kbit/128Kbit/256Kbit/512Kbit 无限次擦写高速铁电存储器
安全与加密（可选）	内置硬件加密与硬件验证机制
RTC	内置 RTC 电池座，使用 CR1220 电池供电
按钮	1x 可编程用户中断按钮
	1x RESET 按钮
LED 指示灯	3x 电源指示灯
	2x 可编程指示灯
启动模式选择开关	1x 6 位启动模式选择拨码开关
电源接口	1x 5V 2A DC 电源接口
	1x 5V 2A 机内电源接口
扩展接口	1x 3.3V LCD 模块接口（支持触摸）
	1x 1.8V WIFI 模块接口
	1x 1.8V GOIO 扩展接口
	1x 1.8V GPMC 高速总线接口



2.1.3. SBC35x 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 3.5A (备注 1)
工作功耗	1.5W	2.5W (备注 2)	17.5W (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：该值为 4 路 USB 接口全部对外供电，并且接上 LCD 模块、数字摄像头、WIFI 等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. SBC35x 机构资料

设计目的	单板机
PCB 尺寸	4.528 x 6.496 英寸 (115.00 x 165.00 mm)
兼容标准	EPIC 标准
安装孔数目	4 个

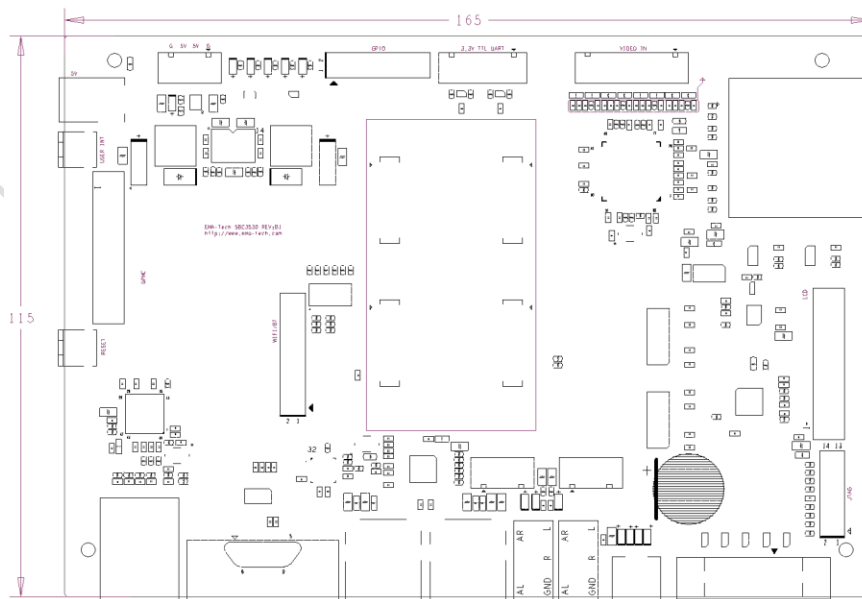


图 2.2 PCB 尺寸图



2.2. SBC35x 的接口说明

2.2.1. 接口一览表

接口序号	接口描述	备注
CON1	5V 直流电源输入接口	DC-208 接口，内正外负，最大电流 2A
CON2	10/100M 以太网接口	高性能以太网
CON3	5 线 RS232 接口	
CON4	USB OTG 2.0 接口	
CON5	3.5mm 音频输出接口	
CON6	3.5mm 麦克风接口	
CON7	MMC/SD/SDHC/SDIO 卡座	3V/1.8V 兼容
CON8	双层 USB Host 2.0 接口	
CON9	双层 USB Host 2.0 接口	
CON10	LCD 模块接口	50Pin 扁平排线接口
CON11	VGA 输出接口	
CON12	GPMC 总线接口	50Pin 扁平排线接口
BT1	RTC 电池座	使用 CR1220 电池
J1~J4	SOM35X 接口	
J5	5V 2A 机内电源接口	
J6	5 线 UART 接口	3.3V I/O
J7	TI 标准 14Pin JTAG 接口	
J8	机内音频输出接口	
J9	机内 USB Host 2.0 接口	
J10	视频输入接口	
J11	WIFI/BT 模块接口	
J12	GPIO 扩展接口	



2.2.2. 按键/开关一览表

按键序号	按键描述	备注
SW1	6 位启动选择拨码开关	
SW2	可编程中断按键	GPIO18
SW3	RESET 按键	

2.2.3. LED 一览表

LED 序号	LED 描述	备注
D4	5V 供电指示	
D5	SOM3530 供电指示	
D6	3.3V 供电指示	
D13	可编程用户指示灯	GPIO126
D14	可编程用户指示灯	GPIO127

2.3. 接口详述

1) +5V 直流电源输入插座 CON1

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 CON2

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 SW1 (备注：1：开关向上拨 0：开关向下拨)



通过拨码开关设置 SBC35x 启动设备的顺序。常用的拨码顺序，可查看 SBC35x 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		
---------	---	--------------------------	-------------------	--	--

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	SB (only on GP devices)	UART3 (only on GP devices)		



	monitoring ON (only for GP devices)				
--	---	--	--	--	--

4) 耳机输出插座 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 SW3

当按下并释放时，会使 SBC35x 进入上电复位状态，可以使开发板重新启动。

7) SD 卡插座 CON7

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 CON3

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) VGA 插座 CON11

标准 VGA 输出接口，支持所有 VESA 标准分辨率，SBC35x 可通过此端口连接液晶和 CRT 显示器。



10) LCD 接口座 CON10

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 J12

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 CON4

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) USB Host 2.0 接口 CON8/ CON9

4 个 USB 2.0 高速 Host（每个口提供标准 500mA 供电）

14) SDIO 接口/WIFI 扩展模块接口 J11

扩展插座提供部分可外接的扩展信号，用户在需要使用这些信号时，可用相应的插座取得这些信号来利用。

15) GPIO 扩展接口 J12

通用输入/输出接口 GPIO，每个 GPIO 端口可通过软件分别配置成输入或输出，引脚定义见 2.7.11 节

16) SPI 扩展接口 J12

串行外围设备接口 SPI，引脚定义见 2.7.14 节



17) 4 路视频输入接口 J10

4 通道视频输入最大支持：

4 路 CVBS 或 1 路 CVBS、1 路 YPbPr 或 1 路 CVBS、1 路 RGB (内置复合同步信号) 或 2 路 CVBS、1 路 S-Video 或 2 路 S-Video。

另外，CVBS 可支持 NTSC/PAL/SECAM 制式。

18) D4、D5 电源指示灯

当 SBC35x 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. SBC35x 接口引脚定义

J1~J4 SOM35X 接口

该接口用于接入 SOM35X 模块

J5 电池模块接口

该接口用于接入电池模块

引脚	功能描述	备注
1	地	
2	直流 5V 输入	
3	直流 5V 输入	
4	地	



J6 UART3 接口

该接口用于接入 UART3

引脚	功能描述	备注
1	3.3V I/O 参考电压输出	I/O 参考电压
2	地	
3	CTS	3.3V I/O
4	RTS	3.3V I/O
5	TX	3.3V I/O
6	RX	3.3V I/O

J7 OMAP35X JTAG 调试接口

该接口用于调试 OMAP35X 及其内置 DSP。

引脚	功能描述	备注
1	TMS	1.8V I/O
2	nTRST	1.8V I/O
3	TDI	1.8V I/O
4	地	
5	1.8V I/O 参考电压输出	I/O 参考电压
6	/	防误插引脚
7	TDO	1.8V I/O
8	地	
9	RTCK	1.8V I/O
10	地	
11	TCK	1.8V I/O



12	地	
13	EMU0	1.8V I/O
14	EMU1	1.8V I/O

J8 机内音频输出接口

该接口用于输出到额外的音频功放模块。

引脚	功能描述	备注
1	左声道输出	25mW 驱动力
2	左声道地	
3	右声道地	
4	右声道输出	25mW 驱动力

J9 机内 USB Host 接口

该接口用于接入 USB 机内 USB 设备。

注：该接口与 CON9 的 USB4 复用。

引脚	功能描述	备注
1	5V 电源输出	用于 USB 设备供电
2	DM	USB 差分数据（负）
3	DP	USB 差分数据（正）
4	地	



J10 视频输入接口

该接口用于视频输入

引脚	功能描述	备注
1	第 4 视频输入通道：CVBS/Y 输入	
2	模拟信号地	第 4 通道地
3	第 3 视频输入通道：CVBS/Pr/R/C 输入	
4	模拟信号地	第 3 通道地
5	第 2 视频输入通道：CVBS/Y/G 输入	
6	模拟信号地	第 2 通道地
7	第 1 视频输入通道：CVBS/Pb/B/C 输入	
8	模拟信号地	第 1 通道地

J11 WIFI 模块扩展接口

引脚	功能描述	备注
1	3.3V 电源输出	
2	地	
3	1.8V I/O 参考电压输出	I/O 参考电压
4	地	
5	SYS_nRESPWRON	
6	MMC2_CMD	1.8V I/O
7	MMC2_CLK	1.8V I/O
8	MMC2_DAT1	1.8V I/O
9	MMC2_DAT0	1.8V I/O
10	MMC2_DAT3	1.8V I/O



11	MMC2_DAT2	1.8V I/O
12	MMC2_DAT5/GPIO137	1.8V I/O
13	MMC2_DAT4/GPIO136	1.8V I/O
14	MMC2_DAT7/GPIO139	1.8V I/O
15	MMC2_DAT6/GPIO138	1.8V I/O
16	MCBSP3_DX/UART2_CTS	1.8V I/O
17	MCBSP3_FSX/UART2_RX	1.8V I/O
18	MCBSP3_CLKX/UART2_TX	1.8V I/O
19	MCBSP3_DR/UART2_RTS	1.8V I/O
20	MCBSP1_DR/GPIO159	1.8V I/O
21	MCBSP1_DX/GPIO158	1.8V I/O
22	MCBSP1_CLKX/GPIO162	1.8V I/O
23	MCBSP1_FSX/GPIO161	1.8V I/O
24	/	/

J12 GPIO 扩展接口

引脚	功能描述	备注
1	3.3V 电源输出	
2	地	
3	1.8V I/O 参考电压输出	I/O 参考电压
4	地	
5	SYS_CLKOUT1/GPIO10	1.8V I/O
6	SYS_nRESPWRON	1.8V I/O
7	SYS_CLKOUT2/GPIO186	1.8V I/O
8	MCBSP1_CLKR/GPIO156	1.8V I/O
9	HSUSB1_D0/MCSPI3_SIMO/GPIO14	1.8V I/O



10	MCBSP1_FSR/GPIO157	1.8V I/O
11	HSUSB1_D1/MCSPI3_SOMI/GPIO15	1.8V I/O
12	HSUSB1_CLK/GPIO13	1.8V I/O
13	HSUSB1_D2/MCSPI3_CS0/GPIO16	1.8V I/O
14	HSUSB1_STP/GPIO12	1.8V I/O
15	HSUSB1_D7/MCSPI3_CLK/GPIO17	1.8V I/O
16	地	
17	3V3_I2C3_SCL	3.3V I/O
18	AV/SVIDEO_Y	CVBS 输出/S-Video Y 输出
19	3V3_I2C3_SDA	3.3V I/O
20	SVIDEO_C	S-Video C 输出



第三章

3. SBC35x 的基本使用



3.1. 以 SBC35x 为核心的计算机系统

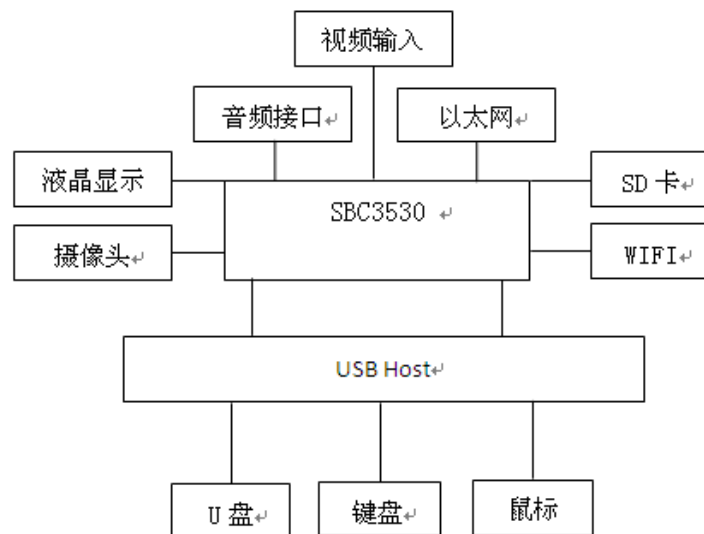


图 3.1

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) SD卡一块
- 6) 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. SBC35x 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SBC35x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SBC35x

取出SBC35x，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SBC35x插座CON3，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、VGA接口的液晶显示器。取出VGA视频电缆，将一端连接到SBC35x上CON11接口。再将另端插接到液晶显示器的VGA输入接口上。

5) 连接LCD触摸屏

取出LCD触摸屏，用屏线连接SBC35x CON10接口，屏线的金手指对应LCD触摸屏的驱动板接口和SBC35x 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SBC35x 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. SBC35x 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SBC35x电源

取出SBC35x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SBC35x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. SBC35x 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SBC35x系统的供电电源
- 4) 关闭外设电源



第四章

4. SBC35x 的 Linux 系统使用与设置



4.1. 系统启动方法和过程

SBC35x 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 SBC35x 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 SBC35x，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），并将三个镜像文件放到 SD 卡的 FAT32 分区里，将文件系统压缩包（evm_fs.tar.bz2）解压到 EXT3 分区里。
- 2) 连接硬件，请参考 **SBC35x 用户手册 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 **SBC35x 用户手册 5.3 串口终端软件的使用**。
- 4) 连接并接通 SBC35x 电源，请参考 **SBC35x 用户手册 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 SBC35x 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SBC35x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

SBC 35x 支持多种显示输出,系统默认是从 VGA 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. VGA 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 VGA、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 boardmodel SBC35X-B1-1880-LUAC0  
# setenv dvmode  
# setenv defaultdisplay lcd043
```

参数的意义是指不用设置 VGA 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 boardmodel SBC35X-B1-1880-LUAC0  
# setenv dvmode  
# setenv defaultdisplay lcd070
```

参数的意义是指不用设置 VGA 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 VGA 接口输出, 要想 LCD 输出显示, 可查看 4.2 节的设置)

```

-----
|      |      | | | | | | |
|  |  |-----|  |  |-----|
|      |  |__|  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|-----|-----|-----|
          |

```



- 用户名是：root，密码是空，直接按回车就可以登陆
- 进入到 opt/gfxsdkdemos/ogles 目录，执行大写“O”字母开头的文件。

4.3.2. Quake3 游戏演示

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

```

-----
|      |                                     .
|  |  |-----|-----||  -----|-----| | | | | | |
|      |  |  _  | --'|'--|  '|  |  |  |
|  |  |  |  |  |-- || --'|  |  |  |  |  |
|'-----|'-----|'-----"'-----|'-----|'-----|
          .

```



stalker login: root

- ```
root@stalker:~# cd /usr/lib/quake3/
root@stalker:/usr/lib/quake3# ./run.sh
```

- 执行运行脚本后就可以在 VGA 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 VGA 接口输出, 要想 LCD 输出显示, 可查看 4.2 节的设置)

```

OMAP3 Stalker # setenv mmcargs 'setenv bootargs console=${console} vram=${vram}
omapfb.mode=dvi:${dvimode} omapdss.def_disp=${defaultdisplay} root=${mmcroot} init=/init
mpurate=${mpurate} rootfstype=${mmccrootfstype} boardmodel=${boardmodel}
mem=99M@0x80000000 mem=128M@0x88000000'
OMAP3 Stalker # saveenv
Saving Environment to NAND...
Erasing Nand...
Erasing at 0x260000 -- 100% complete.
Writing to Nand... done
OMAP3 Stalker # boot
Starting Connection Manager
.-----
| | .-
		-----	-----			-----	-----					
		__	---	'-	.-'							
					---		-'			'		
'-----	'-----	-----"-----	'-----	'-----								

```



```

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。
- 先执行加载脚本，然后再解码播放视频。

```

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

```

#### 4.3.4. 视频输入演示

视频输入演示是播放 DVD 机的光盘，通过 SBC35x 开发板的 VGA 接口显示出来。需  
要将 DVD 机的 CVBS 信号和地信号引出两个条线接在 SBC35x J8 接口的引脚 7 和引脚 8，  
默认的程序通道号是选择 0，其他的程序通道号对应的模拟通道和视频模式选择如下表所  
示：

模拟通道和视频模式选择

| 模式      | 输入选择                    | 程序通<br>道号 | Cvbs | Cvbs<br>GND | S-video<br>Y | S-video<br>C | S-video<br>GND |
|---------|-------------------------|-----------|------|-------------|--------------|--------------|----------------|
| CVBS    | VI_1_A(default)         | 0         | PIN7 | PIN8        | /            | /            | /              |
|         | VI_2_A                  | 1         | PIN5 | PIN6        | /            | /            | /              |
|         | VI_3_A                  | 2         | PIN3 | PIN4        | /            | /            | /              |
|         | VI_4_A                  | 3         | PIN1 | PIN2        | /            | /            | /              |
| S-video | VI_2_A(Y),<br>VI_1_A(C) | 4         | /    | /           | PIN5         | PIN7         | PIN8           |
|         | VI_2_A(Y),<br>VI_3_A(C) | 5         | /    | /           | PIN5         | PIN3         | PIN4           |



|  |                         |   |   |   |      |      |      |
|--|-------------------------|---|---|---|------|------|------|
|  | VI_4_A(Y),<br>VI_1_A(C) | 6 | / | / | PIN1 | PIN7 | PIN8 |
|  | VI_4_A(Y),<br>VI_3_A(C) | 7 | / | / | PIN1 | PIN3 | PIN2 |

**视频输入 DEMO 编译步骤（下面操作都是在虚拟机 Linux 系统下进行）：**

- a) 复制 analog\_vi\_demo.tar.bz2 到/home/ema/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 analog_vi_demo.tar.bz /home/ema/
```

- b) 解压 analog\_vi\_demo.tar.bz2 压缩到当前文件夹

```
ema@ema3530:~ $ cd /home/ema/Source
ema@ema3530:~ $ sudo tar analog_vi_demo.tar.bz
ema@ema3530:~ $ cd analog_vi_demo/video
```

- c) make clean 命令清除以前编译产生的文件

```
ema@ema3530:~/analog_vi_demo/video$ make clean
Cleaning binaries...
```

- d) 接着配置和编译，在 Makefile 文件里修改对应自己交叉编译工具链库文件的路径

```
LDFLAGS = -L/usr/local/arm/arm-2009q1/arm-none-linux-gnueabi/libc/lib/
```

在 Rules.make 文件里修改自己内核源码的路径和交叉编译工具链的路径

```
KERNEL_DIR = /home/ema/linux-03.00.01.06
```

```
CROSS_COMPILE = /usr/local/arm/arm-2009q1/bin/arm-none-linux-gnueabi-
```



```
ema@ema3530:~/analog_vi_demo/video $ make
```

- e) 编译完成后，bin 目录下会生成很多可执行文件，将 saMmapLoopback 复制出来，  
放到开发板系统上运行。

```
ema@ema3530:~/analog_vi_demo/video /bin$ cp saMmapLoopback /mnt/hgfs/share
```

saMmapLoopback.c 这个程序采用的是默认的程序通道号“0”，如果想修改其他的程序通道号，可以在 main()函数里，把“capt\_input = 0”改成对应的程序通道号，保存退出。重新编译一次。

#### 开发板运行步骤：

- 上电启动开发板，输入用户名和密码进入文件系统。

```
Starting Connection Manager
.,-----.,
| | |..
		-----	-----			-----	-----					
		__	---	'---	.'							
				---		-'			'			
'|-----|-----| |-----|-----|
 -' |
 '---'

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。



- 执行“saMmapLoopback”。

```
root@beagleboard:~# ./saMmapLoopback
[44.942169] tvp514x 2-005d: tvp5146m2 found at 0xba (OMAP I2C adapter)

Capture: Opened Channel
Capture: Current Input: COMPOSITE
Capture: Input changed to: COMPOSITE
Capture: Current standard: NTSC
Capture: Capable of streaming
Capture: Number of buffers request = 2
Capture: Number of requested buffers = 2
Capture: Init done successfully

Display: Opened Channel
Display: Capable of streaming
Display: Number of buffers request = 2
Display: Number of requested buffers = 3
Display: Init done successfully

Display: Stream on...
Capture: Stream on...
```

#### 4.3.5. USB 摄像头演示

USB 摄像头演示,需要重新制作一个 SD 卡系统,将 Camera\_FS 文件夹的 MLO,u-

boot.bin,uImage 放在 SD 卡的 FAT32 分区, camera\_fs.tar.bz2 解压到 SD 卡的 EXT3 分区。

(可查看用户手册 6.2. Linux 系统镜像更新)





- [illegible]

- ```
root@beagleboard:~# cd /dev
```

- ```
root@beagleboard:~# cd mjpg-streamer-r63
root@beagleboard:~/mjpg-streamer-r63# vi mjpg-streamer-r63.sh
```



```
./mjpg_streamer -o "output_http.so -w `pwd`/www -p 8080" -i "input_uvc.so -d /dev/video0 -r 640x480 -f 30" &
```

```
./mjpg_streamer -o "output_http.so -w `pwd`/www -p 8081" -i "input_uvc.so -d /dev/video2 -r 640x480 -f 5" &
```

8080 和 8081 是端口号，/dev/video0 和 /dev/video2 是 USB 摄像头对应的设备文件名，

640x480 是分辨率。这个脚本是同时测试两个摄像头。如果测试一个，可以屏蔽一个。

➤ 执行 mjpg-streamer.sh 脚本

```
root@beagleboard:~/mjpg-streamer-r63# ./mjpg-streamer-r63.sh
root@beagleboard:~/mjpg-streamer-r63# MJPG Streamer Version.: 2.0
i: Using V4L2 device.: /dev/video2
i: Desired Resolution: 640 x 480
i: Frames Per Second.: 5
i: Format.....: MJPEG
o: www-folder-path...: /home/root/mjpg-streamer-r63/www/
o: HTTP TCP port.....: 8081
o: username:password.: disabled
o: commands.....: enabled
```

➤ 连接网线，查看开发板 IP。

```
root@beagleboard:~/mjpg-streamer-r63# ifconfig
eth0 Link encap:Ethernet HWaddr 02:00:39:BC:00:04
 inet addr:192.168.1.180 Bcast:255.255.255.255 Mask:255.255.255.0
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:3219 errors:0 dropped:0 overruns:0 frame:0
 TX packets:57 errors:0 dropped:0 overruns:0 carrier:0
```





## 第五章

### 5. 构建嵌入式 Linux 的软件开发环境



#### 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SBC35x系统的开发环境。在这一章中概述了装载SBC35x系统相关的软件到开发主机所需要的步骤。你将需要一张SBC35x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，SBC35x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SBC35x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。

SBC35x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

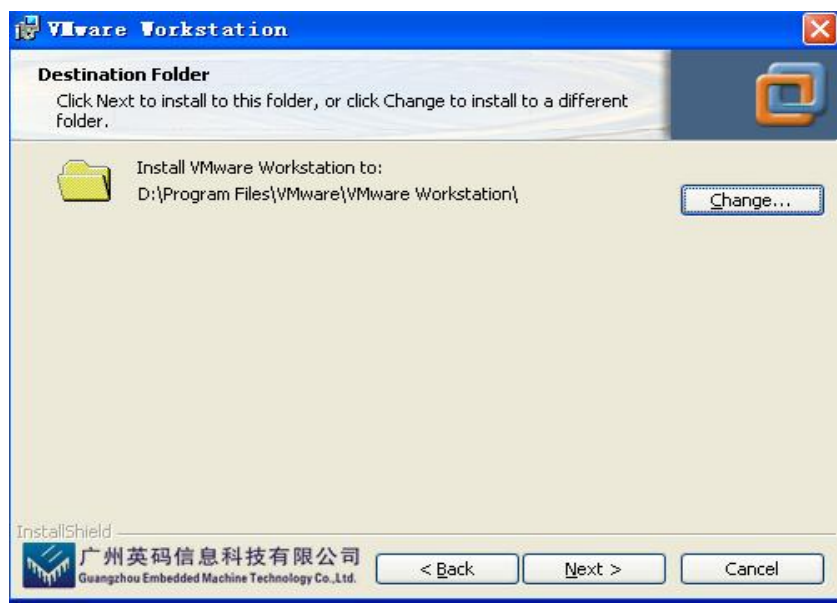


图 4.3

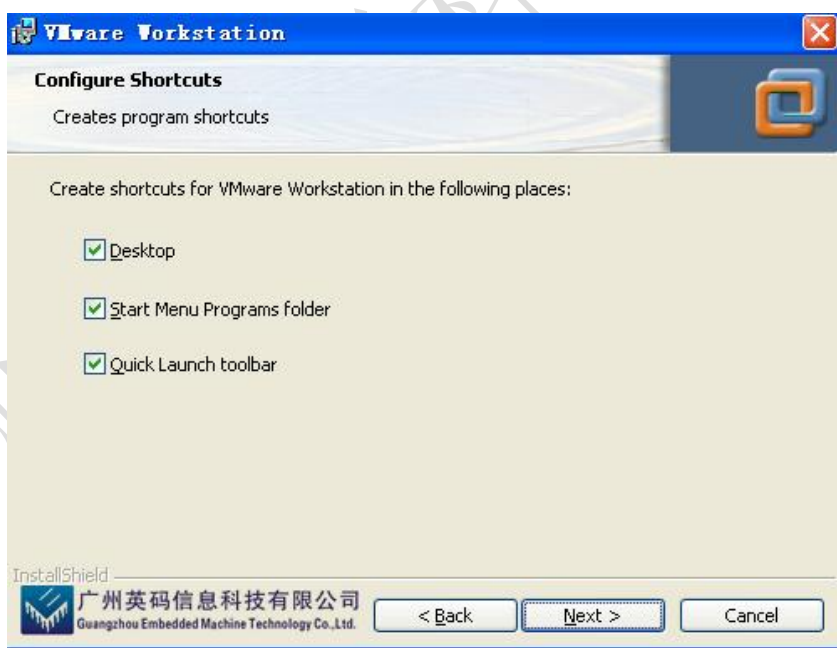


图 4.4

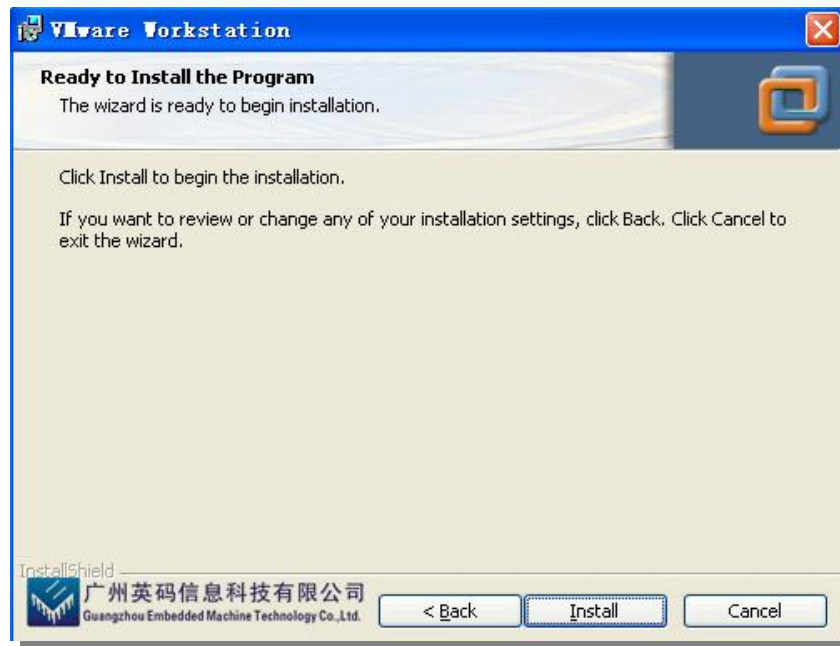


图 4.5

4. 单击 install 后，安装开始启动如图 4.6 所示

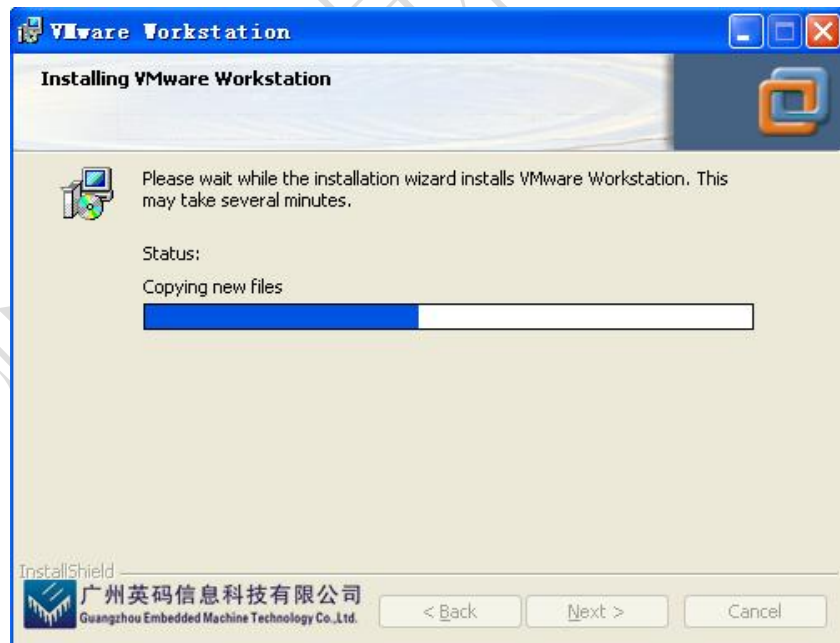


图 4.6



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

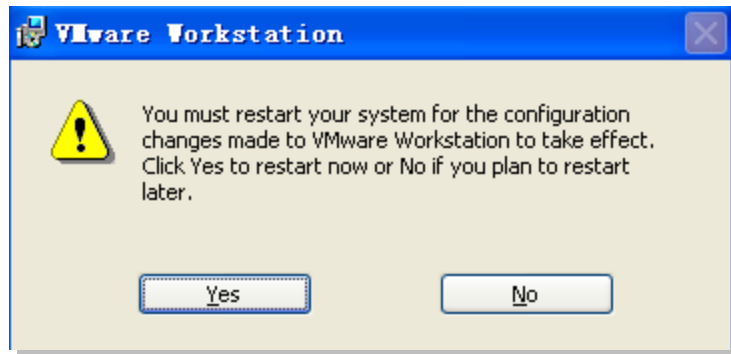


图 4.7

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

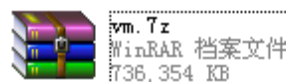


图 4.9





2) 打开 VMware-workstation 图标 ( 如 4.10 )

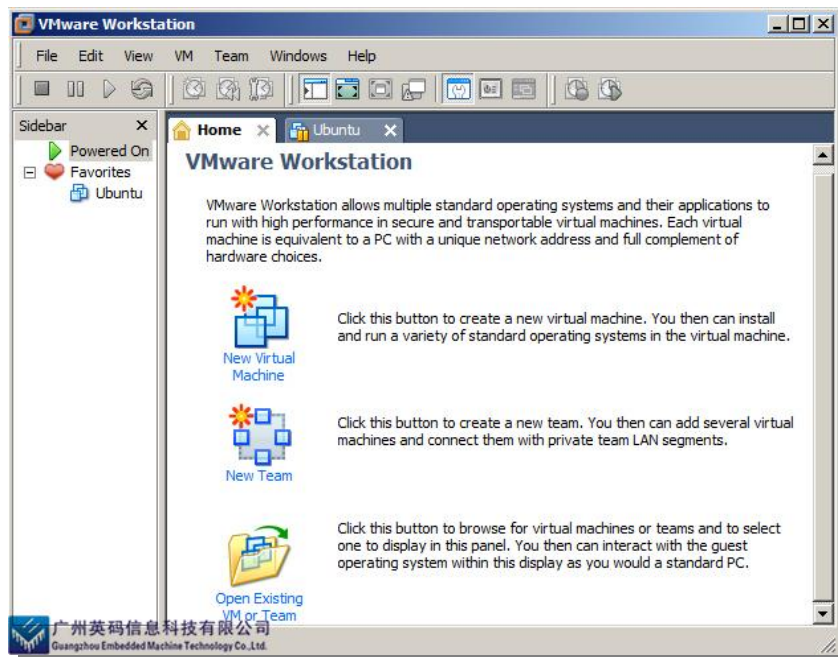


图 4.10

3) 打开 File 下的 Open 如图 4.11 所示

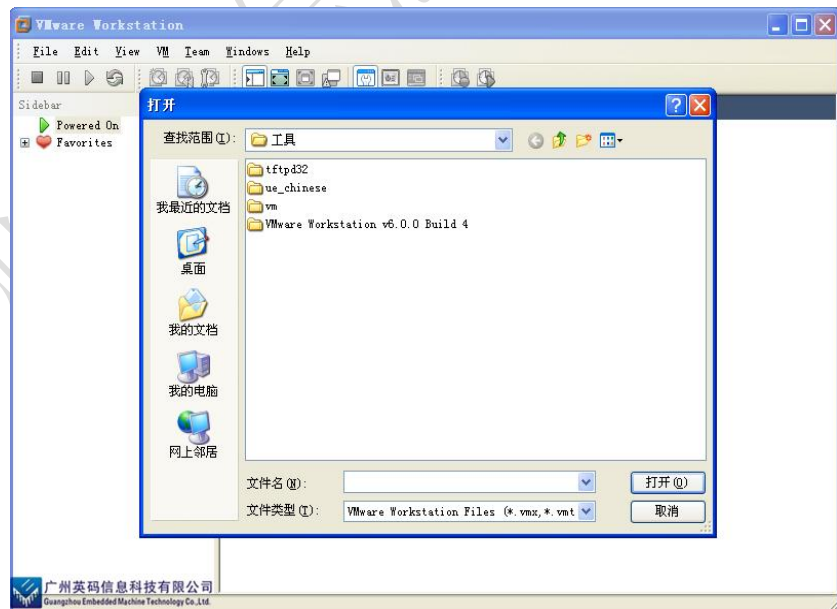


图 4.11



4) 选择刚刚解压的 vm 文件如图 4.12 所示

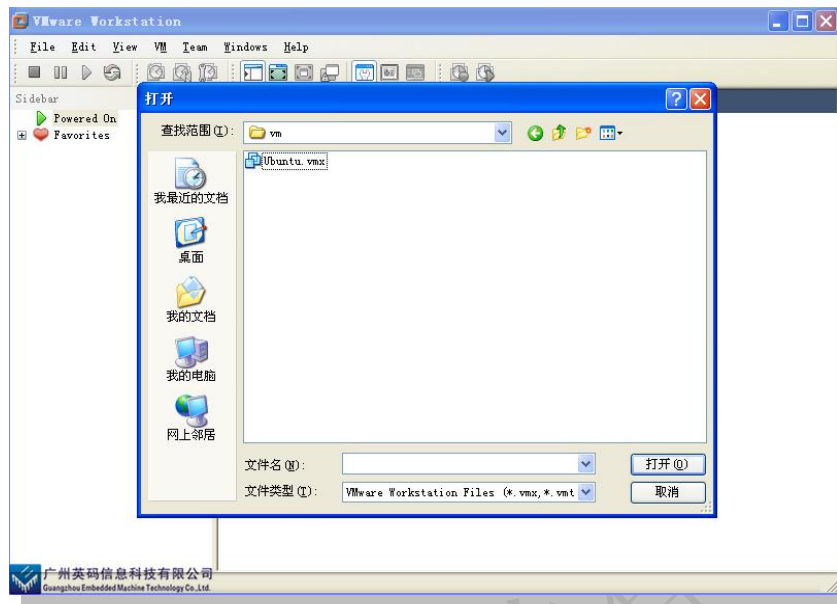


图 4.12

5) 选择左上角的绿色三角，启动 ubuntu 如图 4.13 所示

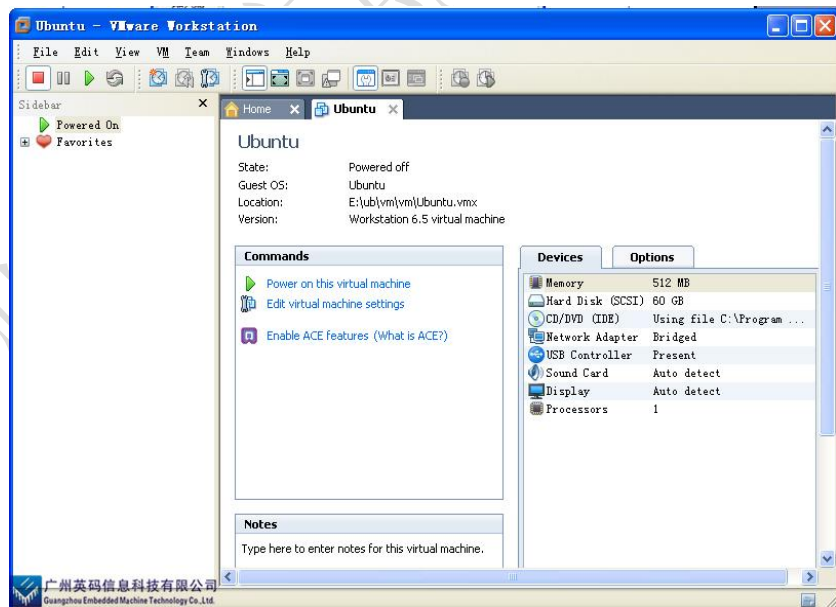


图 4.13



6) 登陆 ubuntu 如图 4.14 所示

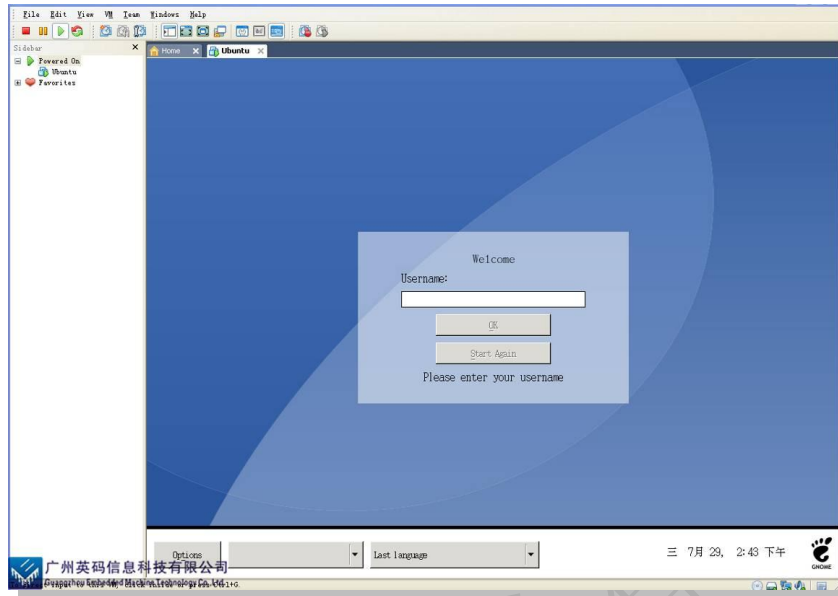


图 4.14

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

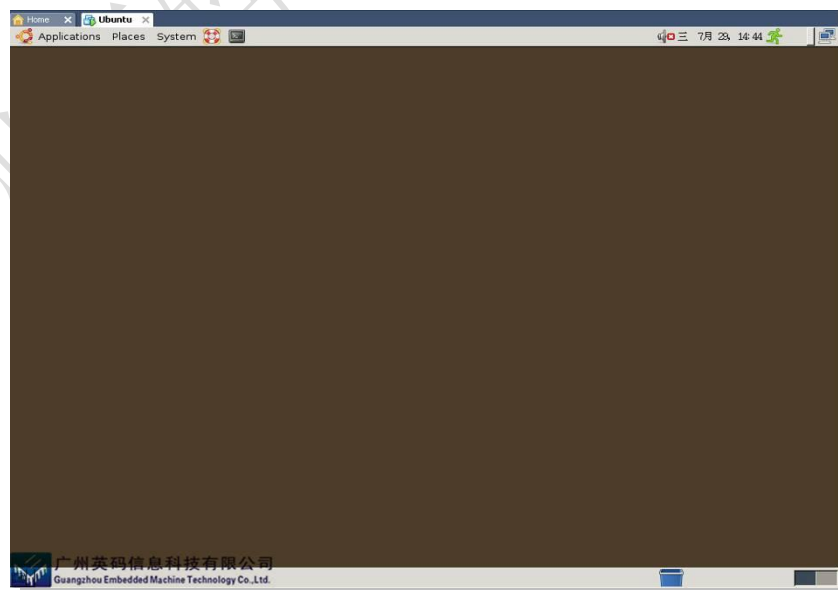


图 4.15



### 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

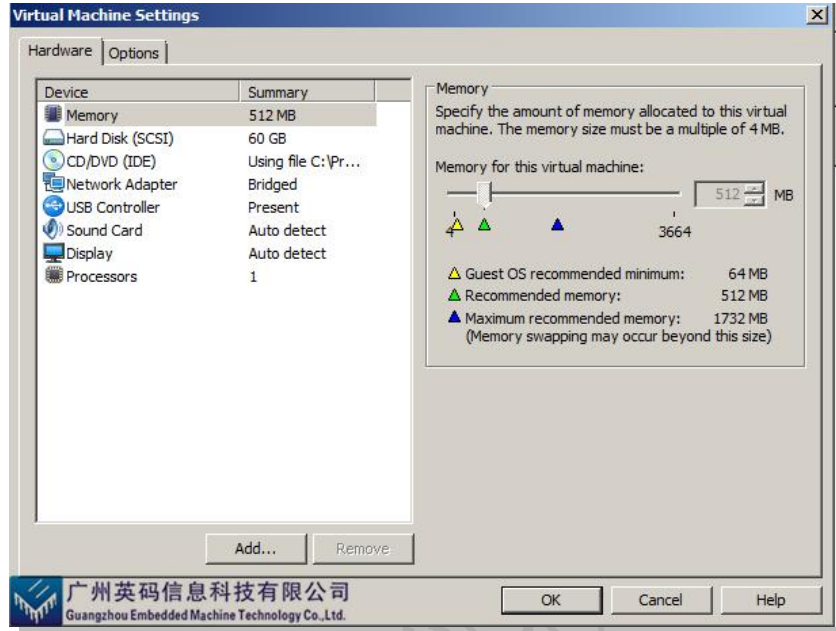


图 4.16

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

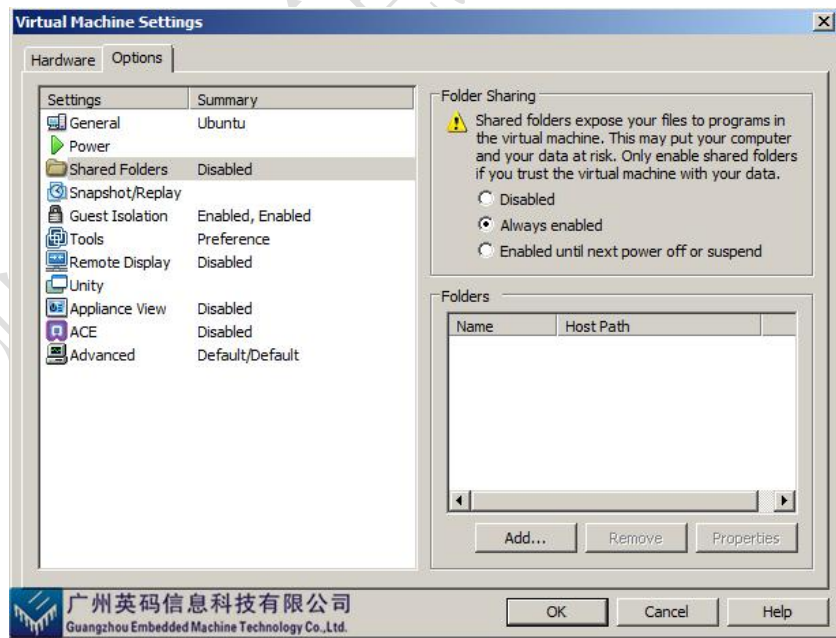


图 4.17



- 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

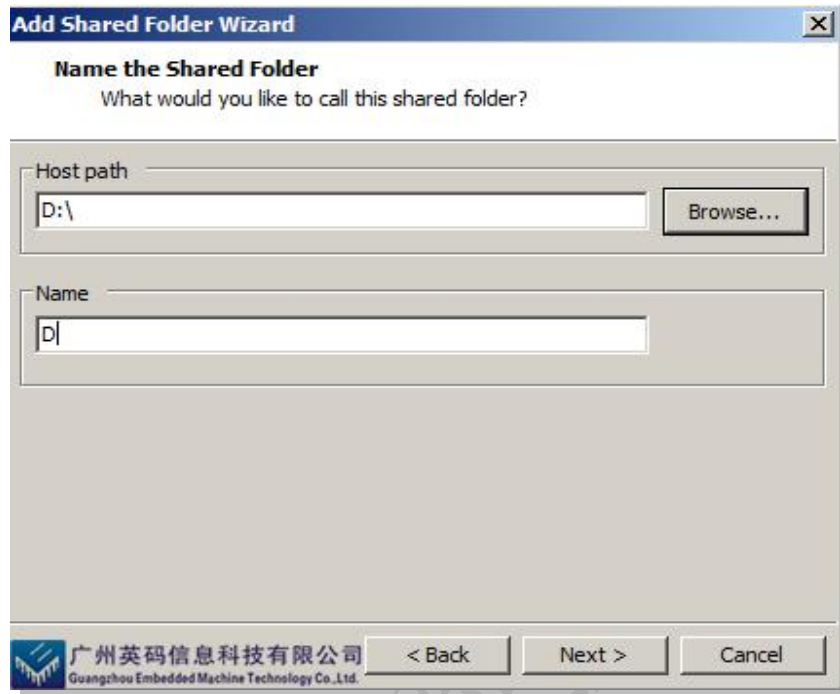


图 4.18

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

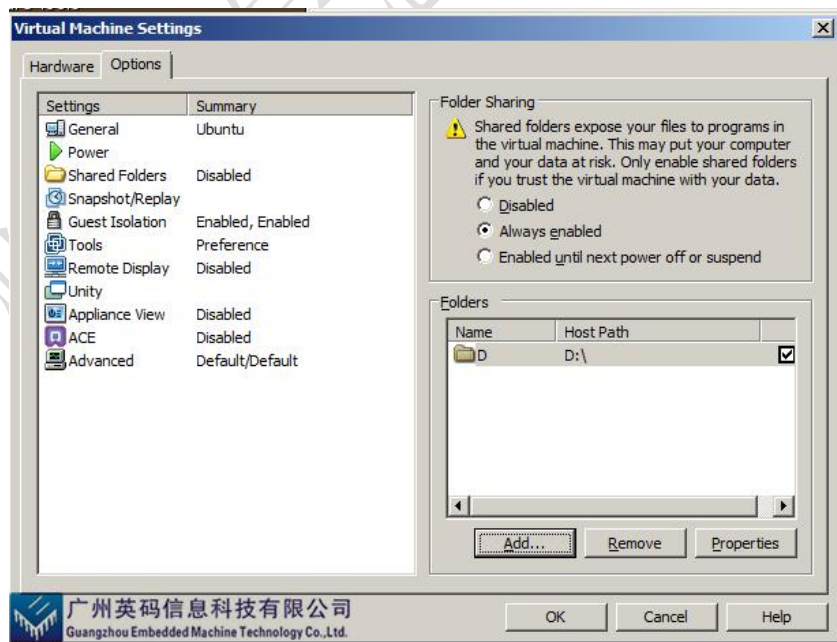


图 4.19



- 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 WindowXP 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位“连接 ema”启动超级终端
- 在 WindowXP 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。

```
/home/ema/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 /home/ema/nfs_share
ema@ema3530:~$ touch /home/ema/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:/home/ema/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:/home/ema/nfs_share /tmp/
Can't set permissions on mtab: Operation not permitted
root@beagleboard:~# ls /tmp/
test
```



## 第六章

# 6. 构建 SBC35x 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
```

f) 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
```

g) 解压 xloader.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
```

h) 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
```

i) 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
```

j) 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.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.tar.bz2 /home/ema/Source
```



- b) 解压 u-boot-release.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.tar.bz2
ema@ema3530:~/Source$ cd u-boot-release
```

- c) 配置编译

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

linux-03.00.01.06.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 linux-03.00.01.06.tar.bz2 /home/ema/Source
```

- b) 解压 linux-03.00.01.06.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 linux-03.00.01.06.tar.bz2
ema@ema3530:~/Source$ cd linux-03.00.01.06
```



c) 配置编译

```
ema@ema3530:~/Source/ linux-03.00.01.06$ make CROSS_COMPILE=arm-none-linux-
gnueabi- ARCH=arm omap3_stalker_defconfig
ema@ema3530:~/Source/ linux-03.00.01.06$ make CROSS_COMPILE=arm-none-linux-
gnueabi- ARCH=arm
ema@ema3530:~/Source/ linux-03.00.01.06$ 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（mkcard.sh脚本在光盘Linux/Tools文件夹里）。

- 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/gfxsdkdemos/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；

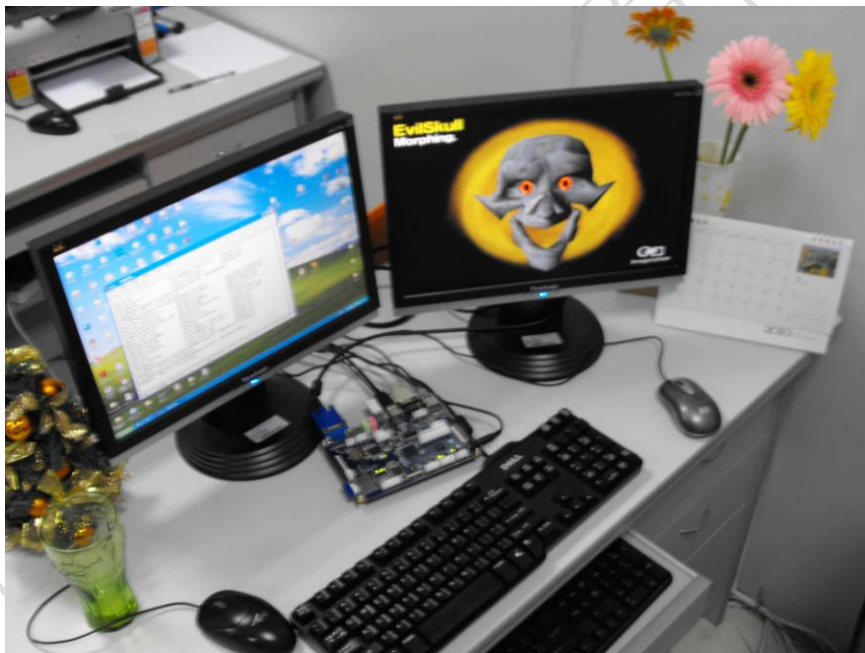
可以将可执行文件通过网络下载到开发板上，[请查看 5.4 挂载网络文件系统 NFS](#)；

可以将可执行文件通过软件下载到开发板上，例如 SSH, WinSCP3。

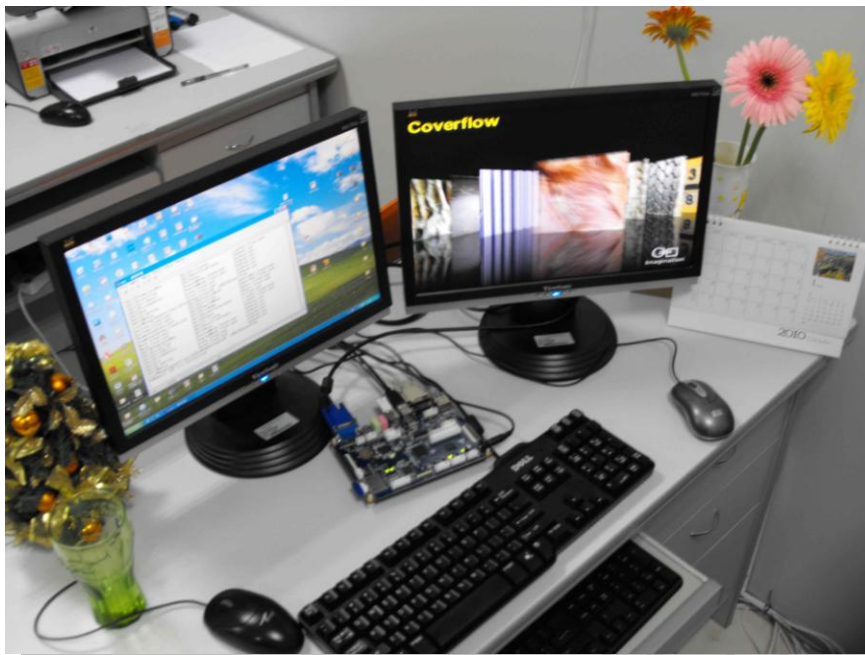
```
./Hello
```

## 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VGA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 SBC35x 系统附带光盘中
- SBC35x 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](http://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/>