



Building Networks For Device

Making Usability For People

DG931x-IT 用户手册

(Ver 1.0.1)

最后修改时间：2005 年 12 月 12 日

版权所有，不得复制

电话：020-62815745，62815746

<http://www.devicegate.com>

广州市联智电子科技有限公司

广州市天河区中山大道 268 号天河广场天威阁 12D

Service@devicegate.com DeviceGate@hotmail.com

目 录

一、DG931x 概述

1.1 DG931x 开发套件构成

1.2 DG931x 开发板组件

1.3 软件特性

二、DG931x 硬件描述

2.1 Cirrus Logic®的 EP931x®微处理器

2.2 Flash 存储器

2.3 SDRAM 存储器

2.4 异步串行通讯口

2.5 1M/10M/100M 以太网接口

2.6 USB 接口

2.7 ADC 接口

2.8 JTAG 接口

三、Linux 在超级终端的启动

四、跳针及连接器指示

五、Linux 的烧写与远行

5.1 RedBoot 概要

5.2 RedBoot 的烧写

5.3 通过 RedBoot 下载，烧写，运行 Linux

5.4 使用单片 16 bit flash 启动与执行

六，系统配置及编译指南

6.1 Linux 内核配置及编译指南

6.2 Cos 配置及编译指南

6.3 Win Ce 配置及编译指南

七，开发板中执行用户应用程序

7.1 使用 TFTP 程序下载

7.2 添加到 Ramdisk 中

附录

技术支持

一、DG931x 概述

首先欢迎选购广州联智电子有限公司出品的 **DG931x** 工控开发套件。

DG931x 工控开发套件为联智电子科技有限公司最新推出的低成本, 适于网络, 音频及 USB Host, IDE, TFT LCD, STN LCD 等应用开发的 **ARM9** 开发套件! **DG931x** 工控开发套件现在已全面支持 **Linux 2.4.19, Linux 2.4.21, eCos 2.0, Wince.Net 4.2, Wince.net 5.0** 等系统。

1.1 **DG931x** 开发套件构成

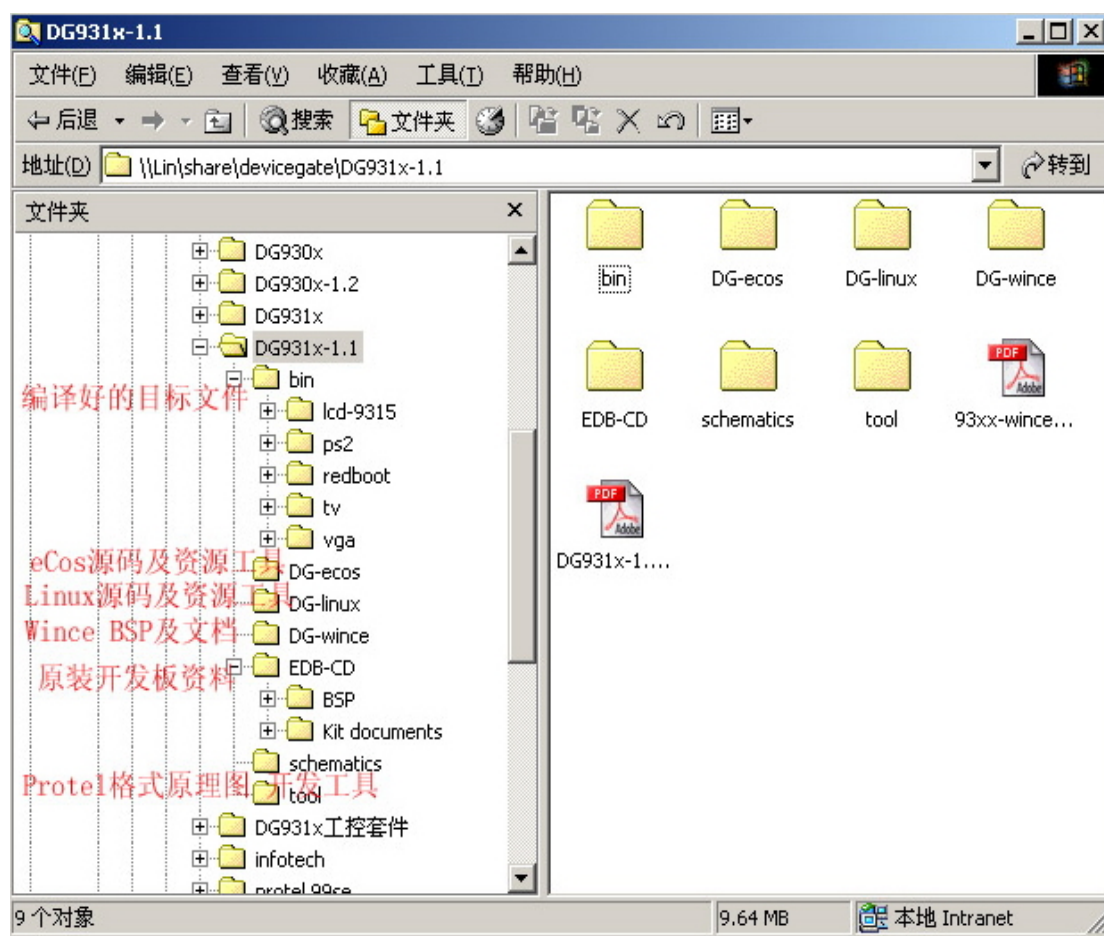
DG931x 开发套件是一个完整的 **ARM9** 嵌入式系统开发平台, 包含开发嵌入式系统需要的所有部件:

——**DG9312/DG9315** 核心板一块

——**DG9312/DG9315** 底板一块

——详细的资料光盘 (包含 **linux 2.4.21** 内核源代码及 **Dg931x** 底板各接口驱动源码, **Wince.Net 4.2 BSP, eCos 2.0** 内核源代码及各种辅助开发工具等)

详细的光盘内容如下图



——9 针的 RS-232C 的串口线

——9V 直流输出的稳压源

1.2 DG931X 开发板组件

DG931X 开发板主要包含以下组件

- Cirrus Logic 公司的 Ep9312/Ep9315 Arm9 处理器
- 64M SDRAM
- 32M FLASH(4-32MB 自由安装)
- 实时时钟
- 三个 USB HOST 接口
- 一个 1/10/100M 的以太网接口
- STEREO 音频的输入输出接口
- 三个串行接口

- LCD 连触摸屏接口
- VGA 接口输出
- IDE 接口（可同时接硬盘，CD，DVD 任两个）
- PS2 键盘接口
- 串行 EEPROM 接口
- 全引出的 16 个 GPIO 接口
- 引出的 8x8 的键盘接口

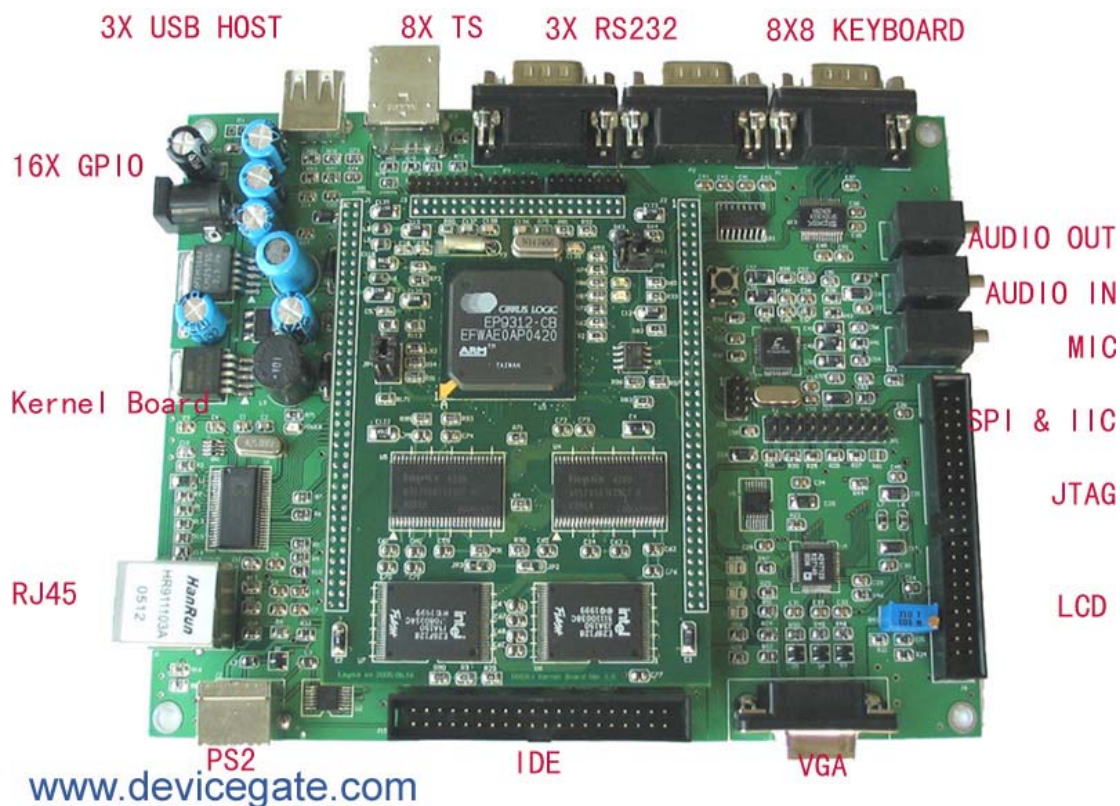
1.3 软件资源

- 支持 Linux2.4.21
- 支持 Linux2.6.8,
- 支持 Wince.net 4.2
- 支持 Wince.net 5.0
- 可提供 wince.net 4.2 BSP
- 可提供 wince.net 5.0 BSP
- 可提供 linux2.4.21 for Arm 内核源代码
- 可提供 linux2.6.8 for Arm 内核源代码
- 可提供 eCos 2.0 for cirrus logic arm9 源代码及系统库文件
- Linux,wince download 工具
- Linux,windows 环境下 Flash 烧写工具
- Redboot, eboot bootload 源代码
- 提供驱动程序如下：
 - 1:LCD 及 VGA,TV 显示驱动
 - 2:AC97 音频驱动
 - 3:网络接口驱动
 - 4:USB HOST 驱动
 - 5:CF /IDE 卡驱动
 - 6:触摸屏驱动
 - 7:flash 驱动

- 8:802.11CF,PCMCIA,USB 无线网卡驱动
- 9:8* 8 键盘接口驱动
- 10:PS2 接口驱动
- 11:U 盘驱动
- 12:USB 键盘鼠标驱动串口驱动
- 13:CF 存储卡驱动

二、DG931X 硬件描述

DG931X 开发板外观如下图：



资源描述如下：

2.1 Cirrus Logic[®] 的 EP931X[®] 微处理器

EP931x 是一种基于 ARM920T 的片上系统，为下一代消费和工业电子产品的开发开辟了新的道路。200-MHz ARM920T 处理器的高速性能以及广泛的外围接口使得 EP931x 成为各种大批量生产型产品的理想选择。

ARM920T 微处理器以单独的 16K 字节、64 路结合组(set-associative)指令以及数据高速缓存为核心。MaverickCrunch™协处理器更进一步增强了其性能。EP931x 的 MaverickCrunch 引擎是一种先进的混合模式数学协处理器，可大大提高 ARM 内核的单/双精度整数及浮点处理能力。当对数字音频和视频格式进行编码、通过以太网处理数据、或执行其它运算密集型计算和数据处理功能时，该引擎可为 EP931x 提供必需的速度和性能。

MaverickKey™独特的硬件编程 ID 是解决网上内容和电子商务安全问题的一个可行方案。互联网安全已逐渐成为通过互联网传送书籍和音乐等数字媒体的重要议题，而传统的软件方法已无法满足安全性的要求。MaverickKey 独特的

ID 为 OEM 厂商提供了一种利用特定硬件 ID 的可行办法,比如分配给 SDMI (安全数字音乐计划)或其它数字版权管理机制的硬件 ID。

EP931x 内置一个高性能 1/10/100 Mbps 以太网媒体存取控制器,以及外部接口,可连接各种外设,包括 SPI 和 I²S 音频、Raster/LCD、IDE 存储外设、小键盘和触摸屏等。该器件还集成有三个 USB 2.0 全速(12 Mbps)主端口和三个 UART。

ARM920T 的 32 位微控制器结构带有一个 5 阶管线,可以极低的功耗提供优异的性能。ARM 内核工作电压为 1.8 V,输入/输出为 3.3 V。依据不同的速度和工作模式,功耗从 45 mW 到 750 mW 不等。

EP931x 适于多种应用,包括:数字媒体服务器和自动点唱机、远程信息控制系统、瘦客户机、机顶盒、POS 终端、工业控制、生物测定安全系统以及 GPS 设备等。

- 200 MHz ARM920T 处理器
 - o 16KB 数据高速缓存和 16 KB 指令数据高速缓存
 - o MMU 可支持 Linux®和 Microsoft® Windows® CE
 - o 100 MHz 系统总线
- MaverickCrunch 数学运算引擎
 - o 浮点、整数和信号处理指令
 - o 优化的数字音乐压缩算法
 - o 硬件互锁允许在线编码
- 用于数字版权管理或 IP 系统安全设计的 MaverickKey ID
 - o 32 位唯一 ID
 - o 128 位随机 ID
- 集成的外设接口
 - o EIDE, 至多可接两个设备
 - o 1/10/100 Mbps 以太网 MAC
 - o 3 端口 USB 2.0 全速主机接口(OHCI)
 - o 3 个 UART (16550 型)
 - o IrDA 接口, 慢速、中速和快速模式
 - o 带专用 SDRAM 通道的 LCD 接口
 - o 触摸屏接口
 - o SPI (串行外设接口) 端口
 - o AC'97 接口
 - o 8X8 小键盘扫描仪
- 外部存储器选项
 - o 32 位 SDRAM 接口, 至多可与 4 组器件相连接
 - o 32/16/8 位 SRAM/FLASH/ROM 接口
 - o 串行 EPROM 接口
- 内置外设
 - o 带软件微调的实时时钟
 - o 12 个可最优化系统性能的数据传输 DMA 通道

- o Boot ROM
- o 双 PLL 可控制所有时钟域
- o Watchdog 计时器
- o 2 个通用 16 位计时器
- o 通用 32 位计时器
- o 40 位 debug 计时器
- 通用 I/O
 - o 16 个具有中断能力的增强型通用 I/O
 - o 额外增加 49 个多路复用在外设上的可选通用 I/O
- 封装: 352 引脚 PBGA; 可选择无铅装配
- o 可适用于消费和工业温度条件

其中 EP9315 更自带 PCMCIA 接口, 并有 2D 图形加速能力。

2.2 Flash 存储器

DG931X 系统主板包含 32MB 的 Flash 存储器, 用于存放启动代码、Linux®内核、用户程序等。7

2.3 SDRAM 存储器

DG931X 包含 64MB SDRAM, 作为程序的运行空间。

2.4 异步串行通讯口

DG931X 外接三个 UART (Universal Asynchronous Receiver/Transmitter), UART0 和 UART1, UART2

UART0 用于软件调试与系统开发。

UART1, UART2 完成与 PC、Modem 及其它支持串行通讯的设备通信。

RS-232C 接口定义见下表:

引脚	名称	功能描述
1	DCD	数据载波检测
2	RXD	数据接收
3	TXD	数据发送

4	DTR	数据终端准备好
5	GND	地
6	DSR	数据设备准备好
7	RTS	请求发送
8	CTS	清除发送
9	RI	振铃指示

2.5 10M/100M 以太网接口

MAC 是 OSI 参考模型中介于物理层（PHY）与逻辑链路层（LLC）之间的 MAC 子层的硬件实现，以太网 MAC 支持 MII(Media Independent Interface) 和 RMII (Reduced Media Independent Interface) 模式的数据传输。

DG9307 外接了一块物理层芯片 ICS 1893AF，这是网络物理层（PHY）芯片，它配合 MCP 提供高效的 10/100Mbps 网络自适应能力，负责网络信号的传送，方便用户接入局域网或者是连上宽带网。

2.6 USB 接口

DG931x 利用 931X 片内集成的 USB 通讯控制器，扩展了三个 USB HOST. 用于支持对鼠标，键盘，U 盘等 USB 从设备的读写。

2.7 ADC 接口

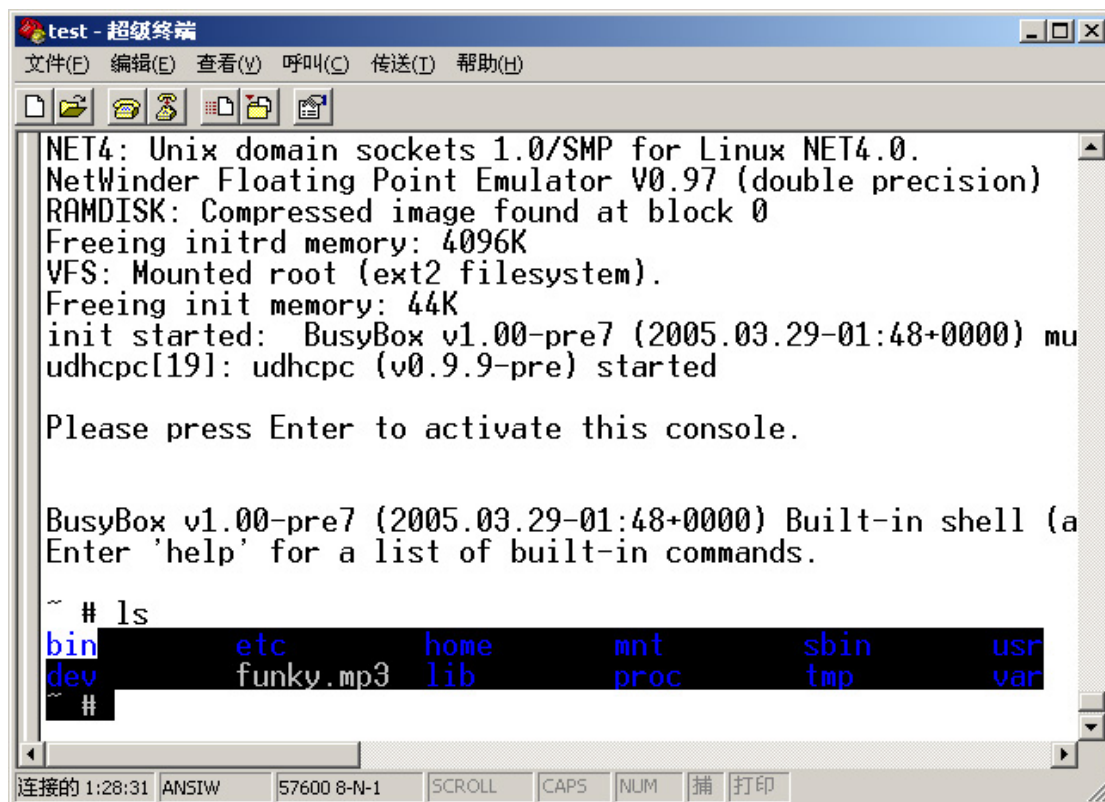
EP9307 片内集成了一个高速的 ADC，支持 5 通道模拟量的输入。

2.8 JTAG 接口

为方便用户的调试，DG931X 设计有 20 针标准 JTAG 接口。20 针接口与 ARM Multi-ICE 接口兼容。

三、Linux 在超级终端的启动

- 1: 打开包装盒，取出 DG931X 开发板
- 2: 确信 DG931X 的 JP9 跳针在 2-3 脚，参见跳针指示章节
- 3: 用附带的串口线连接你的 PC 跟 DG931X 开发板的串口
- 4: 配置你的串口通讯程序，打开 windows 下超级终端（开始—程序—附件—通讯），相应的串口设置为波特率 57600，8 位数据位，1 个停止位，无奇偶校验，无握手。
- 5: 用随板附带的 AC/DC 变压器接通电源，这时你将看到类似下面的画面



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

NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
NetWinder Floating Point Emulator V0.97 (double precision)
RAMDISK: Compressed image found at block 0
Freeing initrd memory: 4096K
VFS: Mounted root (ext2 filesystem).
Freeing init memory: 44K
init started: BusyBox v1.00-pre7 (2005.03.29-01:48+0000) mu
udhcpc[19]: udhcpc (v0.9.9-pre) started

Please press Enter to activate this console.

BusyBox v1.00-pre7 (2005.03.29-01:48+0000) Built-in shell (a
Enter 'help' for a list of built-in commands.

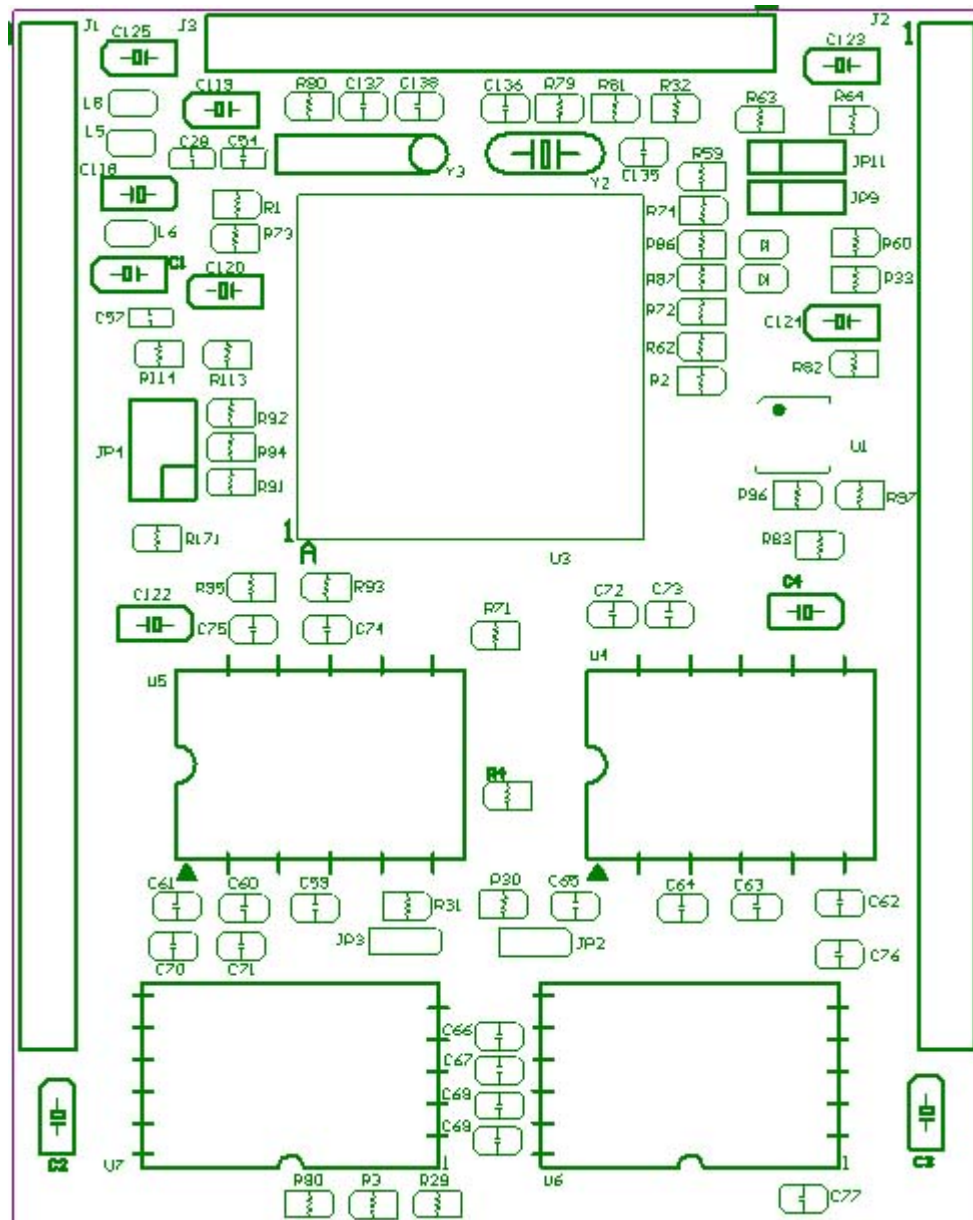
~ # ls
bin      etc      home     mnt      sbin     usr
dev      funky.mp3 lib      proc     tmp      var
~ #
```

连接的 1:28:31 ANSIW 57600 8-N-1 SCROLL CAPS NUM 捕 打印

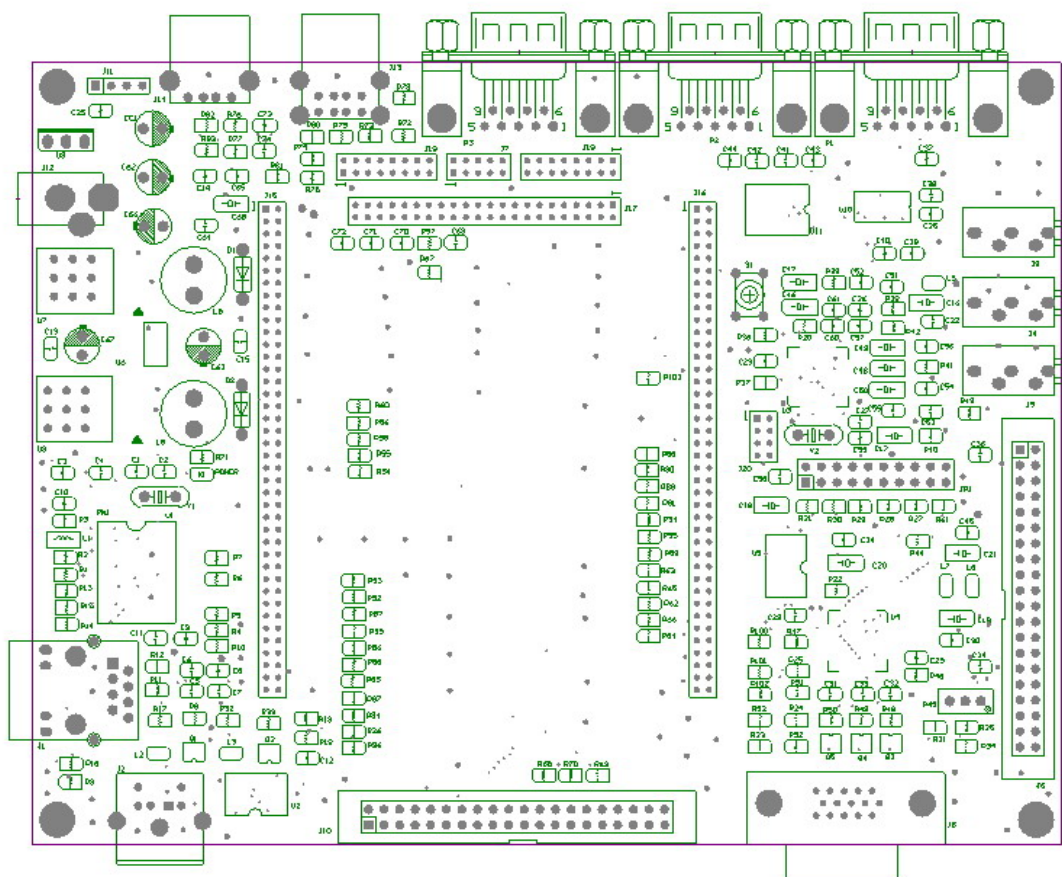
四、跳针及连接器指示

DG931X 的跳针布局图请参看图

核心板布局图：



底板布局图：



核心板连接器功能定义

连接器	功能	连接器	功能
J1	与底板接口	J2	与底板接口
J3	与底板接口	JP4	系统设置
JP2	FLASH 选择	JP3	FLASH 选择

底板连接器功能定义

连接器	功能	连接器	功能
J14	USB HOST	J13	USB HOST
J19	16 EGPIO	P2	第二个串口
JP1	仿真器连接	P1	第一个串口
J2	PS2 接口	P3	第三个串口
J8	VGA 输出	J6	LCD 接口
J10	硬盘接口	J18	8*8 键盘
J12	电源输入	J1	网络接口
J7	8 线触摸屏接口	J3	音频输出

J4	音频输入	J5	麦克输入
----	------	----	------

核心板 JP9 跳线设置

位置	功能	备注
2: 3	正常启动	缺省
1: 2	下载 REDBOOT	

核心板 JP11 跳线设置

跳线	位置	功能	备注
JP11	1: 2	启动选择（内/外）	缺省

JP4 功能设置

引脚	功能	引脚	功能
1: 2	CS7		
3: 4	CS1	5: 6	CS2

CS1	CS2	启动选项
Installed	Installed	Watchdog disabled; reset duration disabled
Installed	Not Installed	Watchdog disabled; reset duration active
Not Installed	Installed	Watchdog active; reset duration disabled
Not Installed	Not Installed	Watchdog active; reset duration active

Cs7	启动宽度
Not Installed	32bit flash
Installed	16bit flash

J11 硬盘电源提供

引脚名	信号名	引脚名	信号名
1	12v	2	地
3	地	4	5v

核心板 JP2，JP3 跳线设置

R18	JP2	JP3	FLASH 选择
INSTALLED	2: 3	2: 3	32BIT FLASH
NOT INSTALL	1: 2	1: 2	16BIT FLASH

JP1 仿真器连接

引脚名	信号名	引脚名	信号名
1	3.3V	2	3.3V
3	TRSTN	4	GND
5	TDI	6	GND
7	TMS	8	GND
9	TCK	10	GND
11	TCK	12	GND
13	TDO	14	GND
15	RSTN	16	GND
17	NC	18	GND
19	NC	20	GND

J7 触摸屏连接

引脚	信号名	引脚	信号名
1	sXp	2	Xp
3	Xm	4	sXm
5	Ym	6	sYm
7	sYp	8	Yp
9	DGND	10	DGND

J18 8x8 键盘连接

引脚	信号名	引脚	信号名
1	ROW0	2	ROW1

3	ROW2	4	ROW3
5	ROW4	6	ROW5
7	ROW6	8	ROW7
9	COL0	10	COL1
11	COL2	12	COL3
13	COL4	14	COL5
15	COL6	16	COL7

J19 EGPIO 连接

引脚	信号名	引脚	信号名
1	EGPIO15	2	EGPIO14
3	EGPIO13	4	EGPIO12
5	EGPIO11	6	EGPIO10
7	EGPIO9	8	EGPIO8
9	EGPIO7	10	EGPIO6
11	EGPIO5	12	EGPIO4
13	EGPIO3	14	EGPIO2
15	EGPIO1	16	EGPIO0

核心板连接器 J1 定义

引脚	信号名	引脚	信号名
1-10	3.3V	11-13	DGND
14-21	1.8V	22	TRST
23	DGND	24	USBM2
25	USBP2	26	DGND
27	WAIT	28	ASDIN
29	ARST	30	EGPIO12
31	EGPIO13	32	EGPIO14
33	EGPIO15	34	DGND
35	IORDY	36	DMACK
37	DIOW	38	DIOR
39	DGND	40	CRS
41	CLD	42	TXERR
43	MIITXD3	44	MIITXD2
45	MIITXD1	46	MIITXD0
47	TXEN	48	TXCLK
49	DGND	50	RXERR

51	DGND	52	RXCLK
53	DGND	54	RXDVAL
55	MIIRXD0	56	MIIRXD1
57	MIIRXD2	58	MIIRXD3
59	MDC	60	MDIO
61	DGND	62	DGND
63	DD2	64	DD3
65	DD11	66	DD4
67	DD10	68	DD5
69	DD9	70	DD6
71	DD8	72	DD7
73	DGND	74	DGND
75	DGND	76	DGND
77	DGND	78	DGND
79	DGND	80	DGNG

核心板连接器 J2 定义

引脚	信号名	引脚	信号名
1	RXD2	2	TXD2
3	RXD1	4	TXD1
5	RESET	6	PBRST
7	TXD0	8	CTS
9	RXD0	10	RTS
11	ASDOUT	12	ABITCLK
13	ASYNC	14	DGND
15	USBP0	16	USBM0
17	DGND	18	USBP1
21	USBM1	20	DGND
21	PWMOUT	22	RDLED
23	GRLED	24	INT3
25	INT2	26	INT1
27	INT0	28	SFRM1
29	SCLK1	30	SSPTX1
31	SSPRX1	32	EEDAT
33	EECLK	34	TMS
35	TDO	36	TCK
37	TDI	38	DTR
39	DSR	40	DGND
41	DD12	42	DD13
43	DD14	44	DD15
45	DD1	46	DD0
47	INT3	48	IDEDA1
49	IDEDA2	50	IDEDA0
51	IDECS1	52	IDECS0
53	DGND	54	DGND
55	SPCLK	56	BLANK
57	HSYNC	58	V_CSNC
59	DGND	60	DGND
61	P12	62	P13
63	P14	64	P15
65	P16	66	P17
67	P6	68	P7
69	P8	70	P9
71	P10	72	P11
73	P0	74	P1
75	P2	76	P3
77	P4	78	P5
79	EGPIO1	80	BRIGHT

核心板连接器 J3 定义

引脚	信号名	引脚	信号名
1	ROW0	2	ROW1
3	ROW2	4	ROW3
5	ROW4	6	ROW5
7	ROW6	8	ROW7
9	COL0	10	COL1
11	COL2	12	COL3
13	COL4	14	COL5
15	COL6	16	COL7
17	RSTN	18	SRST
19	DGND	20	YM
21	YP	22	SXM
23	SXP	24	XM
25	XP	26	SYM
27	SYP	28	DGND
29	EGPIO10	30	EGPIO11
31	EGPIO8	32	EGPIO9
33	EGPIO6	34	EGPIO7
35	EGPIO4	36	EGPIO5
37	EGPIO2	38	EGPIO3
39	EGPIO0	40	EGPIO1
41	DGND	42	DGND
43	DGND	44	DGND

J6 LCD 连接

引脚	信号名	引脚	信号名
1	3.3V	2	5V
3	3.3V	4	5V
5	GND	6	GND
7	BCLK	8	BLANK
9	HSYNC	10	VSYNC
11	GND	12	GND
13	RED0	14	RED1
15	RED2	16	RED3
17	RED4	18	RED5
19	GREEN0	20	GREEN1

21	GREEN2	22	GREEN3
23	GREEN4	24	GREEN5
25	BLUE0	26	BLUE1
27	BLUE2	28	BLUE3
29	BLUE4	30	BLUE5
31	NC	32	NC
33	GND	34	GND
35	+12V	36	+12V
37	/SLEEP	38	BRIGHT
39	SET1	40	SET2

五. Linux 的烧写与运行

5. 1 RedBoot 概要

Redboot 最初由 Redhat 开发，是嵌入式操作系统 eCos 的一个最小版本，集 Bootloader、调试、Flash 烧写于一体。支持串口、网络下载，执行嵌入式应用程序。既可以用在产品的开发阶段（调试功能），也可以用在最终的产品上（Flash 更新、网络启动）。

Redboot 的具体使用方法，参看有关文档或在 `redboot>` 下输入：`help`

5. 2 Redboot 的烧写

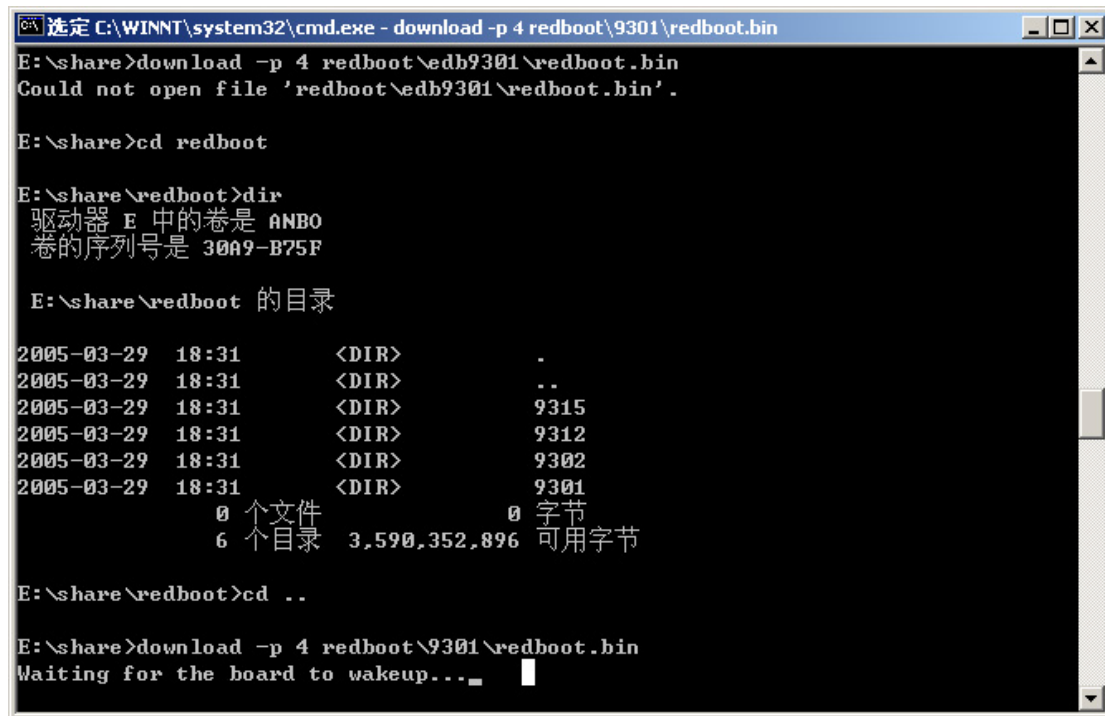
当开发板的系统被破坏或者你需要修改 redboot 以便更符合你的设计要求，你只要按照下列步骤

- 1: 断电，设置 JP9 的跳针为 1-2，参看图 2。
- 2: 连接 DG931X 开发板的串口 1 跟你的 PC 串口，确保 PC 上没有运行任何与开发板相关的串口程序，在 PC 的 WINDOW 命令窗口中运行：

DOWNLOAD -P1 REDBOOT.BIN

其中 windows 版本的 download 在资源光盘的 tool 目录中，linux 版本的 download 在解压 DG-linux.tar.gz 文件后的的相关产品目录中，redboot.bin 在资源光盘的/bin/redboot/931x 目录中

你将看到



```
选定 C:\WINNT\system32\cmd.exe - download -p 4 redboot\9301\redboot.bin
E:\share>download -p 4 redboot\edb9301\redboot.bin
Could not open file 'redboot\edb9301\redboot.bin'.

E:\share>cd redboot

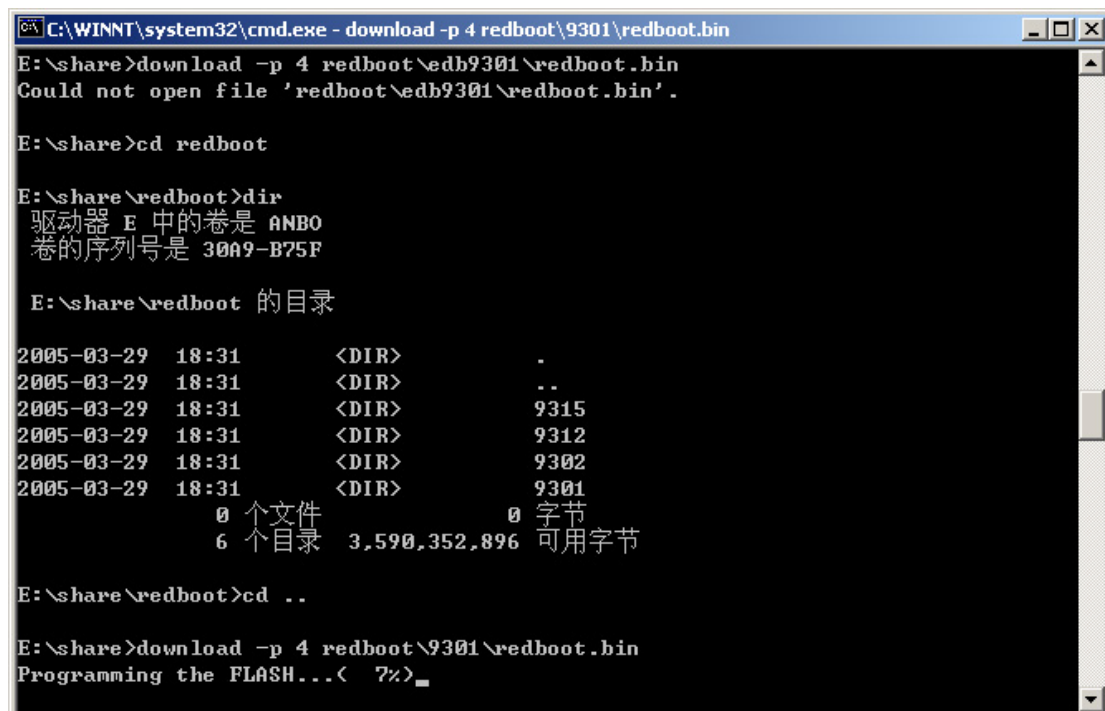
E:\share\redboot>dir
驱动器 E 中的卷是 ANBO
卷的序列号是 30A9-B75F

E:\share\redboot 的目录
2005-03-29 18:31 <DIR> .
2005-03-29 18:31 <DIR> ..
2005-03-29 18:31 <DIR> 9315
2005-03-29 18:31 <DIR> 9312
2005-03-29 18:31 <DIR> 9302
2005-03-29 18:31 <DIR> 9301
                0 个文件                0 字节
                6 个目录 3,590,352,896 可用字节

E:\share\redboot>cd ..

E:\share>download -p 4 redboot\9301\redboot.bin
Waiting for the board to wakeup..._
```

3: 给 DG931X 板子上电, 你将看到



```
C:\WINNT\system32\cmd.exe - download -p 4 redboot\9301\redboot.bin
E:\share>download -p 4 redboot\edb9301\redboot.bin
Could not open file 'redboot\edb9301\redboot.bin'.

E:\share>cd redboot

E:\share\redboot>dir
驱动器 E 中的卷是 ANBO
卷的序列号是 30A9-B75F

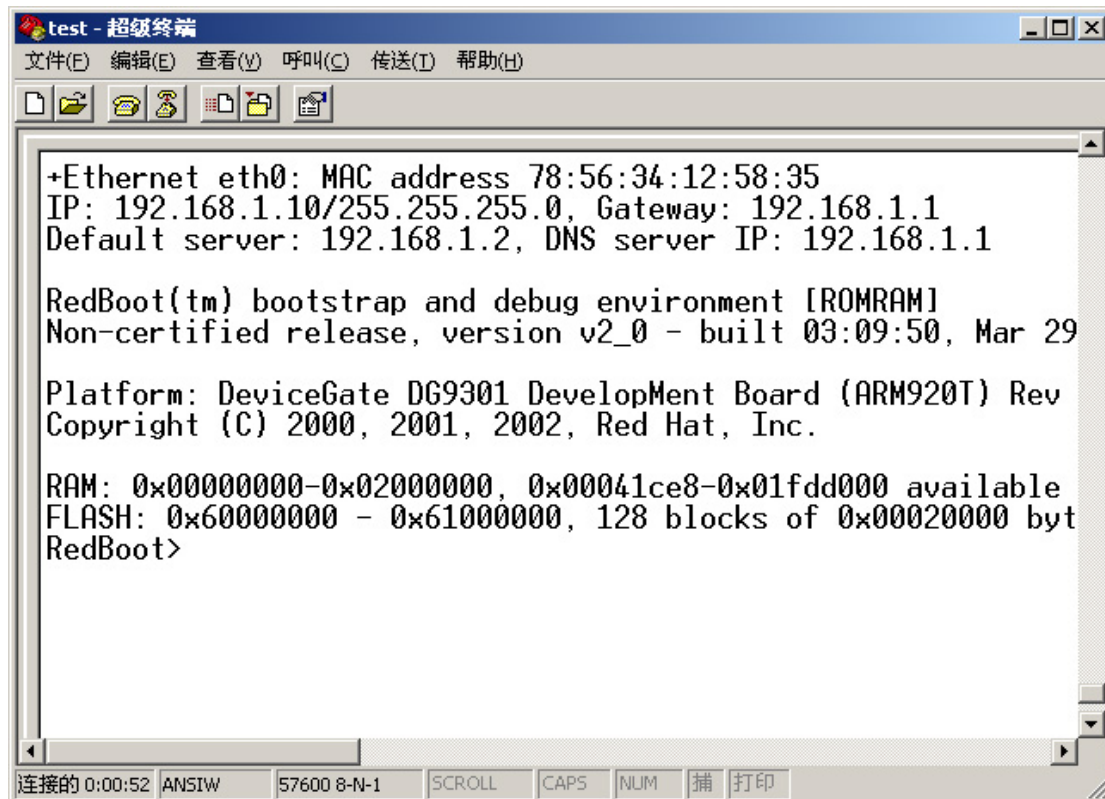
E:\share\redboot 的目录
2005-03-29 18:31 <DIR> .
2005-03-29 18:31 <DIR> ..
2005-03-29 18:31 <DIR> 9315
2005-03-29 18:31 <DIR> 9312
2005-03-29 18:31 <DIR> 9302
2005-03-29 18:31 <DIR> 9301
                0 个文件                0 字节
                6 个目录 3,590,352,896 可用字节

E:\share\redboot>cd ..

E:\share>download -p 4 redboot\9301\redboot.bin
Programming the FLASH...< 7%>_
```

5. 3 通过 Redboot 下载，烧写，运行 LINUX

- 1: DOWNLOAD 完成后，断电，把 JP9 设为 2-3。
- 2: 配置你的串口通讯程序，打开 windows 下超级终端（开始—程序—附件—通讯），相应的串口设置为波特率 57600，8 位数据位，1 个停止位，无奇偶校验，无握手。
- 3: 重新上电你将看到 redboot 启动画面为：



The screenshot shows a Windows SuperTerminal window titled "test - 超级终端". The window contains the following text output from RedBoot:

```
+Ethernet eth0: MAC address 78:56:34:12:58:35
IP: 192.168.1.10/255.255.255.0, Gateway: 192.168.1.1
Default server: 192.168.1.2, DNS server IP: 192.168.1.1

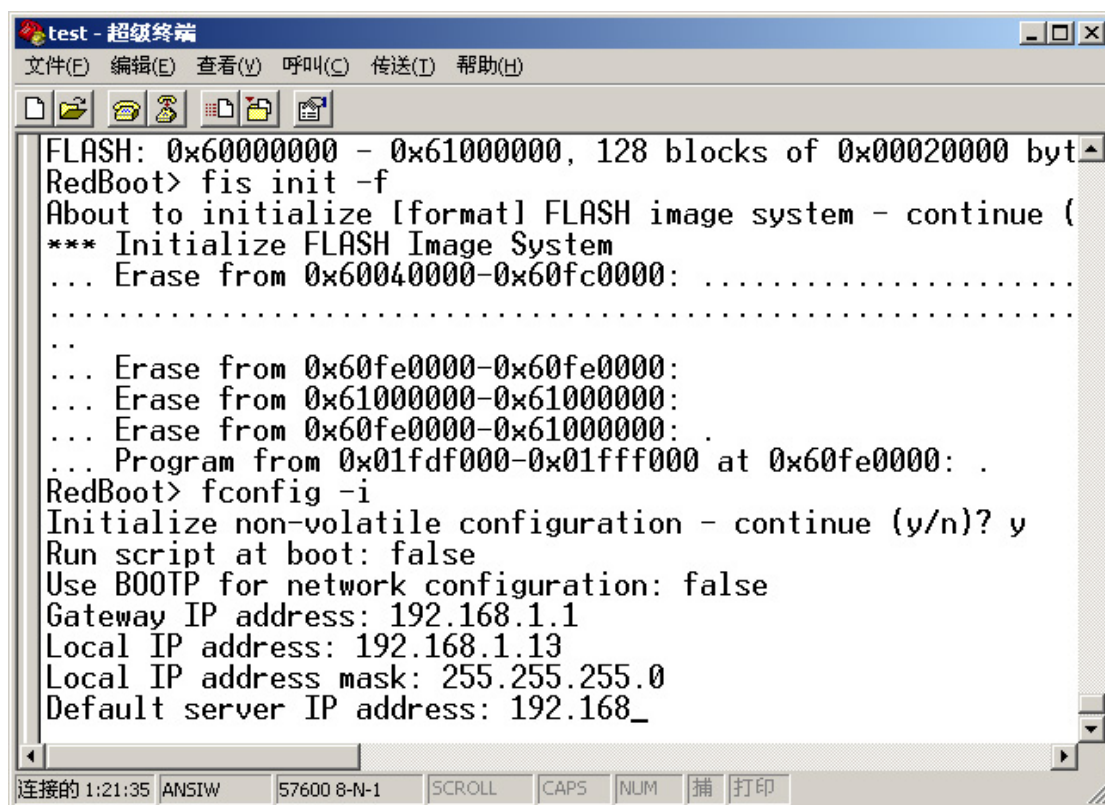
RedBoot(tm) bootstrap and debug environment [ROMRAM]
Non-certified release, version v2_0 - built 03:09:50, Mar 29

Platform: DeviceGate DG9301 Development Board (ARM920T) Rev
Copyright (C) 2000, 2001, 2002, Red Hat, Inc.

RAM: 0x00000000-0x02000000, 0x00041ce8-0x01fdd000 available
FLASH: 0x60000000 - 0x61000000, 128 blocks of 0x00020000 byt
RedBoot>
```

The status bar at the bottom of the window shows: 连接的 0:00:52, ANSIW, 57600 8-N-1, SCROLL, CAPS, NUM, 捕, 打印.

- 4: REDBOOT 启动后，下一步的任务是烧录 LINUX KERNEL 及 RAMDISK
- 5: 按照你的 linux 版本配置你的 tftp 服务器或者在 windows 中使用我们提供的 tftpd32.exe 程序（在光盘的 tool 目录中），并且确认 ramdisk.gz, zImage, 在 TFTP 的工作目录下。
- 6: 确认你的 tftp 服务器已经运行，在 redboot 的提示符下，首先输入 fis init -f 格式化 flash.
然后请输入 fconfig -i, 你将看到

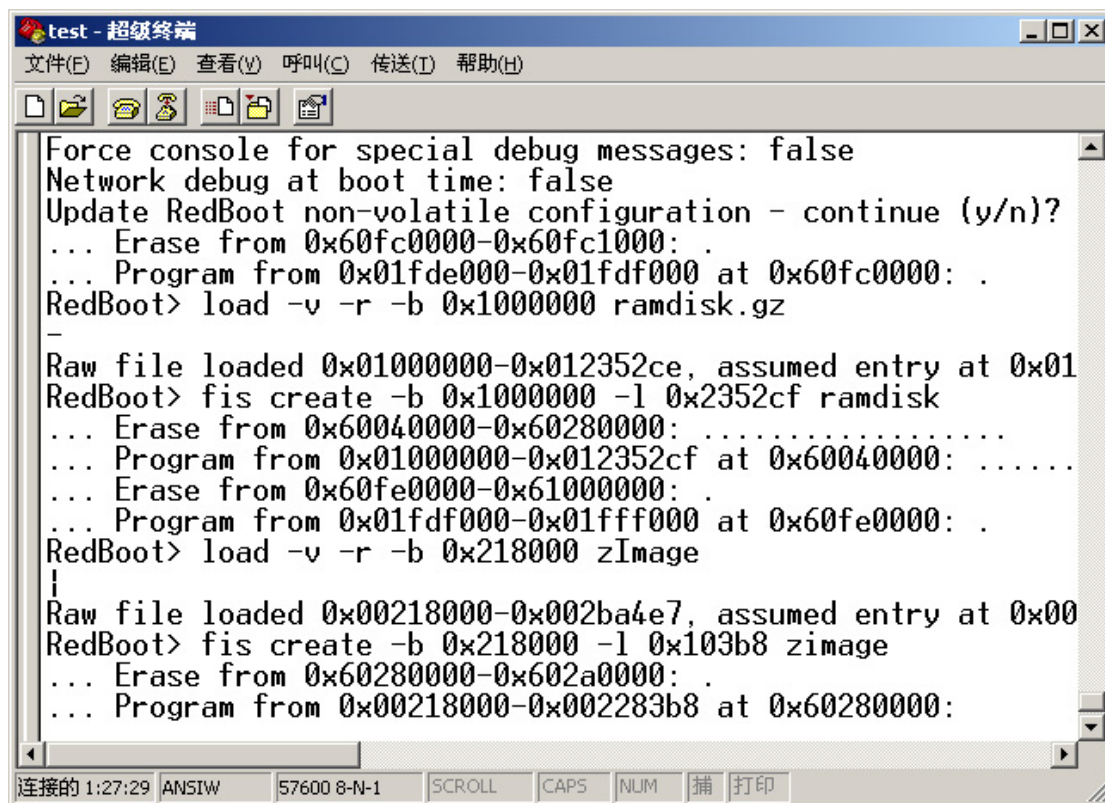


```
test - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

FLASH: 0x60000000 - 0x61000000, 128 blocks of 0x00020000 bytes
RedBoot> fis init -f
About to initialize [format] FLASH image system - continue (y/n)? y
*** Initialize FLASH Image System
... Erase from 0x60040000-0x60fc0000: .....
...
... Erase from 0x60fe0000-0x60fe0000:
... Erase from 0x61000000-0x61000000:
... Erase from 0x60fe0000-0x61000000: .
... Program from 0x01fdf000-0x01fff000 at 0x60fe0000: .
RedBoot> fconfig -i
Initialize non-volatile configuration - continue (y/n)? y
Run script at boot: false
Use BOOTP for network configuration: false
Gateway IP address: 192.168.1.1
Local IP address: 192.168.1.13
Local IP address mask: 255.255.255.0
Default server IP address: 192.168_

连接的 1:21:35 ANSIW 57600 8-N-1 SCROLL CAPS NUM 捕 打印
```

- 7: 如果你不想使用 bootp 启动开发板, 请配置好开发板的 ip 配置信息, 其中 Default server IP address 为运行 tftp 服务器的 PC 的 IP 地址。
- 8: 按住 DG930x 开发板的复位键, 或者直接断电源再开电源启动开发板, 让网络设置生效。
- 9: 在 REDBOOT 的提示符下, 运行 `load -v -r -b 0x800000 ramdisk.gz`, 如果你想把 ramdisk.gz 烧录到 flash 中, 请输入 `fis create -b 0x800000 -l <ramdisk.gz 的文件大小> ramdisk`, 你将看到
- 10: 等烧录 ramdisk 完成后, 再输入 `Load -v -r -b 0x800000 zImage` 如果你想烧录 zImage 到 flash 中, 请输入 `Fis create -b 0x800000 -l <zImage 的文件大小> zImage` 后你将看到



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

Force console for special debug messages: false
Network debug at boot time: false
Update RedBoot non-volatile configuration - continue (y/n)?
... Erase from 0x60fc0000-0x60fc1000: .
... Program from 0x01fde000-0x01fdf000 at 0x60fc0000: .
RedBoot> load -v -r -b 0x1000000 ramdisk.gz
-
Raw file loaded 0x01000000-0x012352ce, assumed entry at 0x01
RedBoot> fis create -b 0x1000000 -l 0x2352cf ramdisk
... Erase from 0x60040000-0x60280000: .....
... Program from 0x01000000-0x012352cf at 0x60040000: .....
... Erase from 0x60fe0000-0x61000000: .
... Program from 0x01fdf000-0x01fff000 at 0x60fe0000: .
RedBoot> load -v -r -b 0x218000 zImage
-
Raw file loaded 0x00218000-0x002ba4e7, assumed entry at 0x00
RedBoot> fis create -b 0x218000 -l 0x103b8 zimage
... Erase from 0x60280000-0x602a0000: .
... Program from 0x00218000-0x002283b8 at 0x60280000:
-
连接的 1:27:29 ANSIW 57600 8-N-1 SCROLL CAPS NUM 捕 打印
```

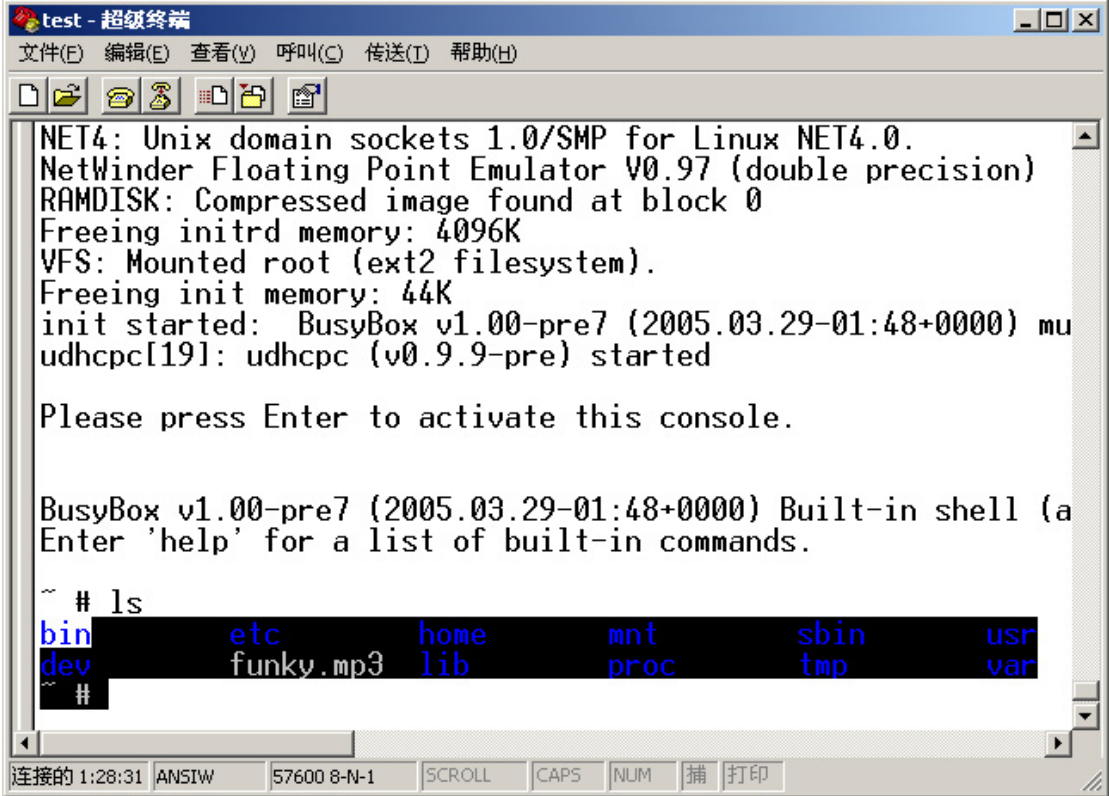
11: 如果你想在系统上电后自动运行 linux,请重新运行 `fconfig -i`, 当系统提示 Run script at boot: 你回答 yes.然后你将输入

Fis load ramdisk

Fis load zImage

Exec -r 0x800000 -s 0x600000 回车, 再回车。

至此你已经完成了所有的烧录步骤。按复位键或者重新给板子上电, 你将看到 linux 启动后的相面为



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

NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
NetWinder Floating Point Emulator V0.97 (double precision)
RAMDISK: Compressed image found at block 0
Freeing initrd memory: 4096K
VFS: Mounted root (ext2 filesystem).
Freeing init memory: 44K
init started: BusyBox v1.00-pre7 (2005.03.29-01:48+0000) mu
udhcpc[19]: udhcpc (v0.9.9-pre) started

Please press Enter to activate this console.

BusyBox v1.00-pre7 (2005.03.29-01:48+0000) Built-in shell (a
Enter 'help' for a list of built-in commands.

~ # ls
bin      etc      home    mnt      sbin     usr
dev      funky.mp3 lib      proc     tmp      var
~ #
```

5. 4 使用单片 16 bit flash 启动与执行

首先进行以下跳线设置：

JP4 插上跳线，JP11 放到 2-3 脚，去掉 R3 电阻，JP2,JP3 焊到 2-3 脚。
具体操作步骤参照 5.2,5.3 节，把/bin/16flash 目录下的 Redboot 烧到开发板，并下载 ramdisk，zimage 执行。

六，系统配置及编译指南

6.1 LINUX 内核配置及编译指南

为了方便用户加快开发过程，联智电子特意整理好各种必备的工具在开发工具的资源光盘中。你只需要按照下面步骤即可：

1:把资料光盘中的 **TOOL** 目录中的 **arm-linux-gcc-3.3.tar.bz2** 解包到你的开发机的根目录，具体如下：

```
tar -xvjf arm-linux-gcc-3.3.tar.bz2
```

2: 把资料光盘中的 **TOOL** 目录中的 **arm-elf-gcc-3.2.1-full.tar.bz2** 解包到你的开发主机的根目录，具体如下：

```
tar -xvjf arm-elf-gcc-3.2.1-full.tar.bz2
```

3: 为了不每次设置 **PATH**, 加下面这行到你的开发主机相关用户的 **.bash_profile** 文件中

```
PATH=/usr/local/arm/3.3/bin:/usr/local/arm/3.2.1-elf/bin:$PATH
```

4: 然后把资料光盘中 **DG-Linux** 目录中的 **DG-Linux.tar.gz** 文件解包到开发主机你想要的目录。

5: 进入解包后你的相关产品命名的目录，（如你是购买的 **9301** 就进入 **9301** 目录），执行 **make**，系统将自动在你所购产品命名的目录下生成你所需要的 **redboot.bin**(引导系统的 **bootload**), **ramdisk.gz**(运行 **linux** 所需要的压缩版本 **ramdisk**), **zImage**(**linux** 内核)。

每个生成的目标都有特定的配置文件，你可以单独修改各自的配置，譬如你仅仅需要修改 **busybox** 的配置，只要运行 **make busyboxconfig**, 按照你的需求配置好 **busybox**, **busybox** 的帮助文档在随板 **CD** 的 **busybox** 目录或者网络上都可以找到。

如果你想修改 **linux** 的配置，请运行 **make linuxconfig** 即可，具体配置方法，另有详细的说明文档在随板 **CD** 中。

当然你就可以单独编译生成你所想要的目标

在你所购产品的相关目录下键入：

```
make linux 将生成 linux kernel
make ramdisk 将生成 ramdisk.gz
make redboot 将生成 redboot.bin
执行 make 将生成上面三个目标文件
```

6.2 Cos 配置及编译指南

为了方便用户，联智电子已经把所有的开发工具及代码包整理在随板的开发资源光盘中，你只需要按照下面步骤即可：

- 1: 把资料光盘中 **DG-ecos** 目录中的 **DG-ecos.tar.gz** 用下列方法解压到你的开发主机：

```
tar -xvzf DG-ecos.tar.gz
```

- 2: 进入到解包后你购买产品的目录下(如果你购买 **9301**, 请进入 **9301** 目录), 执行 **make**, 即可在你购买产品命名的目录下生成可执行映像所需的头文件及库文件。
- 3: 如果你想配置 **ecos**, 运行 **configtool**, 这是 **ecos** 图形配置工具, 无论是 **linux** 还是 **cygwin** 中都有版本, 非常好用, 光盘中有详细的说明文档。或者使用人机对话形式的版本 **ecosconfig**

6.3 Win Ce 配置及编译指南

Wince 的使用在光盘中另有文档详细记载，本文不作说明！

七，开发板中执行用户应用程序

7. 1 使用 TFTP 程序下载

第一步：在开发主机上使用 `arm-linux-gcc` 编译程序，使用“`hello .c`”实例程序来说明

```
Arm-linux-gcc -o hello hello.c
```

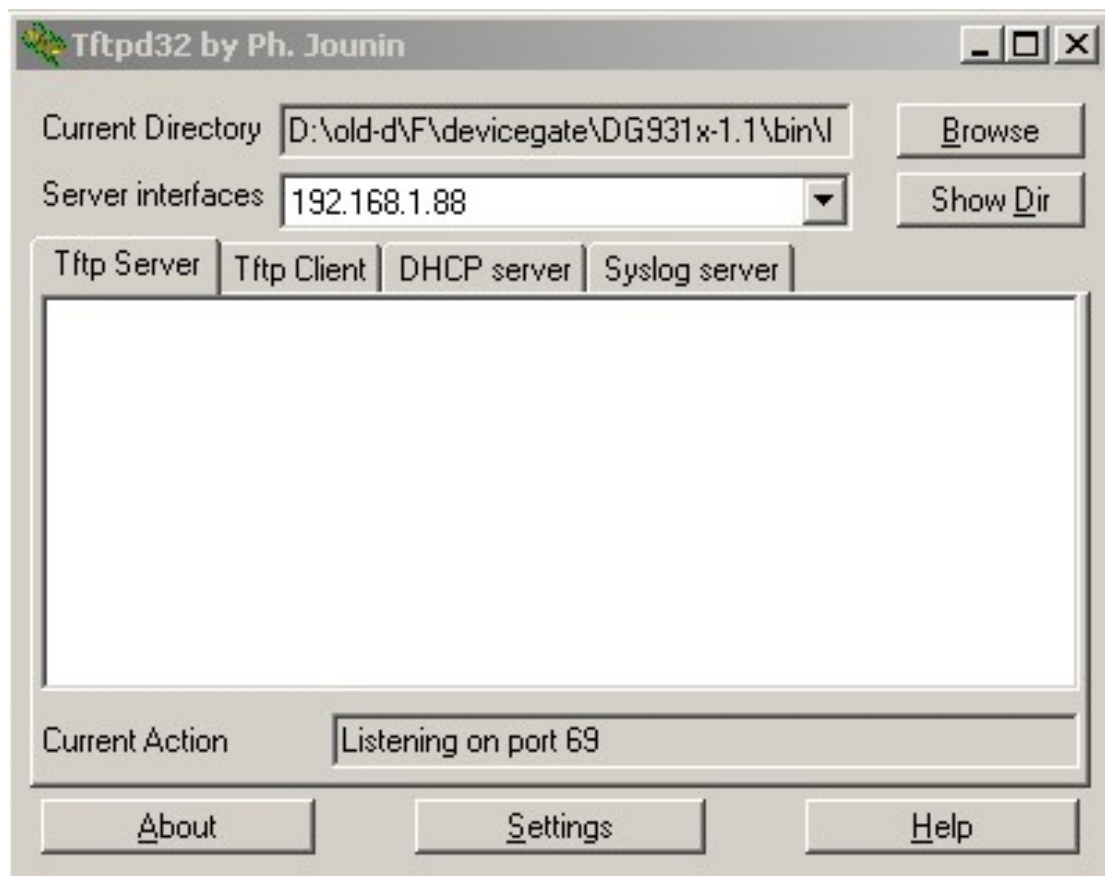
第二步：如果网络上没有 TFTP 服务器，你需要给开发板分配 IP 地址，如：

```
Ifconfig eth0 192.168.1.2
```

如果在 linux 环境下，你需要配置好 linux 的 TFTP 服务，具体操作请参考 linux 相关文档。

如果在 windows 环境下，请打开光盘下的 `Tftpd21 by ph.Jounin` 按下面的设置方式设置好 TFTP 服务：

把 TFTP server 选项的 Current Directory 设为“`hello.c`”所在的目录，如下图所示：



第三步：在开发板的控制平台，创建一个目录，用于存放程序的结果，如：

```
Mkdir /usr/hello
```

第四步：在你的开发板控制平台，需要 TFTP 通过以太网从主机上传最新的编译程序到目标板。`Tftp -g -r hello 192.168.1.1`（192.168.1.1 是开发主机的 ip 地址）

第五步：然后，使用 **Chmod** 添加运行属性。如：

```
Chmod +x hello
```

第六步：现在，你编译的程序在开发板上执行，如： **./hello**

7. 2 添加到 **Ramdisk** 中执行

第一步：使用 **arm-linux-gcc** 编译程序，如：

```
Arm-linux-gcc -o hello hello.c
```

第二步：在开发板目录下编辑 **build.sh** 文件。在这个例子中，在你所购产品的相关目录下键入。

```
gedit build.sh
```

在 **build.sh** 中找到 “see if the ramdisk”

try mkdir -p root /usr/share/udhcp 之后加入下面一行，

```
try mkdir -p /root/usr/hello
```

在 “ **try cp `find www -type f -maxdepth 1 -print /root/home/www`** ”之后增加下面一行：

```
try cp `find hello -type f -maxdepth 1 -print /root/usr/hello`
```

第三步：在 **ep9301** 目录下执行：

```
Make
```

第四步：当完成后，把 **ramdisk.gz** 和 **zimage** 文件移动到主机的 **/ftp** 目录下。
然后，参考我们的文档把映象文件运行在开发板上。

第五步：进入 **fartor** 目录执行一下命令

```
cd /usr/hello
```

```
./hello
```


附录

1: 安装 U 盘

本开发板已烧入的系统内核已经包含了 USB HOST, IIS, 大容量存储系统的支持, 你只需要执行下面命令

mount /dev/sda1 /mnt 即可

2: 播放 mp3(直接播放或者安装播放)

本开发板已烧入的 ramdisk 已经包含 playmp3 执行软件, 你只需要执行下面命令

playmp3 播放文件.mp3 /dev/audio

或者 **playmp3 播放文件.mp3 /dev/dsp** 即可, 用耳机或者有源音箱即可听到锐耳的 MP3 音乐

如果你想直接像 mp3 机一样循环播放 U 盘中的 mp3, 只要用第四章所用方法把 CD 中 /bin/u-disk-mp3 目录中的 ramdisk.gz 及 zImage 烧入开发板或者下载到开发板, 等 linux 启动完毕后, 运行 **playusb** 即可欣赏音乐。

3: 录制音频文件

本开发板已烧入的 ramdisk 已经包含音频录制的测试软件, 你只需要执行下面命令

brec -w -S -s 44100 -b 16 -5 test.wav

如果你想回放, 输入

bplay test.wav

再次感谢选购广州联智电子有限公司的产品! 祝好运!

技术支持

MSN: DeviceGate@hotmail.com

电话: 020-62815745, 62815746

QQ 群: 16909687

论坛: www.devicegate.com/bbs