

JS9331 开发板使用手册

v1.4 (2016.02.15)

智盛电子

网站: www.joysince.com

淘宝店铺: joysince.taobao.com

E-mail: wurobinson@joysince.com

前言

这份《JS9331 开发板使用手册》主要用于介绍 JS9331 开发板的硬件资源、开发板的使用方法、如何解决使用过程中的可能的问题、使用注意事项。

目录

前言.....	2
1. 开发板介绍.....	5
1.1 开发板概述.....	5
1.2 开发板特性.....	5
1.3 开发板硬件资源.....	5
1.4 开发板套件说明.....	7
1.5 开发板配套资料说明.....	8
2. 开发板功能测试.....	8
2.1 基本功能测试.....	8
2.1.1 开机测试.....	8
2.1.2 wifi 测试.....	9
2.2 开发板高级功能测试.....	10
2.2.1.1 安装串口驱动.....	10
2.2.2 安装 SecureCRT.....	10
2.2.3 安装 Putty(可选).....	10
2.2.4 安装 WinScp.....	10
2.2.5 测试调试串口.....	12
2.2.6 测试拓展串口.....	15
2.2.7 按键测试.....	17
2.2.8 USB 测试.....	18
2.2.9 温度传感器测试.....	18
2.2.10 红外发射接收测试.....	18
3. 开发板固件镜像烧录说明.....	20
3.1 openwrt 固件烧写说明.....	20
3.1.1 openwrt 网络烧写.....	21
3.1.2 安装 tftp server.....	22
3.1.3 openwrt 串口+网络烧写.....	23
3.2 uboot 烧写说明.....	25
3.3 ART 烧写说明.....	26

3.4 修改默认 MAC 地址.....	27
4. 常见问题及解答.....	28
5. 修改说明.....	28

1. 开发板介绍

1.1 开发板概述

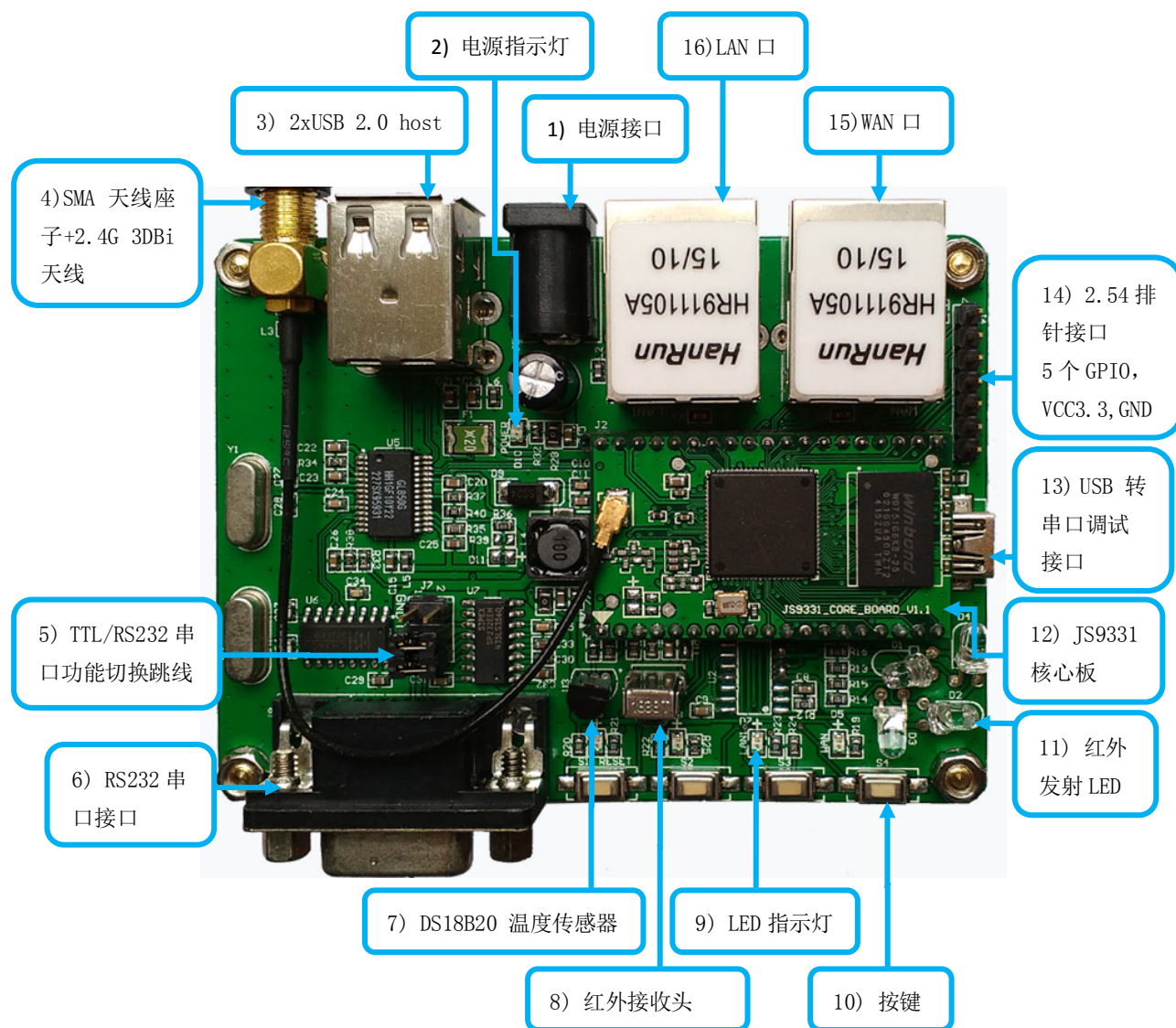
JS9331 开发板是智盛电子开发的基于 Atheros AR9331 SOC 方案的学习开发评估板。本开发板的系统采用 openwrt (linux) 源码完全开放, 板子功能丰富, 同时提供嵌入式 linux 学习资料、openwrt 学习资料, 是学习嵌入式 linux、openwrt、物联网、DIY 智能家居网关、企业前期产品评估的理想选择。JS9331 开发板包含一个底板 (JS9331 底板) 和一个核心板 (JS9331 核心板, 有关该核心板的介绍, 请看《JS9331 核心板介绍手册》)。

1.2 开发板特性

- 1) 400MHz 系统主频
- 2) 64MByte DDR2 RAM
- 3) 8MByte(可升级为 16MByte) SPI flash
- 4) 802.11n 1x1 最高 150Mbps 无线传输速率
- 5) 2 个 100/10Mbps 支持自动翻转的 LAN/WAN 以太网口
- 6) 2 路标准的高速 USB 2.0 host 接口, 支持 U 盘、USB 转串口线、USB 摄像头、USB 移动硬盘、USB 3G 网卡、USB 无线网卡等超多 USB 外
- 7) 1 路 TTL/RS232 串口, 1 路 usb mini 调试串口接口
- 8) 4 个可配置 LED 指示灯, 4 个可配置功能按键
- 9) 1 个板载温度传感器
- 10) 1 路红外接收, 1 路红外发射
- 11) 5 路未用 GPIO 口 (核心板总共有 16 个 GPIO 口, 其他 GPIO 口, 在开发板上已做其他用途)
- 12) DC 5V 2A 电源输入, 3.3V 1A 电源输出
- 13) 整机静态功耗仅 0.9W
- 14) 支持路由、AP、中继、桥接等模式
- 15) 系统采用基于 linux 的 openwrt 开源操作系统, 支持超多高级应用

1.3 开发板硬件资源

下图标识出了 JS9331 的硬件资源。



1) 电源接口

请务必采用 DC 5V 电源。因用错电源导致开发板损坏的，不属于保修范围。尽量采用开发板自带的 DC 5V 2A 电源。

2) 电源指示灯

3) 2xUSB 2.0 host 接口

可接 U 盘、USB 转串口线、USB 摄像头、USB 移动硬盘、USB 3G 网卡、USB 无线网卡等超多 USB 外设。

4) SMA 天线座子+2.4G 3DBi 天线

5) TTL/RS232 功能切换跳线帽

如图，插上跳线帽，则启用 RS232 接口。拔掉跳线帽，则可接 3.3V 串口。默认跳线帽已插上。

6) RS232 串口接口

需按照 5) 的说明启用该接口。

7) 温度传感器—DS18B20

开发板已支持该温度传感器的 openwrt 演示配置界面。具体用法请参考第 5 章。

8) 红外接收头

开发板提供相应的驱动和演示程序源码，具体用法请参考第 5 章。

9) LED 指示灯（所有指示灯功能，均可自由配置）

开发板默认的 LED 功能，从左往右依次是：

- 1) 系统运行指示灯。系统上电约 15s 后快速闪烁，表示系统正在启动，进入慢闪后标示系统已经启动完成。
- 2) wifi 指示灯。wifi 功能开启后，常亮。如果有数据传输，则闪烁。
- 3) LAN 口指示灯。连接后，常亮。如果有数据传输，则闪烁。
- 4) WAN 口指示灯。连接后，常亮。如果有数据传输，则闪烁。

10) 按键（所有指示灯功能，均可自由配置）

开发板默认的按键功能，从左往右依次是：

- 1) 复位/恢复出厂设置按键。开发板启动完成后，短按（小于 3s），开发板重启。**长按（大于 8s），开发板恢复到出厂设置**
- 2) 未定义功能，用户自定义按键
- 3) 未定义功能，用户自定义按键
- 4) 未定义功能，用户自定义按键

11) 红外发射 LED

开发板提供相应的驱动和演示程序源码，具体用法请参考第 5 章。

12) JS9331 核心板

有关该核心板的介绍，请看开发板配套资料中的《JS9331 核心板介绍手册》。

13) Mini USB 转串口调试接口

开发板通过底板板载 USB 转串口芯片（GL850G），实现了核心板的调试串口转到 USB 接口功能。只需外接一根 mini USB 线，即可对开发板进行串口调试、下载程序。

14) 7 pin 2.54 排针接口

从上到下，各个引脚定义（详见底板原理图）依次是

- 1) GPIO22/I2S_MICIN（默认功能为 GPIO）
- 2) GPIO21/I2S_MCK（默认功能为 GPIO）
- 3) GPIO18/I2S_CK（默认功能为 GPIO）
- 4) GPIO19/I2S_WS（默认功能为 GPIO）
- 5) GPIO20/I2S_SD（默认功能为 GPIO）
- 6) GND
- 7) VCC 3.3V 输出

15) WAN 口

默认为 WAN 口，可配置成 LAN 口。支持自动翻转功能。

16) LAN 口

默认为 LAN 口，可配置成 WAN 口（可实现多拨功能）。支持自动翻转功能。

1.4 开发板套件说明

JS9331 开发板配套的配件如下表所示。

型号	数量	备注
JS9331 底板	1	
JS9331 核心板	1	

以太网线	1	
Mini USB 数据线	1	
2.4G 3DBi 棒状天线	1	
5V 2A 电源	1	
红外遥控器	1	
开发板配套透明外壳(选配)	1	
USB 声卡(选配)	1	可做《JS9331 开发板 openwrt 入门教程》中“打造无线音乐播放器”实验。
红外遥控 LED 七彩灯(选配)	1	使用方法在本说明“红外发射接收”一节中有介绍
USB 摄像头(选配)	1	可做《JS9331 开发板 openwrt 入门教程》中“挂载摄像头实现远程监控”实验。

1.5 开发板配套资料说明

下图列出了 JS9331 开发板配套资料主目录。

 JS9331开发板固件镜像安装包	2015/6/10 7:31	文件夹
 JS9331开发板使用手册	2015/6/2 7:56	文件夹
 JS9331开发板硬件原理图	2015/5/30 8:34	文件夹
 JS9331开发板源码	2015/6/10 17:25	文件夹
 开发工具	2015/6/9 7:42	文件夹
 芯片手册	2015/5/31 9:18	文件夹
 学习资料	2015/6/10 8:04	文件夹
 使用前必读.txt	2015/5/30 8:10	文本文档

有关各个目录的内容介绍，请查看各文件夹下的“readme.txt”文件，这里不再赘述。

2. 开发板功能测试

2.1 基本功能测试

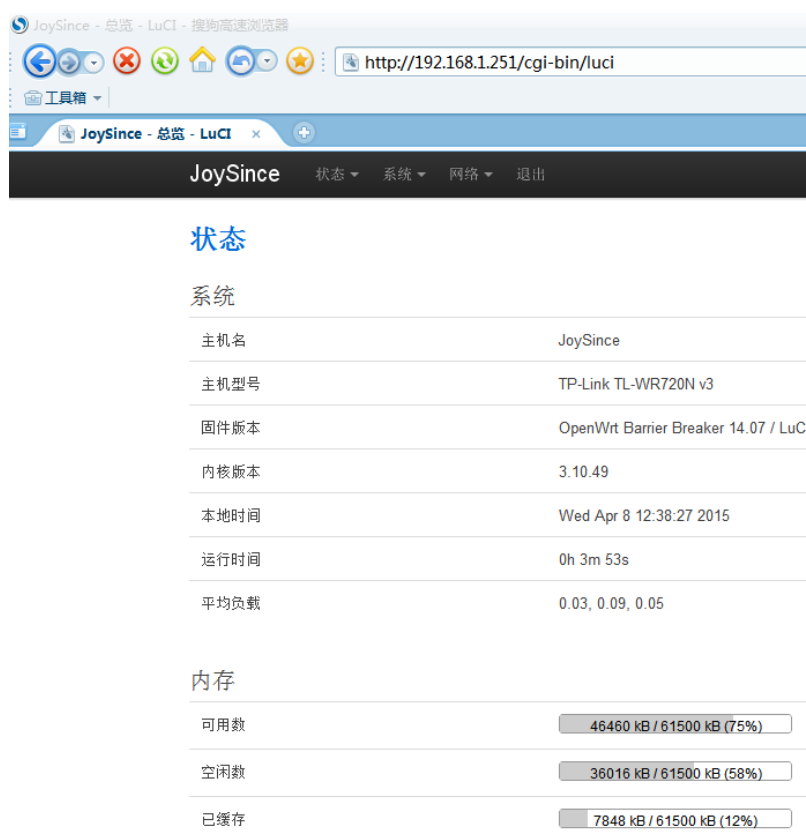
JS9331 开发板出厂时已烧写完整的 openwrt 系统，用户完全可以将本开发板当成普通路由器来使用。本节主要介绍开发板的路由器功能。

2.1.1 开机测试

- 1) 将开发板配套的 5V 电源插到开发板的电源接口，给开发板上电，电源指示灯立刻亮起（如未亮起，请检查电源是否接好）。
- 2) 大约 15 秒后，系统运行指示灯快速闪烁，表示系统正在启动，进入慢闪后标示系统已经启动完成。如果 wifi 开启，则 wifi 指示灯也一并亮起。
- 3) 将开发配套的网线插入开发板 LAN 口和电脑局域网相连，此时开发板上 LAN 指示灯亮起（网线插入到 WAN 口，则 WAN 指示灯亮起，但可能无法登陆系统配置页面）。在网页中输入 192.168.1.251，即可进入开发板配置界面，如下所示。（注意：如无法连接路由器，请检查网线是否插好，电脑和开发板 IP 网段的一致。开发板默认开启 DHCP，可能会与现有局域网网关的 DHCP 功能相冲突，请根据情况开启关闭）



- 4) 用户名和密码，默认都是 root。输入完成后，点击“登录”即可进入系统配置页面。如下图所示。



至此，开机测试完成。

2.1.2 wifi 测试

JS9331 开发板出厂默认 wifi 开启，wifi 名为“JoySince”，未加密。笔记本或手机可以直接连接该名称的 wifi。如果已经将开发板连接到有互联网的网络中，连接 wifi 后，设

备即可上网。

可点击系统配置页面的顶部菜单栏的“网络”->“无线”->“修改”可对 wifi 加密、名称等进行配置。如下图



2.2 开发板高级功能测试

高级功能测试主要有调试串口测试、按键测试、USB 测试、温度传感器测试、红外发射接收测试、拓展串口测试。

2.2.1.1 安装串口驱动

开发板上板载了 USB 转串口系统（采用 ch340 芯片），用于串口调试。此时需要安装相应驱动，该串口才能被系统正确的识别。

将“JS9331 开发板配套资料/开发工具/CH341SER.zip”解压后，运行“SETUP.EXE”完成安装。

2.2.2 安装 SecureCRT

为了实现开发板的串口调试，我们需要安装串口调试软件，常见的串口调试软件有 sscom32、超级终端、串口调试助手、SecureCRT 等。这里我们选择功能比较强大的 SecureCRT，它支持 SSH1, SSH2, Telnet, RLogin, Serial 和 TAPI 协议，界面也很友好。

将开发板提供的软件资料“JS9331 开发板配套资料/开发工具/SecureCRT_CN.rar”直接解压到需要安装的目录中，并运行安装根目录下的“SecureCRT.exe”即可，推荐在桌面创建快捷方式。

2.2.3 安装 Putty(可选)

Putty 和 SecureCRT 一样具有 ssh、telnet、serial 等功能，但是它功能更为精简，无需安装即可使用，请读者自行选择。

读者可以在“JS9331 开发板\项目输出\JS9331 开发板配套资料\开发工具\putty_V0.63.0.0.43510830.exe”找到。

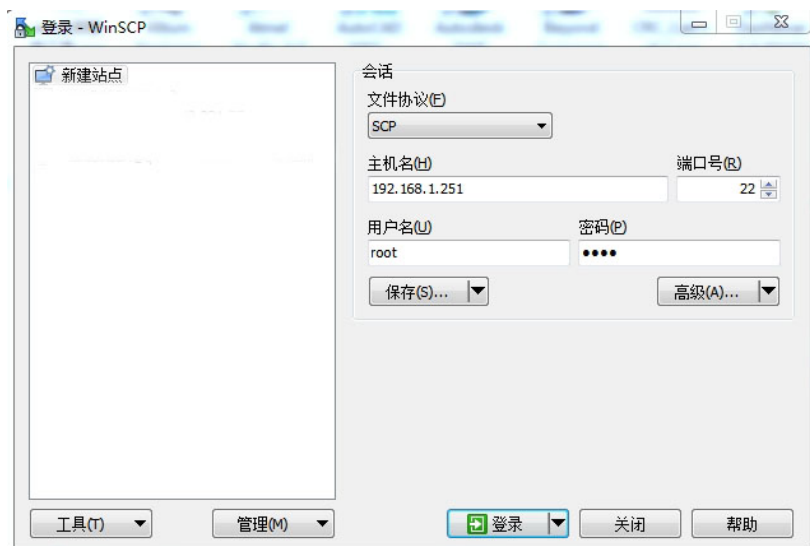
2.2.4 安装 WinScp

为了方便我们需要将文件传输到开发板上，我们需要安装 winscp（基于 SSH）。winscp 安装包在“JS9331 开发板配套资料\开发工具\winscp_V5.7.1.5235_setup.exe”，安装步骤是

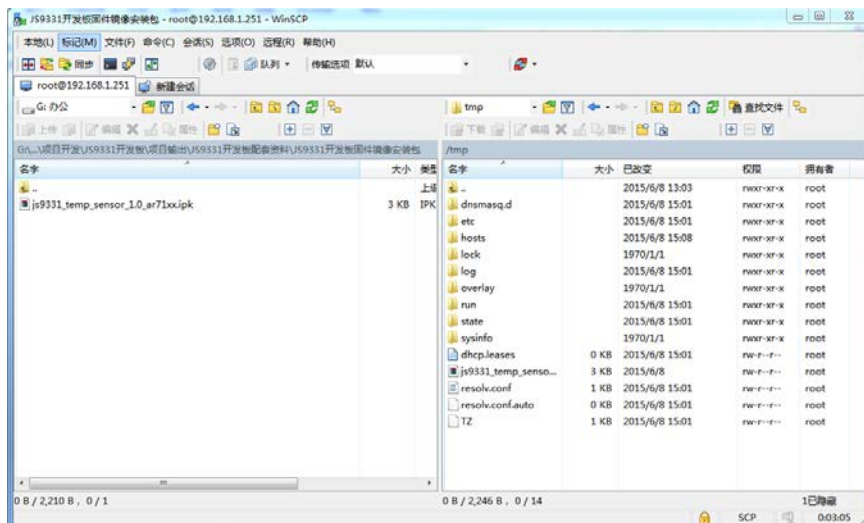
典型的 window 安装，这里不再赘述。安装完成后，点击图标



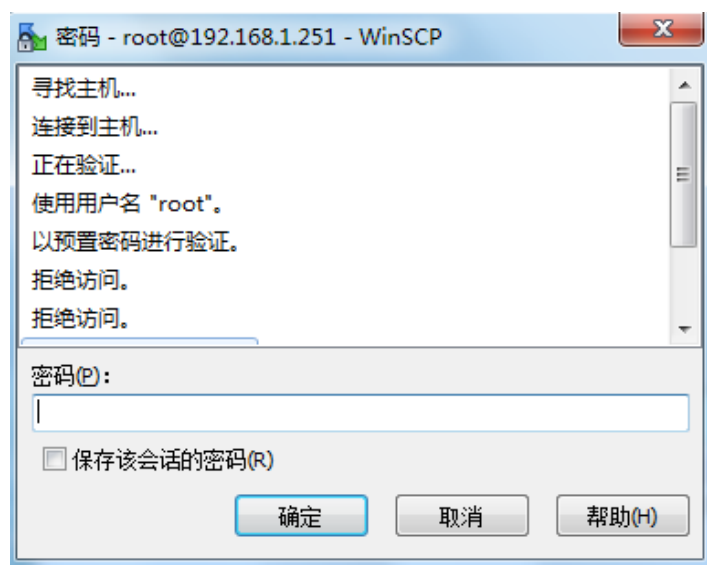
进入 winscp 界面，如下图所示。



输入和上图一致的配置（用户名和密码都是 root），然后点击“登录”。出现类似下图



注意：恢复出厂设置后，openwrt 默认是不开启 ssh 服务器的，此时用 winscp 登录时，如果出现提示下图所示



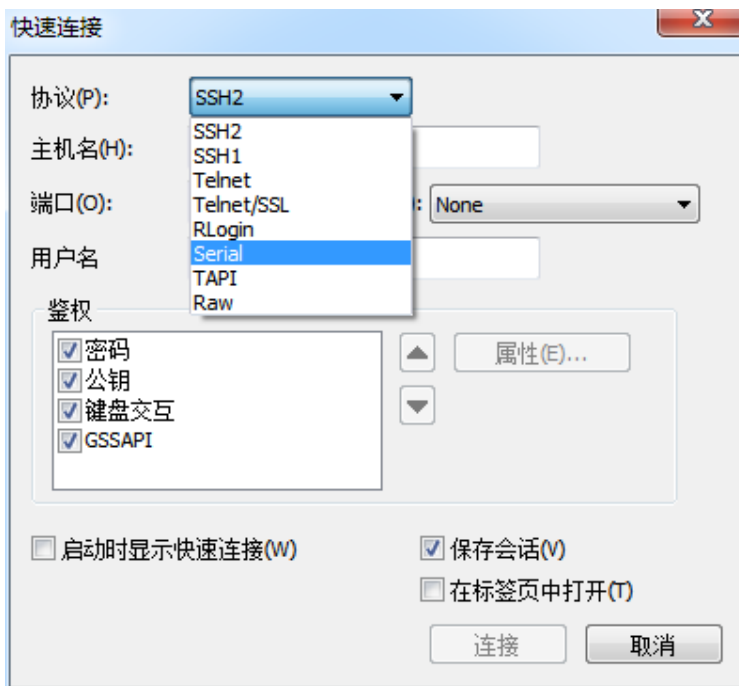
则用户需要到网页设置里面设置一下系统密码即可。

2.2.5 测试调试串口

- 1) 将开发板配套的 mini USB 线插入开发板的 mini USB 接口，另一端连接电脑。（如果未进行 2.2.1 节的步骤，电脑可能会提示无法识别的设备。此时，需要完成 2.2.1 节的步骤方可使用调试串口。）
- 2) 打开之前安装的 SecureCRT，如下图所示。



点击上图所示的红圈按钮所标的“快速连接”，出现下图

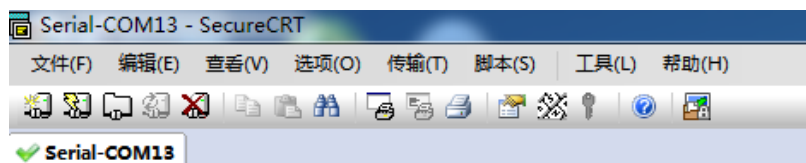


选择红圈中的“Serial”，出现下图



其中，“端口”请根据电脑中实际的串口号设置。其他设置请和上图一致。

注意:上图中的 RTS/CTS 选项一定要去掉,不然会导致终端无法输入
最后点击“连接”。出现下图



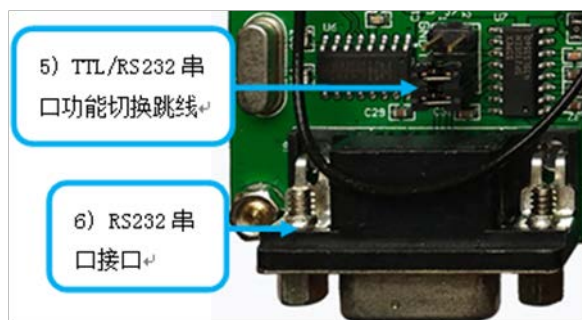
3) 如果开发板尚未上电, 请将开发板配套的 DC 5V 电源连接到开发板, 将其启动。启


```
root@Joysince:/# ls
bin      etc      mnt      proc      root      sys      usr      www
dev      lib      overlay  rom       sbin      tmp      var
```

上图我们利用一个 linux 中非常常用的命令“ls”列出了系统根目录下的文件列表，至此调试串口测试完成。

2.2.6 测试拓展串口

开发板上拓展出了一路串口（“开发板资源”一节中，标号为 5、6 所示）



如果该路串口，在标号“5)”所示位置用跳线帽跳上，那么它可以接 RS232 串口线与外部通信，如果把跳线帽拔掉，可以通过板上所标“TX”、“RX”、“GND”插针与外部进行串口通信（5V）。

下面我们就用 linux 上常用的 minicom 来测试一下该串口。

- 1) 首先搜索目录“JS9331 开发板配套资料\JS9331 开发板固件镜像安装包\openwrt IPK 安装包”找到

“terminfo_5.9-1_ar71xx.ipk”、
“libncurses_5.9-1_ar71xx.ipk”、
“minicom_2.7-1_ar71xx.ipk”

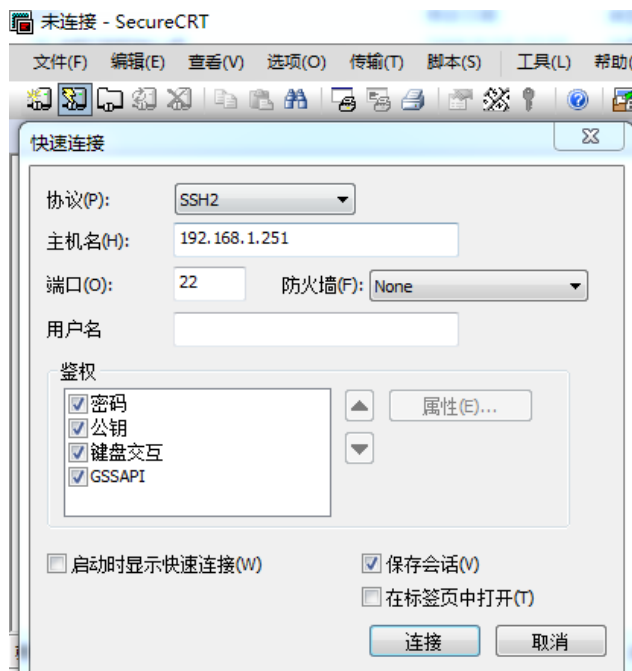
用先前安装过的 winscp，将以上文件复制到开发板的“/tmp”目录下，并用 opkg 安装命令依次安装，安装结果如下图所示。

```
root@Joysince:/# opkg install /tmp/terminfo_5.9-1_ar71xx.ipk
Installing terminfo (5.9-1) to root...
Configuring terminfo.
root@Joysince:/# opkg install /tmp/libncurses_5.9-1_ar71xx.ipk
Installing libncurses (5.9-1) to root...
Configuring libncurses.
root@Joysince:/# opkg install /tmp/minicom_2.7-1_ar71xx.ipk
Installing minicom (2.7-1) to root...
Configuring minicom.
```

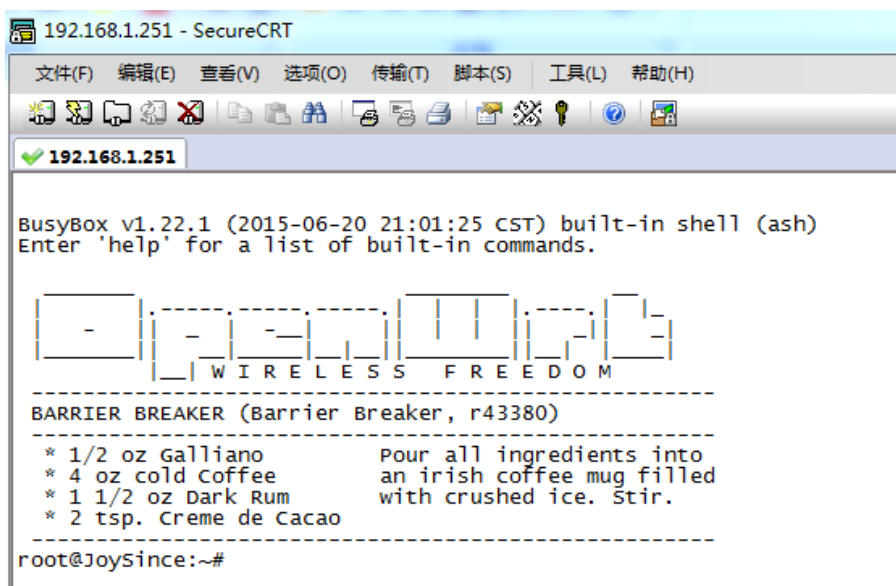
上图表示，“minicom”已经安装成功。

- 2) 我们需要一根 USB 转 RS232 串口线（开发板配件中未有该配件，请读者自备）将其连接到开发板上的 RS232 串口（标号为“6”）接口，并确保此时跳线帽已经接上。新建另外一个 SecurCRT 窗口，并连接该串口号，设置波特率 115200，数据位 8，停止位 1，奇偶校验无，无流控，这里我们称之为通信串口（作者电脑上该串口号为 COM14）。
- 3) 本来我们可以直接通过调试串口终端，打开 minicom 让开发板和通信串口进行通信，但是 minicom 在串口终端下，由于界面显示问题，会导致调试界面“乱码”，影响我们的控制，所以我们这里采用“ssh”来控制 minicom。同样我们采用 SecurCRT，

新建一个 ssh 终端。



如上图所示，我们新建一个协议为“SSH2”的终端，连接到我们的开发板地址192.168.1.251。在随后提示需要输入的用户名和密码都是“root”。连接后如下图所示。



- 4) 在 ssh 终端下我们输入 “minicom -D /dev/ttyUSB0 -b 115200”，然后我们将进入 minicom 控制台。

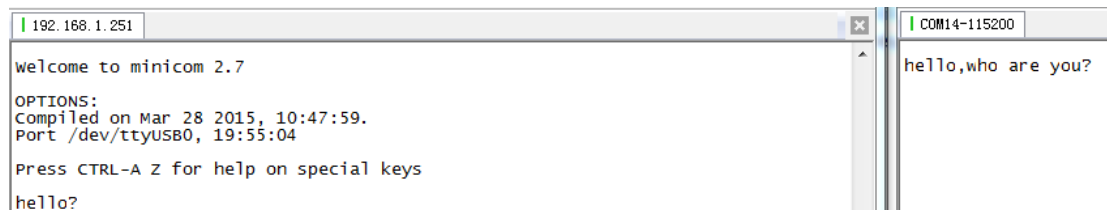

```
192.168.1.251

Welcome to minicom 2.7

OPTIONS:
Compiled on Mar 28 2015, 10:47:59.
Port /dev/ttyUSB0, 19:55:04

Press CTRL-A Z for help on special keys
```

5) 现在我们将COM14的控制终端和ssh控制终端同时显示。在两边测试输入一些东西。



我们输入在右侧窗口“hello?”，结果在左侧窗口显示了。在左侧窗口输入“hello,who are you?”结果在左侧窗口显示了。这表示我们的开发板和外部串口通信成功！

有关该串口的基本测试，这里就结束了，在《JS9331 开发板教程》将介绍“wifi 串口透传”高级应用。

2.2.7 按键测试

开发板上一共有 4 个按键，它们的出厂功能已于 1.3 10) 中介绍过了。短按下第一个按键，系统会重启，如下图所示。

```
Serial-COM13 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)
Serial-COM13

root@Joysince:/# proc: - shutdown -
[ 734.560000] br-lan: port 2(wlan0) entered disabled state
[ 734.560000] br-lan: port 1(eth1) entered disabled state
[ 734.570000] device eth1 left promiscuous mode
[ 734.570000] br-lan: port 1(eth1) entered disabled state
[ 734.590000] eth1: link down
[ 734.590000] IPv6: ADDRCONF(NETDEV_UP): eth1: link is not ready
[ 734.590000] device wlan0 left promiscuous mode
[ 734.600000] br-lan: port 2(wlan0) entered disabled state
proc: - reboot -
[ 734.950000] Removing MTD device #3 (rootfs_data) with use count 1
[ 734.970000] Res
U-Boot 1.1.4 (Apr 6 2015 - 14:18:05)

*****
*                智盛电子                *
*                网站:www.joysince.com        *
*                淘宝地址:joysince.taobao.com *
*****
AP121 (ar9331) U-boot

DRAM: 64 MB
Top of RAM usable for U-Boot at: 84000000
Reserving 139k for U-Boot at: 83fdc000
Reserving 192k for malloc() at: 83fac000
Reserving 44 Bytes for Board Info at: 83fabfd4
Reserving 36 Bytes for Global Data at: 83fabfb0
Reserving 128k for boot params() at: 83f8bfb0
Stack Pointer at: 83f8bf98
```

长按（大于 5s）下第一个按键，系统恢复到出厂设置（慎用），另外 3 个按键未定义功能。

注：开发板上 4 个按键功能均可由用户自行配置，具体方法见《JS9331 开发板 openwrt 入门教程》

提示：另外一种恢复出厂设置的方法是在终端输入“firstboot”命令，该命令执行完成后，重启开发板。

2.2.8 USB 测试

将 USB 设备（比如 U 盘）插入开发板的 USB 接口中（两个 USB 接口任意一个），出现下图所示提示信息

```
root@JoySince:/# [ 256.420000] usb 1-1.1: new high-speed USB device number 4 using ehci-platform
```

出现类似上图提示，表示开发板已经识别该 USB 设备了。

关于如何使用开发板挂载 U 盘，具体方法见《JS9331 开发板 openwrt 入门教程》。

2.2.9 温度传感器测试

开发板上板载有一个 DS18B20 温度传感器（见 1.3 节）。智盛电子开发了一个用于测试该温度传感器的 openwrt 页面（软件开源，见“JS9331 开发板配套资料\JS9331 开发板源码\温度传感器源码”）。下面我们来测试一下温度传感器的功能。

1) 传输温度传感器 IPK 包。

发板出厂默认未安装温度传感器测试程序，需要用户手动安装相应的 IPK 包。安装包位于“JS9331 开发板配套资料\JS9331 开发板固件镜像安装包\温度传感器\js9331_temp_sensor_1.0_ar71xx.ipk”。在 winscp 左侧打开其所在目录，将其拖动到开发板的“/tmp”目录下，即完成了文件的传输。

2) 安装温度传感器 IPK 包。

在串口终端下输入“opkg install /tmp/js9331_temp_sensor_1.0_ar71xx.ipk”即完成安装。如下图

```
root@JoySince:/# opkg install /tmp/js9331_temp_sensor_1.0_ar71xx.ipk
Installing js9331_temp_sensor (1.0) to root...
Configuring js9331_temp_sensor.
```

安装完成后，重启开发板，以使配置生效。

3) 测试温度传感器。

重启完成，登录开发板配置页面，顶板菜单栏出现“服务”菜单，点击下拉的“温度传感器”选项，即出现下面温度传感器页面。



其中“传感器温度”即为 DS18B20 的实际温度，其他的请看配置页面上的文字注释。

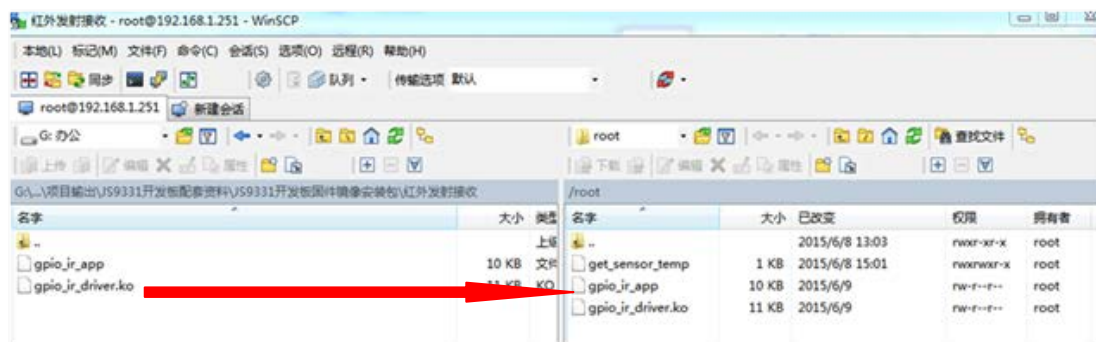
2.2.10 红外发射接收测试

开发板上板载有 1 个红外接收头（见 1.3 节 8）和 4 个红外发射灯（见 1.3 节 11）。为了与红外发射接收硬件配套，智盛电子专门编写了驱动和应用程序（软件开源，见“JS9331 开发板配套资料\JS9331 开发板源码\红外接收发送源码”），下面我们来测试一下它们的功能。

1) 复制红外发射接收软件。

开发板出厂默认未安装红外发射接收测试程序，需要用户手动安装相应的内核模块和执行文件。

用 winscp 将“JS9331 开发板配套资料\JS9331 开发板固件镜像安装包\红外发射接收\”目录下的“gpio_ir_app”、“gpio_ir_driver.ko”拷贝到“/root”目录下。如下图所示。



2) 安装测试

执行以下命令

```
insmod /root/gpio_ir_driver.ko          //将红外接收发送模块插入内核
chmod +x /root/gpio_ir_app              //将红外接收发送应用程序加上可执行权限
/root/gpio_ir_app&                       //让红外接收发送应用程序在后台运行
```

执行结果如下图所示。

```
root@Joysince:/# insmod /root/gpio_ir_driver.ko
[ 1080.030000] **gpio_ir module initiation OK**
root@Joysince:/# chmod +x /root/gpio_ir_app
root@Joysince:/# /root/gpio_ir_app&
root@Joysince:/# **/tmp/gpio_ir_fifo[ 1098.820000] ***gpio_ir_open***
[ 1098.820000] **gpio_ir_out_pin:0**
[ 1098.820000] **gpio_ir_in_pin:12**
o not exit,now create it**

root@Joysince:/# █
```

3) 开发板红外接收测试

执行命令

```
echo 'start_ir_receive' > /tmp/gpio_ir_fifo //启动接收功能
```

然后将开发板配套的红外遥控器，对着开发板的红外接收头，按下红外遥控器的任意按键，串口终端会出现类似下图所示信息（以下信息会根据按键的不同而变化）。

```
root@Joysince:/# ***get nec code_data:0xF50AFF00***
***address:0x00***
***not_address:0xFF***
***command:0x0A***
***not_command:0xF5***
***get repeat code***
***get nec code_data:0xF50AFF00***
***address:0x00***
***not_address:0xFF***
***command:0x0A***
***not_command:0xF5***
```

如果一直按则会类似下图。

```
***get nec code_data:0xFA05FF00***
***address:0x00***
***not_address:0xFF***
***command:0x05***
***not_command:0xFA***
***get repeat code***
***get repeat code***
***get repeat code***
***get repeat code***
***get repeat code***
```

其中“***address:0x00***”即为 NEC 中的地址码 (address), “***not_address:0xFF***”即为 NEC 中的地址反码 (address), 其他同理。“***get repeat code***”是接收到了“重复码 (repeat code)”

4) 开发板红外发送测试

将开发板配套的红外遥控灯 (该配件为选配, 如果需要请到智盛电子店铺中购买) 对着开发板 (的红外发射管), 确保中间无遮挡物, 并将其通电。

执行命令

```
echo "send_ir_data 0xF807FF00" > /tmp/gpio_ir_fifo //开发板发送红外编码 “开灯”
echo "send_ir_data 0xF906FF00" > /tmp/gpio_ir_fifo // “关灯”
echo "send_ir_data 0xF609FF00" > /tmp/gpio_ir_fifo // “红色”
echo "send_ir_data 0xF708FF00" > /tmp/gpio_ir_fifo // “绿色”
echo "send_ir_data 0xF50AFF00" > /tmp/gpio_ir_fifo // “蓝色”
echo "send_ir_data 0xF40BFF00" > /tmp/gpio_ir_fifo // “白色”
.....
```

更多按键编码请读者按照“开发板红外接收测试”一节中介绍的方法获取。如果操作正确, 在各个命令的执行后, 可以看到灯的颜色发生了相应的变化。



有兴趣的读者, 可以尝试通过类似的办法读取电视遥控器编码, 然后控制电视机。

有关 NEC 协议, 请参考开发资料中的“JS9331 开发板配套资料\学习资料\红外遥控协议\NEC_Protocol-NEC 协议.doc”。

3. 开发板固件镜像烧录说明

3.1 openwrt 固件烧写说明

这里介绍两种 openwrt 固件烧写方法, 一种是串口+网络烧写, 另一种是网络烧写。下面我们就以在 windows 平台下为例, 介绍如何进行烧写 openwrt 固件, 读者根据实际情况选择。**注意, 开发板出厂时已烧录了完整的 openwrt 固件, 一般情况无需再次烧写。**

3.1.1 openwrt 网络烧写

这种方法适用于 openwrt 系统可以正常启动的情况,如果系统无法正常启动,请采用“串口+网络烧写”方法。

- 1) 登录开发板地址 “192.168.1.251”
- 2) 点击系统配置页面的顶部菜单“系统”->“备份/升级”

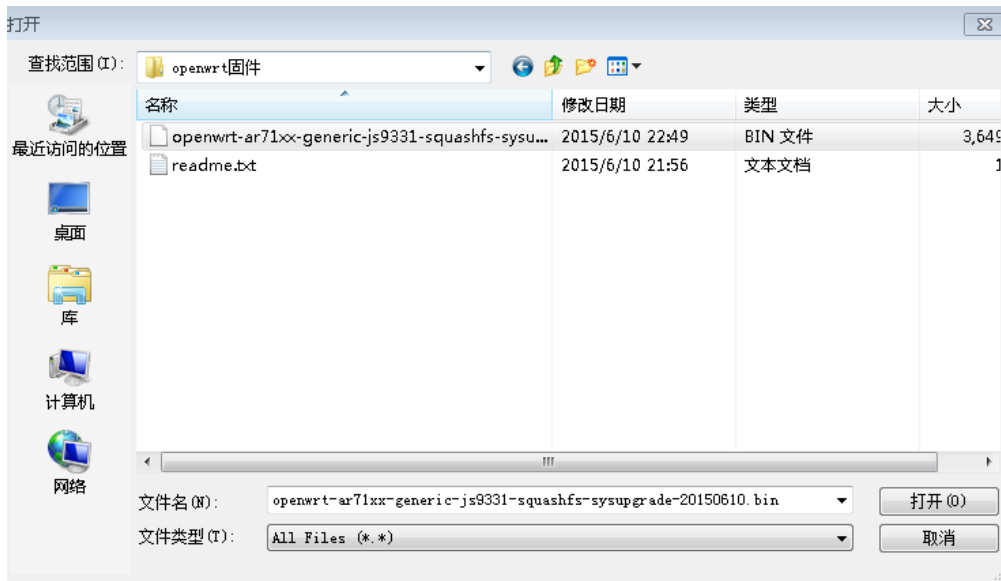


- 3) 选择页面下的“固件文件”一栏“选择文件”按钮,如下图红圈所示。



提示: 去掉“保留配置”勾选,则系统刷新固件后,则不保留之前所做的更改。如果勾选,则保留。

- 4) 在随后跳出的文件选择页面中,选中要上传的固件,并点击“打开”,如下图所示



- 5) 点击“刷写固件”
- 6) 如果固件正确,则会出现下图提示



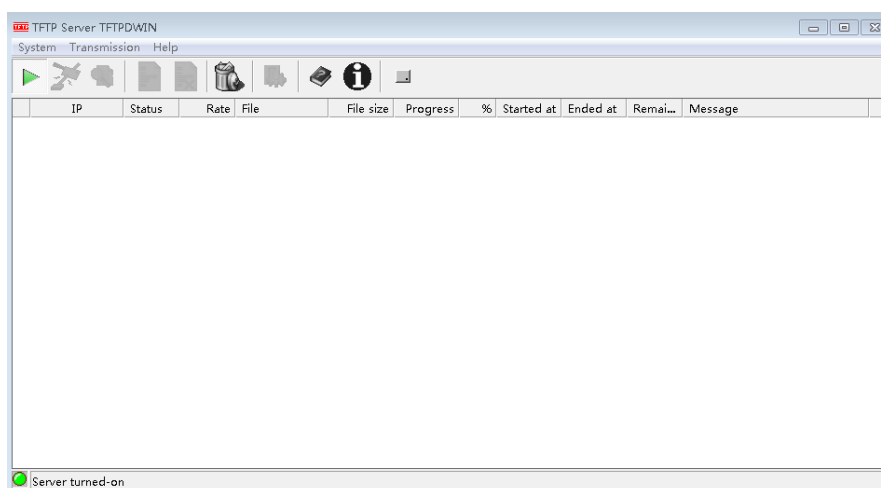
点击上图右下角“执行”，开始烧写固件。等待几分钟后系统刷写完成，系统自动重启。




3.1.2 安装 tftp server

1) 安装 tftp 软件。

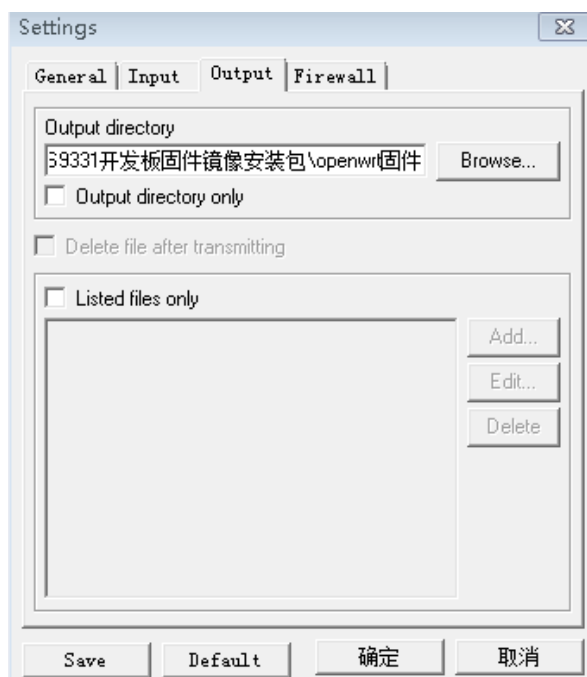
安装包在“JS9331 开发板配套资料\开发工具\TFTP_Server.rar”。安装完成后双

击图标 ，开启 tftp 服务器。

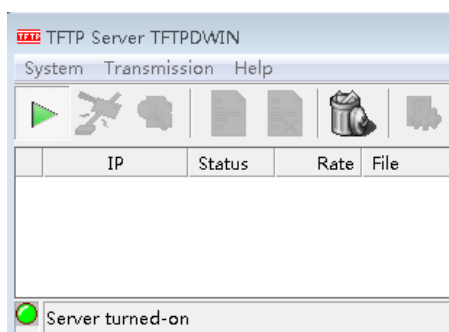


在对其进行配置前，需关闭 TFTP 功能，点击 ，使其变成 ，然后点击  进行服务器配置。

读者需要根据实际情况将下图所示的“Output directory”设置为 openwrt 固件所在目录。



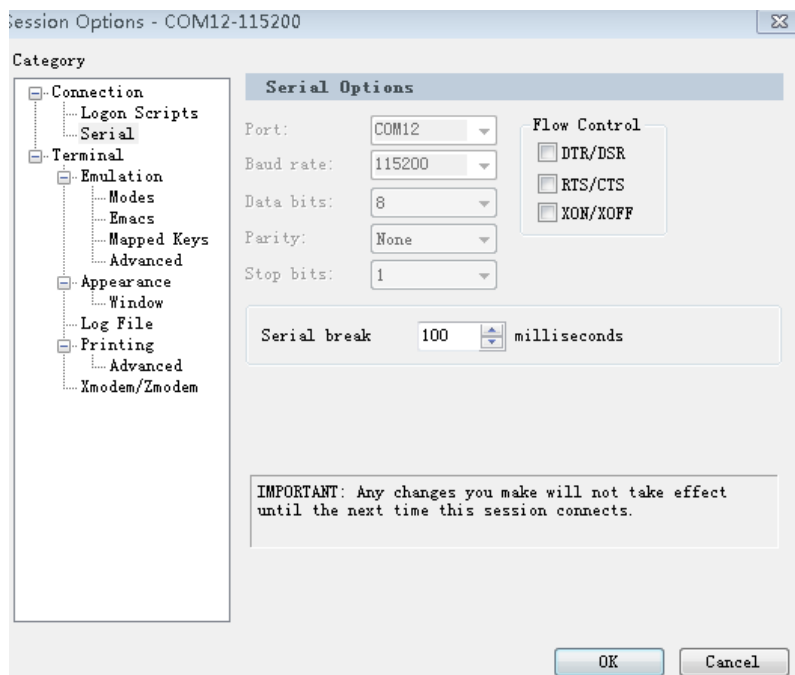
然后点击，“确定”，重新点击  使其变成  启动 tftp 服务器。



3.1.3 openwrt 串口+网络烧写

这种方法适用于系统已烧写了 uboot，还未烧写 openwrt 或由于误操作导致 openwrt 无法使用的情况。首先将开发板和 PC 用 mini USB 线、网线相连接。

- 1) 正确设置 tftp server，启动 tftp server 服务。
- 2) 打开 SecureCRT，并连接相应串口。设置串口如下图



上图中串口号需根据实际情况而定，其他配置请和图中一致。

- 3) 给开发板上电（如果开发板已上电，请重新上电），如果操作正确将出现下图



请在 Uboot 倒数结束前，敲下回车。从而进入 uboot 控制台。

- 4) 执行下列几条指令。

```
setenv serverip 192.168.1.11 根据实际情况设置 PC 端 IP 地址
setenv ipaddr 192.168.1.251 设置开发板 IP
```



```
tftp 0x80002000 openwrt-ar71xx-generic-j9331-squashfs-sysupgrade-20150610.bin
```

erase 0x9f020000 +0x400000 可根据实际固件大小调整，flash 擦除区域大小

cp.b 0x80002000 0x9f020000 0x400000

效果如下图所示

```
ar7240> setenv serverip 192.168.1.110
ar7240> setenv ipaddr 192.168.1.251
ar7240> tftp 0x80002000 openwrt-ar71xx-generic-j9331-squashfs-sysupgrade-20150610.bin
eth0 link down
FAIL
dup 1 speed 1000
Using eth1 device
TFTP from server 192.168.1.110; our IP address is 192.168.1.251
Filename 'openwrt-ar71xx-generic-j9331-squashfs-sysupgrade-20150610.bin'.
Load address: 0x80002000
Loading: #####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
done
Bytes transferred = 3735556 (390004 hex)
ar7240> erase 0x9f020000 +0x400000
Erase Flash from 0x9f020000 to 0x9f41ffff in Bank # 1
First 0x2 last 0x41 sector size 0x10000
65
Erased 64 sectors
ar7240> cp.b 0x80002000 0x9f020000 0x400000
Copy to Flash... write addr: 9f020000
done
```

注意:如果电脑防火墙开启,可能会导致 `tftp` 服务无法进行文件传输,建议关闭电脑防火墙。

最后，执行“reset”开发板重启，直至进入 openwrt 系统。

```
BusyBox v1.22.1 (2015-05-02 11:53:13 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.
```

```
- - - W I R E L E S S F R E E D O M
```

```
BARRIER BREAKER (Barrier Breaker, r43380)
```

```
* 1/2 oz Galliano      Pour all ingredients into  
* 4 oz cold Coffee     an irish coffee mug filled  
* 1 1/2 oz Dark Rum    with crushed ice. Stir.  
* 2 tsp. Creme de Cacao
```

```
root@JoySinc:/# [   32.700000] br-lan: port 2(wlan0) entered forwarding state  
root@JoySinc:/#
```

3.2 uboot 烧写说明

注意，开发板出厂时已烧录了完整的固件，除非智盛电子发布新版本的 JS9331 开发板 uboot，其他情况，一般情况无需再次烧写 uboot。请不要烧写除了 JS9331 开发板配套的 uboot 以外其他的 uboot，否则，有可能使开发板“变砖”。

下面介绍 JS9331 开发板 uboot 烧写步骤。

- 1) 初始步骤同 3.1.3 节 1) ~3), 这里不再赘述。
- 2) 执行以下命令

```
setenv serverip 192.168.1.110
```

根据实际情况设置 PC 端 IP 地址

```
setenv ipaddr 192.168.1.251
```

设置开发板 IP

```
tftp 0x80002000 tuboot 8M 20150528 18.bin
```

```
erase 0x9f000000 +0x10000
```

可根据实际镜像大小调整 flash 擦除区域大小

```
cp.b 0x80002000 0x9f000000 0x10000
```

效果如下图所示。

```

ar7240> setenv serverip 192.168.1.110
ar7240> setenv ipaddr 192.168.1.251
ar7240> tftp 0x80002000 tuboot_8M_20150528.bin
eth0 link down
FAIL
Using eth1 device
TFTP from server 192.168.1.110; our IP address is 192.168.1.251
Filename 'tuboot_8M_20150528.bin'.
Load address: 0x80002000
Loading: #####
done
Bytes transferred = 63158 (f6b6 hex)
ar7240> erase 0x9f000000 +0x10000
Erase Flash from 0x9f000000 to 0x9f00ffff in Bank # 1
First 0x0 last 0x0 sector size 0x10000
0
Erased 1 sectors
ar7240> cp.b 0x80002000 0x9f000000 0x10000
Copy to Flash... write addr: 9f000000
done

```

最后，执行“reset”开发板重启，直至进入 openwrt 系统。

注:16MB flash 的 uboot 烧写命令和上面介绍的一致。

3.3 ART 烧写说明

开发板出厂时已烧录了完整的固件包括 ART，一般不用再次烧写。这里介绍 JS9331 开发 ART (Atheros Radio Test) 分区的烧写，作为参考。

- 1) 初始步骤同 3.1.3 节 1) ~3)，这里不再赘述。
- 2) **注意：16MB flash 的开发板和 8MB flash 开发板 ART 区域存储位置不同。**

8MB flash 的开发板执行以下命令

```

setenv serverip 192.168.1.110 根据实际情况设置 PC 端 IP 地址
setenv ipaddr 192.168.1.251 设置开发板 IP
tftp 0x80002000 js9331_art.bin
erase 0x9f7f0000 +0x10000
cp.b 0x80002000 0x9f7f0000 0x10000

```

16MB flash 的开发板执行以下命令

```

setenv serverip 192.168.1.110 根据实际情况设置 PC 端 IP 地址
setenv ipaddr 192.168.1.251 设置开发板 IP
tftp 0x80002000 js9331_art.bin
erase 0x9fff0000 +0x10000
cp.b 0x80002000 0x9fff0000 0x10000

```

8MB flash 烧写 ART，执行效果如下图所示。

```

ar7240> setenv serverip 192.168.1.110
ar7240> setenv ipaddr 192.168.1.251
ar7240> tftp 0x80002000 js9331_art.bin
eth0 link down
FAIL
dup 1 speed 1000
Using eth1 device
TFTP from server 192.168.1.110; our IP address is 192.168.1.251
Filename 'js9331_art.bin'.
Load address: 0x80002000
Loading: #####
done
Bytes transferred = 65536 (10000 hex)
ar7240> erase 0x9f7f0000 +0x10000
Erase Flash from 0x9f7f0000 to 0xffffffff in Bank # 1
First 0x7f last 0x7f sector size 0x10000
127
Erased 1 sectors
ar7240> cp.b 0x80002000 0x9f7f0000 0x10000
Copy to Flash... write addr: 9f7f0000
done

```

ART 烧写完成。

3.4 修改默认 MAC 地址

有时为了避免 MAC 地址冲突或其他原因，用户需要更改开发板默认的 MAC 地址。在 openwrt 系统里面有两种修改 MAC 地址的方式，下面介绍这两种方式。

方法 1:

在 uboot 里面设置 MAC 地址。进入 uboot，执行命令 "setmac xx:xx:xx:xx:xx:xx"（“xx:xx:xx:xx:xx:xx”为 MAC 地址）如下图所示。

```
ar7240> setmac 00:ca:01:00:00:03
** Notice:
   you should always make a backup of your device
   entire FLASH content before making any changes

Executing: erase 0x9F010000 +0x10000; cp.b 0x80800000 0x9F010000 0x10000

Erase Flash from 0x9F010000 to 0x9F01ffff in Bank # 1
First 0x1 last 0x1 sector size 0x10000
1
Erased 1 sectors
Copy to Flash... write addr: 9f010000
done
ar7240>
```

如上图所示，本例中设置起始 MAC 地址为“00:CA:01:00:00:03”，只需设置一个 MAC 地址，在 openwrt 系统启动后，系统自动按照升序给各个网卡分配 MAC 地址，总共 3 个 MAC 地址，如下图所示。

```
root@JoySince:/# ifconfig
br-lan Link encap:Ethernet Hwaddr 00:CA:01:00:00:05
       inet addr:192.168.1.251 Bcast:192.168.1.255 Mask:255.255.255.0
       inet6 addr: fd02:f0aa:c9c0::1/60 Scope:Global
       inet6 addr: fe80::2ca:1ff:fe00:5/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
       RX packets:0 errors:0 dropped:0 overruns:0 frame:0
       TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:0
       RX bytes:0 (0.0 B) TX bytes:796 (796.0 B)

eth0 Link encap:Ethernet Hwaddr 00:CA:01:00:00:04
      UP BROADCAST MULTICAST MTU:1500 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
      Interrupt:4

eth1 Link encap:Ethernet Hwaddr 00:CA:01:00:00:05
      UP BROADCAST MULTICAST MTU:1500 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
      Interrupt:5

lo Link encap:Local Loopback
   inet addr:127.0.0.1 Mask:255.0.0.0
   inet6 addr: ::1/128 Scope:Host
   UP LOOPBACK RUNNING MTU:65536 Metric:1
   RX packets:98 errors:0 dropped:0 overruns:0 frame:0
   TX packets:98 errors:0 dropped:0 overruns:0 carrier:0
   collisions:0 txqueuelen:0
   RX bytes:7296 (7.1 KiB) TX bytes:7296 (7.1 KiB)

wlan0 Link encap:Ethernet Hwaddr 00:CA:01:00:00:03
       inet6 addr: fe80::2ca:1ff:fe00:3/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

由上图可看到

无线网卡“wlan0”的 MAC 地址为“00:CA:01:00:00:03”

有线网卡“eth0”的 MAC 地址为“00:CA:01:00:00:04”

有线网卡“eth1”的 MAC 地址为“00:CA:01:00:00:05”

方法 2:

修改 openwrt 的“/etc/config/network”文件。在“config interface 'lan'”和“config

interface 'wan' ”下面分别添加“macaddr”字段，指定 LAN 口和 WAN 口的 MAC 地址。如下图所示。

```
config interface 'lan'
    option ifname 'eth1'
    option force_link '1'
    option type 'bridge'
    option proto 'static'
    option ipaddr '192.168.1.252'
    option netmask '255.255.255.0'
    option ip6assign '60'
    option macaddr '00:ca:01:0e:00:08'

config interface 'wan'
    option ifname 'eth0'
    option proto 'dhcp'
    option macaddr '00:ca:01:0e:00:01'
```

在“/etc/config/wireless”文件中的“config wifi-iface”下面添加“macaddr”字段，指定无线的 MAC 地址，如下图所示。

```
config wifi-iface
    option ssid 'openwrt'
    option encryption 'none'
    option device 'radio0'
    option mode 'sta'
    option network 'wan'
    option macaddr '00:ca:0f:00:00:01'
```

修改完成后执行“/etc/init.d/network restart”命令，使之前的网络配置生效。执行 ifconfig 查看各网卡信息，可以看到之前的配置已生效。

```
eth0      Link encap:Ethernet  HWaddr 00:CA:01:0E:00:01
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:4

eth1      Link encap:Ethernet  HWaddr 00:CA:01:00:00:05
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:3013 errors:0 dropped:18 overruns:0 frame:0
          TX packets:1653 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:351694 (343.4 KiB)  TX bytes:926831 (905.1 KiB)
          Interrupt:5

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2636 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2636 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:182821 (178.5 KiB)  TX bytes:182821 (178.5 KiB)

wlan0     Link encap:Ethernet  HWaddr 00:CA:0F:00:00:01
          inet addr:192.168.1.225  Bcast:192.168.1.255  Mask:255.255.25
          inet6 addr: fe80::2ca:fff:fe00:1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

注意：不建议用此方法修改 MAC 地址，因为用这种方式设置的 MAC 地址，在开发板恢复出厂设置后随即失效，并且在操作的时候会因为删除接口而消失。

4. 常见问题及解答

待添加。

5. 修改说明

版本	时间	修改说明
V1.0	2015.06.01	JS9331 开发板使用手册初始版本

V1.1	2015.07.06	增加了拓展串口的说明。修正了一些错误的说明。
V1.2	2015.09.01	增加了 tftp 传输有关防火墙的说明。增加了开发板红外发射控制的说明。修正了一些其他说明。增加了有关 64 位 ubuntu 无法编译通过的说明。
V1.3	2015.11.01	增加恢复出厂设置后,winscp 无法登陆的说明,和其他一些问题的说明。增加了修改 MAC 地址的方法。
V1.4	2016.02.15	增加了 16MB flash ART 的烧写方法。增加多种恢复出厂设置的说明。