

Banana Pi Router - BPI-R1

Manual for HW setup and basic router functionalities

This manual is OLD, you have it much easier and more fun with this new release : [armbian 4.2 k3.4.108 -01](#)

Table of contents

[Information](#)

[About the BPI-R1](#)

[Reading this guide](#)

[Instruction about how to use this manual](#)

[Installing the OS](#)

[with LINUX or Mac OSX](#)

[with WINDOWS](#)

[Prerequisites](#)

[Preparation](#)

[Make sure the system is up to date:](#)

[Install needed software:](#)

[Some commands you might find useful](#)

[nano text-editor tuning](#)

[Backup 2GB Image](#)

[Configuration files](#)

[swconfig](#)

[wpa supplicant](#)

[isc-dhcp](#)

[Alternative: Using dnsmasq](#)

[interfaces](#)

[IPtables](#)

[To secure a little bit the router:](#)

[before you go for compilation \(hostapd & driver\)](#)

[compile from Git](#)

[Compile RealTek WiFi driver & hostapd](#)

[hostapd](#)

[Internet resources](#)

version	date	change
0.6	06. May	isc-dhcp & RealTek added more information
0.7	08. June	added nethogs, intro for beginner, warning overheating
0.8	09. Aug	updated Information & Internet resources
0.9	27. Sep	isc-dhcp added 'routers'
0.10	06. Oct	isc-dhcp & interfaces updated
0.11	08. Dec '15	currently ongoing changes, by now added back 'dnsmasq'

Please help, add your comments, suggestions - just do it :)

Information

The BPi-R1 is a special version in the family of Banana Pis. This document is intended to provide and share information on how to make use of its full router functionality. As is the norm, all of the information here should not be considered absolute. It is a guide only.

About the BPi-R1

The BPi-R1 uses a A20 SoC (System on Chip) in junction with a BCM53125 PHY (7 port ethernet chip) to provide 5 physical external ethernet ports. A further 6th port is used by the A20 itself. So physically all 6 external ethernet-ports are interconnected.

Reading this guide

The physical layout of the ethernet ports doesn't follow the logical numbering for the ports. The logical numbering goes thusly:

3, 0, 1, 2, 4

Where port 3 is the port that is physically slightly separated from the rest. For the purposes of this guide, port 3 will be the WAN (Wide Area Network) and ports 0, 1, 2 and 4 will be the LAN ports. The WLAN (Wireless Local Area Network) will be bridged with the LAN ports. This functionality can be accomplished without the use of the NAT (Network Address Translation) is also covered in this guide.

If you want to use it like a classical router with WAN and LAN you must separate it in the configuration. Therefore you have to create a virtual-LAN (VLAN) and a bridge (br0) to connect the VLAN and WLAN. No! firewall configuration with these steps.

Let's think about if we have everything:

Software:

1. driver wireless RealTek
2. driver BCM53125
3. 8021q = VLAN module
4. swconfig = to control BCM53125 switch (from openWRT) /usr/sbin
5. bridge-utils = bring WiFi & Ethernet together
6. isc-dhcp = dhcp server
7. hostapd = WiFi security /usr/sbin/
8. hostapd_cli
9. IPtables = the Kernel needs to know how to route the traffic

Configuration:

1. swconfig
2. isc-dhcp
3. interfaces
4. IPtables
5. hostapd

I read somewhere that the MALI GPU does not play nice with other Kernel than 3.4 (Feb. 2015)

Instruction about how to use this manual

By accident you might insert or delete something, then simply press: **ctrl + z**

Installing the OS

The following section covers installing the Bananian 15.04 on a SDcard.

You will download and install a fresh [Bananian 15.04](#) on your SDcard (based on Debian 7 wheezy), my old SDcard from 2010 wouldn't boot. Make sure you have two to test in case one doesn't work. The SDcard should be at least 4GB big.

with LINUX or Mac OS X

After you have downloaded the .zip file from the link above, you will create a folder on your PC and unzip /unrar the files with your distributions archive program.

This will take some time as it is quite a large file. Go to the folder with the extracted files (you can try in your file-manager a 'right-click' to open a terminal straight from this location).

Or simply open a terminal (usually Ctrl-Alt-T in most Linux desktop environments. In Mac OS X, it is located in Applications > Utilities) and go to the directory in which you keep the extracted files.

the image file: `bananian-1504.img`

Insert your SDcard into the card reader

write the image to the SDcard;

As you can see red highlighted, you have to tell the PC to which device /SDcard it shall write to, the number of the partition is not necessary. To find this information you can try the command: `df -h`

```
sudo dd if=bananian-1504.img of=/dev/sdj bs=1M && sync
```

After entering your password, you will not see changes to the terminal until it is finished. The time it takes is largely dependent on the speed of the SDcard and SDcard reader (ca 2 min). In other words, be patient.

So you have the time to get some [water](#), the following steps are a complete walk-through and I recommend to work through without interruption. At the end you have a working system.

When dd has finished you can remove the card. The sync command ensures all information has been written to the SD Card (and not just cached in RAM).

Put the SDcard into your BPi-R1 and turn on the power.

with WINDOWS

After you have downloaded the .zip file from the link above, you will create a folder on your PC and unzip with 'right-click' extract.

your extract: `bananian-1504.img`

Download the [Etcher Disk Imager](#) be careful not install anything else

Insert your SD card into the card reader, start Etcher.

Prerequisites

You'll need the IP address of your BPi-R1.

- If you're sitting in front of your Pi with a monitor plugged in, plug a network cable in and type:
`ip r`
- If you're doing this via ssh (or [PuTTY](#) in Windows), you'll need some other means like using your router's web frontend.

Check internet connection: `ping www.google.com` (stop with `Ctrl + c`).

Preparation

Make sure the system is up to date:

```
apt-get update
apt-get upgrade
```

Run `bananian-config`

Your current hardware configuration is: BananaPi (change this)

Do you want to expand the root file system (No) - faster backup.

Install needed software:

What is already installed ?

```
dpkg -l hostapd vlan bridge-utils iptables-persistent nethogs isc-*
dnsmasq
```

Install needed packages:

```
apt-get install bridge-utils iptables-persistent nethogs (say 'yes' twice)
```

Explanation:

bridge-utils - We'll use this to set up our bridge between the wireless LAN and LAN.

iptables-persistent - Make your firewall rules persistent

nethogs - see a summary of traffic usage

Reboot

```
shutdown -r now
```

To copy files use an FTP-client (i.e. gFTP) on your PC and SSH2 or SFTP on Banana Pi

Some commands you might find useful

`lsmod` (Show the status of modules in the Linux Kernel)

`ps ax` (to see every running process of the system)

Check your log files:

`less /var/log/syslog` (type capital G to jump to last entry - q to exit)

`less /var/log/messages`

`tail -f /var/log/syslog` (open an additional ssh and let it run to see the current logfile)

You can check the dpkg log files to see which packages you installed recently:

`grep 'install*' /var/log/dpkg.log`

list all partitions

`lsblk -o NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL`

nano text-editor tuning

We will use the nano text editor often and it helps to have different colors in the config files.

You can turn it **on** or **off** at anytime with: Alt + y

create nano /usr/share/nano/conf.nanorc **and add these lines:**

```
# config file highlighting

syntax "conf"
"(\.(conf|config|cfg|cnf|rc|lst|list|defs|ini|desktop|mime|types|preset|
cache|seat|service|htaccess)$|(^|/)(\w*crontab|mirrorlist|group|hosts|pa
sswd|rpc|netconfig|shadow|fstab|inittab|inputrc|protocols|sudoers|interf
aces|swconfig|hostapd)$|conf.d|\.config/)"

# default text
color magenta "^.*$"
# special values
icolor brightblue "(^|\s|=)(default|true|false|on|off|yes|no)(\s|$)"
# keys
icolor cyan "^\s*(set\s+)?[A-Z0-9_\/\.\%\@+-]+\s*([:]\|>)"
# commands
color blue "^\s*set\s+\<"
# punctuation
color blue "[.]"
# numbers
color red "(^|\s|[[/:|<>(){}=,]|\])[-+]?[0-9](\.[0-9])%?($|\>)"
# keys
icolor cyan "^\s*(\${if })?([A-Z0-9_\/\.\%\@+-]|\s)+="
# punctuation
color blue "/"
color green "[\]|[( )<>[{ },;:=])"
color brightwhite "(^|\\[|\\(|\\:)\s*-(\s|$)"
# section headings
icolor brightyellow "^\s*(\\([([A-Z0-9_\\. -]|\s)+\\))+\s*$"
color brightcyan "^\s*((Sub)?Section\s*(=|\>)|End(Sub)?Section\s*$)"
color brightcyan "^\s*\$(end)?if(\s|$)"
# URLs
icolor green "\b(([A-Z]+://|www[.]) [A-Z0-9/:#?&$=_\.\-]+)(\b|$| )"
# XML-like tags
icolor brightcyan
"</?\w+((\s*\w+\s*=)?\s*("[^"]*"|'['']*'|!?\s*[A-Z0-9_:/]))*(\s*/)?>"
# strings
color yellow "\"(\\.|[^\"])*\""'(\\.|[^\'])*'"
# comments
color white "#.*$"
color blue "^\\s*##.*$"
color white "^;.*$"
color white start="<!--" end="-->"
```

Ctrl + o **to save the file**, Ctrl + x **to close the application.**

Then you include this configuration to nano. Go to the last line of nano /etc/nanorc file and add:

```
## Configuration files (catch-all syntax)
include "/usr/share/nano/conf.nanorc"
```

In the first codeline, at the end you can add more filename if you miss some.
And now it is much easier to read configuration files, as long as you have no inabilities to see colors.

Backup 2GB Image

with Linux

`shutdown -h now` to turn off the Pi - place the SDcard in your PCs CardReader.

`dd` will not make an image of the size of the sum of the two partitions, because it has no idea that there are partitions. It is fed input data, it delivers output data. It does no interpretation at all.

Go to the folder where you want to backup the image, (you can try in your file-manager a 'right-click' to open a terminal straight from this location).

```
fdisk -l
```

i.e. Bananian on 16GB Stick

Gerät	boot.	Begin	End	Block	Id	System
/dev/sdj1		2048	43007	20480	83	Linux
/dev/sdj2		43008	3710936	8141824	83	Linux

sector size is given, 512 bytes

So you have to find the last Sector and let `dd` just copy necessary Sectors = `fdisk -l`

```
dd if=/dev/sdj of=bananian-1504_2GB_bkup_01.img bs=512 count=3710937
```

I've got an extra sector there to avoid any kind of 'off by one' misunderstanding of how `dd` works.

Start the command with `sudo`.

Configuration files

swconfig

switch configuration

```
nano /etc/network/if-pre-up.d/swconfig
```

```
#!/bin/sh

#-----#
# BPI-R1 VLAN configuration #
#-----#
#
# This will create the following ethernet ports:
# - eth0.101 = WAN (single port)
# - eth0.102 = LAN (4 port switch)
#
# You have to adjust your /etc/network/interfaces
#
# Comment out the next line to enable the VLAN configuration:
# exit 0

ifconfig eth0 up

swconfig dev eth0 set reset 1
swconfig dev eth0 set enable_vlan 1
swconfig dev eth0 vlan 101 set ports '3 8t'
swconfig dev eth0 vlan 102 set ports '4 0 1 2 8t'
swconfig dev eth0 set apply 1

# If you want to address some ports individually, change these lines.
# Use each port only once. The order used in this document is the order
# the ports appear physically (3 4 0 1 2) where 3 is the one that is
# separated out.
# i.e.
# swconfig dev eth0 vlan 201 set ports '3 8t'
# swconfig dev eth0 vlan 202 set ports '2 8t'
# swconfig dev eth0 vlan 203 set ports '4 0 1 8t'
```

wpa supplicant

It is suitable for both desktop/laptop computers and embedded systems.

Supplicant is the IEEE 802.1X/WPA component that is used **in the client stations**.

security mechanism - wpa_supplicant is designed to be a "daemon" program that runs in the background and acts as the backend component controlling the wireless connection.

NOTE: While wpa_supplicant can usually be used on wired connections using TTLS/PAP (or similar) authentication mechanism, this doesn't seem to be the case with BPi-R1. Authenticating on such networks seems to be impossible with BPi-R1.

isc-dhcp

This is the server from the Internet Software Consortium's implementation of DHCP.

```
apt-get install isc-dhcp-server
```

configuration file `nano /etc/default/isc-dhcp-server`

```
INTERFACES="br0"
```

```
mv /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.old
```

configuration file `nano /etc/dhcp/dhcpd.conf`

option definitions common to all supported networks...

```
#option domain-name "example.org";
```

```
#option domain-name-servers ns1.example.org, ns2.example.org;
```

the numbers represent seconds

```
default-lease-time 3601;
```

```
max-lease-time 44000;
```

Use this to send dhcp log messages to a different log file (you also

have to hack syslog.conf to complete the redirection).

```
log-facility local7;
```

This is a very basic subnet declaration, 192.168.10.1 is my NETGEAR router connected to the modem.

```
subnet 192.168.9.0 netmask 255.255.255.0 {  
    range 192.168.9.150 192.168.9.250;  
    option routers 192.168.9.2;  
    option domain-name-servers 192.168.10.1;  
}
```

Checks:

```
service isc-dhcp-server status (it does not run at this stage, continue..)
```

Location of the IP-Address lease file:

```
cd /var/lib/dhcp/
```

```
less dhclient.eth0.101.leases
```

```
less dhcpd.leases (WLAN, actually br0)
```

Alternative: Using dnsmasq

Instead of using the above mentioned `isc-dhcp-server...`

`dnsmasq` is able to provide a simple (it should be seen as consumer grade) way of providing dhcp and DNS (if you want to be able to access your router as "router" rather than using it's IP-Address for example).

To install:

```
apt-get install dnsmasq
```

Configure the IP-Addresses:

```
mv /etc/dnsmasq.conf /etc/dnsmasq.conf.example  
nano /etc/dnsmasq.conf
```

Interface to listen to

```
interface=br0
```

use self for dns lookups

```
listen-address=127.0.0.1
```

```
domain=home
```

```
domain-needed
```

```
bogus-priv
```

**# Change the amount of time used for a lease for high usage sites
(schools) i.e. the following says "I am allowing a maximum of 100
devices on my network within a 24h period.**

```
dhcp-range=192.168.9.150,192.168.9.250,24h
```

static leases

dhcp-host=mac address:ip address

```
dhcp-host=dc:9f:dc:dc:dc:dc,192.168.9.149
```

**# Please note: addresses used for static leases should not exist within
your dhcp address lease pool.**

Save and exit. Configure DNS:

```
nano /etc/hosts
```

#These must always exist in here.

```
127.0.0.1      localhost
```

```
::1           localhost ip6-localhost ip6-loopback
```

```
ff02::1       ip6-allnodes
```

```
ff02::2       ip6-allrouters
```

#Custom entries

```
192.168.9.2    router
```

Save and exit nano.

Enable dnsmasq (default = 1): nano /etc/default/dnsmasq

Look for the line that says:

```
ENABLED=0
```

and change it to:

```
ENABLED=1
```

```
apt-get --purge remove isc-dhcp-server
```

interfaces

Bridge, will be used to bridge the Ethernet and Wireless interfaces together.

Lines starting with `#` are ignored. Note that end-of-line comments are NOT supported, comments must be on a line of their own.

```
nano /etc/network/interfaces
```

```
# interfaces(5) file used by ifup(8) and ifdown(8)
```

```
# auto = Start the interface(s) at boot.
```

```
# iface = interface
```

```
auto lo
```

```
iface lo inet loopback
```

```
#allow-hotplug <interface> - Start the interface when a "hotplug" event is detected.
```

```
#In real world, is used on the same situations of auto but the difference is that it will wait an event like "plug the cable" on ethernet interfaces.
```

```
allow-hotplug eth0
```

```
allow-hotplug wlan0
```

```
#link aggregation = use 2 Ethernet-Connections (bond0) to increase the speed
```

```
##### ** LAN config VLAN 1** #####
```

```
##### receive IP-Address from your DSL modem or cablemodem
```

```
auto eth0.101
```

```
    iface eth0.101 inet dhcp
```

```
##### * LAN config VLAN 2, for the 4 ports * #####
```

```
##### generate IP-Address for connected devices
```

```
auto eth0.102/b
```

```
    iface eth0.102 inet manual
```

```
##### ** WLAN config ** #####
```

```
##### generate IP-Address for connected devices
```

```
auto wlan0
```

```
    iface wlan0 inet manual
```

```
# you can take this option if you have a /etc/init.d/hostapd
```

```
# post-up service hostapd start
```

```
# -B run daemon in the background
```

```
# post-up hostapd -dB /etc/hostapd/hostapd.conf
```

```
# post-down, actions taken right after the iface is down
```

```
post-down rm -rf /run/hostapd/wlan0
```

```
##### ** Bridge config ** #####
```

```
auto br0
```

```
    iface br0 inet static
```

```
        bridge_ports eth0.102 wlan0
```

```
        bridge_waitport 0
```

```
        address 192.168.9.2
```

```
        network 192.168.9.0
```

```
        netmask 255.255.255.0
```

```
# adjust the gateway to your configuration, my gateway is my NETGEAR router. Comment (#) if your R1 is directly connected to the DSL- or cablemodem.
```

```
    gateway 192.168.10.1
```

Checks:

```
netstat -uap
cat /etc/network/interfaces
ip r
ip link (successor of ifconfig)
ifconfig -a
ifconfig eth0.101
brctl show
brctl showmacs br0
less /var/log/syslog (type capital G to jump to last entry - q to exit)
less /var/log/messages
```

IPtables

iptables are used to set up, maintain, and inspect the tables of IPv4 packet filter rules in the Linux kernel. Several different tables may be defined.

Each table contains a number of built-in chains and may also contain user-defined chains.

Each chain is a list of rules which can match a set of packets.

Each rule specifies what to do with a packet that matches. This is called a 'target', which may be a jump to a user-defined chain in the same table. `man iptables`

<http://ubuntuforums.org/showthread.php?t=716192>

```
iptables -S (see what you have)
iptables -L
```

For bridged interfaces you need these:

```
iptables -A FORWARD -i br0 -s 192.168.9.0/255.255.255.0 -j ACCEPT
iptables -A FORWARD -i eth0.101 -d 192.168.9.0/255.255.255.0 -j ACCEPT
iptables -t nat -A POSTROUTING -o eth0.101 -j MASQUERADE
iptables-save > /etc/iptables/rules.v4
```

To secure a little bit the router:

Be careful to understand what a rule does, before you apply it. If you are connected to your Banana via SSH you might lose your connection and have to edit files directly with HDMI monitor and USB-Keybord or put the SDcard in your PC to correct a wrong rule.

These lines allow incoming connection to `loopback` and `br0` (the 4 LAN and WLAN if you followed this tutorial) :

```
iptables -I INPUT 1 -i lo -j ACCEPT
iptables -I INPUT 1 -i br0 -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

These lines defines the default policy.

In this case to DROP **every** packet that is not explicitly told to be accepted.

So before you apply these rules, read below: open a specific port

Banana Pi Router - BPi-R1 - 15.04

```
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
```

If you need, you can open a specific port:

```
iptables -A INPUT -p 1 --dport 2 -i eth0.101 -j ACCEPT
```

allow incoming SSH connection to WAN port and give the rule top priority (no WiFi).

```
iptables -I INPUT 1 -i eth0.101 -p tcp -m tcp --dport 22 -j ACCEPT
```

1: tcp or udp

2: the port to open

to check your configuration:

```
iptables -L
iptables -S
iptables -L -t nat
```

And we need to turn on IP forwarding in the Kernel:

```
less /proc/sys/net/ipv4/ip_forward (0)
echo 1 > /proc/sys/net/ipv4/ip_forward
```

to make this change permanent edit `nano /etc/sysctl.conf` and uncomment line
`net.ipv4.ip_forward=1`

`shutdown -h now` -> it's time for a **Backup** now ;)

i.e.:

```
dd if=/dev/sdj of=bananian-1504_2GB_bkup_02.img bs=512 count=3710937
```

if you start now, you must plug the Ethernet-Cable to the WAN port

before you go for compilation (hostapd & driver)

This is important information - please read it.

It is written in several posts, that the DEBIAN RealTek driver 8192cu v3.3.2 doesn't play nice. Neither does the hostapd play nice - it won't let you connect (a daemon that handles access point management and authentication (wpa2)).

So you would get the original hostapd & driver from RealTek, compile it and you are done.

Unfortunately, according to several posts - this driver v4.0.2 does may be work in combination with a modified hostapd and /or 'the right' configuration-file.

So what we need is a working driver & hostapd for WiFi!

You can find several source files on Git, remember we need ARM-architecture.

```
apt-get update
apt-get install build-essential git
```

Install the Kernel source:

```
apt-get install linux-headers-$(uname -r)
```

symlink, copy both line as one (it is a long command)

```
ln -s /usr/src/linux-headers-3.4.104-bananian/arch/arm
/usr/src/linux-headers-3.4.104-bananian/arch/armv7l
```

```
cd /usr/src/linux-headers-3.4.104-bananian/
make modules_prepare (may return an error - ignore it)
make headers_check
make headers_install
make scripts/kallsyms
```

shutdown -h now → it's time for a **Backup** now.

i.e.:

```
dd if=/dev/sdj of=bananian-1504_2GB_bkup_03.img bs=512 count=3710937
```

jekader - just hostapd-2.4 for the RTL driver v4.0.2

[jekader's mixture](#)

install the right driver

<https://sites.google.com/site/easylinuxtipsproject/reserve-7>

create the driver with this git and use the hostapd from RealTek

[Wannabe Nerd Randomness](#)

<https://github.com/dz0ny/rt8192cu> v4.0.2 may be modified

Jens hostapd (and the RealTek driver is enough?)

<http://www.jenssegers.be/blog/43/realtek-rtl8188-based-access-point-on-raspberry-pi>

<https://github.com/jenssegers/RTL8188-hostapd>

kernel.org about rtl819x

<https://wireless.wiki.kernel.org/en/users/drivers/rtl819x>
[driver rtl8188C 8192C 8192D usb linux v3.4.2](#)

compile from Git

To do so you need to install the software.

```
apt-get install git dkms (Dynamic Kernel Module Support Framework)
mkdir -p /home/git && cd /home/git
git clone https://github.com/dz0ny/rt8192cu.git --depth 1
cd rt8192cu
make
make dkms
```

or follow the next page and try the original one from RTL

Compile RealTek WiFi driver & hostapd

In case your BPi-R1 does now and then stop working, look into the log-files.

While you are testing, do a reboot from now and then `shutdown -r now` sometimes it helps ;)

To get hostapd running with rtl871xdrv you need the proprietary kernel module 8192cu (without rtl*)

For the proprietary 8192cu you have to set driver=rtl871xdrv

```
nano /etc/hostapd/hostapd.conf
```

```
modinfo 8192cu          (... v3.3.2, better v3.4 or v4.0)
lsmod | grep 8192       (8192cu          453340 0)
dmesg | grep rtl81      (usbcore: registered new interface driver rtl8192cu)
```

```
cd /etc/modprobe.d/
nano blacklist.conf
blacklist rtl8192cu (add this line)
```

The binary from Debian's repository doesn't include the latest driver so we have to download it from Realtek's website and compile it ourselves. I downloaded it on my PC, unzipped it and with gFTP I copied it to the 'home' directory on the SDcard - or simply put the SDcard into your PC to copy.

<http://www.realtek.com/downloads/downloadsView.aspx?Langid=1&PNid=48&PFid=48&Level=5&Conn=4&DownTypeID=3&GetDown=false&Downloads=true#2772> (10MB)

```
apt-get update
apt-get install build-essential
```

Install the Kernel source:

```
apt-get install linux-headers-$(uname -r)
```

symlink, copy both line as one (it is a long command)

```
ln -s /usr/src/linux-headers-3.4.104-bananian/arch/arm
/usr/src/linux-headers-3.4.104-bananian/arch/armv7l
```

```
cd /usr/src/linux-headers-3.4.104-bananian/
make modules_prepare (may return an error)
make headers_check
make headers_install
make scripts/kallsyms
```

```
cd /home/RTL81xxxxx
ls -la
chmod 775 install.sh
./install.sh (get a tea or coffee, takes about 2min)
```

The driver is compiled and can be found:

```
/lib/modules/3.4.104-bananian/kernel/drivers/net/wireless/
```

Banana Pi Router - BPi-R1 - 15.04

To rename the existing, old driver, follow these steps:

```
cd /lib/modules/3.4.104-bananian/kernel/drivers/net/wireless/rtl8192cu/
```

copy three lines as one (it is a long command, rename the current (old) driver)

```
mv  
/lib/modules/3.4.104-bananian/kernel/drivers/net/wireless/rtl8192cu/8192  
cu.ko 8192cu.ko.old
```

depmod creates a list of module dependencies by reading each module under /lib/modules/version and determining what symbols it exports and what symbols it needs.

```
/sbin/depmod -a 3.4.104-bananian
```

Now you compile the hostapd & hostapd_cli

```
cd /home/RTL8188C*/wpa_supplicant_*/  
tar zxvf wpa_supplicant_hostapd-0.8_rtw_*.tar.gz  
cd wpa_supplicant_hostapd-0.8_*/hostapd/  
make  
make install (/usr/local/bin/)
```

fyi: hostapd hostapd_cli /usr/sbin/ (Debian standard location)

```
ln -s /usr/local/bin/hostapd /usr/sbin/ (can be linked)
```

```
ln -s /usr/local/bin/hostapd_cli /usr/sbin/ (can be linked)
```

```
ln -s /usr/local/bin/hostapd /usr/local/sbin/ (can be linked)
```

```
ln -s /usr/local/bin/hostapd_cli /usr/local/sbin/ (can be linked)
```

find all files with the word 'hostapd' in it: `find / -name hostapd` check if the configuration file points to your file location

hostapd

Host AccessPoint Daemon

This service must start after the wlan0, after isc-dhcp !!

I am looking after a solution, if you have a working one, for:

with `service networking restart`

hostapd must be the last to restart. For your information, this section in the Manual 15.01 is newer and followed by 'boot order control' or simply do a reboot instead of service restart.

An example configuration file can

be found at `/usr/share/doc/hostapd/examples/hostapd.conf.gz` (`gzip -d hostapd.conf.gz`)

Nice information: <http://wiki.gentoo.org/wiki/Hostapd>

hostapd; is a daemon that handles access point management and authentication

Which hostapd do you have installed:

`hostapd -v` (returns the version)

`nano /etc/default/hostapd`

activate the daemon, set the absolute path to your hostapd configuration

`DAEMON_CONF="/etc/hostapd/hostapd.conf" # point to configuration file`

`nano /etc/init.d/hostapd`

and complete the following lines with your path to the config file (if you copied it to another location in the previous step) you can search for it: `find / -name hostapd`

`DAEMON_SBIN=/usr/sbin/hostapd`

`DAEMON_CONF=/etc/hostapd/hostapd.conf`

hostapd.conf

`mkdir -p /etc/hostapd/` (create folder, if it is not there)

`nano /etc/hostapd/hostapd.conf`

`interface=wlan0`

`bridge=br0`

`driver=rtl871xdrv`

This advertises the country_code and the set of allowed channels and transmit power

`# country_code=DE`

`# ieee80211d=1`

Maximum data rate 54Mbps in 802.11g and 300Mbps in 802.11n

The RTL8192CU provides simple legacy and 20MHz/40MHz co-existence mechanisms to ensure backward and network compatibility. Maximum PHY data rate up to 144.4Mbps using 20MHz bandwidth, 300Mbps using 40MHz bandwidth

if you don't have a wireless N capable device = 0

`ieee80211n=1`

2,4GHz bandwidth

`channel=4`

`hw_mode=g`

`wmm_enabled=1`

Identification and Login

`ssid=bpir1`

```
wpa_passphrase=supersecret
#wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP CCMP
#rsn_pairwise=CCMP
auth_algs=1
wpa=2

macaddr_acl=0
beacon_int=100
max_num_sta=8
ht_capab=[SHORT-GI-20][SHORT-GI-40][HT40]
ignore_broadcast_ssid=0
ctrl_interface=/run/hostapd
noscans=1
```

check your config file: `hostapd -dd /etc/hostapd/hostapd.conf`
is the service running? `service hostapd status`
`ps ax | grep [h]ostapd`
`less /var/log/daemon.log`

prevent the hostapd from automatic starting, so you can control it somewhere else:
`update-rc.d -f hostapd remove`

To start the daemon service every time on boot (if you want that):

```
update-rc.d hostapd defaults
update-rc.d hostapd enable
```

scan networks: `iwlist wlan0 scan`
show available channels: `iwlist wlan0 channel`
find least used channel: `cat /proc/net/rtl819xC/wlan0/best_channel`

How much traffic do you have per process?

```
nethogs eth0.101 q to quit
```

```
shutdown -r now
```

of course, you don't have to, but consider how many hours I have invested into this document.

So if you have login for lemaker.org - I would appreciate it

if this manual was helpful for you, please leave a 'I like' at

<http://forum.lemaker.org/forum.php?mod=viewthread&tid=13877&fromuid=72927>

Warning

One last word about the housing, it is tight - make sure the temperature does not rise too high.

To check the temperature, run the following:

```
echo $(( $(cat  
/sys/devices/platform/sunxi-i2c.0/i2c-0/0-0034/temp1_input)/1000 ))
```

Even better, install RPi-Monitor

This gives you the temperature in °C. The Banana Pi is designed to work in temperatures up to 75°C but in practise, you want to try and keep that temperature below 65°C - particularly if you have a mechanical hard drive plugged in.

To manage the temperature, the board should stand vertically, instead of horizontally (with the smallest face at the bottom), to allow better air flow (convection).

The acrylic case sold with some Banana Pis can limit ventilation. One way to improve ventilation is to remove a couple of the side panels (the longest ones).

Another option is to add “active” cooling i.e. a fan. Bear in mind that this does have issues:

- Fans use power and so your low power Pi could end up using more power than is necessary.
 - Fans are just motors in which case, they add “noise” to your power supply. If you want to use that power supply for some other purpose at the same time, like a set of speakers for example, it can make those speakers crackle.
-

Internet resources

Openwrt CHAOS CALMER (V1.0.0522) - Bug fixes and other fixes

<http://www.bananapi.com/>

Banana PI / PRO / R1 Debian SD image - including Kernel Header for compiling

<http://www.igorpecovnik.com/2014/09/07/banana-pi-debian-sd-image/>

Why the SATA disk doesn't work on BPi-R1

<http://www.bananapi.com/index.php/forum/general/391-why-the-sata-disk-doesnt-work-on-bpi-r1?start=12#1317>

Banana Pi R1 as firewall /router - short but well explained in each detail

<http://rolfebozier.com/archives/51>

Configuring a Linux home internet gateway - well explained what you need each part for

http://www.freesoftwaremagazine.com/articles/home_internet_gateway

RsPi as WAP

<http://www.linuxuser.co.uk/tutorials/set-up-a-wireless-access-point-with-a-raspberry-pi>

Realtek RTL8188 based access point on Raspberry Pi

<http://www.jenssegers.be/blog/43/realtek-rtl8188-based-access-point-on-raspberry-pi>

BananaPi as a fileserver -- some personal thoughts and experiences.

<http://forum.lemaker.org/thread-7102-1-1-.html>

Banana Pro – WLan-AccessPoint (german)

<http://frank-mankel.de/>

Netzwerk unter Linux (german)

<http://www.stefanux.de/wiki/doku.php/linux/netzwerke>

Housing for BPi - R1

on [Google+](#) add files to the print and diagrams in one month (approx end of April '15)

Files for 3D printing available on <http://www.thingiverse.com/thing:735741> this is an early version.

Access Point - Bash Script

<http://crunchbang.org/forums/viewtopic.php?id=35427>