

# Skipper's Mate

## Why You Need One and How to Get Going

Software Version	1.0.0.1	2 May 2015
Documentation Version	1.0.0.3	6 May 2015

### Table of Contents

Document Conventions.....	3
<b>Section 1 - All About The System.....</b>	<b>5</b>
Chapter 1. Why Would You Want One ?.....	6
Chapter 2. Feature List.....	7
Chapter 3. How It Works.....	8
Chapter 4. Using the Full System.....	10
<b>Section 2 – Getting Going The Easy Way.....</b>	<b>11</b>
Chapter 5. How To Buy And Get The Full Kit Working.....	12
Chapter 6. Using Skipper's Mate.....	13
<b>Section 3 – Getting Going with the Free Version.....</b>	<b>14</b>
The Raspbian users.....	14
Supervisor Rights – root and sudo – and the Raspberry Pi.....	14
How to Install a Software Package under Linux.....	14
<b>Chapter 7. How to Download and Get the Free Version Working.....</b>	<b>17</b>
What hardware you need to get going :.....	17
Login and Configure Raspberry.....	19
Run System Update.....	21
Install a Text Editor - joe.....	21
Install and Configure the Wireless Access Point.....	21
Install and configure DHCP server.....	24
Make Your Wireless Device an Access Point.....	25
Configure IP Routing between Wireless and Ethernet.....	26
Enable Packet Forwarding.....	28

# Skipper's Mate

## Why You Need One and How to Get Going

Setup RAM disk.....	29
Create pcl Group and Members.....	30
Install Motion software.....	30
Install PHP.....	33
Install Web Server – Lighttpd.....	33
Install the SQLite Database.....	35
Install the PostgreSQL Database.....	35
Install the PostgreSQL Administration Tool - phpPgAdmin (optional).....	36
Install Skipper's Mate.....	36
Load the Skipper's Mate Databases.....	37
Install the Network Bridge.....	38
Set Up the Routing Script.....	39
Install the VNC server.....	41
<b>APPENDIX A – Hardware Diagram for WiFi HotSpot Only.....</b>	<b>48</b>
<b>APPENDIX B – Hardware Diagram for Skipper's Mate – Next Version.....</b>	<b>49</b>
<b>APPENDIX C – joe Editor Commands – a simple sub-set.....</b>	<b>50</b>

# Skipper's Mate

## Why You Need One and How to Get Going

### Document Conventions



### Command line :

```
sudo apt-get install joe
```

### Command line output :

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  joe
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 474 kB of archives.
After this operation, 1,269 kB of additional disk space will be
used.
Processing triggers for menu ...
```

# Skipper's Mate

## Why You Need One and How to Get Going

### Editor (joe)

```
# Uncomment the next line to enable packet forwarding for IPv4
# Comment forward line in Skippers Mate
net.ipv4.ip_forward=1
```

The items in bold are the changes you need to make

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlan0
# change next line manual to static add address and netmask lines
# and gateway
iface wlan0 inet static
#iface wlan0 inet manual
address 192.168.51.254
netmask 255.255.255.0
gateway 10.0.1.254
# end of mod

wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

A larger sub-set of joe commands can be found in **Appendix C**.

```
Save and exit - <ctrl> KF
Quit without saving - <ctrl> KQ
```

# **Skipper's Mate**

## **Why You Need One and How to Get Going**

### **Section 1 - All About The System**

# Skipper's Mate

## Why You Need One and How to Get Going

### **Chapter 1. Why Would You Want One ?**

The Skipper's Mate Boat System provides you with :

- ◆ a fully-featured, powerful desktop computer running a mature operating system – capable of running almost anything a standard desktop can run, including office applications, web streaming and marine applications
- ◆ hardware particularly suitable for marine applications with very low power consumption – typically less than 0.5 amps per hour at 12 volts
- ◆ a system which will run from 12V (or 24V) power via a step-down
- ◆ a system accessible either by tablet/smartphone/laptop wirelessly (through a web browser and/or virtual network server) or or via a USB keyboard and mouse
- ◆ able to act as a local wi-fi hotspot providing web access from multiple devices (e.g. tablet, laptop) from your boat
- ◆ complete with specialised marine applications – Skipper's Mate is a fully-featured boat alarm system and wi-fi hotspot enabling you to monitor on-board devices remotely. It will shortly be offering remote control of on-board devices e.g. switch on your heating and refrigerator while you're still on the motorway ... arrive at your cosy boat to find a cold beer waiting. For details of the features of Skipper's Mate go to Section 2
- ◆ includes many other marine applications e.g. OpenCPN – a fully-featured chart plotting system which is AIS-capable
- ◆ multi-user simultaneous access e.g. chart table, cockpit and even the skipper's berth
- ◆ they're sweet :)

# Skipper's Mate

## Why You Need One and How to Get Going

### **Chapter 2. Skipper's Mate Feature List**

- ◆ general terms, the more powerful the antenna the more susceptible to movement.
- ◆
- ◆
- ◆
- ◆
- ◆
- ◆
- ◆
- ◆
- ◆

# Skipper's Mate

## Why You Need One and How to Get Going

### Chapter 3. How It Works

Plug the following into your Raspberry Pi :

1. **SD or micro SD card** (as supplied ready-prepared or burned by you from a download)
2. **Wireless Access Point** (as supplied pre-configured or supplied by you)
3. Power to your Raspberry Pi




**DO NOT PLUG IN THE WIRELESS RECEIVER YET !**

### Step 2 - Boot your Raspberry Pi

This can take a couple of minutes – watch the lights. The ACT light should stop flashing when the it has booted.

#### Lights – Raspberry Pi Model B



*The model B+ has only the ACT and PWR lights.*

ACT		SD card access light – should flash intermittently while booting and when activity occurs at other times
PWR		should show permanently – if not then you have problem with the power – try re-seating the power lead. Then check the power to the RPi
FDK		Full Duplex (LAN) connected



# Skipper's Mate

## Why You Need One and How to Get Going

LNK		Link/Activity (LAN)
100		100 Mbit LAN connected – will not light if 10Mbit

*This step will be run once only. The Wireless AP will be stored as the first USB device (wlan0) and its characteristics stored with it. If you need to change the AP there are steps in a document available in **Section 5 Troubleshooting** to help. At the moment all you need to care about is that your first USB device is the Wireless Access Point and that the default SSID (Hotspot Name) is **SkippersMate**.*

# **Skipper's Mate**

## **Why You Need One and How to Get Going**

### ***Chapter 4. Using the Full System***

Using any wireless-enabled device (e.g. tablet, smartphone, laptop) and look for the wireless

# **Skipper's Mate**

## **Why You Need One and How to Get Going**

### **Section 2 - Getting Going The Easy Way**

# Skipper's Mate

## Why You Need One and How to Get Going

### ***Chapter 5. How To Buy And Get The Full Kit Working***

The full Skipper's Mate Kit can be purchased as a complete hardware and software bundle ready for you to plug in on your boat direct from our web site.

It includes the following :

#### **Hardware**

- ◆ Raspberry Pi
- ◆ Step-down from 12V or 24V to the required current for the system
- ◆ Wi-Fi Access Point
- ◆ 3G/4G mobile broadband dongle (\* SIM not included)
- ◆ 32Gb Micro SD card
- ◆ Camera (choice of USB or IP)

#### **Hardware optional :**

- ◆ GPS receiver
- ◆ Additional camera

#### **Software**

- ◆ Skipper's Mate Boat System – security,

# Skipper's Mate

## Why You Need One and How to Get Going

### ***Chapter 6. Using Skipper's Mate***

What hardware you need to get going :

- ◆ Raspberry Pi 2 \* *Note 1*
- ◆

# Skipper's Mate

## Why You Need One and How to Get Going

### Section 3 - Getting Going with the Free Version

This section is by far the largest in this document. Is intended to help you use the free and open source software available for the Raspberry Pi – including our own offering, Skipper's Mate.

If you are already familiar with a version of Linux then this should not be too difficult for you – although there are quite a number of software packages to install.

#### The Raspbian users

Raspbian comes with 2 installed users :

**root** – the superuser (initially supplied with no password)

**pi** - a standard user (initial password of **raspberry**)

#### Supervisor Rights – root and sudo – and the Raspberry Pi

Linux systems have a super-user named **root**. You can temporarily emulate supervisor use by typing the command **sudo**. This stands for 'do as superuser'.

The standard operating system for the Raspberry Pi is called Raspbian. Since the Pi is intended to be used as an educational tool, the designers of Raspbian have provided a quick shortcut in the initial version of Raspbian which you will install – the ability to use superuser rights without a password.

The notes in the chapters below will show you how to change this and 'harden' this version of the operating system but it does make the initial steps easy.

#### How to Install a Software Package under Linux

If you are a Windows user then we hope to make this possible for you by including some notes on the differences between installing under Linux and Windows. Believe it or not, it is actually easier to install most software under Linux – partially because most software comes from a single central source for a given version of the operating system.

Linux is free and so there is usually no need to hunt around the web first for one software package and then for another. Kind people assemble the software packages into what is known as a 'repository' and so a single installation command will normally obtain any package you might need. Specialised software (e.g. Skipper's Mate) is not included in a repository but you will find that everything from a web server to a database mentioned in this document can be obtained directly from the Raspbian repositories.

You can login as **root** to carry out the installation but this is a bad habit to develop so almost all steps can be carried as the temporary superuser by preceding the command with the text **sudo**. If you attempt to run as user pi :

```
apt-get install joe
```

you will see an error message because you do not have permission, as a standard end-user, to

# Skipper's Mate

## Why You Need One and How to Get Going

install software :

```
E: Could not open lock file /var/lib/dpkg/lock - open (13:
Permission denied)
E: Unable to lock the administration directory (/var/lib/dpkg/),
are you root?
```

The single command you need is :

```
sudo apt-get install <package name>
```

e.g. to install the database SQLite you only need to enter a single line

```
sudo apt-get install sqlite3
```

Here is a full example of installing an editor called **joe**. This stands for Joe's Own Editor. Yes, Linux is a friendly operating system !

This is what you type :

```
sudo apt-get install joe
```

And this is the reply. You will see that it is installing one new package – **joe**. That's it !

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
 joe
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 474 kB of archives.
After this operation, 1,269 kB of additional disk space will be
used.
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main joe
armhf 3.7-2.3 [474 kB]
Fetched 474 kB in 1s (384 kB/s)
Selecting previously unselected package joe.
(Reading database ... 76938 files and directories currently
installed.)
Unpacking joe (from ../archives/joe_3.7-2.3_armhf.deb) ...
Processing triggers for menu ...
Processing triggers for man-db ...
Setting up joe (3.7-2.3) ...
update-alternatives: using /usr/bin/joe to provide /usr/bin/editor
(editor) in auto mode
```

# Skipper's Mate

## Why You Need One and How to Get Going

```
Processing triggers for menu ...
```

In some instances you may be asked to confirm that you wish to proceed, by typing in **y** for yes.

So Linux isn't purely for geeks !

Having said that, there is a fair amount of work in installing everything yourself but you will have the satisfaction of knowing all about the system and getting everything FREE.



# Skipper's Mate

## Why You Need One and How to Get Going

### Chapter 7. How to Download and Get the Free Version Working

#### What hardware you need to get going :

- ◆ Raspberry Pi 2 \* *Note 1*
- ◆ Step-down from 12V or 24V to the required current for the system
- ◆ 8Gb or larger Micro SD card \*\*\*\* *Note 2*
- ◆ 3G/4G mobile broadband dongle and SIM \*\*\* *Note 3*
- ◆ USB wireless adapter capable of acting as a Wi-Fi Access Point \*\* *Note 4*
- ◆ Camera (optional – USB or Ipcam – your choice)

\* **Note 1** - Can be done with a lower spec. Raspberry. However you need to consider what you're going to run since less memory may not be sufficient for all the applications you may want to run.

\*\*\*\* **Note 2** – The type of card will be dictated by the Raspberry you're using – either micro SD or SD. The size of card is a matter of choice – if you wish to add Open Source software such as OpenCPN with charts. Skipper's Mate and the Rasbian operating system occupy approximately 3.1Gb.

\*\*\* **Note 3** – dongles come in 2 flavours – those using ppp which involves usb-modeswitch and wvdial (both of which are truly hard going) and those which emulate

\*\* **Note 4** – Check that the wireless device you plan to use is compatible with the Raspberry Pi and will act as an Wireless Access Point.

#### **Warning !**

Only a limited set of wireless devices can act as access points (to provide a hotspot). Check that the wireless device you plan to use is compatible with the Raspberry Pi and can act as an Wireless Access Point. This page has tables for the Rpi showing working and problematic wireless devices, including their capabilities : **RPi USB Wi-Fi Adapters** [http://elinux.org/Rpi\\_USB\\_Wi-Fi\\_Adapters](http://elinux.org/Rpi_USB_Wi-Fi_Adapters) .

# Skipper's Mate

## Why You Need One and How to Get Going

*These notes assume you have broadband access.*

- ◆ Download Rasbian Debian Wheezy Operating System from <https://www.raspberrypi.org/downloads/> Rasbian - Debian Wheezy and follow the image installation guides referred to immediately above the RASPBIAN image download. When you have finished this you should have a bootable SD card.

### *Pi, power, ethernet cable*

- ◆ Attach Raspberry Pi to your router via an ethernet cable. Most routers are set up to issue DHCP. The Rasbian installation is expecting to receive DHCP.
- ◆ Boot RPI with newly written SD card

### For Linux Users

Enter ifconfig on the command line :

```
ifconfig
```

You should see something like this where your ethernet connection (which in this case happens to be eth3) will show the inet address (IP address) of the PC you're using :

```
eth3  Link encap:Ethernet HWaddr 00:1d:7d:e9:91:fc
      inet addr:10.0.1.5 Bcast:10.0.1.255 Mask:255.255.255.0
      inet6 addr: fe80::21d:7dff:fee9:91fc/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:103371 errors:0 dropped:0 overruns:0 frame:0
      TX packets:101110 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:85057640 (85.0 MB) TX bytes:21920760 (21.9 MB)
```

Enter the following, where your IP address is 10.0.1.5 and your subnet is 10.0.1.0 :

```
sudo nmap -sn 10.0.1.0/24
```

Among the nmap details returned you should see one which refers to the Raspberry Pi Foundation e.g. :

```
Nmap scan report for 10.0.1.122
Host is up (0.00085s latency).
```

# Skipper's Mate

## Why You Need One and How to Get Going

```
MAC Address: B8:27:EB:F4:C0:0C (Raspberry Pi Foundation)
```

**For Windows users**, either look in the router for attached clients. If not, browse to <http://nmap.org/book/inst-windows.html>, install the software and follow the instructions on that page and then issue the Linux command, without the sudo i.e.

```
nmap -sn <your IP subnet e.g. 10.0.1>.0/24 e.g. 10.0.1.0/24
```

For Mac users, either look in the router for attached clients. If not, browse to

<http://nmap.org/book/inst-macosx.html> , install the software and follow the instructions on that page and then issue the Linux command, without the sudo i.e.

```
nmap -sn <your IP subnet e.g. 10.0.1>.0/24 e.g. 10.0.1.0/24
```

Contact your RPi , using the IP address returned by nmap. e.g. 10.0.1.122, using **ssh** or **WinSCP** or **putty**.

### For Linux users

The best tool to work on the Raspberry under Linux is **ssh**. See the next section for details.

### For Windows users

Search for putty with a browser by visiting this link :

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

This will should show you :

*A Windows installer for everything except PuTTYtel*

Installer: [putty<version>-installer.exe](#)      [\(RSA sig\)](#)    [\(DSA sig\)](#)

Double click on the .exe link and take all the defaults. This should install an icon on the desktop. Now ouble click on the right hand side - Host name or IP address : 10.0.1.122

Leave port 22 untouched

Bumbles about hostkey not cached in registry – hit Yes

### For Mac Users

Google for “mac putty” to find a useful equivalent.

# Skipper's Mate

## Why You Need One and How to Get Going

### Login and Configure Raspberry

Login as : pi pw: raspberry

Into command line

Configure raspberry

```
ssh pi@10.0.1.122
```

```
The authenticity of host '10.0.1.122 (10.0.1.122)' can't be
established.
ECDSA key fingerprint is
a5:da:6b:a2:72:47:fe:a4:76:2b:9b:ef:8b:aa:52:1c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.1.122' (ECDSA) to the list of
known hosts.
```

```
pi@10.0.1.122's password: raspberry
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 29 13:39:11 2015 from 10.0.1.9

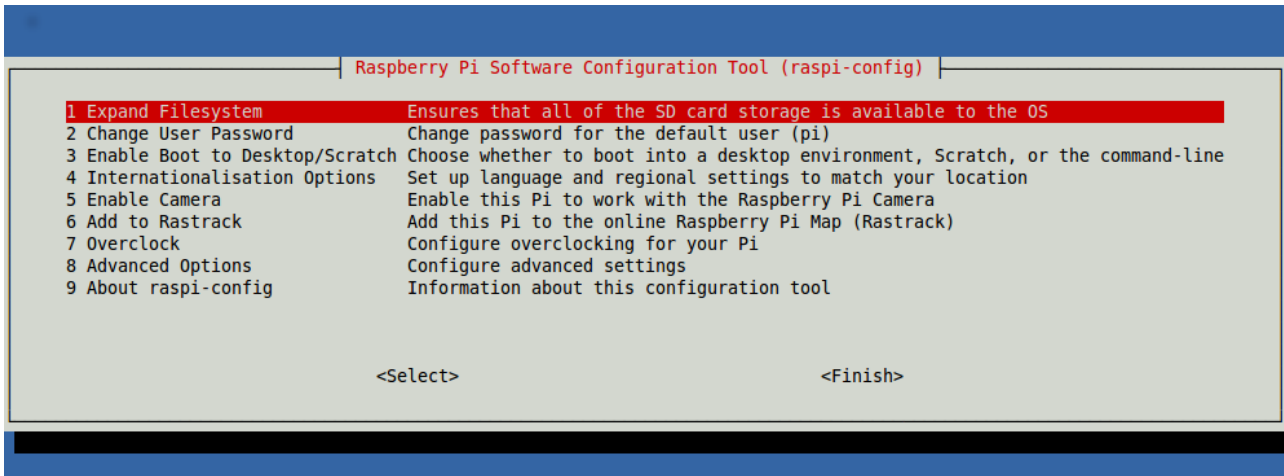
NOTICE: the software on this Raspberry Pi has not been fully
configured. Please run 'sudo raspi-config'
```

```
sudo raspi-config
```

Will display this screen :

# Skipper's Mate

## Why You Need One and How to Get Going



Highlight option 1) Expand Filesystem and tab to highlight the Select button. Hit <Enter>. This option will expand the image you downloaded to fill the entire SD/micro SD card. You should then see the following :

```
Root partition has been resized.  
The filesystem will be enlarged upon the next reboot
```

Hit <Enter> on the <OK> button to return to the previous screen.

Tab to the <Finish> button and hit <Enter>. Then select <Yes> to reboot.



Hit Enter on highlighted <Yes> to Reboot. You should see a broadcast message :

```
The system is going down for reboot NOW!
```

# Skipper's Mate

## Why You Need One and How to Get Going

Wait and then ssh in again

```
ssh pi@10.0.1.122
```

To check that the installation has been resized you can enter :

```
df -h
```

This should show you the File system usage – this example is for an 8Gb card. You can see that the total Size is the total usable size of the card – 7.2 Gb.

```
Filesystem      Size  Used Avail Use% Mounted on
rootfs          7.2G  2.4G  4.5G  35% /
/dev/root       7.2G  2.4G  4.5G  35% /
devtmpfs        460M   0    460M   0% /dev
tmpfs           93M    208K  93M    1% /run
tmpfs           5.0M   0    5.0M   0% /run/lock
tmpfs          186M   0    186M   0% /run/shm
/dev/mmcblk0p1  56M    15M   42M   26% /boot
```

### Run System Update

Obtain the latest updates to the Raspbian operating system :

```
sudo apt-get update
```

If you have a reasonable broadband connection (2Mb+ download) this will probably take less than 5 minutes.

### Install a Text Editor - joe

You will need this for editing various configuration files in this tutorial :

```
sudo apt-get install joe
```

### Install and Configure the Wireless Access Point

Do this before plugging in wireless adapter or anything else ! This will force the system to enter the details of your network devices into a file which retains this information - /etc/udev/rules.d/70-persistent-net.rules.

Edit the configuration file 75-persistent-net-generator.rules :

```
sudo joe /lib/udev/rules.d/75-persistent-net-generator.rules
```

# Skipper's Mate

## Why You Need One and How to Get Going

Add into the whitelist eth\* and wlan\*[0-9] :

```
# device name whitelist
KERNEL!="eth*|ath*|wlan*[0-9]|msh*|ra*|sta*|ctc*|lcs*|hsi*", \

GOTO="persistent_net_generator_end"
```

Insert your USB Wireless Access Point device.

### Warning !

Only a limited set of wireless devices can act as access points (to provide a hotspot). Check that the wireless device you plan to use is compatible with the Raspberry Pi and can act as an Wireless Access Point. This page has tables for the Rpi showing working and problematic wireless devices, including their capabilities : **RPi USB Wi-Fi Adapters** [http://elinux.org/Rpi\\_USB\\_Wi-Fi\\_Adapters](http://elinux.org/Rpi_USB_Wi-Fi_Adapters).

Check that the Wireless USB Device is recognised as a usb device :

```
lsusb
```

You should see a result similar to this in the Results window :

```
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 148f:5370 Ralink Technology, Corp. RT5370
Wireless Adapter
```

The line you're after is something like the one in bold (this is the line for the AP we supply). If nothing like that appears then the wireless device isn't recognised. Try the usual – ensure plugged in etc. There is a short list of chipsets which we have tested with the Rpi – see the Hardware page on the website for Skipper's Mate. If that doesn't help, try checking that your wireless device is compatible with the Raspberry Pi – Google for RPi Verified Peripherals – eLinux.

Let's assume that your wireless AP is seen by the Rpi. Configure wireless adapter with static IP by editing the interfaces configuration file :

# Skipper's Mate

## Why You Need One and How to Get Going

```
sudo joe /etc/network/interfaces
```

The items in bold are the changes you need to make

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlan0
# change next line manual to static add address and netmask lines
# and gateway
iface wlan0 inet static
#iface wlan0 inet manual
address 192.168.51.254
netmask 255.255.255.0
gateway 10.0.1.254
# end of mod

wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

Then reboot :

```
sudo reboot; exit
```

and then ssh in again

```
ssh pi@10.0.1.122
```

If you do an :

```
ifconfig
```

you will should see that you now have eth0 as the original IP – 10.0.1.122 – and the wireless adapter, wlan0, as 192.168.51.254 :

```
eth0      Link encap:Ethernet  HWaddr b8:27:eb:f4:c0:0c
          inet addr:10.0.1.122  Bcast:10.0.1.255
Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:115 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12044 (11.7 KiB)  TX bytes:11924 (11.6 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
```



# Skipper's Mate

## Why You Need One and How to Get Going

```
TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:1104 (1.0 KiB) TX bytes:1104 (1.0 KiB)

wlan0    Link encap:Ethernet  HWaddr 7c:dd:90:57:25:de
         inet addr:192.168.51.254  Bcast:192.168.51.255
Mask:255.255.255.0
         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)
```

### Install and configure DHCP server

Install the DHCP server which will issue IP addresses to any devices you wish to attach e.g. laptops, tablets :

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

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  isc-dhcp-server-ldap
The following NEW packages will be installed:
[FAIL] Starting ISC DHCP server: dhcpd[....] check syslog for
diagnostics. ... failed!
failed!
invoke-rc.d: initscript isc-dhcp-server, action "start" failed.
```

You will see that it fails to start the DHCP server. Ignore this. Raspbian tries to start server and we don't yet have a suitable configuration file. So now we edit dhcpd.conf :

```
sudo joe /etc/dhcp/dhcpd.conf
```

At the top of the file comment out the two option domain-name lines :

```
# option definitions common to all supported networks...
# comment out next 2 lines - Skippers Mate
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;
```

# Skipper's Mate

## Why You Need One and How to Get Going

Search for this section and comment out authoritative:

```
# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
# comment in authoritative - Skippers Mate
#authoritative;
```

Immediately after this line add in this section :

```
# Add in Skippers Mate
subnet 192.168.51.0 netmask 255.255.255.0 {
range 192.168.51.100 192.168.51.110;
option broadcast-address 192.168.51.255;
# option routers IP should match the bcast address of wless device
option routers 192.168.51.254;
default-lease-time 600;
max-lease-time 7200;
option domain-name "local";
option domain-name-servers 8.8.8.8,8.8.4.4;
}
```

*Note that the lease times are not duplications but apply to this bracketed section.*

Make sure dhcp server is listening on the wireless interface by editing the isc-dhcp-server file :

```
sudo joe /etc/default/isc-dhcp-server
```

... at end of file change INTERFACES line to read INTERFACES="wlan0"

```
# On what interfaces should the DHCP server (dhcpd) serve DHCP
requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="wlan0"
```

Your dhcp server should now be set up – don't start it yet.

### Make Your Wireless Device an Access Point

Install and configure the Access Point daemon :

```
sudo apt-get install hostapd
```

Now create the configuration file :

```
sudo joe /etc/hostapd/hostapd.conf
```

# Skipper's Mate

## Why You Need One and How to Get Going

Using joe, create a new file and insert the following :

```
interface=wlan0
driver=nl80211
# Some network devices will require the driver on the following line
# driver=rtl871xdrv - if so, comment out the nl80211 driver and
# comment in the rtl871drv line
ssid=SkippersMate
hw_mode=g
channel=5
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=skippersmate
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Now tell hostapd where to find its configuration file :

```
sudo joe /etc/default/hostapd
```

```
# Uncomment DAEMON_CONF line and insert "/etc/hostapd/hostapd.conf"
# Skippers Mate
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Check that your dhcpd.conf edit is correct by trying to start the dhcp server :

```
service isc-dhcp-server restart
```

Any errors will be displayed e.g. :

```
/etc/dhcp/dhcpd.conf line 22: expecting class name.
Add in
  ^
Configuration file errors encountered -- exiting
```

If you go to line 22 you will see that the comment symbol # has been omitted :

```
# Add in Skippers Mate
subnet 192.168.51.0 netmask 255.255.255.0 {
```

# Skipper's Mate

## Why You Need One and How to Get Going

Check that your edit on the configuration file has worked by trying to start hostapd :

```
sudo hostapd /etc/hostapd/hostapd.conf
```

If there are errors this will report these e.g.:

```
Configuration file: /etc/hostapd/hostapd.conf
Line 1: unknown configuration item 'nterface'
1 errors found in configuration file '/etc/hostapd/hostapd.conf'
```

### Configure IP Routing between Wireless and Ethernet

You need to set this up as root so you need to be able to login as the root user. Try changing to the superuser and you will see the following error message :

```
su -
Password:
```

```
su: Authentication failure
```

So use the sudo passwd root command to set your password – and make sure you write it down !

```
sudo passwd root
Enter new UNIX password: <your_choice_WRITE_IT_DOWN>
Retype new UNIX password: <your_choice_WRITE_IT_DOWN>
passwd: password updated successfully
```

Now you can change to the superuser :

```
su -
```

Enter the password you've just chosen and you will be presented with a root prompt :

```
root@raspberrypi:~#
```

You will now need to edit the routing script.

#### Tip

This uses the bash shell and iptables so if you experience any problems you can locate these using the which command :

```
which bash
```

which in this instance returns :

# Skipper's Mate

## Why You Need One and How to Get Going

```
/bin/bash
```

or :

```
which iptables
```

returns :

```
/sbin/iptables
```

### # end of Tip

So, to edit the script dorouting :

```
joe dorouting
```

Enter the details from the box below.

This uses the bash command shell, sets up three variables – one for the path to the command iptables and one each for the WAN and LAN side. The latter will be changed later but you need them set as this now. Then adds the routing in the last three lines.

Make sure you copy the lines correctly – iptables is very powerful and very finicky.

```
#!/bin/bash
IPT=/sbin/iptables
WAN0=eth0
LAN0=wlan0

$IPT -t nat -A POSTROUTING -o $WAN0 -j MASQUERADE
$IPT -A FORWARD -i $WAN0 -o $LAN0 -m state --state RELATED,ESTABLISHED -j ACCEPT
$IPT -A FORWARD -i $LAN0 -o $WAN0 -j ACCEPT
```

It is essential that the lines appear exactly as they do in the box above – which is why this has been shown in a small font so it does not have any extra line feeds. Joe may insert extraline feeds. If you cut-and-paste, make sure that \$IPT is the first thing on the line and use the backspace before any ACCEPT commands to make sure that they are at the end of the line. If this is incorrect, when you come to run the script (see below) you will get an error.

Now make the script executable by adding the permission x :

```
chmod +x dorouting
```

And now run dorouting

```
./dorouting
```

If you see this error, or something similar, then check the dorouting file in the editor joe against the one shown above.

```
iptables v1.4.14: option "-j" requires an argument
Try `iptables -h' or 'iptables --help' for more information.
```

# Skipper's Mate

## Why You Need One and How to Get Going

```
./dorouting: line 8: ACCEPT: command not found
```

If the script is successful it will simply return to a clean command line :

```
root@raspberrypi:~#
```

You can check what iptables has done by listing iptables :

```
iptables -L
```

This is what you should see :

```
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination
ACCEPT     all  --  anywhere              anywhere             state RELATED,ESTABLISHED
ACCEPT     all  --  anywhere              anywhere
ACCEPT     all  --  anywhere              anywhere             state RELATED,ESTABLISHED
ACCEPT     all  --  anywhere              anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
```

Once you're happy that works, you can insert the call to dorouting into crontab so that it runs at boot time :

```
crontab -e
```

Add the following line, making sure you add a carriage return/line feed (EOL). crontab takes the same commands to save or quit without saving, as Joe (<ctrl>x or <ctrl>q).

```
@reboot /root/dorouting
```

```
crontab: installing new crontab
new crontab file is missing newline before EOF, can't install.
Do you want to retry the same edit? (y/n)
```

Don't reboot yet !

### Enable Packet Forwarding

To do this you need to edit the file /etc/sysctl.conf (still as user root) :

```
joe /etc/sysctl.conf
```

# Skipper's Mate

## Why You Need One and How to Get Going

At about line 26 you should see these lines. Remove the comment so that `net.ipv4.ip_forward=1` is an active line.

```
# Uncomment the next line to enable packet forwarding for IPv4
# Comment forward line in Skippers Mate
net.ipv4.ip_forward=1
#net.ipv4.ip_forward=1
```

Now, from the Linux shell on your computer run this command and cross your fingers !

*Tip – the reboot; exit command means that you can re-use the shell instead of having to close it and open a new shell. You should still be user root – if not, then precede with a sudo.*

```
reboot ; exit
```

### Setup RAM disk

**What is a RAM disk ?** Also known as a RAM drive, this is a block of memory (volatile memory not disk or card storage) that is treated as temporary storage.

**Why does Skipper's Mate use a RAM disk ?** It is used for temporary files and database files. This not only speeds up the writes but also reduces the number of writes to the SD card. Flash memory, such as SD/micro SD cards, permit a limited number of writes. This number is very high but by putting those files which have frequent writes – such as log files, video and snapshots and the details of GPS position changes – into memory you extend the life of the card. Skipper's Mate writes selected information to the SD card, and/or to the Skipper's Mate server, as appropriate but does not write all information which is on the RAM disk.

We suggest a 200 Mb RAM disk if you are using a Raspberry Pi 2. If you're using an earlier model with less memory you may wish to change this to, say, 100Mb i.e. `size=100m`

```
sudo joe /etc/fstab
```

Add the following line, ensuring you add a line break after the second zero :

```
asbo /mnt/ramdisk tmpfs size=200m 0 0
```

**Why asbo ?** You need to assign a name for the purposes of `fstab` – and our dog is called Asbo. Go figure !

Create a directory for the mountpoint for the ramdisk under `/mnt` :

```
sudo mkdir /mnt/ramdisk
```

Mount the directory :

```
sudo mount -a
```

# Skipper's Mate

## Why You Need One and How to Get Going

List the contents of the ramdisk :

```
ls /mnt/ramdisk
```

The ls should return no files – i.e. you should simply see another shell prompt. If you have not created the ramdisk successfully you will see a message like this :

```
ls: cannot access /mnt/ramdisk: No such file or directory
```

### Create pcl Group and Members

Skipper's Mate requires a Linux user group called pcl which has permission to access the Skipper's Mate directories.

**Why pcl ?** Project Computers Ltd – the company which produces Skipper's Mate.

There need to be two Linux users as members of this group – the existing Raspberry user **pi** and the user which the web server, Lighttpd, and Raspbian use to access web pages – **www-data**.

```
sudo groupadd pcl
```

Successful creation will return you a clean command prompt.

Add the pi and www-data users to the pcl group :

```
sudo usermod -a -G pcl www-data
```

Check that this has been completed successfully :

```
getent group | grep -i 'pcl'
```

If successful, you should see a line like the one below :

```
group:password:groupnumber:user list
```

You can ignore the password and groupnumber. The important bit is that the group pcl has users pi and www-data as members.

```
pcl:x:1004:pi,www-data
```

### Install Motion software

**What is Motion ?** Motion software is a great piece of Open Source software which monitors a video signal for motion detection written by Kenneth Lavresen. It runs under Linux and produces .jpg or .ppm files and video sequences. To find out more about the software visit Kenneth's Motion web site <http://www.lavrsen.dk/foswiki/bin/view/Motion/WebHome> . There is plenty of documentation on Motion on his web site.



# Skipper's Mate

## Why You Need One and How to Get Going

Skipper's Mate, which is also Open Source, uses Motion as part of its security features. We are very grateful to Kenneth for his hard work. Skipper's Mate provides a GUI front end, and default configuration, which should make it seamless to get your camera(s) going with

Motion is present in the Raspbian repositories so you can install it with apt-get :

```
sudo apt-get install motion
```

This is quite a large download so may take two or three minutes.

Ignore the warning message. The Motion software can run in two modes – foreground and background (silent – no messages to the command line).

```
[warn] Not starting motion daemon, disabled via
/etc/default/motion ... (warning).
Setting up ffmpeg (6:0.8.17-1+rpil) ...
```

To edit the Motion configuration file :

```
sudo joe /etc/motion/motion.conf
```

This is a large file – over 600 lines. It helps to use the search function in joe (<ctrl> k f) but the approximate line number is shown. Obviously this file may change over time so the line numbers may not be exact matches for the version you install.

Set the daemon (run automatically in background) to **on** (about **line 11**) :

```
# Start in daemon (background) mode and release terminal (default:
off)
#daemon off
daemon on
```

Change the target directory to use RAM disk – target\_dir /mnt/ramdisk to save writes and increase speed (about **line 359**) :

```
# Target base directory for pictures and films
# Recommended to use absolute path. (Default: current working
directory)
#target_dir /tmp/motion
target_dir /mnt/ramdisk
```

Set webcam\_motion to 1 (i.e. on). (*This is not a typo – the options are off, 1, 2 etc.*) (about **line 407**) :

# Skipper's Mate

## Why You Need One and How to Get Going

```
# Output frames at 1 fps when no motion is detected and increase to
the
# rate given by webcam_maxrate when motion is detected (default: off)
#webcam_motion off
webcam_motion 1
```

Set the webcam\_maxrate to 20 (about *line 412*) :

```
# Maximum framerate for webcam streams (default: 1)
#webcam_maxrate 1
webcam_maxrate 20
```

Ensure you make the webcam accessible externally or you won't be able to see them from your on-board computers/tablets/phones or when you access your boat via the Skipper's Mate web server (about *line 415*):

```
# Restrict webcam connections to localhost only (default: on)
#webcam_localhost on
webcam_localhost off
```

Similarly you want to be able to control the webcam(s) from a device other than the Raspberry itself. Access will be user/password controlled for security (about *line 431*).

```
# Restrict control connections to localhost only (default: on)
#control_localhost on
control_localhost off
```

Each camera has a separate configuration file. Skipper's Mate calls them fred1.conf, fred2.conf etc. so edit in the line thread /usr/local/etc/fred1.conf (about *line 635*) :

```
#####
# Thread config files - One for each camera.
# Except if only one camera - You only need this config file.
# If you have more than one camera you MUST define one thread
# config file for each camera in addition to this config file.
#####

# Remember: If you have more than one camera you must have one
# thread file for each camera. E.g. 2 cameras requires 3 files:
# This motion.conf file AND thread1.conf and thread2.conf.
# Only put the options that are unique to each camera in the
# thread config files.
# Add fred.1 line - for Skippers Mate - makes it so much more
personal
thread /etc/motion/fred1.conf
; thread /usr/local/etc/thread1.conf
```

# Skipper's Mate

## Why You Need One and How to Get Going

```
; thread /usr/local/etc/thread2.conf
; thread /usr/local/etc/thread3.conf
; thread /usr/local/etc/thread4.conf
```

Save your edited file.

Edit the configuration file for the first camera :

```
sudo joe /etc/motion/fred1.conf
```

The default configuration file created for Skipper's Mate is nice and simple. These settings can be accessed via the Skipper's Mate web pages. Details of these settings are available in in the Skipper's Mate Help (available from the Help button anywhere in Skipper's Mate once you have installed it and browse to the system).

```
text_left Cam1
text_double off
ffmpeg_cap_new off
locate off
gap 20
pre_capture 3
post_capture 0
ffmpeg_video_codec mpeg4
threshold 1500
framerate 2
minimum_frame_time 0
output_normal off
height 240
webcam_quality 50
snapshot_interval 0
width 320
```

### Install PHP

In order to run the web pages you will need to install the PHP software including associated packages for cgi (Common Gateway Interface allowing a web server to interact with programs) :

```
sudo apt-get install php5-common php5-cgi php5
```

### Install Web Server – Lighttpd

**Why Lighttpd ?** Our company standard for large servers is the world's most popular web server – Apache. However, the web server on the Raspberry is only intended for serving web pages and not the huge range of facilities that Apache offers. Lighttpd is an excellent small, light-weight web

# Skipper's Mate

## Why You Need One and How to Get Going

server. That's why.

Before installing Lighttpd, we want to create an html directory under /var/www as a document root and give it permissions for user pi and group pcl.

```
cd /var
sudo mkdir www
cd www
sudo mkdir html
sudo chown pi:pcl html
```

To install Lighttpd :

```
sudo apt-get install lighttpd
```

This should start the web server and we want to configure it so stop it first :

```
sudo service lighttpd stop
```

```
[ ok ] Stopping web server: lighttpd.
```

To enable the fastcgi module for the web server, Lighttpd enter :

```
sudo lighty-enable-mod fastcgi-php
```

You should see output like this :

```
Met dependency: fastcgi
Enabling fastcgi-php: ok
Enabling fastcgi: ok
Run /etc/init.d/lighttpd force-reload to enable changes
```

Don't reload yet – we need to configure Lighttpd first. Now configure Lighttpd by editing lighttpd.conf :

```
sudo joe /etc/lighttpd/lighttpd.conf
```

# Skipper's Mate

## Why You Need One and How to Get Going

Add this section immediately before the `server.document-root` line :

```
alias.url += ("/cgi-bin" => "/var/www/cgi-bin")
alias.url += ( "/phppgadmin" => "/usr/share/phppgadmin/")
$http["url"] =~ "^/cgi-bin" {
    cgi.assign = ( "" => "" )
}

cgi.assign = (
    ".cgi" => ""
)
```

Change the `server.document-root` line by adding the `html` directory :

```
server.document-root = "/var/www/html"
```

Comment out the `upload-dirs` :

```
#server.upload-dirs = ( "/var/cache/lighttpd/uploads" )
```

Save your edits.

Now restart the web server :

```
sudo service lighttpd start
```

You should see that the web server has started successfully :

```
[ ok ] Starting web server: lighttpd.
```

These steps should have enabled `fastcgi` and `fastcgi-php`. If you list the conf-enabled directory beneath `lighttpd` :

```
ls /etc/lighttpd/conf-enabled -la
```

You should see something like this :

```
lrwxrwxrwx 1 root root 33 May 3 12:20 10-fastcgi.conf -> ../conf-available/10-fastcgi.conf
lrwxrwxrwx 1 root root 37 May 3 12:20 15-fastcgi-php.conf ->
../conf-available/15-fastcgi-php.conf
```

These are symbolic links to the packages available in `lighttpd/conf-enabled` indicating that those packages are in use.

### Install the SQLite Database

**Why SQLite ?** This light-weight database is used on the RAMdisk for rapid access and temporary storage.

Install the SQLite server :

```
sudo apt-get install sqlite3
```

# Skipper's Mate

## Why You Need One and How to Get Going

### Install the PostgreSQL Database

**Why PostgreSQL ?** This is an Open Source industry-standard relational database which we use on our servers. This is installed on the Raspberry to keep the local database in step with that on the web servers which permit remote access to your boat.

Install PostgreSQL. This can take some minutes – there is a lot to be installed and configured.

```
sudo apt-get install postgresql
```

Edit the configuration file `pg_hba.conf` :

```
sudo joe /etc/postgresql/9.1/main/pg_hba.conf
```

The commented-in section normally begins near the end of the file (about line 85). Change the METHOD entries to trust (4) : local all postgres, local all all, and both IPv4 and IPv6 entries. It is better to edit rather than cut-and-paste as the tabs can confuse the editor.

```
# Database administrative login by Unix domain socket
#local all postgres peer
local all postgres trust

# TYPE DATABASE USER ADDRESS METHOD

# "local" is for Unix domain socket connections only
#local all all peer
local all all trust
# IPv4 local connections:
#host all all 127.0.0.1/32 md5
host all all 127.0.0.1/32 trust
# IPv6 local connections:
#host all all ::1/128 md5
host all all ::1/128 trust
```

Restart the PostgreSQL server :

```
sudo service postgresql restart
```

If this starts successfully you should see :

```
[ ok ] Restarting PostgreSQL 9.1 database server: main.
```

Followed by a clear prompt line.

### Install the PostgreSQL Administration Tool - phpPgAdmin (optional)

You may find it useful to have the web-based administration tool for PostgreSQL. If so use :

```
sudo apt-get install phpPgAdmin
```

# Skipper's Mate

## Why You Need One and How to Get Going

### Install Skipper's Mate

Change directory to the RAM disk :

```
cd /mnt/ramdisk
```

Download the compressed tar file from the Skipper's Mate web site :

```
wget www.skippersmate.co.uk/skippersmate/downloads/sm.tar.gz
```

Unpack tar file. You should be in the root (/) of /mnt/ramdisk.

```
tar -zxvf sm.tar.gz
```

This should leave you with the following files and directories - if you list the contents using **ls** :

```
etc sm.tar.gz usr var
```

Now copy the unzipped/untarred directories to their homes on the Raspberry.

Copy the usr directory tree to /usr

```
sudo cp -r -v usr/lib/* /usr/lib
```

```
sudo cp -r usr /usr
```

Copy etc directory tree to /etc

```
sudo cp -r etc /etc
```

Copy www directory tree to /var

```
sudo cp -r www /var
```

You will need to change ownership and permissions under the /var directory tree as follows.

Change ownership of all files and directories under /var/www to owner pi and group pcl

```
sudo chown -R pi:pcl /var/www
```

Now change to the /var/www/cgi-bin directory :

```
cd /var/www/cgi-bin
```

Change permissions on the following cgi files which need to be executable. You need to cut and paste as two statements as the command line does not accept a line break :

```
sudo chmod 4755 apconnect.cgi driverupd.cgi pclcmd.cgi reboot.cgi  
sudo chmod 4755 smdownload.cgi wireless.cgi writehostapd.cgi
```

# Skipper's Mate

## Why You Need One and How to Get Going

### Load the Skipper's Mate Databases

Change to the pcl database directory :

```
cd /var/www/html/pcl/db
```

Create the PostgreSQL users and database and load the base tables :

```
sh pcldoitall.sh
```

You should see lots of messages on the command line, including multiple occurrences of NOTICE statements like this example, ending in a series of GRANT statements :

```
psql:pclcreatestdtables.sql:585: NOTICE: CREATE TABLE will create
implicit sequence "pcl_report_parms_reportid_seq" for serial
column "pcl_report_parms.reportid"
INSERT 0 1
GRANT
```

```
cd /var/www/html/pcl/sm/db
```

Run the script :

```
sh smdoitall.sh
```

You should see more NOTICE: CREATE TABLE statements and plenty of INSERT 0 1 statements, finishing with lots of INSERTs. At this point your database should be complete.

You should now be able to attach with a browser, either via the machine you are using to load the Raspberry or via a wireless device such as your tablet.

Browse to <http://192.168.51.254/pcl/sm/smsignin.html>

Login as

### Install the Network Bridge

**Why do I need a bridge ?** A bridge is a clever way to connect wired and wireless access.

Install the bridge utilities :

```
sudo apt-get install bridge-utils
```

Create the bridge :



# Skipper's Mate

## Why You Need One and How to Get Going

```
sudo brctl addbr br0
```

Add the network interfaces to be bridged :

```
sudo brctl addif br0 eth0 wlan0
```

Edit the dhcp server configuration to include the bridge :

```
sudo joe /etc/default/isc-dhcp-server
```

Change INTERFACES from wlan0 to the bridge, br0 :

```
# On what interfaces should the DHCP server (dhcpcd) serve DHCP
requests?
#     Separate multiple interfaces with spaces, e.g. "eth0 eth1".
# Add wlan0 for Skippers Mate
#INTERFACES="wlan0"
INTERFACES="br0"
```

Edit the interfaces configuration file :

```
sudo joe /etc/network/interfaces
```

Comment out the iface dhcp line. Add auto wlan0 and the section for the bridge (br0) shown below. Comment out :

```
auto lo

iface lo inet loopback
#iface eth0 inet dhcp

# add auto wlan0 for SM and comment out the iface wlan0 section SM
2nd edit
allow-hotplug wlan0
auto wlan0

auto br0
iface br0 inet static
address 192.168.51.254
network 192.168.51.0
netmask 255.255.255.0
broadcast 192.168.51.255
bridge-ports eth0 wlan0

# change next line manual to static add address and netmask lines
and
# gateway for SM 1st edit
#iface wlan0 inet static
#address 192.168.51.254
#netmask 255.255.255.0
```

# Skipper's Mate

## Why You Need One and How to Get Going

```
#gateway 10.0.1.254
# end of mod

# comment out wpa-roam and iface for SM 2nd edit
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
#iface default inet dhcp
```

### Set Up the Routing Script

Change to superuser :

```
su -
```

Edit the dorouting script :

```
root@raspberrypi:~# joe dorouting
```

```
#!/bin/bash
IPT=/sbin/iptables
# change WAN0 from eth0 to eth1 SM
WAN0=eth1
LAN0=wlan0
# SM Add LAN1 and loop for dongle (could add wireless receiver if
they have one in Wans)
LAN1=eth0
declare -A Wans=(
    [DONGLE]=$WAN0
)
# replace three end statements with do loop
$IPT -F
$IPT -t mangle -X
for i in "${!Wans[@]}";
-t <table name> e.g. -t nat
do
    Strg="${Wans[$i]}";
    $IPT -t nat -A POSTROUTING -o $Strg -j MASQUERADE
    $IPT -A FORWARD -i br0 -o $Strg -m state --state
RELATED,ESTABLISHED -j ACCEPT
    $IPT -A FORWARD -i $Strg -o br0 -j ACCEPT
done

#$IPT -t nat -A POSTROUTING -o $WAN0 -j MASQUERADE
#$IPT -A FORWARD -i $WAN0 -o $LAN0 -m state --state
RELATED,ESTABLISHED -j ACCEPT
#$IPT -A FORWARD -i $LAN0 -o $WAN0 -j ACCEPT
```

# Skipper's Mate

## Why You Need One and How to Get Going

Change back from the root user to the pi user :

```
sudo pi
```

Edit the network interfaces configuration :

```
sudo joe /etc/network/interfaces
```

```
auto lo
iface lo inet loopback
# change dhcp from eth0 to eth1
#iface eth0 inet dhcp
iface eth1 inet dhcp
```

Add auto wlan0 for Skipper's Mate.

```
allow-hotplug wlan0
auto wlan0
```

Add in the auto br0 section as shown below :

```
auto br0
  iface br0 inet static
  address 192.168.51.254
  network 192.168.51.0
  netmask 255.255.255.0
  broadcast 192.168.51.255
  bridge-ports eth0 wlan0
```

Comment out the iface wlan0 section which we inserted earlier when we changed from a manual wlan0 to static :

```
# change next line manual to static add address and netmask lines
and
# gateway for SM 1st edit
#iface wlan0 inet static
#address 192.168.51.254
#netmask 255.255.255.0
#gateway 10.0.1.254
# end of mod
```

Comment out wpa-roam and iface lines :

```
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
#iface default inet dhcp
```

# Skipper's Mate

## Why You Need One and How to Get Going

Edit the hostapd configuration file to include the bridge :

```
sudo joe /etc/hostapd/hostapd.conf
```

Add a line for the bridge immediately after the interface line :

```
interface=wlan0  
bridge=br0  
driver=nl80211
```

*Tip – ensure that there are no spaces at end of line or hostapd will not start. To test this XXX*

### Install the VNC server

**Why do I need a VNC Server ?** A VNC (Virtual Network Computing) server is a graphical desktop sharing system. You **DO NOT** need this for Skipper's Mate as this is controlled via a browser. However, if you wish to use other software which comes in desktop form rather than as thin client (via a browser) then you may find this useful. For example, the great Open Source navigational system is desktop and runs under Raspbian. To use the Raspberry as a server you need to install VNC server software here and then one of the many VNC viewers on the device(s) you want to run on – such as your tablet, laptop or even smartphone. See the section on **Installing a VNC Viewer**

Install TightVNCServer:

```
sudo apt-get install tightvncserver
```

Start the TightVNCServer

```
tightvncserver
```

You will be prompted for a password – don't forget to write it down !

```
You will require a password to access your desktops.
```

```
Password:  
Verify:  
Would you like to enter a view-only password (y/n)? n
```

You should now see a message like this – the VNC server should be running.

```
xauth: file /root/.Xauthority does not exist  
New 'X' desktop is raspberrypi:1  
Creating default startup script /root/.vnc/xstartup  
Starting applications specified in /root/.vnc/xstartup  
Log file is /root/.vnc/raspberrypi:1.log
```

# Skipper's Mate

## Why You Need One and How to Get Going

Once you have installed your VNC Viewer on a device you will be able to access any desktop applications you choose to install on the Raspberry – such as the free Open Source navigation system OpenCPN (available from <http://www.opencpn.org>).

### How to Install a VNC Viewer

#### For Android users

Visit the Google Play Store and search for VNC Viewer. At the time of writing the VNC Viewer from RealVNC is free and we find it easy to set up and use. There are also other free VNC Viewers available.

Open the app and hit the **+** sign to create a new connection.

Enter the IP address of your Raspberry Pi followed by : **5901** to access the VNC server over port 5901 :

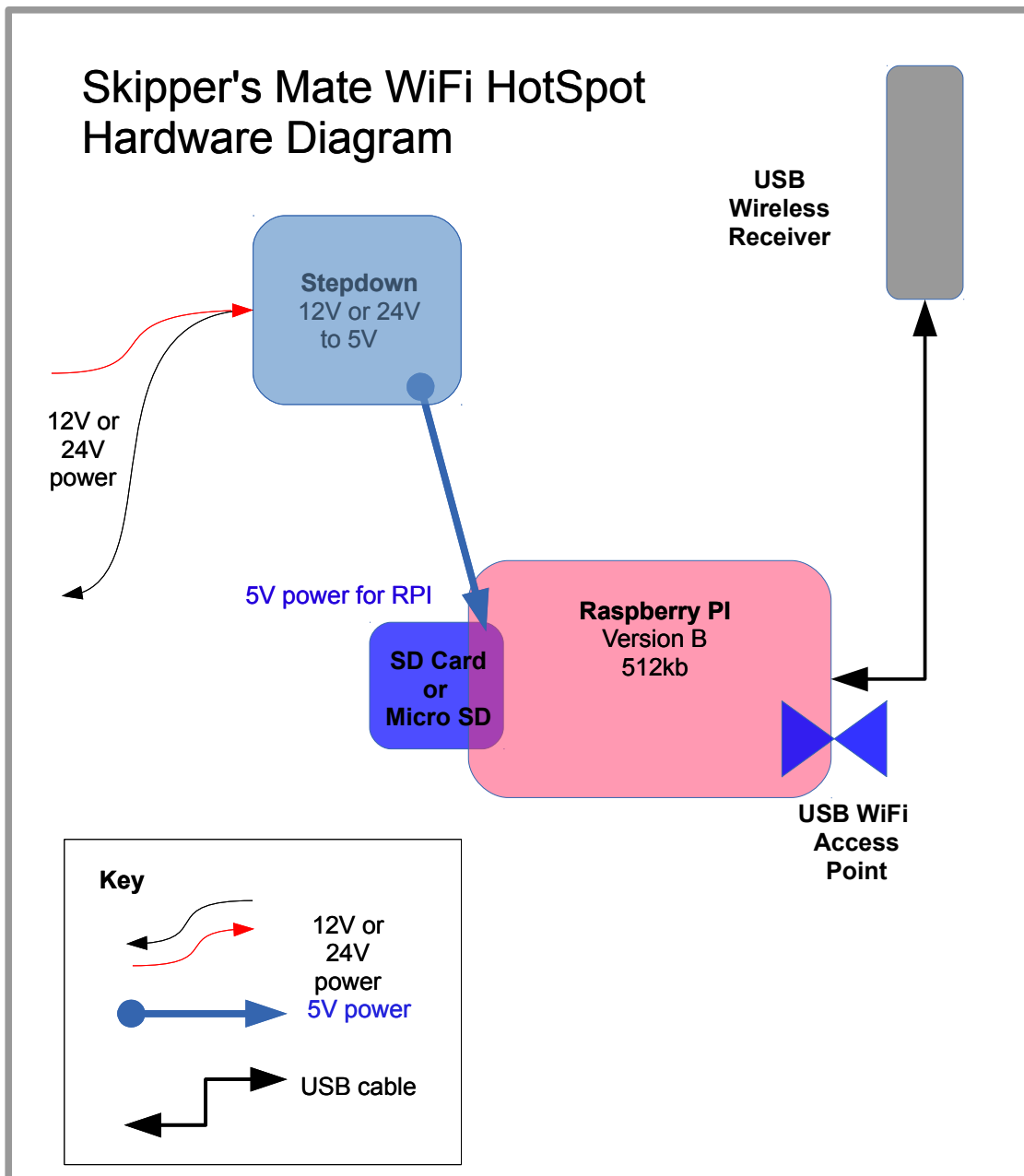
Address 10.0.1.122:5901

XXX Check this when I have an Rpi up

# Skipper's Mate

## Why You Need One and How to Get Going

### APPENDIX A - Hardware Diagram for WiFi HotSpot Only



# **Skipper's Mate**

## **Why You Need One and How to Get Going**

### ***APPENDIX B - Hardware Diagram for Skipper's Mate - Next Version***

# **Skipper's Mate**

## **Why You Need One and How to Get Going**

### ***APPENDIX C - joe Editor Commands - a simple sub-set***