1 Introduction

This document provides a step by step guide to get rocrail (www.rocrail.net) running on a raspberry pi (www.raspberrypi.org/).

It also contains specific configuration details regarding specific hardware I intend to use: touch screen, video capture usb and interface board to digital IO in raspberry.

The final objective is to get the rpi running on a very simple configuration, starting rocrail as a daemon, and enabling the launching of rocview through the activation of a digital input at the rpi. The DDX software command station from rocrail will be used to send the commands to trains and accessories, whilst the Loconet protocol will be used for the sensors in the layout.

My intention is to use this set up with a touch screen (Packard bell Viseo 200T) and an interface board to rpi. This interface board should take care of getting the serial signal from the rpi to a booster and enabling a switch to start rocview.

In the future, it could be studied if a new software command station interfacing one of the digital outputs at the rpi could be used instead of DDX (serial output).

1.1 Your rpi

I will be assuming you are running the recommended Raspbian "wheezy" distro. You can get it here: http://www.raspberrypi.org/downloads. The only thing you need to know is the user and password: Username: pi Password: raspberry.

Eventually, I will publish my own image of my system.

One of the things to study is if any of the other distros being used by the rpi users can improve the performance of rocrail.

Follow the instructions on the raspberry pi website and you will not have any problem. The first time you star your rpi, raspi-config is called to set up locations, keyboard, language, overclocking, etc.

Some of the additional things you probably will like to do are:

- *Extend the size of your partitions. The distro for Raspbian is in a 2GB image, if you are using a bigger SD (recommended) you will need to extend the main partition.*
- *Set a fix IP.*
- *Setting up the sound. In Wheezy the sound is set up by default and operates on the hdmi port. You may need to redirect the sound to the headphones output.*
- *Increase the video signal at the hdmi port. Some TVs and screens needs it.*
- *Get rid of the big black borders around the image. Again, a typical problem with some TVs and screens.*
- *Share a folder.*
- *Clean some of the daemons starting with the system, so it starts up quicker.*
- *Change the name of your rpi.*
- *Backup your system.*

All this procedures can be found in section 4.1 of this document.

2 Compile Rocrail

I normally use windows and eclipse to compile rocrail. I am not familiar with linux, so the instructions hereafter are mainly lessons learnt when trying to get rocrail running.

First, I studied if cross-compiling from windows was an option. I have not found any one doing any kind of cross-compilation for rpi in windows – not without emulating the raspberry. I do think it should be possible… but I do not know how.

On the other hand, cross-compiling from linux seems to be a standard procedure for the people developing for the raspberry pi, so it should be possible – and easy? – to achieve it.

Considering the information in the rocrail forum, and my current limited knowledge on linux, I decided to go the "easiest way" and compile directly on the rpi.

2.1 Prerequisites

2.1.1 Prepare a place to download everything

This is not absolutely necessary, but in order to make clear where each of the components will be downloaded and installed, it is convenient to create some folders. This guide will refer to them.

```
cd /home/pi
mkdir rocrail
cd rocrail
mkdir source
mkdir wxGTK
```

You should be already in your home directory (/home/pi)… pi is the name of the user, but if you have created another user, you could use it.

2.1.2 Install Bazaar

This should be the easy part:

```
sudo apt-get install bzr
```

Accept all dependencies and request to install new software you may receive.

2.1.3 Build wxwidgets 2.8x from source

Source: http://wiki.rocrail.net/doku.php?id=buildwx-en
http://forums.wxwidgets.org/viewtopic.php?f=23&t=34891

1. Download: http://prdownloads.sourceforge.net/wxwindows/wxGTK-2.8.12.tar.gz
   Note that the last version may be different (2.8.x). Currently I am using 2.8.12.

```
cd/home/pi/rocrail/wxGTK
wget http://prdownloads.sourceforge.net/wxwindows/wxGTK-2.8.12.tar.gz
```

2. **Install gtk2-devel and all other missing development tools from your distribution!!!**

```
sudo apt-get install libgtk2.0-dev
```

3. Unzip the source tarball:

```
gzip -d wxGTK-2.8.12.tar.gz
```

4. Un tar it:

```
tar -xf wxGTK-2.8.12.tar
```

5. Configure the compilation:

```
cd wxGTK-2.8.12
/configure --enable-unicode --enable-graphics_ctx
```

   This could take a while.

6. Make.

```
make
```

   This could take a while (even longer than configuring).

7. install the wx headers and libraries:

```
sudo make install
```

8. update the ld.so.conf:

```
sudo /sbin/ldconfig
```

2.2 Compiling Rocrail

Source: http://wiki.rocrail.net/doku.php?id=build-en

This is also quite easy with the information in the rocrail web site:

1. *Download the sources.*

```
cd /home/pi/rocrail/source
bzr checkout --lightweight https://launchpad.net/rocrail Rocrail
```

2. *Make the files.*

```
cd /home/pi/rocrail/source/Rocrail
make PLATFORM=LINUX fromtar
```

3. *Install.*

```
cd /home/pi/rocrail/source/Rocrail
sudo make install
```

*Rocrail is installed in **/opt/rocrail**.*

2.3 Creating a debian distribution package

Source: http://wiki.rocrail.net/doku.php?id=mkdeb-en
This step is completely optional, and actually I have not tried its results (feedback is welcome).
There is a script to build the package already provided by rocrail. I have not been able to follow step by step the instructions, so here is a possible alternative:

1. *Create a control file for raspberry with hard float (armhf).*

```
cd /home/pi/rocrail/source/Rocrail/rocrail/package
cp control control-armhf
```

2. *Edit this file and change the architecture to "armhf" (just to make it nice).*

```
nano control-armhf
```

3. *Run the script.*

```
./mkdeb.sh debian armhf
```

   Instead of debian, you can call it as you want… in my case "rocpi".
4. *The resulting file is in **/home/pi/rocrail/source/Rocrail/package** and looks like this: **rocrail-setup-rev4082-rocpi-armhf.deb**.*
   The only problem is that it is recovering the revision number from the repository in internet… so it may be the case that the revision noted in the name is not the same as the one you have compiled – depending on how long it has taken you to compile. To be sure, follow step 5.
5. *Check the revision.*

```
cd /home/pi/rocrail/source/Rocrail/common
nano version.h
```

   You can see there the revision you have actually compiled, and if necessary rename the file:

```
cd /home/pi/rocrail/source/Rocrail/package
mv rocrail-setup-rev4082-rocpi-armhf.deb rocrail-setup-rev4011-rocpi-armhf.deb
```

   In the first file the old name, in the second file the new name.
6. *Now you should be able to install the file (I have not tried this last step.*

```
cd /home/pi/rocrail/source/Rocrail/package/
sudo dpkg -i rocrail-setup-rev4011-rocpi-armhf.deb
```

3 Running your rocpi

3.1 Preparing the environment

3.1.1 Getting all the files needed

The first thing you need to do is to copy your ***image*** *and* ***svn*** *folders to /opt/rocrail/*

It seems that after compiling, they are not installed with the rest of the program, and at least the svg folder is critical.

```
cd /opt/rocrail
sudo mkdir svn
sudo mkdir images
sudo cp /home/pi/rocrail/sources/Rocrail/rocrail/package/images/*.* images
sudo cp -rf /home/pi/rocrail/sources/Rocrail/rocview/svg/ .
```

Note the last "." at the end

3.1.2 Releasing the use of the serial port

Source: http://elinux.org/RPi_Serial_Connection

To make use of the DDX you will need to clear the use of the serial port in the rpi. By default, there are a number of messages going to the serial port.

The first is ***kernel boot messages*** *at startup, those are caused by the kernel parameter console=ttyAMA0,115200 set in /boot/cmdline.txt.* ***Kernel debuging*** *is also on the serial port. To remove them:*

1. *Make a copy of the cmdline.txt file*

```
cd /boot/
sudo cp cmdline.txt cmdline.bck
```

2. *Remove the following parameters:* ***console=ttyAMA0,115200*** *and* ***kgdboc=ttyAMA0,115200***

```
sudo nano cmdline.txt
```

Next is the login prompt, set in /etc/inittab

1. *Make a copy of the file*

```
cd /etc/
sudo cp inittab inittab.bck
```

2. *Comment (#) the line the line* ***T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100***

```
sudo nano inittab
```

WIP:

"Some distro's may be setup differently, just double check 'ps aux|grep ttyAMA0' and 'cat /proc/cmdline' to verify ttyAMA0 is not in use anywhere".

3.2 Launching rocrail

Running rocrail should be easy at this point. Just remember it has to be run with root privileges.

```
cd /opt/rocrail/
sudo ./rocrail
```

I have tried to run it from another folder, but I always get errors related to dependences with other files. Since in the typical setting, you will be running rocrail as a daemon you will not usually need to follow the above procedure.

3.3 Running rocrail as a daemon

This is an optional configuration, and it is an alternative to launching rocrail as in section 3.2. This procedure will install a daemon in the rpi running rocrail each time the system is boot.

Source: *http://wiki.rocrail.net/doku.php?id=rocrail-daemon-en*

http://forum.rocrail.net/viewtopic.php?t=2569&highlight=daemon

http://www.thegeekstuff.com/2012/03/lsbinit-script/

http://ubuntuforums.org/showthread.php?t=1028066

The configuration should be straightforward following the wiki on rocrail website:

1. *Copy the files needed to the right places.*

   ```
   sudo cp /home/pi/rocrail/source/Rocrail/rocrail/package/rocraild.sh
   /opt/rocrail/
   sudo cp /home/pi/rocrail/source/Rocrail/rocrail/package/rocraild
   /etc/init.d/
   ```

2. *Make the files executable.*

   ```
   sudo su
   chmod 755 /etc/init.d/rocraild
   chmod 755 /opt/rocrail/rocraild.sh
   ```

3. *And here the difference I had to make in order to get the daemon running on boot. I do not know the reasons; I just followed the instructions/suggestions in the forum -it should have worked without this modification.*
   a. *Edit the file*

      ```
      sudo nano /etc/init.d/rocraild
      ```

   b. *And include the following levels in the header part from,*

      ```
      # Default-Start:     3
      # Default-Stop:
      ```

      to

      ```
      # Default-Start:     2 3 4 5
      # Default-Stop:      0 1 6
      ```

4. *Finally add the appropriate symbolic links to cause the daemon to be run when the system goes down, or comes up.*

   ```
   sudo update-rc.d rocraild defaults
   ```

   If you do not receive a clean message, you may have arrived to the same problem I had, and you will need step 3.But before, you will need to remove the link:

   ```
   sudo update-rc.d -f rocraild remove
   ```

   And then create the symbolic links again.

   ```
   sudo update-rc.d rocraild defaults
   ```

5. *With everything in place, rocrail should be launched during start up, and stop when shutting down. It is always possible to start and stop the service manually:*
   Start:

   ```
   sudo /etc/init.d/rocraild start
   ```

   Stop:

   ```
   sudo /etc/init.d/rocraild stop
   ```

3.3.1 Customizing the daemon

This section is not necessary to run the daemon. The configuration hereafter serve to make the daemon look like the rest of processes in startup, and to get a copy of rocrail trace in a second terminal (tty2).

3.3.1.1 Colors and messages

If you see the boot sequence of the rpi, you will see that all services are marked with an [OK], [fail],[info], etc. tag, depending if they are launched correctly. In order to get this feature, you would need to edit the rocraild file in /etc/init.d/.

Note that the following code is just to make the message look nice. Adapting the daemon to the debian template would probably require more modifications:

1. *Open the file.*

```
sudo nano /etc/init.d/rocraild
```

2. *Include the following line after the header to get access to log functions:* ***. /lib/lsb/init-functions***
   It should look like this.

```
## END INIT INFO

. /lib/lsb/init-functions

rocraild_BIN=/opt/rocrail/rocrail
```

3. *Then replace the echo messages with the log messages.*
   From:

```
if [ ! -x $rocraild_BIN ] ; then
       echo -n "Rocrail not installed ! "
       exit 5
fi

case "$1" in
    start)
       if [ ! -e $rocraild_PID ] ; then
              echo "Starting Rocrail"
       else
              echo "rocraild.pid already exists ! "
              exit 5
       fi
       su - root -c "$rocraild_SH"
       ;;
    stop)
       if [ -e $rocraild_PID ] ; then
              echo "Shutting down Rocrail"
       else
              echo "Rocrail not running or missing PID File ! "
              exit 5
       fi
       su - root -c "kill `head $rocraild_PID`"
       su - root -c "rm $rocraild_PID"
       ;;
*)
```

To:

```
if [ ! -x $rocraild_BIN ] ; then
       log_daemon_msg "Rocrail not installed !" "rocrail"
       log_end_msg 1
       exit 5
fi

case "$1" in
    start)
       log_daemon_msg "Starting Rocrail" "rocrail"
       if [ ! -e $rocraild_PID ] ; then
              su - root -c "$rocraild_SH"
       else
              log_end_msg 1
        echo "rocraild.pid already exists ! "
              exit 5
       fi
       log_end_msg 0
       ;;
    stop)
       log_daemon_msg  "Shutting down Rocrail" "rocrail"
       if [ -e $rocraild_PID ] ; then
       su - root -c "kill `head $rocraild_PID`"
       su - root -c "rm $rocraild_PID"
    else
              log_end_msg 1
        echo "Rocrail not running or missing PID File ! "
              exit 5
       fi
       log_end_msg 0
       ;;
*)
```

3.3.1.2 Copy of nohup in tty2

I want to use the second terminal (tty2) as screen to check the daemon operations. There are several ways to do it, and it is even possible to activate/deactivate the output to tty2, but I have decided to make it a default monitoring screen. In this way, pressing CTRL+ALT+F2 it is possible to monitor the daemon (not start and stop, but see what it is doing).

Additionally, I am removing all messages when starting the daemon, even the ones related to nohup.out file.

1. *To get this last feature you need to edit the rocraild.sh file.*

```
sudo nano /opt/rocrail/rocraild.sh
```

2. *Change the original file from.*

```
#!/bin/bash
cd /opt/rocrail/
rm -f nohup.out
nohup ./rocrail -l /opt/rocrail&
echo "$!" > rocraild.pid
```

To:

```
#!/bin/bash
cd /opt/rocrail/
rm -f nohup.out
touch nohup.out
nohup ./rocrail -1 /opt/rocrail >nohup.out 2>&1 &
echo "$!" > rocraild.pid
tail -f nohup.out >/dev/tty2&
```

3.3.2 Controlling the daemon from another computer

It is always possible to control the daemon from another computer through ssh, just open a session and use the start and stop commands in section 3.3.

*Additionally, if you want to see how the daemon is working, you can use the last command in the previous section (**tail -f nohup.out**).*

If you want to see what the daemon is actually sending to tty2, you need:

1. *Install linuxvnc.*

   ```
   sudo apt-get install linuxvnc
   ```

2. *Grab terminal 2 with linuxvnc.*

   ```
   sudo linuxvnc 2
   ```

3. *Make use of your favorite vnc tool to check tty2 (port in linuxvnc by default is 5900.*

3.4 Launching rocview

3.4.1 Initial configurations

Running rocview for the first time will be also useful to complete the configuration of rocrail.

*It is important you have rocrail already running, as a daemon or not. If you are running it as described in section 0, you need to push **CTRL+ALT+F2.** This will open a second shell, leaving rocrail running in **CTRL+ALT+F1.***

Once in the new shell, introduce again your user and password and open a graphical session.

```
startx
```

This will open a desktop (lxde). You need to select and open a terminal (lxterminal). In the distribution from raspberry, you should already have a link in the desktop. Otherwise, look for it at the program menu (accessories).

In the terminal, run rocview. Note you can always use this procedure to run it – not just for initial configurations.

```
cd /opt/rocrail
sudo ./rocview
```

Now you will get the typical rocview window. There are three things you need to do here:

1. *Configure ddx as describe in: http://wiki.rocrail.net/doku.php?id=ddx-en. Just note that the serial port name is **ttyAMA0**. So you must configure it as **/dev/ttyAMA0.***
   WIP. I have not tested it yet, there are different ways to configure the serial port, and to make the ORD-2 running you need SIGNAL RTD, TXD, etc.
2. *Deactivate the s88 bus in ddx. The rpi does not count with a parallel port, so you will need to use other sensing mechanism – as Loconet. To that end, just set the address to 0.*
3. *Configure Loconet, or any other mechanism you may have to replace the s88*

4. *Make rocview as big as possible. We will be running rocview directly without a desktop and a windows manager. So, in order to make your rocview use the entire screen, go to **View/Full Screen.** This is the easiest way, you could also edit your rocview.ini file.*

It is time to close rocview File/Exit, and end the graphical session (CTRL+ALT+BCK). If you cannot end with the combination of keys, simply close the session as usual (Start/Exit).

3.4.2 Launching rocview without a windows manager and desktop

In order to release as many resources as possible, you do not need to open the desktop to run rocview.

From the second shell you have in CTRL+ALT+F2, or from the main one if you are already running rocrail as a daemon, you can directly start rocview:

```
sudo xinit /usr/bin/ck-launch-session ./rocview
```

Note that the windows manager will not be active, so you will not see the typical close, maximize and minimize buttons. You will not able either to resize the window. That is the reason to make it first in section 3.4.1.

3.4.3 Launching rocview from hardware

WIP

The idea is to use one of the general purpose IO pins at the rpi to activate a script with the above mentioned procedure to start rocview.

4 Tuning up your pi

4.1 General settings

4.1.1 Extend the size of your partitions.

The distro for Raspbian is in a 2GB image, if you are using a bigger SD (recommended) you will need to extend the main partition.

Source : http://elinux.org/RPi_Resize_Flash_Partitions

This is a straightforward procedure coming from the rpi guidelines. Nevertheless I recommend getting a backup of your image (see section 4.1.9).

1. *Start the fdisk tool.*

   ```
   sudo fdisk /dev/mmcblk0
   ```

2. *Then delete partitions with d and create a new with n. You can view the existing table with p.*

   *p to see the current start of the main partition (**this is a very important piece of information if you do not want to lose any data, write down somewhere the current start**).*

   d, 3 to delete the swap partition. In the last version there is no swap, already, so you can skip this step.

   d, 2 to delete the main partition.

   n p 2 to create a new primary partition, next you need to enter the start of the old main partition and then the size (enter for complete SD card).

   w write the new partition table.

3. *Now you need to reboot:*

   ```
   sudo shutdown -r now
   ```

4. *After the reboot you need to resize the filesystem on the partition. The resize2fs command will resize your filesystem to the new size from the changed partition table.*

   ```
   sudo resize2fs /dev/mmcblk0p2
   ```

4.1.2 Set a fix IP

Source: http://elinux.org/RPi_Setting_up_a_static_IP_in_Debian

This setup is quite interesting for running rocrail in the rpi. With this setup, the clients will have a fix IP address to connect.

1. Backup the current version of the interfaces file:

   ```
   sudo cp /etc/network/interfaces /etc/network/interfaces.sav
   ```

2. Edit the file

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

   The interfaces file should have a line such as:

   ```
   iface eth0 inet dhcp
   ```

3. Disable the DHCP client by commenting the above line

   ```
   #iface eth0 inet dhcp
   ```

4. Include the fix address

   ```
   auto lo
   iface lo inet loopback
   auto eth0
   iface eth0 inet static
   #your static IP
   address 192.168.1.118
   #your gateway IP
   gateway 192.168.1.1
   netmask 255.255.255.0
   #your network address "family"
   network 192.168.1.0
   broadcast 192.168.1.255
   ```

Alternatively, you can always force your router to serve the same IP over DHCP to your raspberry MAC address (but this procedure depends on your network configuration).

4.1.3 Setting up the sound.

Source:

http://elinux.org/R-Pi_Troubleshooting#Sound_does_not_work_at_all.2C_or_in_some_applications

In Wheezy the sound is set up by default and operates on the hdmi port. You may need to redirect the sound to the headphones output.

```
sudo amixer cset numid=3 1
```

To get the sound back to hdmi:

```
sudo amixer cset numid=3 2
```

And you can check if it is working:

```
sudo aplay /usr/share/sounds/alsa/Front_Center.wav
```

4.1.4 Increase the video signal at the hdmi port.

Source: http://elinux.org/R-Pi_Troubleshooting#Interference_visible_on_a_HDMI_or_DVI_monitor

Some TVs and screens need it.

1. *Edit the rpi configuration file.*

```
sudo nano /boot/config.txt
```

2. *Add the following line to the configuration file, replace de value already there with 4, or uncomment the line.*

```
config_hdmi_boost=4
```

3. *Exit saving.*

```
Press Control-x
Press y
Press [enter]
```

4. *After exiting the editor, restart using the command.*

```
sudo reboot
```

4.1.5 Get rid of the big black borders around the image.

Again, this is a typical problem with some TVs and screens.

1. *Edit the rpi configuration file.*

```
sudo nano /boot/config.txt
```

2. *Add the following line to the configuration file, replace de value already there with 1, or uncomment the line.*

```
disable_overscan=1
```

3. *Exit saving.*

```
Press Control-x
Press y
Press [enter]
```

4. *After exiting the editor, restart using the command.*

```
sudo reboot
```

4.1.6 Share a folder.

Source: http://elinux.org/R-Pi_NAS

1. *First you need to install all the needed samba libraries.*

```
sudo apt-get install samba samba-common-bin
```

2. *Second step is to comfigure samba.*

```
sudo nano /etc/samba/smb.conf
```

Search for the section marked ##### Authentication #####

Uncomment the following line.

```
#  security = user
```

It should be like: o

```
security = user
```

In the [homes] section, change from

```
  read only = yes
```

to

```
  read only = no
```

Save the file.

3. By default, the username pi is defined. To allow pi to be a samba user.

   ```
   sudo smbpasswd -a pi
   ```

   You will be asked to enter pi's password twice.

4. If you want to use another username for Samba, add a new user.

   ```
   sudo useradd john -m -G users
   sudo passwd john
   ```

5. Restart samba to use the new configuration file.

   ```
   sudo /etc/init.d/samba restart
   ```

You may find some problems to see the share home folder from a Windows machine. In some cases it helps to make a folder public to everyone (note that you will still need to access the folder with a user/password from rpi):

1. Create a directory to store public files.

   ```
   sudo mkdir /home/shares
   sudo mkdir /home/shares/public
   sudo chown -R root:users /home/shares/public
   sudo chmod -R ug=rwx,o=rx /home/shares/public
   ```

2. Edit the samba configuration file.

   ```
   sudo nano /etc/samba/smb.conf
   ```

   At the end of the file, add the following lines.

   ```
   [public]
     comment = Public Storage
     path = /home/shares/public
     valid users = @users
     force group = users
     create mask = 0660
     directory mask = 0771
     read only = no
   ```

You may want to tweak where your drives are mounted so that they mount in the share area. First create a directory then mount the drive there.

```
sudo mkdir /home/shares/public/disk1
sudo mount /dev/sdxx /home/shares/public/disk1
```

4.1.7 Clean some of the daemons starting with the system, so it starts up quicker.

Source: http://theos.in/desktop-linux/removing-unwanted-startup-debian-files-or-services/

In order to speed up the boot process as much as possible, you can deactivate some daemons. An easy way to do it is through the rrconf tool.

```
sudo apt-get intstall rrconf
```

The following list of daemons is candidate to be removed… I have not tested, so feedback is welcome:

- *Lightdm. If no desktop is going to be used, it seems it could be removed. However, I have no idea on the effect when trying to launch a stand-alone rocview windows.*
- *Motf. I would assume, removing this will only affect the message of the day… but I have not tested.*
- *Rsync. As far as I know, this is not in use… unless being used for updates of the rpi.*
- *X11-common. Again, in case of not planning to use rocview.*

4.1.8 Change the name of your rpi-

Source: http://www.simonthepiman.com/how_to_rename_my_raspberry_pi.php

In order to change the name to rocpi (or anything you want) you need to edit the /etc/hostname file.

```
sudo nano /etc/hostname
```

which should look like the entry below

```
raspberry
```

and change it to

```
rocpi
```

Next you need to edit the /etc/hosts file,

```
sudo nano /etc/hosts
```

which should look like the entry below.

```
::1 raspberry localhost6.localdomain6 localhost6
127.0.1.1 raspberry
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

And change the details as below.

```
::1 rpi1 localhost6.localdomain6 localhost6
127.0.1.1 rocpi
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Finally you need to reboot the pi.

```
sudo reboot
```

4.1.9 Backup your system

Sources: http://www.raspberrypi.org/phpBB3/viewtopic.php?f=29&t=12540

http://www.raspberrypi.org/phpBB3/viewtopic.php?f=29&t=10543

Easiest way from windows is to make use of the same tool used to install the rpi image in your sd card: Win32DiskImager. This time, instead of writing, just read.

It may take some time, depending on the size of your card.

4.2 Specific settings for my layout

4.2.1 Two IPs for Loconet Ethernet bridge

Source: http://www.nish.com/2007/12/how-to-set-multiple-ips-on-debian/

To replace the s88 bus for sensors in the layout, I use Loconet, and specifically the GCA101(MGV101) board designed by Peter Giling and Fred Jansen (http://wiki.rocrail.net/doku.php?id=mgv101-en).

The manual of this board explains the benefits of having the Loconet Bridge in a segregated subnet. The instructions to get multiple IPs on debian can be found hereafter (you will need a fix IP a explained in section 4.1.2):

1. *Backup the current version of the interfaces file.*

```
sudo cp /etc/network/interfaces /etc/network/interfaces.sav
```

2. *Edit the file.*

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

3. *Add the following lines.*

```
auto eth0:0
iface eth0:0 inet static

address 192.168.0.10
netmask 255.255.255.0
broadcast 192.168.0.255
```

4. *Restart the network.*

```
sudo /etc/init.d/networking restart
```

4.2.2 Touchscreen

Fortunately the Packard Bell Viseo200T I am using is already recognized by debian distro installed in rpi. Thus, there is no need for any hard work.

However, in order to fine tune the screen. There are a number of things that can be done:

1. *Install a touch keyboard.*
   I am using the following one:
   http://homepage3.nifty.com/tsato/xvkbd/#mainmenu

   WIP: save configuration, launch it from hardware or from output symbol in rocview.
2. *Emulate right click*
   WIP: http://www.touch-base.com/documentation/Virtual%20Keyboards.htm
   http://www.conan.de/touchscreen/evtouch.html
   http://forum.xbmc.org/showthread.php?tid=137852
   http://home.eeti.com.tw/web20/eg/about_EETI.html

4.2.3 Video tuner

WIP

5 Next steps

Some ideas to be check:

- *Build an interface to the GPIO and check if ddx can be used.*
- *Use other GPIOs for other functions:*
    - *Activate the program track (PT).*
    - *Capture through the AD converter the consumption of different boosters.*
    - *Notify with some leds when the system is ready to go (rocrail daemon initialized).*
- *Check if instead of ddx, a new command station acting directly though the GPIO is possible.*

6 Acknowledges

This guide has been created with information from different sources. I have tried to reference all of them within the text.

Specially, it would not have been possible to get rocrail running on a raspberry pi without the information provided by Rob Versluis in the rocrail web site and forum.