A new image was released on 9/18 with a bunch of fixes. My apt-get is messed up so I can't do an upgrade. Time to start over. This adds the RT2800 USB 802.11N adapter with antenna. New instructions at bootom of the file.

Here is the initial boot screen

Debian GNU/Linux wheezy/sid raspberrypi ttyAMA0

raspberrypi login: **pi**
Password: **raspberry**
Linux raspberrypi 3.2.27+ #160 PREEMPT Mon Sep 17 23:18:42 BST 2012 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

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

pi@raspberrypi:~$

Ran the sudo raspi-config but can't easily read the screen as the terminal emulator is not identical on RPi and PC. The serial port is known as ttyAMA0.

Select expand-rootfs to use the entire SD card (after next reboot)
Did not enable overscan (for video screen)
Set Timezone
Selected medium overclook (none=700M, modest=800,medium=900,high=950,turbo=1000)
Made sure SSH is enabled on boot
Set boot behavior no not auto start desktop (Use startx)

Save and reboot

Root has no password so you can't log in. fix by

sudo passwd root

and add the desired password.

Note: Only had the wifi port installed, no Ethernet cable. wlan0 did not come up so will need to make some setup customizations.

Log out and log back in as root

Make some basic adjustments

Add searching the local directory for all users

vi /etc/profile

```
if [ "`id -u`" -eq 0 ]; then
  PATH=".:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
else
  PATH=".:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games"
fi
```

Added .: into each of the 2 PATH strings

```
if [ "$BASH" ] && [ "$BASH" != "/bin/sh" ]; then
      if [ -f /etc/bash.bashrc ]; then
     . /etc/bash.bashrc
     fi
fi
```

Changed $BASH evaluation

```
if [ "`id -u`" -eq 0 ]; then
   PS1='\h # '
else
    PS1='\h $ '
fi
```

Changed prompt depending on UID

vi /etc/bash.bashrc

#PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '

comment out the above line

also changed /home/pi/.bashrc comment out PS1 overwrites

Edit the hostname in /etc/hostname

vi /etc/hostname

RPI-Debian

vi /etc/network/interfaces

 # This file describes the network interfaces available on your system

 # The loopback network interface
auto lo
iface lo inet loopback

 # The primary network interface
 # auto eth0 If using ifplugd, don't use auto here
iface eth0 inet static
metric 10
address 192.168.0.89
gateway 192.168.0.1
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
dns-nameservers 192.168.1.1

 # The wireless interface
auto wlan0
iface wlan0 inet dhcp
metric 200

wpa-proto RSN
 # wan-proto WAP use WAP WAP2 use RSN
wpa-pairwise CCMP
 # pairwise AES use CCMP
wpa-group CCMP
wpa-key-mgmt WPA-PSK
wpa-ssid "JJONES"
wpa-psk "BADBEEF1234"
 # added for RT5370 turn off power-management
wireless-power off

Testing to see what SSID's are available

iwlist wlan0 scan

iwconfig

Load apt-show-versions

apt-get update
apt-get install apt-show-versions

Next, edit /etc/inittab to set default runlevel to 3 (so Mimo /dev/fb1 starts along with /dev/ttyAMA0) and to stop getty from running on tty2 – tty 6 by adding #'s

vi /etc/inittab

 # The default runlevel.
id:3:initdefault:

 # 1:2345:respawn:/sbin/getty --noclear 38400 tty1
 # 2:23:respawn:/sbin/getty 38400 tty2
 # 3:23:respawn:/sbin/getty 38400 tty3
 # 4:23:respawn:/sbin/getty 38400 tty4
 # 5:23:respawn:/sbin/getty 38400 tty5
 # 6:23:respawn:/sbin/getty 38400 tty6

#Spawn a getty on Raspberry Pi serial line
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100

Adding Mimo Display Link USB monitor

Plug in the Mimo to the powered USB Hub and run lsusb to see if it, and the mouse, and the keyboard are all discovered.

lsusb

apt-get update
apt-get install libusb-dev xorg-dev git-core build-essential libncurses5-dev
apt-get install xinit xserver-xorg xserver-xorg-dev xfonts-base x11-xserver-utils

Once this was done, needed to recompile linux to add in support for the DisplayLink drivers. See separate document.

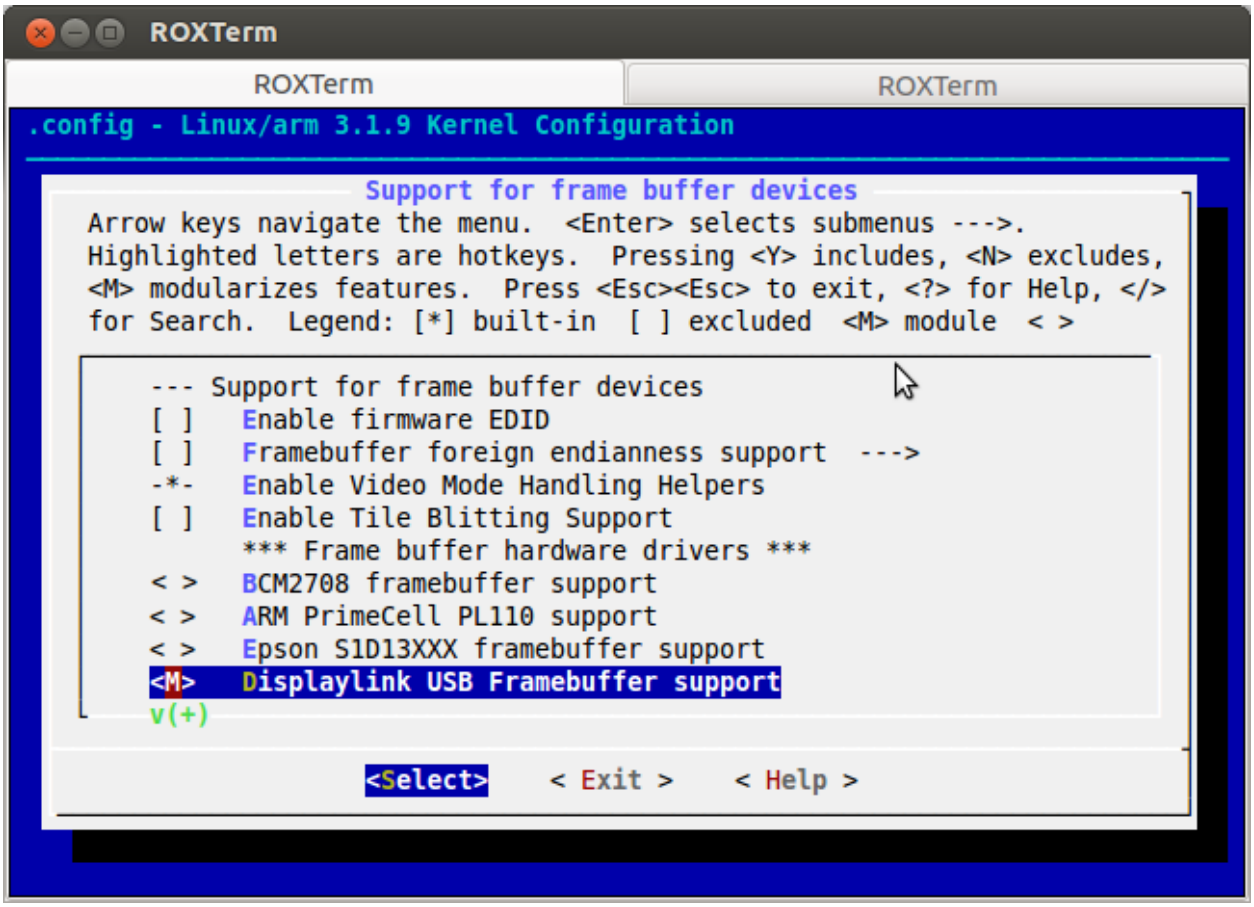
```
cd /opt
git clone --depth 1 git://github.com/raspberrypi/firmware.git
cd firmware/boot
cp arm128_start.elf arm192_start.elf arm224_start.elf bootcode.bin loader.bin
start.elf /boot/


apt-get update
apt-get -y dist-upgrade
apt-get install -y git gcc make tmux
apt-get install -y libncurses5-dev

cd /opt
mkdir raspberrypi
cd raspberrypi
git clone --depth 1 git://github.com/raspberrypi/linux.git
cd linux
zcat /proc/config.gz > .config
```

Make a new kernel

```
cd /opt/raspberrypi/linux
make menuconfig
```

In **menuconfig** we include **udlfb** module, which supports DisplayLink Display.
Follow "Device Drivers -> Graphics Support -> Support for Frame buffer devices" .
Enable "Displaylink USB Framebuffer support" #by selecting M for module or * for kernel built-in.

Optionally if you don't want HDMI port enabled, you # can unselect "BCM2708 framebuffer support".
The device node for displaylink is /dev/fb0 if it is the # only module you enable in framebuffer. In case
you enable BCM2708 also, /dev/fb0 is allocated for that. Displaylink will get /dev/fb1.




```
nice make

##############… 8 to 11 hours later…

nice make modules


cp arch/arm/boot/Image /boot/kernel.img
make ARCH=arm modules_install INSTALL_MOD_PATH=/

shutdown -r now

System rebooted, Mimo screen turned green, first boot took almost a minute
but was successful.
```

Solid Green screen on the Mimo which is good.

Different from first build (maybe older software)

lsusb shows it and lsmod shows udlfb . dmesg reports udlfb: DisplayLink USB device /dev/fb1 . ps –el **does** show udlfb.

Displaylink is assigned /dev/fb1. You need to know this to configure /etc/X11/xorg.conf in following steps.

Update Driver

mkdir /opt/Desktop
cd /opt/Desktop
wget http://projects.unbit.it/downloads/udlfb-0.2.3_and_xf86-video-displaylink-0.3.tar.gz tar xzvf
tar xvzf udlfb-0.2.3_and_xf86-video-displaylink-0.3.tar.gz

cd udlfb

patch udlfb.c by including vmalloc.h near the top of the file and changing the x and y res values on two lines at about line 781. Use vi search / 1280 to find the screen resolution values.

vi /opt/Desktop/udlfb/udlfb.c

```
#include <linux/vmalloc.h>
```

- info->var.xres = 1280; ← change from 1280 to 800

- info->var.yres = 1024; ← change from 1024 to 480

Exit vi

make CROSS_COMPILE=
make install

depmod -a
cd ../xf86-video-displaylink/

vi src/displaylink.c

```
# comment out the following four lines...
//#include "xf86Resources.h"
```

```
//#include "xf86RAC.h"
//      pScrn->racMemFlags = RAC_FB | RAC_COLORMAP | RAC_CURSOR |
RAC_VIEWPORT;
//      pScrn->racIoFlags = RAC_FB | RAC_COLORMAP | RAC_CURSOR |
RAC_VIEWPORT;
//      xf86CrtcScreenInit (pScreen);
```

./configure CROSS_COMPILE=
make CROSS_COMPILE=
make install

Many messages. Not 100% sure this was successful but the driver was created.

```
cp   /usr/local/lib/xorg/modules/drivers/displaylink_drv.so
/usr/lib/xorg/modules/drivers/
```

Create a /etc/ xorg.conf looks like:

```
Section "Device"
      Identifier     "dl0"
      driver        "displaylink"
      Option  "fbdev" "/dev/fb1"
EndSection

Section "Monitor"
      Identifier "monitor0"
      DisplaySize  190 115
EndSection

Section "Screen"
      Identifier "screen0"
      Device "dl0"
      Monitor "monitor0"
EndSection

Section "ServerLayout"
   Identifier "ServerLayout0"
   Option "BlankTime"  "10"
   Option "StandbyTime" "10"
   Option "SuspendTime" "10"
   Option "OffTime" "0"
EndSection
```

Now a login screen appears on the Mimo at power up. There is a choice of Default Xsession, LXDE or OpenBox

If the Mimo is not used for a while, the screen flashes between all red, all green, all blue, all gray, and a checkerboard pattern.

I want it to blank and turn black. Edit /etc/X11/Xsession I think only xset s blank is needed. Need to add –d :0 to select Mimo monitor. Learned by typing echo $DISPLAY from Mimo terminal.

vi /etc/X11/Xsession

 # added by JJ
 # DPMS stuff
turn on monitor
 # xset -d :0 dpms force on
disable sleep modes etc.
#xset -dpms
disable screensaver
 # xset -d :0 s on
set screensaver to blank

xset -d :0 s blank

General commands Note: the display assignment :0 does not seem to be applied until someone logs on on the Mimo.

```
sleep 1; xset dpms force off
xset -d :0 dpms force standby
```

to see the current settings

xset -d :0 q

Make a Clone

Put a 8G micro SD card in a USB-micro-SD adapter and inserted into the USB speaker USB port. The drive showed up as /dev/sda. The copy took a little over an hour.

```
dd if=/dev/mmcblk0 of=/dev/sda  bs=1024 conv=noerror
```

dd if=/dev/mmcblk0 of=/dev/sda bs=1024 conv=noerror
7761920+0 records in
7761920+0 records out
7948206080 bytes (7.9 GB) copied, 3756.31 s, 2.1 MB/s

Make a Disk Image

```
USB memory stick learned as /dev/sda

mount -t ext4 -o rw,users /dev/sda1 /mnt


dd if=/dev/mmcblk0 of=/mnt/image.gz


To compress

dd if=/dev/mmcblk0 | gzip -c > /mnt/image.gz
```

To restore the backup you reverse the commands:

```
dd if=/mnt/image.gz of=/dev/mmcblk0
```

or when compressed:

```
gzip -dc /mnt/image.gz | of=/dev/mmcblk0
```

Adding RALink 2800 USB 802.11N wifi adapter

apt-get update

apt-get install wireless-tools wpasupplicant firmware-ralink

Software configuration

Now we have all the needed software but still the Wi-Fi USB will not appear as device `wlan0` we need to make it working: it is time of some udev voodoo and finally things should work.

First we need to edit the file below by adding the line below

vi `/etc/udev/rules.d/network_drivers.rules`

```
ACTION=="add", SUBSYSTEM=="usb", ATTR{idVendor}=="1737",
ATTR{idProduct}=="0078", RUN+="/sbin/modprobe -qba rt2800usb"
```

Then we need to edit the file below by adding the line below

vi `/etc/modprobe.d/network_drivers.conf`

```
install rt2800usb /sbin/modprobe --ignore-install rt2800usb $CMDLINE_OPTS;
/bin/echo "1737 0078" > /sys/bus/usb/drivers/rt2800usb/new_id

Remove the original USB wifi adapter, reboot and everything works
```