Custom Equipment Monitoring

with OpenWRT and Carambola

Sysadmin Miniconf

Linux.conf.au Perth - 6 January 2014

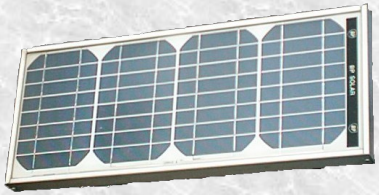
Andrew McDonnell - Software Engineer & Tech Problem Solver
retro@andrewmcdonnell.net
Twitter: @pastcompute
http://blog.oldcomputerjunk.net
https://launchpad.net/~andymc73
https://github.com/andymc73

Overview

- Equipment Monitoring with a Budget
- Introduction to Carambola
- Introduction to OpenWRT
- Monitoring with OpenWRT and Carambola

Use Case

- Everything has a computer in it these days



- And a connection:

bluetooth rs485 i2c CANbus
1-wire wimax ethernet wifi rs232
SPI USB i2s ZigBee GPIO

http://en.wikipedia.org/wiki/File:SolarpanelBp.JPG

http://en.wikipedia.org/wiki/File:2008-07-11_Air_conditioners_at_UNC-CH.jpg

http://en.wikipedia.org/wiki/File:Davis_VantagePro.jpg

Use Case Requirements

- So if you have a <insert widget here>?
- And a small physical space requirement?
- How do you have it talk to <insert toolkit here>?

http://openclipart.org/detail/182810/old-computer-by-jhnri4-182810

http://openclipart.org/detail/188441/sid-chip-by-arvin61r58-188441

Potential Solutions

- Industrial COTS
- Embedded microcontrollers
  - arduino
- Embedded Linux
  - Raspberry Pi
  - Beaglebone Black
  - Carambola2

http://www.openelectrical.org/wiki/images/4/43/Programmable-logic-controller-plc-22971.jpg

Embedded Linux Computers.

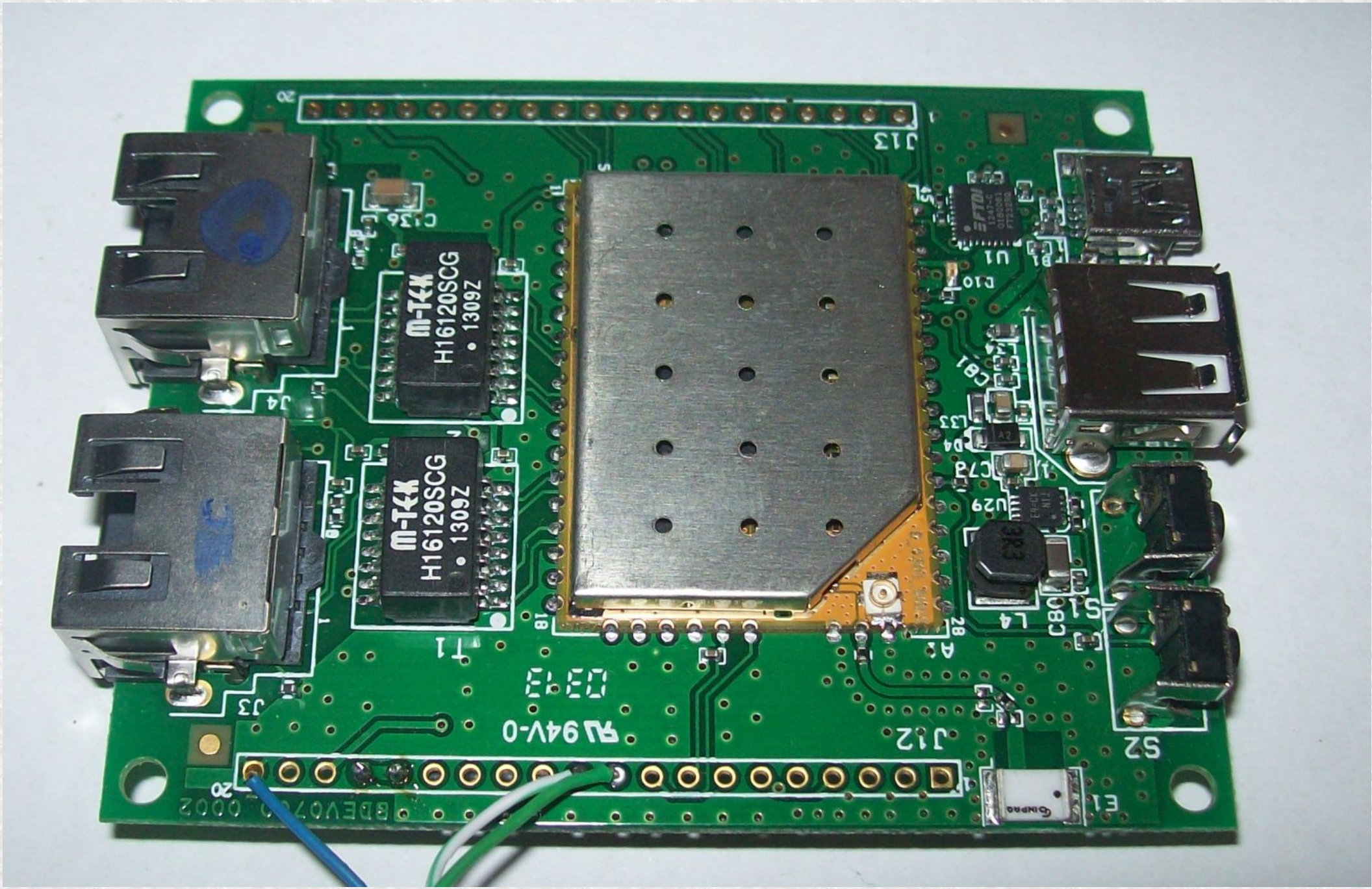
- Raspberry Pi
  - Aimed at education market
  - Small footprint, very cheap
  - Power, Ethernet performance?
  - Runs various distros

- Beaglebone black
  - High performance
  - Good connectivity

Image: Andy Kirkpatrick

http://en.wikipedia.org/wiki/File:RaspberryPi.jpg

Image: andrewmcdonnell.net

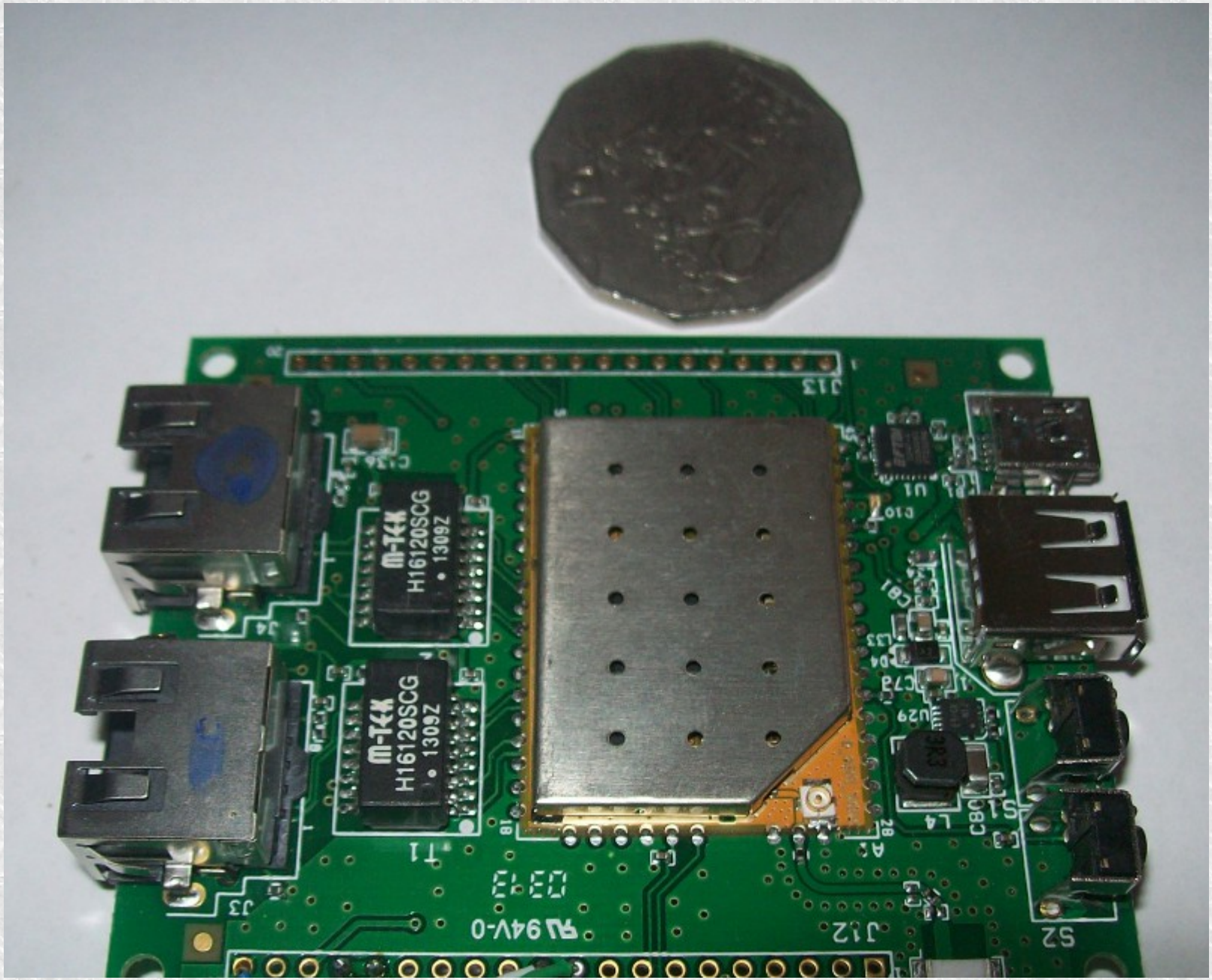


Embedded Linux : Carambola2

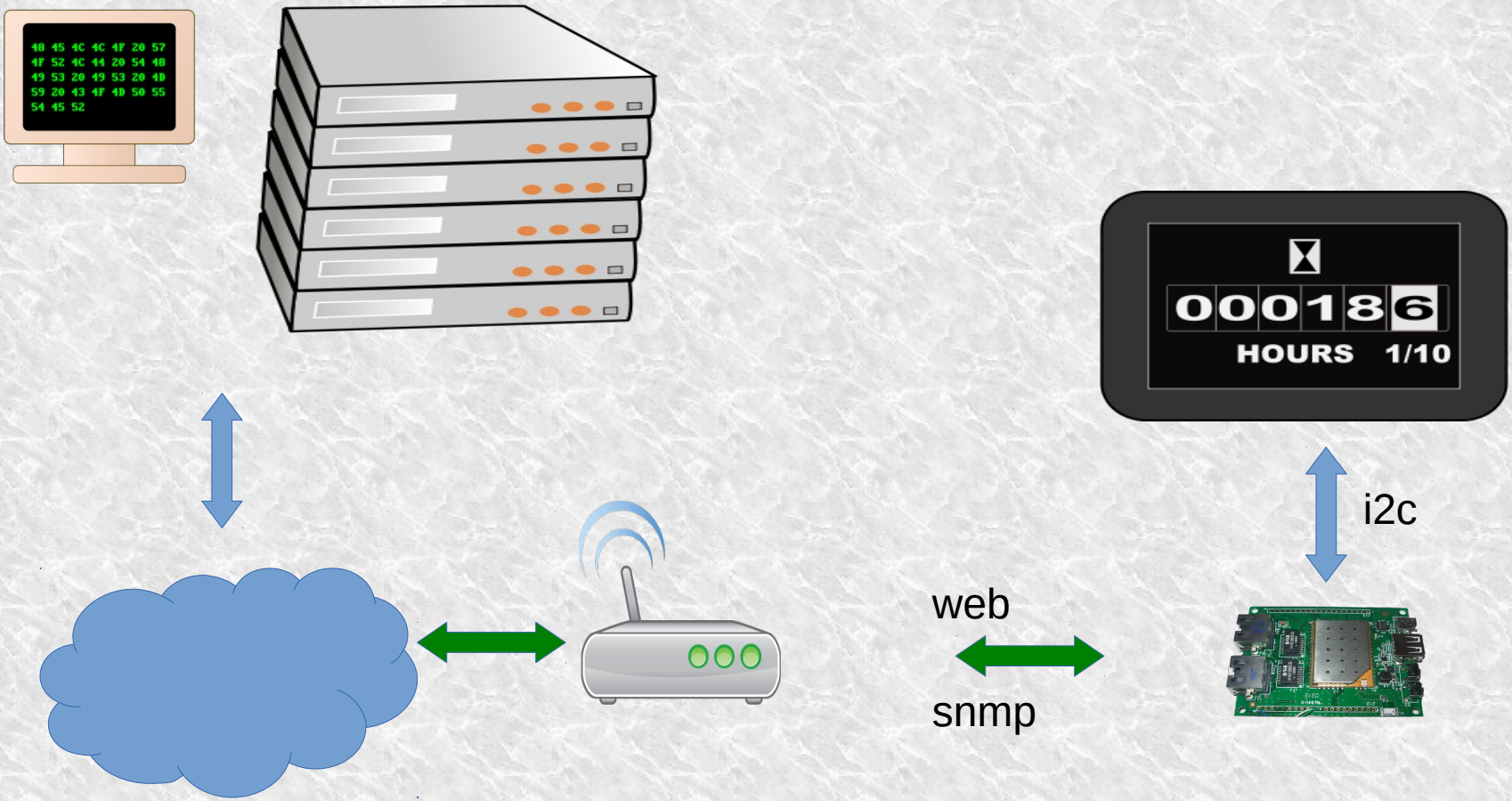
- Carambola2
  - Very cheap < $100 with cabling & casing, small footprint
  - Build from even smaller module and connector board
  - Integrated wifi, plus dual ethernet with switch
  - USB host and USB serial slave
  - Low power usage
  - GPIO: SPI, I2C, serial & RS485, i2s, SPDIF
  - AR9331 SOC – MIPS 400MHz, 64MB RAM
  - Runs OpenWRT Linux
  - https://github.com/8devices/

http://8devices.com/wiki_carambola/doku.php/carambola

Image: andrewmcdonnell.net



Simple Carambola solution example



http://openclipart.org/detail/1723/stacked_servers-by-ajith

http://openclipart.org/detail/18063/hobbs-hour-meter-by-startright

OpenWRT Linux & Carambola

- OpenWRT is an embedded Linux distribution
- OpenWRT has GPLv2 license
- Carambola: designed with ease of use of OpenWRT in mind
- 8devices maintained github fork of OpenWRT
- The Carambola2 ships with OpenWRT installed
- Firmware images and package library at

  http://pkg.8devices.com/?dir=v2.4/carambola2/

What is OpenWRT?

- In their own words:

"Instead of trying to create a single, static firmware, OpenWrt provides a fully writable filesystem with package management. This frees you from the application selection and configuration provided by the vendor and allows you to customize the device through the use of packages to suit any application. For developer, OpenWrt is the framework to build an application without having to build a complete firmware around it; for users this means the ability for full customization, to use the device in ways never envisioned."

http://en.wikipedia.org/wiki/File:Linksys_WRT54G_V1.jpg

OpenWRT features

- Large package library
- OPKG – package management system
- UCI – common configuration interface
- LUCI – polished web interface
- Community project
- Open Source (GPL v2)

OpenWRT Tour

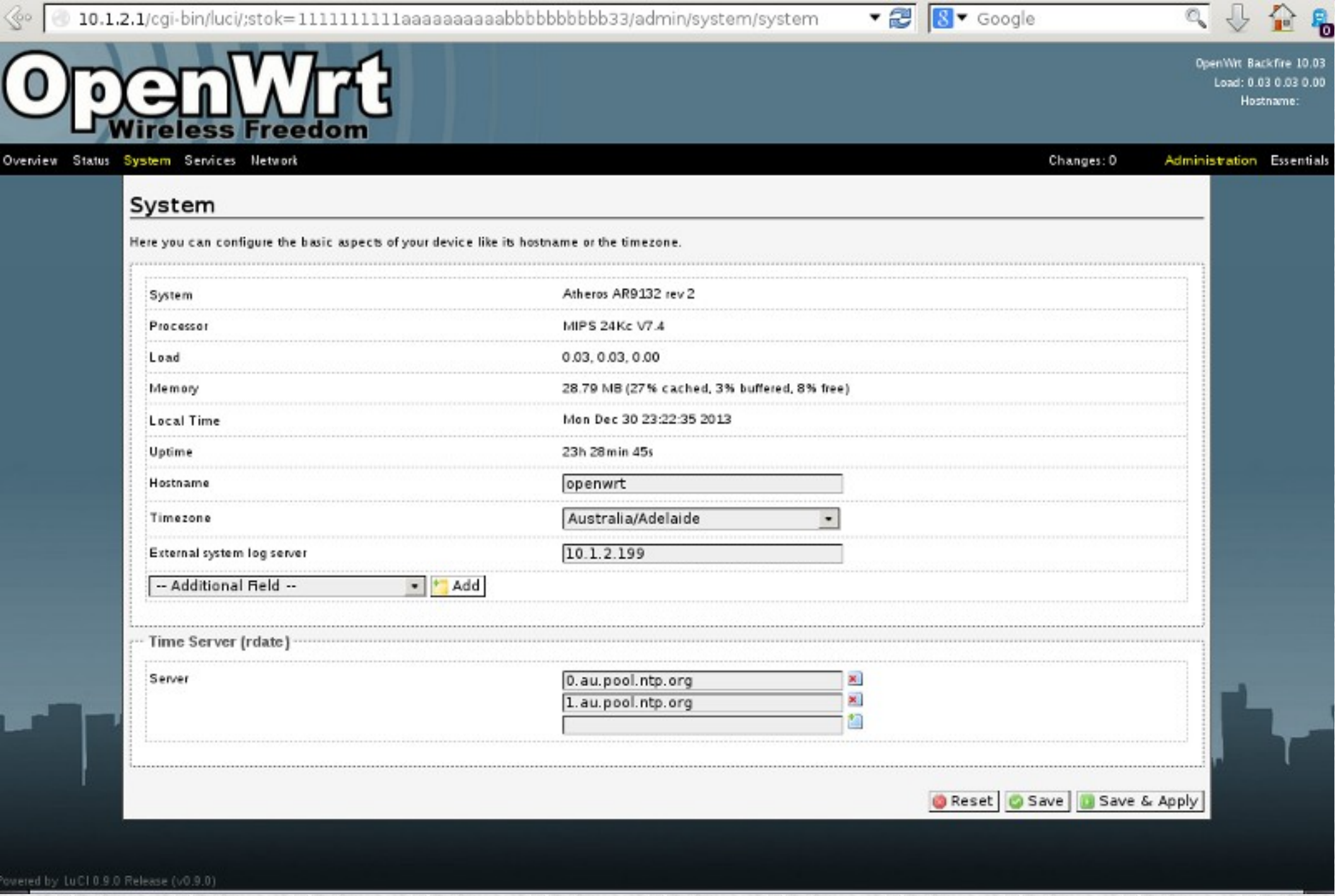
```
$ ssh root@openwrt

BusyBox v1.15.3 (2010-04-06 03:14:11 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.


  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 Backfire (10.03, r20728) -------------------------
  * 1/3 shot Kahlua    In a shot glass, layer Kahlua
  * 1/3 shot Bailey's  on the bottom, then Bailey's,
  * 1/3 shot Vodka     then Vodka.
 ---------------------------------------------------
root@openwrt:~#
```

It really is Linux...

```
# ps
  PID USER       VSZ STAT COMMAND
    1 root      1360 S    init
    2 root         0 SW   [kthreadd]
    3 root         0 SW   [ksoftirqd/0]
    4 root         0 SW   [events/0]
    5 root         0 SW   [khelper]
    8 root         0 SW   [async/mgr]
   41 root         0 SW   [sync_supers]
   43 root         0 SW   [bdi-default]
   45 root         0 SW   [kblockd/0]
   71 root         0 SW   [aio/0]
   84 root         0 SW   [mtdblockd]
   90 root         0 SW   [ar71xx-spi]
  225 root         0 SW   [ipolldevd]
  279 root         0 SWN  [jffs2_gcd_mtd3]
  294 root      1360 S    init
  313 root      1364 S    syslogd -C16 -L -R 192.168.1.27:514
  315 root      1348 S    klogd
  809 root         0 SW   [cfg80211]
 1257 root         0 SW   [phy0]
 4538 root      1136 S    /usr/sbin/dropbear -p 192.168.1.11:22 -P /var/run/dro
 4558 root      2672 S    /usr/sbin/sshd
 4573 root      1464 S    /usr/sbin/uhttpd -f -h /www -r tyto -x /cgi-bin -t 60
 4780 nobody    1220 S    gkrellmd
 4808 root         0 SW<  [loop0]
 4815 root      1356 S    watchdog -t 5 /dev/watchdog
 9253 root      1436 S    hostapd -P /var/run/wifi-phy0.pid -B /var/run/hostapd
13186 root      1332 S    /usr/sbin/ntpd -g -p /var/run/ntpd.pid
13187 root      1440 S    /bin/sh /usr/lib/ddns/dynamic_dns_updater.sh myddns 0
13278 root      1368 S    udhcpc -t 0 -i eth0.2 -b -p /var/run/eth0.2.pid -O ro
13993 nobody     920 S    /usr/sbin/dnsmasq -K -D -f -N -q -y -Z -b -E -z -s ea
14004 root      4440 S    arpwatch -f /sd/var/arpwatch/arp.dat -i br-lan
```


10.1.2.1/cgi-bin/luci/;stok=1111111111aaaaaaaaaabbbbbbbbbb33/admin/system/system
Google
OpenWrt
Wireless Freedom
OpenWrt Backfire 10.03
Load: 0.03 0.03 0.00
Hostname:
Overview Status System Services Network
Changes: 0
Administration Essentials
System
Here you can configure the basic aspects of your device like its hostname or the timezone.
System
Atheros AR9132 rev 2
Processor
MIPS 24Kc V7.4
Load
0.03, 0.03, 0.00
Memory
28.79 MB (27% cached, 3% buffered, 8% free)
Local Time
Mon Dec 30 23:22:35 2013
Uptime
23h 28min 45s
Hostname
openwrt
Timezone
Australia/Adelaide
External system log server
10.1.2.199
-- Additional Field --
Add
Time Server (rdate)
Server
0.au.pool.ntp.org
1.au.pool.ntp.org
Reset
Save
Save & Apply
Powered by LuCI 0.9.0 Release (v0.9.0)

OpenWRT Package Management

```
# opkg list_installed |head
arptables - 0.0.3-3-1
arpwatch - 2.1a15-2
base-files - 42-r20728
bind-dig - 9.7.2-P3-1
bind-host - 9.7.2-P3-1
bind-libs - 9.6.1-P2-2
block-hotplug - 0.1.0-1
block-mount - 0.1.0-1
busybox - 1.15.3-2
Bzip2 – 1.0.5-1
# opkg search /bin/vi
Busybox – 1.15.3-2
# opkg install -d ext tcpdump
Upgrading tcpdump on ext from 4.0.0-2 to 4.1.1-2...
Downloading http://downloads.openwrt.org/.../ar71xx/packages/tcpdump_4.1.1-2_ar71xx.ipk.
Configuring tcpdump
# opkg files tcpdump
Package tcpdump (4.1.1-2) is installed on ext and has the following files:
/ext/usr/sbin/tcpdump
```

- Can select alternative targets
  - e.g. USB stick for storage, RAM (/tmp) for testing
  - Minimize flash wear

OpenWRT Configuration - UCI

```
# uci show wireless.radio0
wireless.radio0=wifi-device
wireless.radio0.type=mac80211
wireless.radio0.macaddr=66:55:44:33:22:11
wireless.radio0.ht_capab=SHORT-GI-40 DSSS_CCK-40
wireless.radio0.disabled=0
wireless.radio0.channel=3
wireless.radio0.country=AU
wireless.radio0.hwmode=11g
wireless.radio0.distance=25
# uci set wireless.radio0.channel=4
# uci commit wireless
# /etc/init.d/network restart
```

- Or use the LUCI web interface
- See http://wiki.openwrt.org/doc/uci
  - Heed the warning about extraneous lines in config files

Sending email

- Easiest way for monitoring
- Use for prototyping before SMS or SNMP
  - Carambola has msmtp which runs a daemon locally
  - I prefer ssmpt, but you need to build OpenWRT from source

```
opkg install msmtp-nossl
# edit /etc/msmtprc here
echo 'Message body' | sendmail me@example.com


# Or:


opkg install ssmtp mailtuils

# edit /etc/ssmtp/ssmtp.conf here

mail -s "subject" monitor@example.com < somefile
```

I2C Monitoring

- OpenWRT includes basic functionality
- You need to map i2c SCL, SDA onto GPIO pins
  - Carambola needs pull-ups
- This lets you do byte/word reads and writes

```
opkg install i2c-tools kmod-i2c-gpio kmod-i2c-gpio-custom
cat > /etc/modules.d/59-i2c-gpio <<EOF
i2c-gpio
i2c-gpio-custom bus0=0,12,13 # SCL=12 SDA=13
EOF
reboot


i2cget 0 0x03 0x45 b
```

- http://wiki.openwrt.org/doc/hardware/port.i2c

GPIO Monitoring

- Modern Linux exposes pins to userspace easily
- Simply set the direction then manipulate in shell

```
echo 12 > /sys/class/gpio/export
echo 13 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio12/direction
echo in > /sys/class/gpio/gpio13/direction
echo 1 > /sys/class/gpio/gpio12/value
while true; do
  sleep 60
  X=$(cat /sys/class/gpio/gpio12/value)
  if [ $X == 1] ; then
    echo "My widget is still high!" | ssmtp helpme@example.com
  fi
done
```

- You can easily wire a LED this way as well

Monitoring Automation

- You can use cron or at, just like on a server
- You can create your own init script
- e.g. the script in the previous page:

```
vi /etc/init.d/gpio-poller
ln -s /etc/init.d/gpio-poller /etc/rc.d/89gpio-poller
```

SNMP & SMS

- You can use snmpd or mini_snmpd package
  - Send a custom trap based on a gpio pin event
- SMS – use the usb serial modules and picocom
  - Or an integrated package such as gnokii
    - Need to build OpenWRT from source

OpenWRT filesystem

- Overlay filesystem:
  - Provides illusion of write access to underlying flash
  - Allows later package install
  - Upgrades / removal by 'white out'
- You could use USB stick for much larger storage
  - I would advise this for experimenting and for logs
  - Or for swap...
    - (not a good idea to swap onto onboard flash)

OpenWRT filesystem

- The Carambola flash is partitioned:
  - Boot loader – not touched by Linux / OpenWRT
  - The boot / root (ROM) filesystem firmware image
  - The flash overlay, using a union filesystem
    - Using jffs2 or whatever is appropriate
- OpenWRT also uses tmpfs quite effectively
    - Not just /tmp;
    - various items in /etc such as resolv.conf dynamically link to /tmp

Monitoring – OOTB limitations

- Depending on your hardware:
  - You may need to custom build the software
  - For example, the shipped i2c tools are limited
  - We ended up writing our own C program using IOCTLs
- You may want to limit the installed software
  - To save space
  - To implement configured custom solution
  - To enforce policy
  - For security – limit attack surface
    - e.g. exclude ability to access wireless by leaving out modules

Nuts and Bolts

- Build custom image to make most of OpenWRT
- Approx wall time: 3 hours on a 8MBit link
  - Downloads, compilations
- Your time – one hour
  - if you can do something else whilst waiting for the build

The basic procedure

```
sudo apt-get install ckermit gawk subversion zlib1g-dev git
# you also need:
# gcc, binutils, patch, bzip2, flex, make, gettext, pkg-config, unzip, libz-dev and libc headers.

git clone https://github.com/8devices/carambola2
# make a coffee

cd carambola2
scripts/feeds update
scripts/feeds install wget msmtp ser2net picocom mini-snmpd i2c-tools gpioctl-sysfs lcdproc luci
make menuconfig
make -j7

# go and have lunch

# connect USB cable – provides power and serial
# connect Ethernet cable
# download image
# cycle power to Carambola2
```

A little more detail...

- Git clone – get OpenWRT core
- scripts/feeds – select packages to download
- make menuconfig – define target & packages
- make
  - Download & build tools (bison, quilt, …)
  - Download & build cross-toolchain (gcc, binutils, …)
  - Download & build kernel & packages
  - Assemble firmware image

OpenWRT build system

- The build system bootstraps itself from source
  - A bit like Gentoo, “kind-of”
  - To avoid re-downloading, make a cache location
    - I keep this in its own local GIT repository (I know its binary...)
  - Simplifies building if no Internet or you don't want to wait
  - Or if you need to build several devices exactly the same

```
cd ..
mkdir DL
cd carambola2
make oldconfig
Echo 'CONFIG_DEVEL=y' >> .config
echo 'CONFIG_DOWNLOAD_FOLDER="`cd ..; pwd`"' >> .config
make oldconfig
```

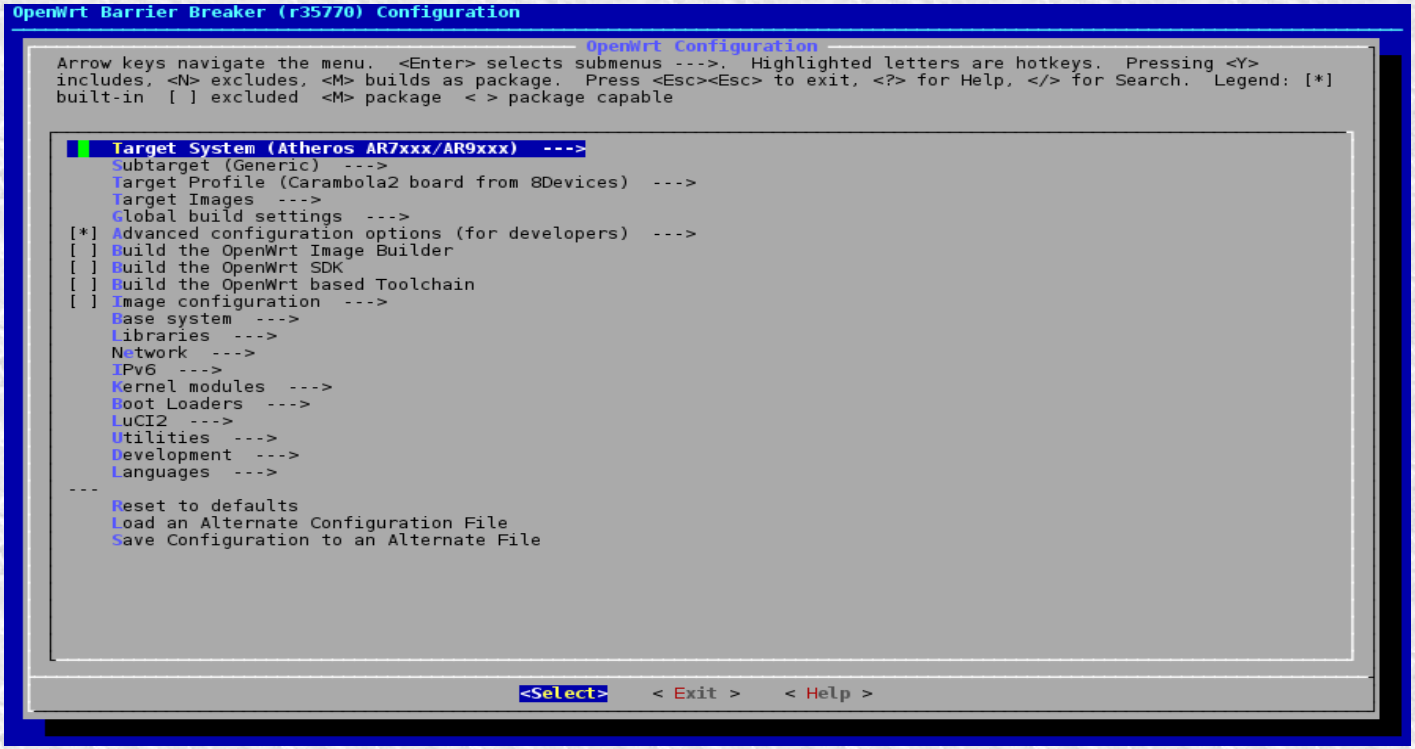
What if I need to build in Windows?

- Often you have some corporate constraints...
- My work laptop is a Windows7 8-core
  - So I installed VirtualBox with the extension pack
  - I used Xubuntu 12.10 for all OpenWRT development
  - USB pass-through works quite well
  - Configuring the Windows firewall to transparently network how I needed took _just_ a bit more work...

OpenWRT build configuration

- Powerful configuration based on Linux kernel text console menu system

```
make menuconfig
```



OpenWRT packages

- Using the menu system, choose what to include
- Packages can be modules or fully installed
  - modules – don't go into initial firmware flash image
  - allows later reconfiguration and uses less space when upgrading

OpenWRT filesystem

- The Carambola flash is partitioned:
  - Boot loader – not touched by Linux / OpenWRT
  - The boot / root (ROM) filesystem firmware image
  - The flash overlay, using a union filesystem
    - Using jffs2 or whatever is appropriate
- OpenWRT also uses tmpfs quite effectively
    - Not just /tmp;
    - various items in /etc such as resolv.conf dynamically link to /tmp

OpenWRT filesystem

```
#  df -k
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/root                 1280      1280         0 100% /rom
tmpfs                    14744      1676     13068  11% /tmp
tmpfs                      512         0       512   0% /dev
/dev/mtdblock3            5440      4852       588  89% /overlay
mini_fo:/overlay          1280      1280         0 100% /
```

OpenWRT Kernel on Carambola

- The USB client is a serial port
- Connect and hit reset button to see boot log:

```
## Booting image at 9f050000 ...
   Image Name:   MIPS OpenWrt Linux-3.7.9
   Created:      2013-09-26  23:41:51 UTC
   Image Type:   MIPS Linux Kernel Image (lzma compressed)
   Data Size:    940982 Bytes = 918.9 kB
   Load Address: 80060000
   Entry Point:  80060000
   Verifying Checksum at 0x9f050040 ...OK
   Uncompressing Kernel Image ... OK
No initrd
## Transferring control to Linux (at address 80060000) ...
## Giving linux memsize in bytes, 67108864

Starting kernel ...

[    0.000000] Linux version 3.7.9 (andrew@carambola-vm) (gcc version 4.7.3 20121205 ...
[    0.000000] bootconsole [early0] enabled
```

Upgrading / Flashing Firmware

- Simplest method – use web interface
- If no web interface installed:
  - Ideally you have at least one of ssh & wget
  - Be connected on the serial port as well
  - Backup config files first if needed

```
scp somewhere/openwrt-ar71xx-generic-carambola2-squashfs-sysupgrade.bin /tmp

sysupgrade -v -n /tmp/openwrt-ar71xx-generic-carambola2-squashfs-sysupgrade.bin
```

Recovering a Bad Flash

- Flashing is pretty robust
  - But when rolling your own mistakes can happen
  - And dont unplug power part way through!
- The Carambola has a Uboot boot loader
  - I experienced problems getting tftp to load, YMMV
  - If all else fails you can unbrick using kermit...
    - Use the 'loadb' command from Uboot
    - CTRL+\ , C to drop to kermit, then send filename.bin
  - This runs about 2kpbs so go have a coffee...
  - After the image is loaded, you still need to flash it

Kermit flash recovery

- To minimise chance of transmission errors
  - Use this .kermrc file

```
$ cat ~/.kermrc
set line /dev/ttyUSB0
set speed 115200
set carrier-watch off
set handshake none
set flow-control none
set flow none
robust
set file type bin
set file name lit
set rec pack 1000
set send pack 1024
set window 5
```

X86 build

```
Apt-get install gnome-screenshot nmap telnet ckermit gawk subversion zlib1g-dev
git clone git://git.openwrt.org/openwrt.git
scripts/feeds update
scripts/feeds install wget luci-app-ddns openssh-server openssh-client luci-app-firewall
make defconfig
echo 'CONFIG_DEVEL=y >> .config
echo 'CONFIG_DOWNLOAD_FOLDER="/scratch/openwrt/DL"' >> .config
# x86, KVM Guest
make
kvm -net nic,model=virtio -net user \
   -drive file=openwrt-x86-kvm_guest-combined-squashfs.img,if=virtio \
   -smp sockets=1,cores=1 \
   -redir tcp:5555::80
```

Inside the VM

```
uci set network.lan.proto=dhcp && uci commit && /etc/init.d/network restart
```

Conclusion

- OpenWRT provides a full Linux O/S tailored for embedded, network solutions
- The Carambola2 is a cheap, small, low-power embedded system
- Supports numerous connectivity, I/O options
- Retain power of Linux

Credits & License

- Content by Andrew McDonnell 2014

andrew@andrewmcdonnell.net
Twitter: @pastcompute
http://oldcomputerjunk.net
https://github.com.andymc73
https://launchpad.net/~andymc73
http://au.linkedin.com/in/amcdonnell

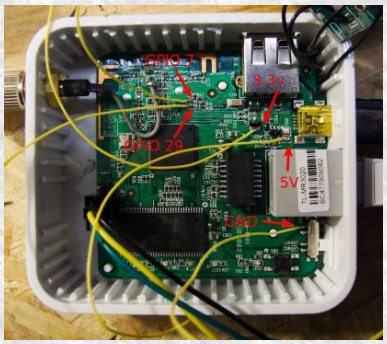
Images / cartoons attributed as used

Appendix

- To make evince open PDF instead of GIMP:
  - Edit ~/.local/share/applications/mimeapps.list, use application/pdf=evince.desktop

DIY Linux Embedded Computers

- You could repurpose something else...
- Many LCA attendees like such things – for fun...



- But we need that stuff such as COTS, and documentation, and engineering, for our day job, now, don't we?

http://en.wikipedia.org/wiki/File:Linksys_WRT54G_V1.jpg

http://wiki.openwrt.org/toh/tp-link/tl-mr3020