The Hot Water Balloon

Wookey
wookey@wookware.org

Debian / Embedded Debian / Balloonboard.org / iEndian / Toby Churchill Ltd

Embedded Linux Conference - Europe 2008
Ede
What is this talk about?

- Solar Thermal crash course
- Controllers
- 1-wire hackery how-to
- Results Analysis
Evacuated Tube

- Vacuum tube with heat-pipe

![Diagram of evacuated tube with heat-pipe](image)
Solar Thermal System
Wook’s System

plate heat exchanger (PHE) layout

Solar Panel
Expansion Vessel
Solar Pump
DHW
Water Cistern
F&E Tank
Cylinder
Radiators
Cold Main
Gas Boiler

The Hot Water Balloon
Panel Fitting

Wookey

The Hot Water Balloon
Thermosiphon

- Existing Tank
- Thermosiphon
- Anti-siphon dip
Commercial Controllers

- Standard: 164€
- with Vbus: 230€
- Datalogger: 260€
- Remote Display: 68€
- 10 sensors, 7 outputs: 388€

- 3 sensors, 2 outputs: 162€
- Ethernet connection: 454€
Balloonboard

- PXA270, 1GB flash
- 384MB RAM, FPGA
- USB host, VGA display
- Open Hardware
- Distro choice, package management

Wookey

The Hot Water Balloon
CUED IO Expansion board

- I2C, 8 digital IO
- 6x 10bit ADC
- 15V/5V crowbarred supply
- USB host/slave
- PWM motor driver
Hardware 1-wire

- 8 multiplexed 1-wire buses
- Data: 14Kbit/s max
- Scan: 13 devices/second
- Measurement: 94-750ms
- I2C 400Kbit/s
Hardware - Switching

- PCF8574a
- 8 channel IO
- I2C interface
- ‘On’ pulls pin low - PNP
- Relay needed for 12V
- puts out 45mA max
Hardware - Building

- 4 wires - I2CData, I2Cclk, GND, +5V
1-wire Sensors

Wookey

The Hot Water Balloon
Fitting Sensors
Wook’s tidy airing cupboard

- Balloon, extra IO board, USB ethernet adaptor
Software - I2C modules and addressing

- modprobe i2c-pxa
- modprobe i2c-dev
- /dev/i2c-0 /dev/i2c-1
- bus 0 is general, bus 1 is power
- I2C addresses fixed.
- 7bit - bottom bit is r/w
- on DS2482 Address is 0011nnn
- 0x18-0x1F
- pull all three pins low to get 0x18
Software - I2C access

- **i2cdump 0 0x18**
  ```
  0 1 2 3 4 5 6 7 8 9 a b c d e f 0123456789abcdef
  00: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX b5 XXXXXXXXXXXXXXX?
  10: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX b5 XX XXXXXXXXXXXXXX?X
  20: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX b5 XX XX XXXXXXXXXXXXX?XX
  30: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX b5 XX XX XX XXXXXXXXXXXX?XXX
  \{\ldots\}
  e0: XX 00 XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XXXXXXXXXXXXXXXX
  f0: 18 XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XXXXXXXXXXXXXXXX
  ```

- **pcf8574a on 0x38**
- **i2cdump from lm-sensors or i2c-tools**
- **On**: echo “0” > /sys/bus/i2c/devices/0-0038/write
- **Off**: echo “255” > /sys/bus/i2c/devices/0-0038/write
FUSE filesystem
configure in /etc/default/owfs
  - SERVER_OPTS="-d /dev/i2c-0"
  - CLIENT_OPTS="-s localhost:4304"
run with /etc/init.d/owfs start
New on lenny (on etch needs older python, swig, libfuse)
apt-get install owfs libowfs26 libfuse
Discover: owdir -s 430 /28.D23974010000

Read: owread -s 4304 /28.D23974010000/temperature
34.25
Create Round Robin Database

Filled in with rrdupdate

```
sudo rrdtool create /var/log/solar.rrd -s 10
  DS:panel:GAUGE:40:−20:125
  DS:tanktop:GAUGE:40:−20:125
  DS:tankbott:GAUGE:40:−20:125
  DS:pump:GAUGE:40:0:1  RRA:AVERAGE:0.5:1:20160
  RRA:LAST:0.5:1:20160  RRA:MAX:0.5:60:1120
```

Should be 9 days, actually 2.5
sudo rrdtool graph --end 18:00d --start 06:00d /tmp/solardetail.png --M --i\t--t "Wook's solar system" --v "Temp (C)" --h 200 --w 800\nDEF:pump=/var/log/solar.rrd:pump:LAST TICK:pump#000000:0.05:'Solar Pump on'\nDEF:panel=/var/log/solar.rrd:panel:LAST LINE2:panel#FF0000:'Panel Temp'\nDEF:tanktop=/var/log/solar.rrd:tanktop:LAST LINE2:tanktop#00FF00:'Top of Tank'\nDEF:tankbott=/var/log/solar.rrd:tankbott:LAST LINE2:tankbott#00ffff:'Bottom of Tank'
Munin

- local and remote munin setups
- hides rrdtool details
- Easy plugin scripts
- Munin-graph uses 85% of cpu
- Munin broken on arm
Munin plugins

- Any language
- Called with config: prints variables
- Called normally: prints values

```
sudo munin-run panels

panelin.value 10.875
panell.value 9.5625
panel2.value 10.4
pump.value 0
```
Munin plugin for panel

sudo munin-run panels config

graph_title Solar Panel Temperatures
graph_order panelin panel1 panel2 pump
graph_vlabel Degrees Celcius
graph_info This graph shows solar panel temperature
graph_period minute
panelin.label panels-in
panelin.draw LINE1
panelin.max 126
panelin.min -2
panelin.type GAUGE
panelin.info Temp flow into solar panels
What did I discover

- It works!
- Thermosiphon reacts very quickly (<30 seconds)
- Control is interesting question
- 15°C rise across panels
- 68°C in October
- Provided hot water about 23 days of 31
if panel >tank_bot + 20 or panel >tank_top + 4 : pump on
if panel <tank_bot + 10 : pump off
if tank_top >80 : pump off
Horrible shell script
Decimal temp readings - bc
DIYzoning... PID control
- Tank stays stratified during loading
- Tank hotter than panel!? 
- Tank stays stratified during loading
- Tank hotter than panel!?
- Hysteresis between +10°C and +20°C
Reliability

- Uptime: 87 days load average: 1.02
- owserver crashed once in 5 months
- Some 1-wire glitches
- Disk full survived
Temp accuracy

- Nominal 0.5°C accuracy (0-55°C)
- 2°C accuracy (below 0°C, above 55°C)
- 3.8°C difference
- Stuck at 85°C, and 10% low
Flowmeter

- Swissflow SF800 - optical sensor
- Battery-backed Counter board
1.3 million pulses
4°C temp drop
6100 pulse/litre
1.6 l/min

\[ \text{Specific heat (kWh/KgC) \times Mass (Kg) \times Temp_{diff} (C) = Energy (Kwh)} \]

\[ 0.00116 \times 1327532/6100 \times 4 = 1 \text{kWh} \]
What’s missing/Future?

- power-fail proofing
- Local User Interface
- Solar: Display tank temp, Bath status, Energy gain
- House: Room temps, Active zones
- Inputs: ‘Make a bath’ button, Leaving house, +1hr
- Configuration: Sensor allocation, Rule adding/adjusting
- How?: Browser, GTK, Misterhouse, Glade
- MythTV plugin, SMS
- Interfaces: xAP, wireless sensors, moon on stick
- Upload data for comparison, to AMEE
Conclusions

- Solar Thermal is great
- 1-wire is great
- UI is harder - needs work
- I have a very tolerant wife