Domestic Power Monitoring

Objective:

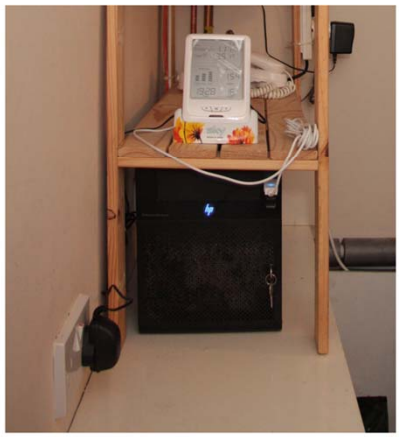
To monitor and graph electrical power usage over time with a view to understanding when power is being used and by what loads

ian@m0ijs.net

Harvesting Numbers

Data collection

Data storage



17/11/2012 M0IJS / Rugby Amateur Transmitting Society 2

- CC128 monitors current and sends this data using a 433MHz module to a display
- Display exports this data, time and temperature as XML strings via a serial port
- Serial port is monitored by a server and data stored into database

RRDtool database

Create the database:

```
#!/bin/sh
# create rrd database for current cost monitoring
rrdtool create powertemp.rrd --step 5 \
DS:Power:GAUGE:180:0:U \
DS:Temperature:GAUGE:180:U:U \
RRA:AVERAGE:0.5:1:3200 \
RRA:AVERAGE:0.5:6:3200 \
RRA:AVERAGE:0.5:36:3200 \
RRA:AVERAGE:0.5:144:3200 \
RRA:AVERAGE:0.5:1008:3200 \
RRA:AVERAGE:0.5:4320:3200 \
RRA:AVERAGE:0.5:52560:3200 \
RRA:AVERAGE:0.5:525600:3200 \
RRA:MIN:0.5:1:3200 \
RRA:MIN:0.5:6:3200 \
RRA:MIN:0.5:36:3200 \
RRA:MIN:0.5:144:3200 \
RRA:MIN:0.5:1008:3200 \
RRA:MIN:0.5:4320:3200 \
RRA:MIN:0.5:52560:3200 \
RRA:MIN:0.5:525600:3200 \
RRA:MAX:0.5:1:3200 \
RRA:MAX:0.5:6:3200 \
RRA:MAX:0.5:36:3200 \
RRA:MAX:0.5:144:3200 \
RRA:MAX:0.5:1008:3200 \
RRA:MAX:0.5:4320:3200 \
RRA:MAX:0.5:52560:3200 \
RRA:MAX:0.5:525600:3200
```


Here we create the database, set the sample interval (step=5, 5 seconds). Also generate and add additional data for averaging, max & min.

RRDtool database

Read from the serial port and send to database:

```
#!/usr/bin/perl

# Reads data from Current Cost and e.on energy monitor devices via serial port.
# Derived from http://www.jibble.org/currentcost/
# and http://www.linuxuk.org/2008/12/currentcost-and-ubuntu/
#
# http://pastebin.com/Waa4ZWAx
# http://oimon.wordpress.com/2011/12/20/power-monitoring-with-linux-the-low-energy-way/
#

use strict;
use Device::SerialPort qw( :PARAM :STAT 0.07 );

my $PORT = "/dev/ttyUSB0";

my $ob = Device::SerialPort->new($PORT);
$ob->baudrate(57600);
$ob->write_settings;

open(SERIAL, "+>$PORT");
while (my $line = <SERIAL>) {
  if ($line =~ m!<tmpr> *([\-\d.]+)</tmpr>.*<watts>0*(\d+)</watts>!) {
     my $temp = $1;
     my $watts = $2;
     #print "Watts: $watts, Temp: $temp\n";
          system("rrdtool update powertemp.rrd N:$watts:$temp")
  }
}
```


PC serial port is a USB UART device:
Bus 004 Device 005: ID 067b:2303 Prolific Technology, Inc. PL2303 Serial Port

This Perl script opens the port, parses the incoming XML string and stores off the values we need.
The script is always running and continuously feeds the database.

Raw XML looks like:

```
<msg>                    start of message
  <src>CC128-v0.11</src>      source and software version
  <dsb>00089</dsb>           days since birth, ie days run
  <time>13:02:39</time>       24 hour clock time as displayed
  <tmpr>18.7</tmpr>          temperature as displayed
  <sensor>1</sensor>         Appliance Number as displayed
  <id>01234</id>             radio ID received from the sensor
  <type>1</type>             sensor Type, "1" = electricity
  <ch1>                   sensor channel
    <watts>00345</watts>      data and units
  </ch1>
 </msg>                     end of message
```

RRDtool database

Create graphs from the data sets:

```
# 10 minute power graph
rrdtool graph /var/www/currentcost/graphs/power-10min.png \
--start end-10m --width 700 --end now --slope-mode \
--no-legend --vertical-label Watts --lower-limit 0 \
--alt-autoscale-max \
DEF:Power=/root/CurrentCost/powertemp.rrd:Power:AVERAGE LINE3:Power#0000FF:Average > /dev/null
# 1hr graph
rrdtool graph /var/www/currentcost/graphs/power-60min.png \
--start end-60m --width 700 --end now --slope-mode \
--no-legend --vertical-label Watts --lower-limit 0 \
--alt-autoscale-max \
DEF:Power=/root/CurrentCost/powertemp.rrd:Power:AVERAGE LINE3:Power#0000FF:Average > /dev/null
# 1 day power graph
rrdtool graph /var/www/currentcost/graphs/power-day.png \
--start end-1d --width 700 --end now --slope-mode \
--no-legend --vertical-label Watts --lower-limit 0 \
--alt-autoscale-max \
DEF:Power=/root/CurrentCost/powertemp.rrd:Power:AVERAGE \
DEF:PowerMin=/root/CurrentCost/powertemp.rrd:Power:MIN \
DEF:PowerMax=/root/CurrentCost/powertemp.rrd:Power:MAX \
CDEF:PowerRange=PowerMax,PowerMin,- \
LINE1:PowerMin: \
AREA:PowerRange#0000FF11:"Error Range":STACK \
LINE1:PowerMin#0000FF33:"Min" \
LINE1:PowerMax#0000FF33:"Max" \
LINE1:Power#0000FF:"Average" > /dev/null
# 1 week power graph
rrdtool graph /var/www/currentcost/graphs/power-week.png \
--start end-7d --width 700 --end now --slope-mode \
--no-legend --vertical-label Watts --lower-limit 0 \
--alt-autoscale-max \
DEF:Power=/root/CurrentCost/powertemp.rrd:Power:AVERAGE \
DEF:PowerMin=/root/CurrentCost/powertemp.rrd:Power:MIN \
DEF:PowerMax=/root/CurrentCost/powertemp.rrd:Power:MAX \
CDEF:PowerRange=PowerMax,PowerMin,- \
LINE1:PowerMin: \
AREA:PowerRange#0000FF11:"Error Range":STACK \
LINE1:PowerMin#0000FF33:"Min" \
LINE1:PowerMax#0000FF33:"Max" \
LINE1:Power#0000FF:"Average" > /dev/null
```

```
# 1 Month power graph
rrdtool graph /var/www/currentcost/graphs/power-month.png \
--start end-30d --width 700 --end now --slope-mode \
--no-legend --vertical-label Watts --lower-limit 0 \
--alt-autoscale-max \
DEF:Power=/root/CurrentCost/powertemp.rrd:Power:AVERAGE \
DEF:PowerMin=/root/CurrentCost/powertemp.rrd:Power:MIN \
DEF:PowerMax=/root/CurrentCost/powertemp.rrd:Power:MAX \
CDEF:PowerRange=PowerMax,PowerMin,- \
LINE1:PowerMin: \
AREA:PowerRange#0000FF11:"Error Range":STACK \
LINE1:PowerMin#0000FF33:"Min" \
LINE1:PowerMax#0000FF33:"Max" \
LINE1:Power#0000FF:"Average" > /dev/null


# ################## TEMPERATURE GRAPHS ######################
# 60 min temp graph
rrdtool graph /var/www/currentcost/graphs/temp-hour.png --start end-60m \
--width 700 --end now --slope-mode --no-legend --vertical-label Celcius \
--lower-limit 0 --alt-autoscale-max \
DEF:Power=/root/CurrentCost/powertemp.rrd:Temperature:AVERAGE LINE3:Power#0000FF:Average >
/dev/null
# 1 day temperature
rrdtool graph /var/www/currentcost/graphs/temp-day.png --start end-1d \
--width 700 --end now --slope-mode --no-legend --vertical-label Celcius \
--lower-limit 0 --alt-autoscale-max \
DEF:Power=/root/CurrentCost/powertemp.rrd:Temperature:AVERAGE LINE3:Power#0000FF:Average >
/dev/null
# 1 week temperature
rrdtool graph /var/www/currentcost/graphs/temp-week.png --start end-7d \
--width 700 --end now --slope-mode --no-legend --vertical-label Celcius \
--lower-limit 0 --alt-autoscale-max \
DEF:Power=/root/CurrentCost/powertemp.rrd:Temperature:AVERAGE LINE3:Power#0000FF:Average >
/dev/null
# 1 month temperature
rrdtool graph /var/www/currentcost/graphs/temp-month.png --start end-30d \
--width 700 --end now --slope-mode --no-legend --vertical-label Celcius \
--lower-limit 0 --alt-autoscale-max \
DEF:Power=/root/CurrentCost/powertemp.rrd:Temperature:AVERAGE LINE3:Power#0000FF:Average >
/dev/null
```


This Perl script generates image files, the graphs, from datasets pulled from the database. This results in files which our web server can then make available. The script is run as a cron task every 5 minutes.

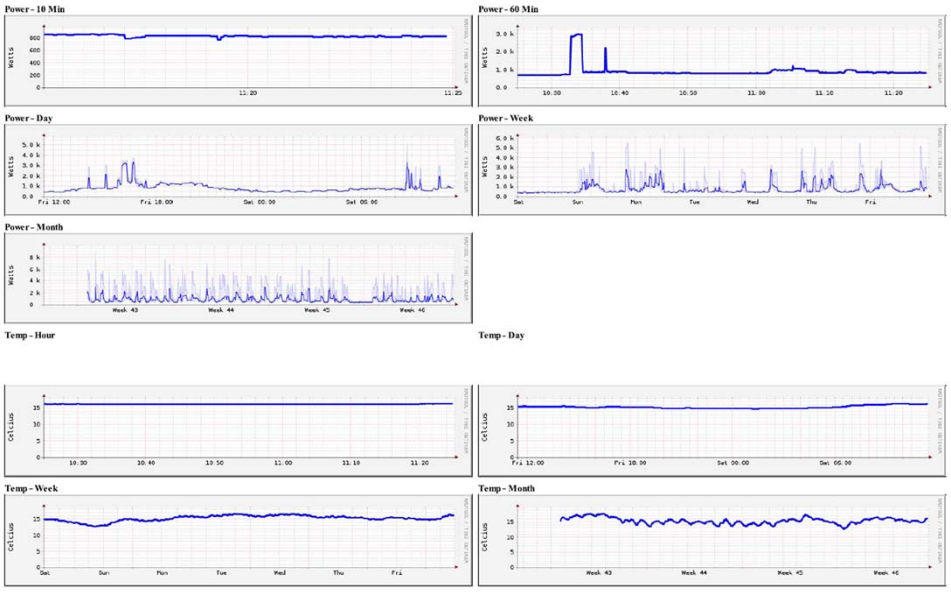
```
Rtask:/var/www/currentcost/graphs# ls -al
total 148
drwxr-xr-x 2 root root  4096 Oct 22 09:01 .
drwxr-xr-x 3 root root  4096 Oct 21 18:59 ..
-rw-r--r-- 1 root root  2295 Oct 27 10:42 index_1.html
-rw-r--r-- 1 root root  8878 Oct 27 11:05 power-10min.png
-rw-r--r-- 1 root root 12012 Oct 27 11:05 power-60min.png
-rw-r--r-- 1 root root 17757 Oct 27 11:05 power-day.png
-rw-r--r-- 1 root root 16642 Oct 27 11:05 power-month.png
-rw-r--r-- 1 root root 23990 Oct 27 11:05 power-week.png
-rw-r--r-- 1 root root  9295 Oct 27 11:05 temp-day.png
-rw-r--r-- 1 root root  8798 Oct 27 11:05 temp-hour.png
-rw-r--r-- 1 root root  9466 Oct 27 11:05 temp-month.png
-rw-r--r-- 1 root root  9624 Oct 27 11:05 temp-week.png
```

Web page

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
   <TITLE>CC128 Current and Temperature logger</TITLE>
   <!-- Command line is easier to read using "View Page Properties" of your browser -->
   <!-- But not all browsers show that information. :-(                            -->
   <meta http-equiv="content-type" content="text/html; charset=iso-8859-15" >
   <META HTTP-EQUIV="Refresh" CONTENT="300" >
   <META HTTP-EQUIV="Cache-Control" content="no-cache" >
   <META HTTP-EQUIV="Pragma" CONTENT="no-cache" >
   <META HTTP-EQUIV="Expires" CONTENT="Wed, 04 Jan 2012 19:57:53 GMT" >
   <LINK HREF="favicon.ico" rel="shortcut icon" >

<style type="text/css">
/* commandline was: /usr/bin/indexmaker --output=/var/www/mrtg/index.html /etc/mrtg.cfg */
/* sorry, no style, just abusing this to place the commandline and pass validation */
</style>
</HEAD>

<BODY bgcolor="#ffffff" text="#000000" link="#000000" vlink="#000000" alink="#000000">


<H1>CC128 Index Page.  Hacked together from an MRTG index page.</H1>

<TABLE BORDER=0 CELLPADDING=0 CELLSPACING=10>
<tr>
<td><DIV><B>Power - 10 Min</B></DIV>
<DIV><A HREF=""><IMG BORDER=1 ALT="" SRC="power-10min.png"></A><BR>
</td>
<td><DIV><B>Power - 60 Min</B></DIV>
<DIV><A HREF=""><IMG BORDER=1 ALT="" SRC="power-60min.png"></A><BR>
</td></tr>

<tr>
<td><DIV><B>Power - Day</B></DIV>
<DIV><A HREF=""><IMG BORDER=1 ALT="" SRC="power-day.png"></A><BR>
</td>
<td><DIV><B>Power - Week</B></DIV>
<DIV><A HREF=""><IMG BORDER=1 ALT="" SRC="power-week.png"></A><BR>
</td></tr>

<tr>
<td><DIV><B>Power - Month</B></DIV>
<DIV><A HREF=""><IMG BORDER=1 ALT="" SRC="power-month.png"></A><BR>
</td></tr>



<tr>
<td><DIV><B>Temp - Hour</B></DIV>
<DIV><A HREF=""><IMG BORDER=1 ALT="" SRC="temp-hour.png"></A><BR>

</td><td><DIV><B>Temp - Day</B></DIV>
<DIV><A HREF=""><IMG BORDER=1 ALT="" SRC="temp-day.png"></A><BR>
</td></tr>
<tr>

<td><DIV><B>Temp - Week</B></DIV>
<DIV><A HREF=""><IMG BORDER=1 ALT="" SRC="temp-week.png"></A><BR>
<SMALL><!--#flastmod file="192.168.1.252_3.html" --></SMALL></DIV>

</td><td><DIV><B>Temp - Month</B></DIV>
<DIV><A HREF=""><IMG BORDER=1 ALT="" SRC="temp-month.png"></A><BR>
<SMALL><!--#flastmod file="192.168.1.252_4.html" --></SMALL></DIV>
</td></tr>

</TABLE>

<BR>
</BODY>
</HTML>
```


HTML generates a web page pulling together the images, graphs, we have previously made.

Let there be graphs!



Resultant rendering of html, with PNGs previously created by the script.

CurrentCost 128

17/11/2012 M0IJS / Rugby Amateur Transmitting Society 8

Sensor on left, 433MHz telemetry module. One hard wired current clamp and provision for two more via jacks.
MCHP MCU.

Display on right. 433MHz telemetry module. PIC MCU, PIC18F85J90. RJ45 socket exposes MCU programming pins and UART.