

I've just bought a new house, and having a geeky interest in home automation and a more serious interest in climate change I bought some lightwaveRF home automation kit including an electricity monitor.

I work in IT and am familiar with the Cacti graphing tool which provides a front end for rrdtool a graphing program specifically designed for displaying time-series data. This seemed like the serious interest in climate change, I wanted to try out some home automation with a view to first measuring our energy usage and then hopefully be able to take some action to reduce it.

I've just bought the LightwaveRF Wifi link and energy meter. Annoyingly, once I logged into the LightwaveRF site, I found merely a teaser about a "concierge" service to show historical power usage. I don't want to wait until September, and since it sounds like there might be a cost associated with this service when it does arrive, I took it upon myself to try to make my own graphing system.ideal solution since I have a Linux box on all the time (home server, media streaming and automation). So I took it upon myself to build a cacti plugin for the LightwaveRF wifi link and energy monitor.

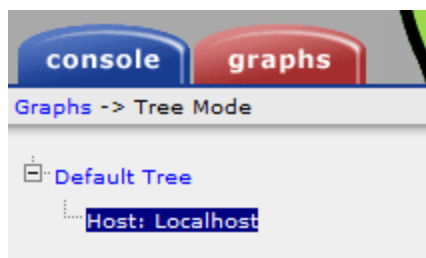
I started with [this useful blog post](#) which got me started with PHP for LightwaveRF automation, and then using the [LightwaveRF open API](#) to work out how to receive the response, I followed the [instructions here](#) about building a PHP cacti plugin. I built a php script able to send a UDP datagram requesting current energy data and then receive and parse the result and pass it back to cacti.

Unfortunately, Cacti have changed their PHP script method since that post was written so I had to follow the cacti docs to port my script into their new format. In the end however I was able to create a working script which could be called by Cacti to retrieve energy consumption data from a LightwaveRF Wifi link with energy meter.

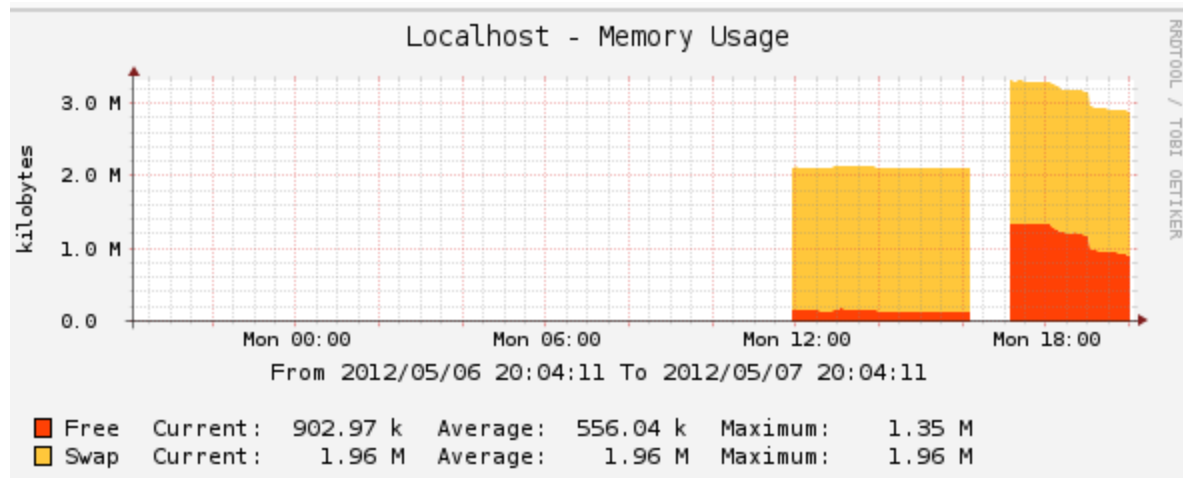
The script is available from Github here:

`git://github.com/stephenjirvine/lightwaverf_energy_poller.git`

In order to use this script you will need a *working* cacti installation, I use ubuntu and was able to install using the ubuntu package manager apt-get. Once installed, you log in, reset your password and away you go. To help you make sure your graphs are working Cacti includes a set of 4 graphs you should be able to see by clicking the Graphs button in the top left.



You should now see some graphs below that look like this:



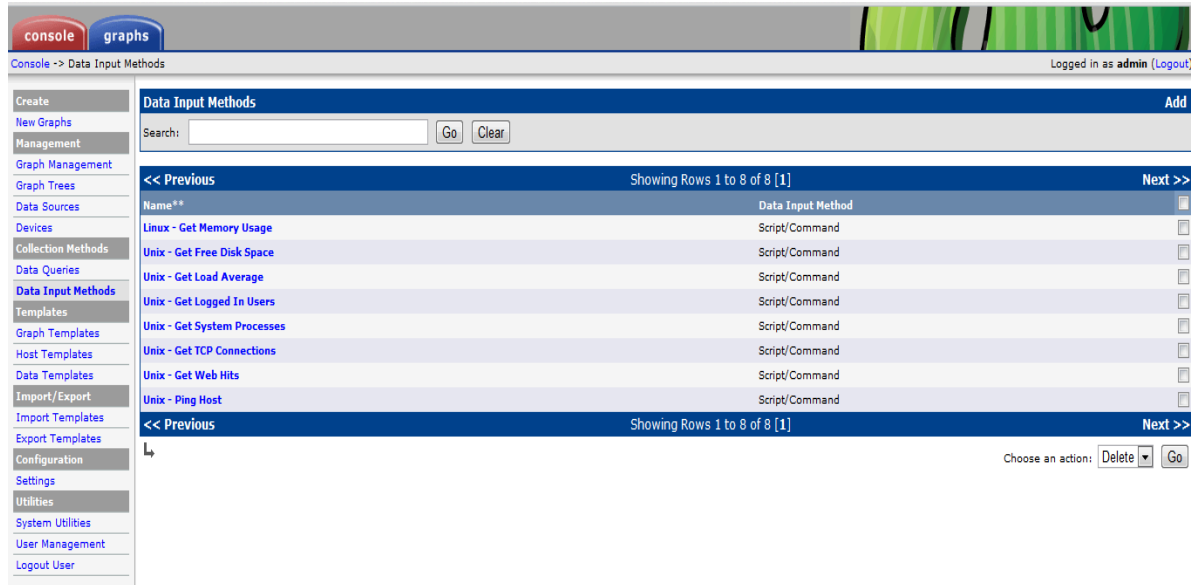
These are the cacti builtin graphs so if you can't see them you need to fix cacti before going any further.

Hint: check the user your apache server is running under has permission to access the directories where cacti stores it's data.

Once you've verified you can see the builtin graphs, copy the php script up to a folder on the cacti computer that is accessible to the cacti process, (I chose the scripts folder in the cacti directory itself /use/share/cacti/site/scripts, but you can put it anywhere).

Important: the file must be called ss_lwrfenergypoller.php, this is a cacti convention.

Once done you need to create a data input method for LightwaveRF, so click Console at the top and click on Data Input method. Then click add in the top right to make a new one to add to the existing list.



Fill in the fields as above, if you want to change the name that's fine but the Input String must read:

```
/usr/share/cacti/site/scripts/ss_lwrfenergypoller.php ss_lwrfenergypoller <myip> <wifilinkip>
<send_port> <recv_port> <broadcast_string>
```

If you put your script in a different place then make sure your path goes above.

Click create and then click Add in the Input Fields section:

Input Fields			Add
Name	Field Order	Friendly Name	
No Input Fields			

You will need to do this 5 times, one for each of the fields in <> brackets in the Input string above, what we're doing here is defining the arguments we're going to supply to the energy poller script.

Select the first Input field "myip" fill in the details as shown here:

Input Fields [edit: LightwaveRF Energy Poller]

Field [Input]

Choose the associated field from the Input field.

myip

Friendly Name

Enter a meaningful name for this data input method.

IP Address of server running the query

Regular Expression Match

If you want to require a certain regular expression to be matched against input data, enter it here (ereg format).

Allow Empty Input

Check here if you want to allow NULL input in this field from the user.

☐ Allow Empty Input

Special Type Code

If this field should be treated specially by host templates, indicate so here. Valid keywords for this field are

'hostname', 'host_id',

'snmp_community',

'snmp_username',

'snmp_password',

'snmp_auth_protocol',

'snmp_priv_passphrase',

'snmp_priv_protocol',

'snmp_context',

'snmp_version',

'snmp_port',

'snmp_timeout'

Cancel

Create

Then click create, do this for each of the input fields:

myip = IP Address of the computer running cacti
wifilinkip = IP Address of the Wifi Link
send_port = UDP port for sending data
recv_port = UDP port for receiving data
broadcast_string = The string to send over UDP

By now your input fields table should look like this:

Input Fields			Add
Name	Field Order	Friendly Name	
myip	1	IP Address of server running the query	✕
wifilinkip	2	IP Address of the Wifi Link	✕
send_port	3	UDP port for sending data	✕
recv_port	4	UDP port for receiving data	✕
broadcast_string	5	The string to send over UDP	✕

I decided to break out these fields as input arguments to the script to allow the same script to be re-used in the event of LightwaveRF adding other pollable devices later. We shouldn't need to edit the script in the future, just call it with different inputs.

Now we need to define the output of the script. Click add in output fields, this should be quicker, there's only one output, the current power measured in Watts:

Output Fields [edit: LightwaveRF Energy Poller]

Field [Output]
Enter a name for this Output field.

Friendly Name
Enter a meaningful name for this data input method.

Update RRD File
Whether data from this output field is to be entered into the rrd file.
☒ Update RRD File

Hit create and you should see Save Successful. in the top.

Save Successful.

Data Input Methods [edit: LightwaveRF Energy Poller]

Name

Enter a meaningful name for this data input method.

Input Type

Choose the method you wish to use to collect data for this Data Input method.

Input String

The data that is sent to the script, which includes the complete path to the script and input sources in <> brackets.

```
/usr/share/cacti/site/scripts/ss_lwrfenergypoller.php  
ss_lwrfenergypoller <myip> <wifilinkip> <send_port>  
<recv_port> <broadcast_string>
```

Input Fields

Add

Name	Field Order	Friendly Name	
myip	1	IP Address of server running the query	✗
wifilinkip	2	IP Address of the Wifi Link	✗
send_port	3	UDP port for sending data	✗
recv_port	4	UDP port for receiving data	✗
broadcast_string	5	The string to send over UDP	✗

Output Fields

Add

Name	Field Order	Friendly Name	Update RRA	
current_watts	0 (Not In Use)	Power Usage in Watts	Selected	✗

You've successfully defined the method by which we collect data from the WiFi link, now we have to set up a Data Source to actually do it.

On the menu on the left click Data Sources, then add in the top right:

Create

New Graphs

Management

Graph Management

Graph Trees

Data Sources

--- RRAs

Devices

Collection Methods

Data Queries

Data Input Methods

Templates

Graph Templates

Host Templates

Data Templates

Import/Export

Import Templates

Export Templates

Configuration

Settings

Data Sources [host: 127.0.0.1] Add

Host: Localhost (127.0.0.1) ▼

Template: Any ▼ Go Clear

Method: Any ▼ Rows per Page: 30 ▼

Search:

<< Previous

Showing Rows 1 to 5 of 5 [1]

Next >>

Name**	ID	Data Input Method	Poller Interval	Active	Template Name	<input type="checkbox"/>
Localhost - Load Average	5	Unix - Get Load Average	5 Minutes	Yes	Unix - Load Average	<input type="checkbox"/>
Localhost - Logged in Users	6	Unix - Get Logged In Users	5 Minutes	Yes	Unix - Logged in Users	<input type="checkbox"/>
Localhost - Memory - Free	3	Linux - Get Memory Usage	5 Minutes	Yes	Linux - Memory - Free	<input type="checkbox"/>
Localhost - Memory - Free Swap	4	Linux - Get Memory Usage	5 Minutes	Yes	Linux - Memory - Free Swap	<input type="checkbox"/>
Localhost - Processes	7	Unix - Get System Processes	5 Minutes	Yes	Unix - Processes	<input type="checkbox"/>

<< Previous

Showing Rows 1 to 5 of 5 [1]

Next >>

Choose an action: Delete ▼ Go

We don't have a Template to work from, and we should leave this data source associated with Localhost, click create and fill the rest in as below. Leave all the RRAs selected. Hit Create.

Another box should pop out of the bottom, now you can populate the inputs we defined earlier:

Custom Data [data input: LightwaveRF Energy Poller]

IP Address of server running the query

192.168.0.200

IP Address of the Wifi Link

192.168.0.100

The string to send over UDP

'001,@?'

UDP port for receiving data

9761

UDP port for sending data

9760

Cancel

Save

Please note these just order themselves alphabetically based on what's in the description string, so be careful what you put where. You will obviously need to change the IP Address values to match your environment.

Once you save this, you should be collecting data, but now we need to define a graph to view it, on the left go to Graph Management., and click add in the top left.

Create

New Graphs

Management

Graph Management

--- CDEFs

--- Colors

--- GPRINT Presets

Graph Trees

Data Sources

Devices

Collection Methods

Data Queries

Data Input Methods

Templates

Graph Templates

Host Templates

Graph Management

Add

Host: Any Template: Any Go Clear

Search: Rows per Page: 30

<< Previous

Showing Rows 1 to 4 of 4 [1]

Next >>

Graph Title**	ID	Template Name	Size	
Localhost - Load Average	2	Unix - Load Average	120x500	
Localhost - Logged in Users	3	Unix - Logged in Users	120x500	
Localhost - Memory Usage	1	Linux - Memory Usage	120x500	
Localhost - Processes	4	Unix - Processes	120x500	

<< Previous

Showing Rows 1 to 4 of 4 [1]

Next >>

Choose an action:

Delete

Go

Leave template blank and assign this graph to localhost:

Graph Template Selection [new]

Selected Graph Template

Choose a graph template to apply to this graph. Please note that graph data may be lost if you change the graph template after one is already applied.

None

Host

Choose the host that this graph belongs to.

Localhost (127.0.0.1)

Then I just gave it a name and left everything else as it was.

Graph Configuration	
Title (--title) The name that is printed on the graph.	<input type="text" value="Home electricity consumption"/>
Image Format (--imgformat) The type of graph that is generated; PNG, GIF or SVG. The selection of graph image type is very RRDtool dependent.	<input type="text" value="PNG"/>
Height (--height) The height (in pixels) that the graph is.	<input type="text" value="120"/>
Width (--width) The width (in pixels) that the graph is.	<input type="text" value="500"/>
Slope Mode (--slope-mode) Using Slope Mode, in RRDtool 1.2.x and above, evens out the shape of the graphs at the expense of some on screen resolution.	<input checked="" type="checkbox"/> Slope Mode (--slope-mode)
Auto Scale Auto scale the y-axis instead of defining an upper and lower limit. Note: if this is checked both the Upper and Lower limit will be ignored.	<input checked="" type="checkbox"/> Auto Scale
Auto Scale Options Use --alt-autoscale to scale to the absolute minimum and maximum --alt-autoscale-max to scale to the maximum value, using a given lower limit --alt-autoscale-min to scale to the minimum value, using a given upper limit --alt-autoscale (with limits) to scale using both lower and upper limits (rrdtool default)	<input type="radio"/> Use --alt-autoscale (ignoring given limits) <input checked="" type="radio"/> Use --alt-autoscale-max (accepting a lower limit) <input type="radio"/> Use --alt-autoscale-min (accepting an upper limit, requires rrdtool 1.2.x) <input type="radio"/> Use --alt-autoscale (accepting both limits, rrdtool default)
Logarithmic Scaling (--logarithmic) Use Logarithmic y-axis scaling	<input type="checkbox"/> Logarithmic Scaling (--logarithmic)
SI Units for Logarithmic Scaling (--units=si) Use SI Units for Logarithmic Scaling instead of using exponential notation (not available for rrdtool-1.0.x). Note: Linear graphs use SI notation by default.	<input type="checkbox"/> SI Units for Logarithmic Scaling (--units=si)
Rigid Boundaries Mode (--rigid) Do not expand the lower and upper limit if the graph contains a value outside the valid range.	<input type="checkbox"/> Rigid Boundaries Mode (--rigid)
Auto Padding Pad text so that legend and graph data always line up. Note: this could cause graphs to take longer to render because of the larger overhead. Also Auto Padding may not be accurate on all types of graphs, consistent labeling usually helps.	<input checked="" type="checkbox"/> Auto Padding
Allow Graph Export Choose whether this graph will be included in the static html/png export if you use cacti's export feature.	<input checked="" type="checkbox"/> Allow Graph Export

Once you save this dialogue, a graph designer pops up above it:

Graph Items [edit: Home electricity consumption]					Add
Graph Item	Data Source	Graph Item Type	CF Type	Item Color	
No Items					

RRDTool Command:

```

/usr/bin/rrdtool graph - \
--imgformat=PNG \
--start=-86400 \
--end=-300 \
--title='Home electricity consumption' \
--base=1000 \
--height=120 \
--width=500 \
--alt-autoscale-max \
--lower-limit=0 \
--vertical-label='' \
--slope-mode \
--font TITLE:10: \
--font AXIS:7: \
--font LEGEND:8: \
--font UNIT:7: \

```



RRDTool Says:

Use the add dialogue to add items to the graph, here we add an area graph of our energy usage, in a nice orange colour:

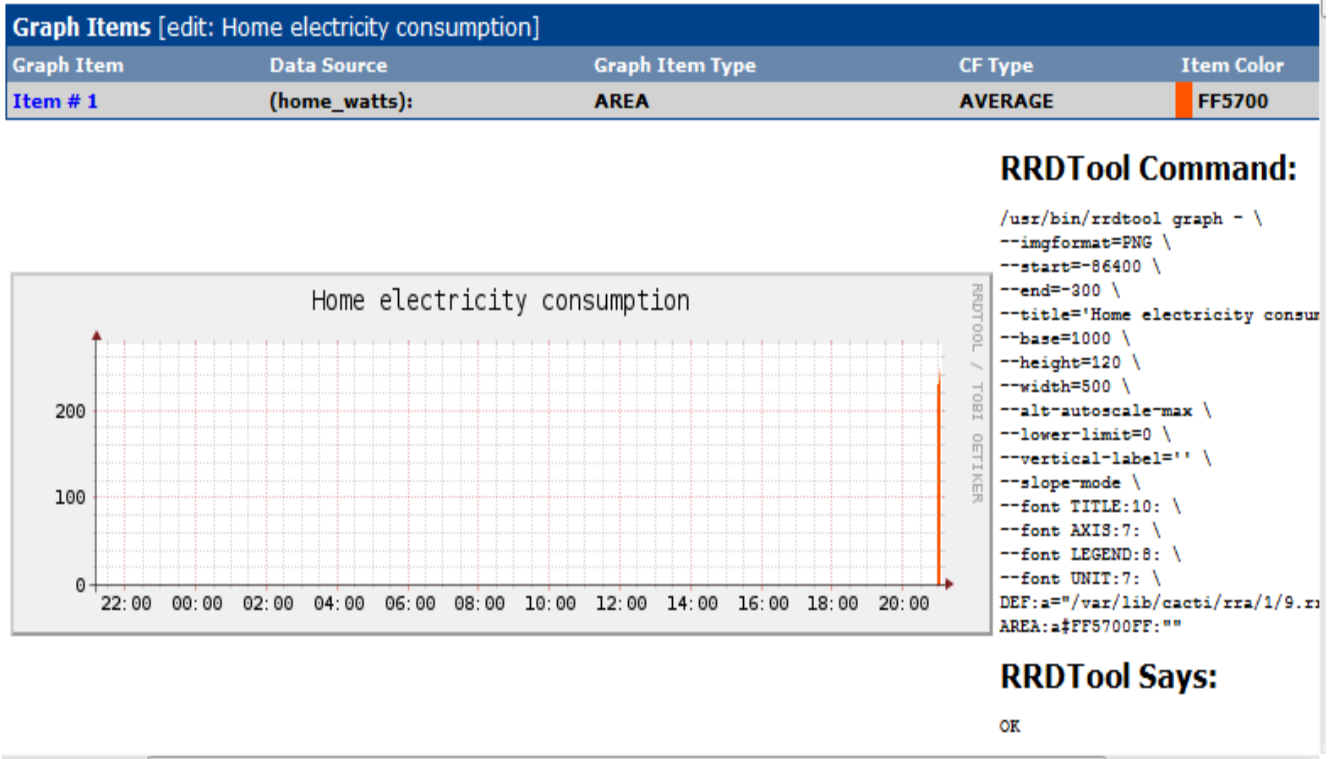
Data Sources [host: 127.0.0.1]	
Host:	Localhost (127.0.0.1) ▼
Data Template:	Any ▼

Graph Items [edit graph: Home electricity consumption]	
Data Source The data source to use for this graph item.	Home LightwaveRF Energy Poller (home_watts) ▼
Color The color to use for the legend.	FF5700 ▼
Opacity/Alpha Channel The opacity/alpha channel of the color. Not available for rrdtool-1.0.x.	100% ▼
Graph Item Type How data for this item is represented visually on the graph.	AREA ▼
Consolidation Function How data for this item is represented statistically on the graph.	AVERAGE ▼
CDEF Function A CDEF (math) function to apply to this item on the graph.	None ▼
Value The value of an HRULE or VRULE graph item.	
GPRINT Type If this graph item is a GPRINT, you can optionally choose another format here. You can define additional types under "GPRINT Presets".	Normal ▼
Text Format Text that will be displayed on the legend for this graph item.	
Insert Hard Return Forces the legend to the next line after this item.	<input type="checkbox"/> Insert Hard Return
Sequence	

Cancel

Create

We should see our new graph appear below:



Ace! It worked. Now we see some data appearing in our first graph.

We can add other items, for example a gprint function to show Average power usage:

Graph Items [edit graph: Home electricity consumption]

Data Source The data source to use for this graph item.	Home LightwaveRF Energy Poller (home_watts) ▼
Color The color to use for the legend.	None ▼
Opacity/Alpha Channel The opacity/alpha channel of the color. Not available for rrdtool-1.0.x.	100% ▼
Graph Item Type How data for this item is represented visually on the graph.	GPRINT ▼
Consolidation Function How data for this item is represented statistically on the graph.	AVERAGE ▼
CDEF Function A CDEF (math) function to apply to this item on the graph.	None ▼
Value The value of an HRULE or VRULE graph item.	
GPRINT Type If this graph item is a GPRINT, you can optionally choose another format here. You can define additional types under "GPRINT Presets".	Normal ▼
Text Format Text that will be displayed on the legend for this graph item.	Average:
Insert Hard Return Forces the legend to the next line after this item.	<input type="checkbox"/> Insert Hard Return
Sequence	2

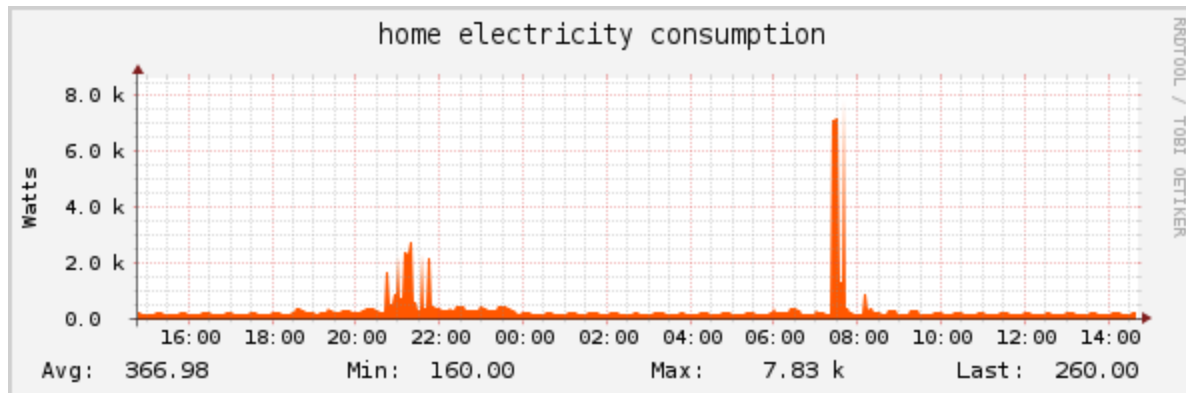
Cancel Save

Now our graph has an average calculated on the fly and added to the bottom left:



You should have a play around here with the different options and labels, there's a lot you can do.

In the end I added a few more GPRINT statements to show min, max etc etc and after a day's logging here's my output.



The 2kw spike around 21:00 is us cooking dinner and the big 7.5kw spike just before 08:00 is morning showers.