

[Edit on GitHub](#)[Edit on GitHub Prose](#)

# Monitoring SolarPV, Heatpump and house electric, EmonTx v2 system upgrade example

2014-08-06 17:25:00 +00002014-08-06 17:25:00 +0000 Trystan Lea

five minutes reading time

- [Heat Pump](#)
- [emonTx V2](#)
- [upgrade](#)

## Comments

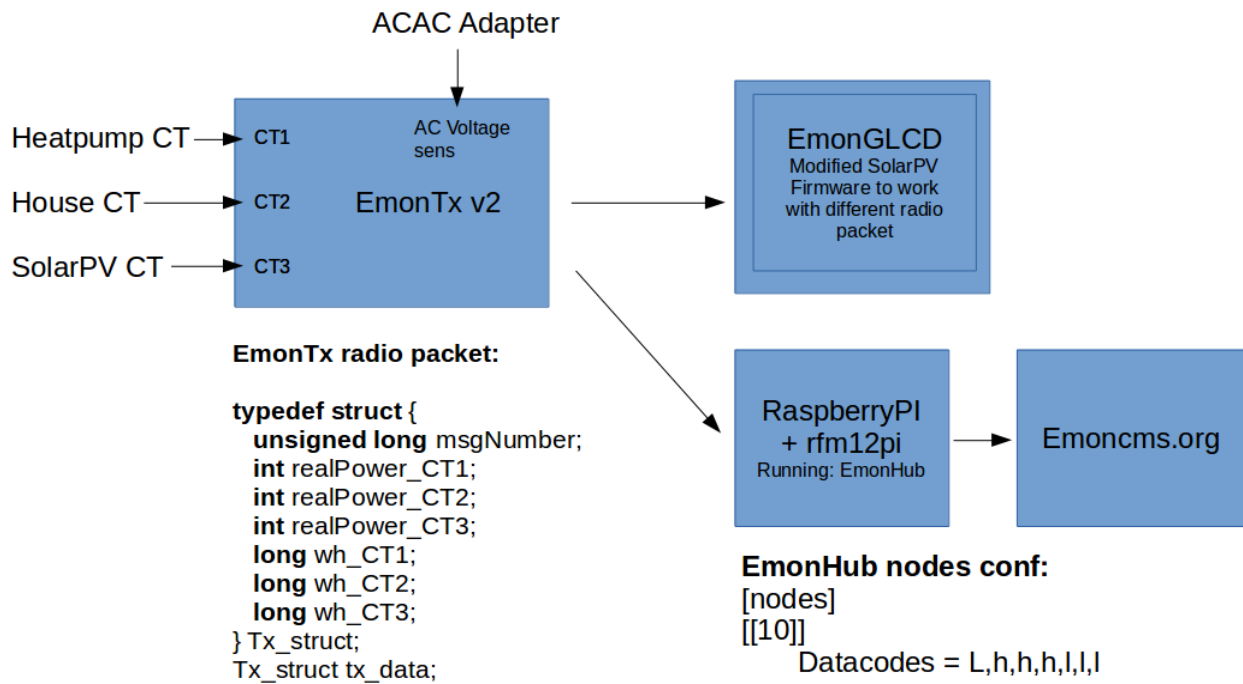
Several years ago we did a community energy monitoring project in our local area, Penrhyndeudraeth, North Wales. It involved installing 20 openenergymonitor energy monitors in 20 households and carrying out energy assessment looking at electricity, heating and transport, there is a little more about it and some of the tools we developed [here](#).

The houses that were most engaged in the monitoring part of the project still have those energy monitors installed and yesterday I went to upgrade one of these systems, I thought I'd write a blog post on what I did (written half as a guide) as it might be useful for those using older hardware such as the emonTx V2 and it's also quite an interesting example of what you can do with the monitoring system.

The upgrade essentially involved replacing a first generation nanode basestation with a RaspberryPI basestation running the latest image and also upgrading the emontx v2 that was in place with new firmware that both does the higher accuracy continuous sampling (developed by Robin Emley) and the watt hour calculation on board.

This particular system is measuring Solar PV generation, household electric consumption and the electrical consumption of a heatpump. It has an EmonGLCD Display running the SolarPV firmware, with the red/green glowing ambient light depending on whether more power is being used than generated or vice versa and the data is also posted to [emoncms.org](#) for graphing.

Here's a system diagram:



## EmonTx:

Upload to the EmonTxV2 the continuous sampling + watt hour firmware:

[https://github.com/openenergymonitor/emonTxFirmware/tree/master/emonTxV2/emonTx\\_continuous\\_watthours](https://github.com/openenergymonitor/emonTxFirmware/tree/master/emonTxV2/emonTx_continuous_watthours)

The same thing can of course be achieved with an EmonTxV3 but with the added benefit of a fourth channel and only needing the one ACAC Power adapter. If you have the EmonTxV3 upload the continuous sampling + watt hour firmware found here:

[https://github.com/openenergymonitor/emonTxFirmware/tree/master/emonTxV3/RFM12B/Examples/emonTxV3\\_continuous\\_kwhtotals\\_noeprom](https://github.com/openenergymonitor/emonTxFirmware/tree/master/emonTxV3/RFM12B/Examples/emonTxV3_continuous_kwhtotals_noeprom)

## EmonGLCD

Open the SolarPV example:

<https://github.com/openenergymonitor/EmonGLCD/tree/master/SolarPV>

On [lines 68 and 69](#) change the emontx radio packet definition from:

```
typedef struct { int power1, power2, power3, Vrms; } PayloadTX;
PayloadTX emontx;
```

to:

```
typedef struct {
    unsigned long msgNumber;
    int heatpump; // heatpump
    int power1; // house power
    int power2; // solar power
    long wh_CT1;
    long wh_CT2;
    long wh_CT3;
} PayloadTX;
PayloadTX emontx;
```

I've named the second variable here heatpump instead of power1 and then powers 1 to 3 because power1 and power2 are used as house power and solar power lower down in the firmware.

## RaspberryPI Basestation

I used the latest (28th of July emonSD image) that has EmonHub (thanks to Jerome and Paul Burnell) and emoncms ready to go but without the local emoncms installation enabled, the PI was just used as an EmonHub gateway forwarder to emoncms.org. I used the nice pimoroni berryblack raspberrypi case so that the installation was tidy.

[Download ready-to-go image: emonSD-13-08-14.img.zip \[link updated\]](#)

To configure EmonHub to post to emoncms.org, SSH into the raspberrypi and put the PI in read/write mode with:

```
rpi-rw
```

and then open the emonhub configuration file:

```
nano /boot/emonhub.conf
```

Set the dispatcher url to: <http://emoncms.org> and the apikey to your emoncms.org account write apikey.

In order to decode the emontx continuous + watt hour radio packet add a node decoder to the nodes section in the emonhub config file at the bottom:

```
[nodes]
[[10]]
  Datacodes = L,h,h,h,l,l,l
```

Secure the pi user by changing the password (the default password is raspberry)

```
passwd
```

Place the PI back in read only mode:

```
rpi-ro
```

## Emoncms configuration

With the emontx running and the raspberrypi configured as above the inputs will now appear in emoncms.org, but they will be un-named. For this particular installation I named them as in the picture below to correspond to the radio packet format and what I knew I was monitoring on each CT channel.

I then logged each of the power inputs to a fixed interval with averaging (PHPFiwa) feed and used the Wh Accumulator input processor to record the Watt hour totals. The Wh Accumulator checks if the watt hour total has reset and if it does it continues the accumulation from the last point. I used the fixed interval without averaging feed engine for the watt hour feeds.

## Inputs

[Input API Help](#)

### Node10: Heatpump Wh config

Input processes are executed sequentially with the result being passed back for further processing by the next processor in the input processing list.

Order	Process	Arg	Actions
1	Wh Accumulator	Node:10: Heatpump Wh (feedvalue:348.00)	

#### Add process:

Log to feed  CREATE NEW:  Heatpump Wh Feed engine: Fixed Interval With Averaging 10s

**Log to feed:** This processor logs the current selected input to a timeseries feed which can then be used to explore historic data. This is recommended for logging power, temperature, humidity, voltage and current data.

**Feed engine:** The fixed interval with averaging (PHPFIIWA) feed engine is the recommended engine to use for logging power, temperature, humidity, voltage and current data. In addition to storing the full resolution data it produces a series of downsampled averaged layers which gives a more accurate representation of the data when viewing the data over a large time range.

**Feed interval:** When selecting the feed interval select an interval that is the same as, or longer than the update rate that is set in your monitoring equipment. Setting the interval rate to be shorter than the update rate of the equipment causes un-needed disk space to be used up.

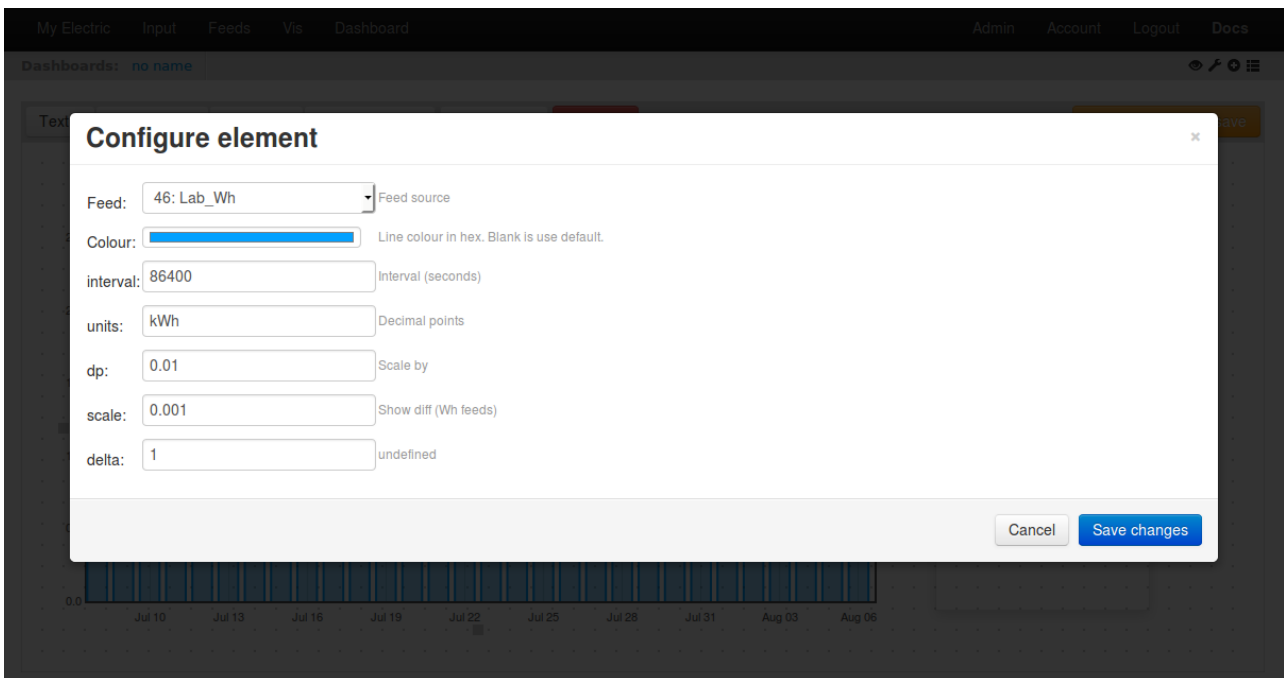
Node:	Key	Name	Process list	last updated	value			
10	1	Message Number		11s ago	50088			
10	2	Heatpump		11s ago	9.00			
10	3	House		11s ago	290			
10	4	SolarPV		11s ago	543			
10	5	Heatpump Wh		11s ago	352			
10	6	House Wh		11s ago	4649			
10	7	SolarPV Wh		11s ago	7165			

To make use of the higher accuracy calculating of watt hours on the emontx and the recording of watt hour feeds, there is a slightly different procedure for creating dashboard kwh per day graphs:

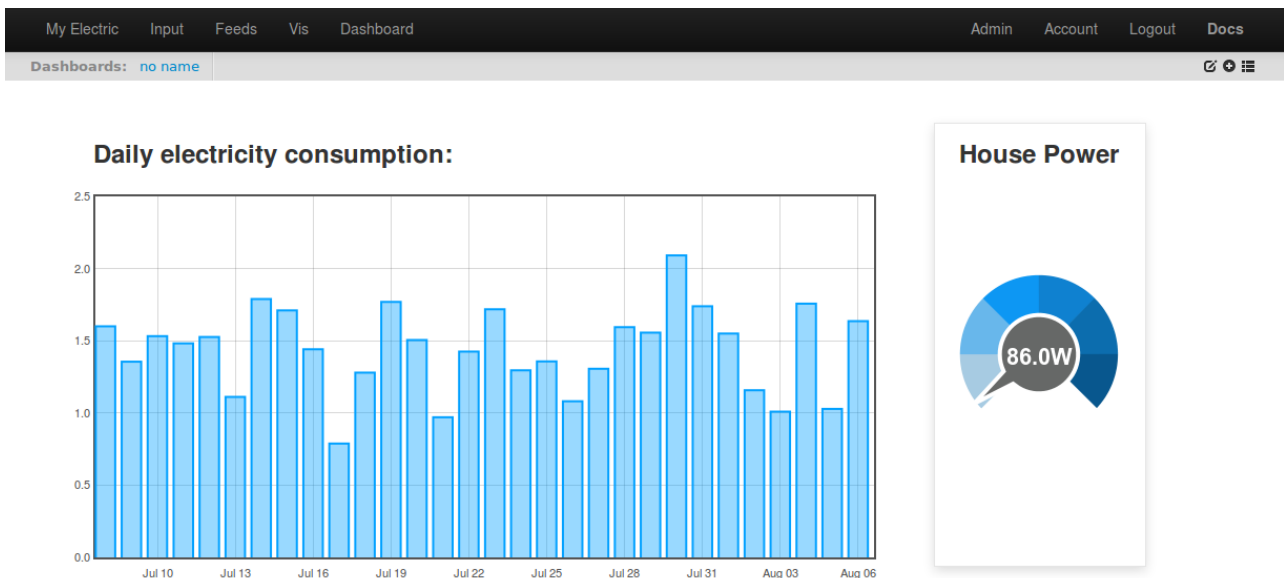
### Creating a daily electricity consumption graph in a dashboard using watt hour feeds (instead of the old power\_to\_kwhd method).

Create a new dashboard and select visualisations > bargraph.

Click on the bargraph and click configure to bring up the configuration dialog. Select the watt hour feed that you want to display daily totals from, enter interval, units, dp, scale and delta=1 as below: (Note that you could change the interval here to any interval: hourly, half day, week etc)



Click save and continue to add any other text and widgets as required. A simple dashboard showing daily consumption and the current power could look like the screenshot below and could be repeated adding a graph for each channel, heatpump, solarpv and house consumption.



To engage in discussion regarding this post, please post on our [Community Forum](#).

## Recent Posts

- [Introducing IoTaWatt](#)
- [CydYnni – Energy Local, community hydro smart grid](#)
- [Advances Wales Magazine Article](#)
- [Emoncms Android App V2](#)
- [emonPi Network Setup Wizard](#)
- [Raspberry Pi blog](#)
- [emonPi as a Home Automation Hub](#)
- [MyHomeEnergyPlanner - open source home energy assessment](#)
- [Launching Learn: Sustainable Energy](#)
- [New home page and Learn website!](#)

Website powered by [Jekyll](#) and [Oscalite](#).  
Hosted by [GitHub](#) and served by [CloudFlare](#).