HeatSet Home Building Monitoring System

Developed by HeatSet (George Robinson) with support from T4 Sustainability (John Beardmore)

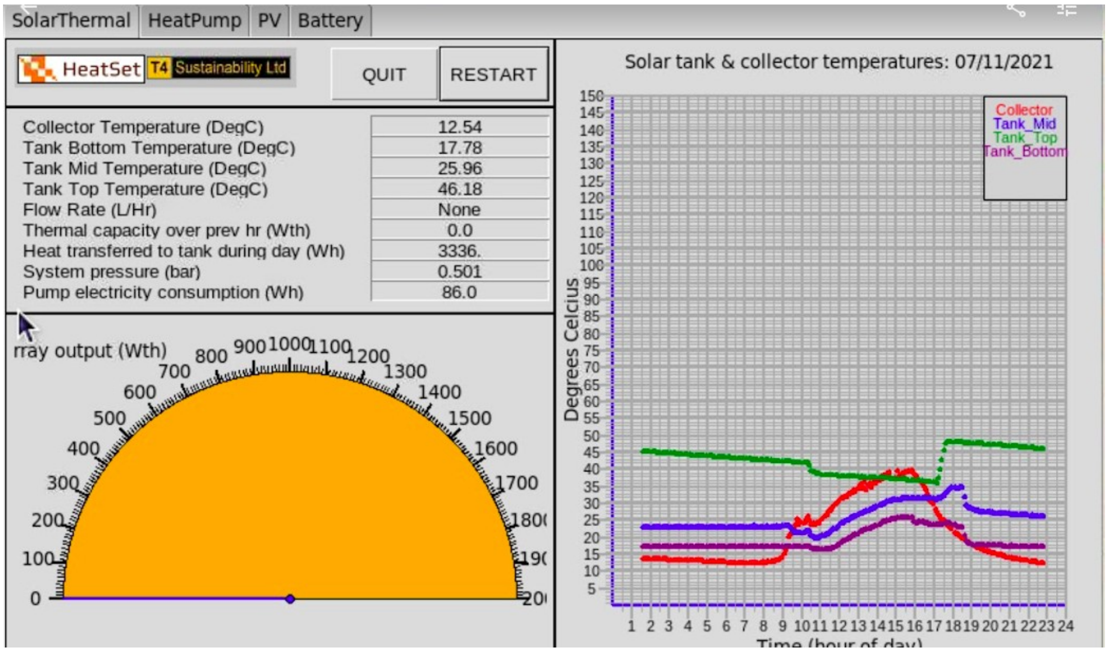


Table of Contents

Project Summary

While there is central government support for monitoring the efficiency of your heat pump (**https://www.ofgem.gov.uk/publications/essential-guide-metering-and-monitoring-service-packages-mmsp**), I wanted to create a one-stop shop to be able to monitor:

- heat generated and stored by domestic solar heating;
- heat generated and electricity consumed by a domestic heat pump;
- electricity charged and discharged by a battery; and
- electricity generated by a PV array

The system uses a Raspberry Pi and is designed to provide a graphical user interface (GUI) that can be accessed via a phone, laptop, tablet etc. capable of integrating these four areas. It also stores all of the measured data in a database on a minute-by-minute basis and exports this each hour to a spreadsheet allowing for subsequent analysis. Each year a new database is created to allow for the previous year's database to be archived.

This project requires the user to install (or have a professional install) measuring instruments for temperature, hot water flow rates, electricity consumption and system pressure. It also requires the user to build and wire a circuit board (see "PCB Assembly ReadMe.pdf") that converts the analogue outputs from the sensors installed into digital values for temperature and pressure readings, as well as monitoring for pulses from hot water and electricity meters installed.

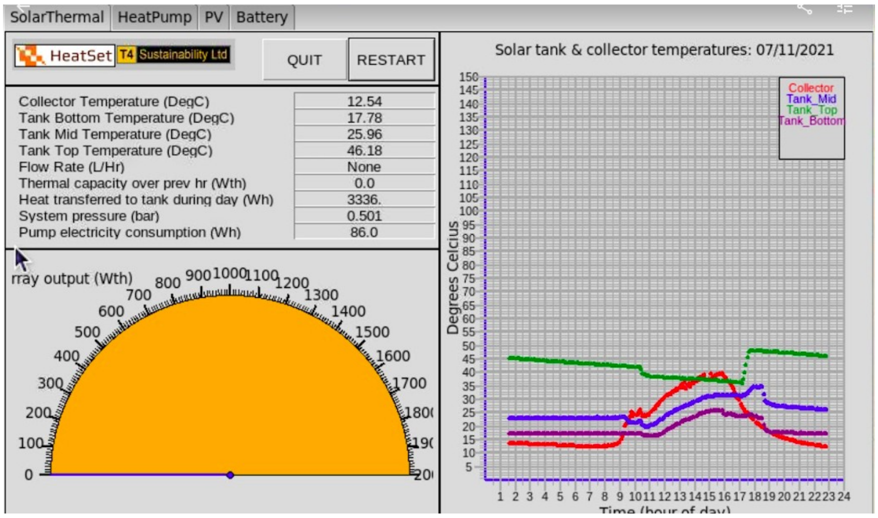
Graphical User Interface

Overview

The GUI has been developed in Python and is run on a Raspberry Pi 2 which can be bought for approximately £30. There are four tabs Solar Thermal, HeatPump, PV and Battery; however, if you are not installing all of these energy types they can be switched off – see Section "*Removing Tabs on the GUI*".

Solar Thermal

Figure 1: Solar Thermal Tab



“Collector temperature (Deg C)”
This is the temperature, in degrees Celsius, of the solar collector on your roof (or ground mounted).

“Tank bottom temperature (Deg C)”
This is the temperature, in degrees Celsius, of your hot water cylinder at the bottom. It is important to note that you will need to specify three additional ½” BSPP threaded monitoring points within the cylinder that you purchase located at the bottom, middle and top of the cylinder – see Section “*Specifying the solar thermal monitoring system*”. Understanding the stratification of heat within your cylinder is helpful information and can be surprising!

“Tank Mid temperature (Deg C)”
This is the temperature, in degrees Celsius, of your hot water cylinder at the middle.

“Tank Top temperature (Deg C)”
This is the temperature, in degrees Celsius, of your hot water cylinder at the top.

“Flow rate (L/Hr)”
This is the measured flow rate of the system over the past hour. The value is updated each 5 minutes. It should not be read as an instantaneous flow rate (i.e. it is not a measure of the current velocity and mass of the fluid being measured, rather it is the total number of litres that have been measured over the past hour).

“Thermal capacity over the past hour (Wth)”
The thermal capacity of the solar array is calculated based on the sum of Watt hours measured over the past hour (i.e. Wh divided by 1 hour = the average wattage of the array over the past hour).

“Heat transferred to tank during the day (Wh)”
This is the total heat transferred from the solar collector to the hot water tank over the duration of the day. The hot water pulse meter measures each 0.25L of hot water that has been pumped from the solar collector to the hot water cylinder. The location of the solar coil is assumed to be at the bottom of the hot water cylinder and as such the temperature difference is taken to be between the solar collector temperature and the bottom of the hot water cylinder (see Section *Setting the approximate location of the Solar thermal circuit within the cylinder* to alter the location assumption). A 25% glycol mix is assumed to avoid winter freezing, thereby lowering the specific heat capacity of the fluid (see Section *Glycol mix is not 25%* to alter the concentration of the glycol mix).

The calculation for heat transfer:

Watt hour = specific heat capacity (J/kg K) * litres measured (kg) * Delta T (K) / 3600 (s in hr)

A database query calculates the total Watt hours recorded in the current day and this is displayed in this section of the GUI.

“System pressure (bar)”
This is the pressure of the solar thermal circuit expressed in bars of pressure.

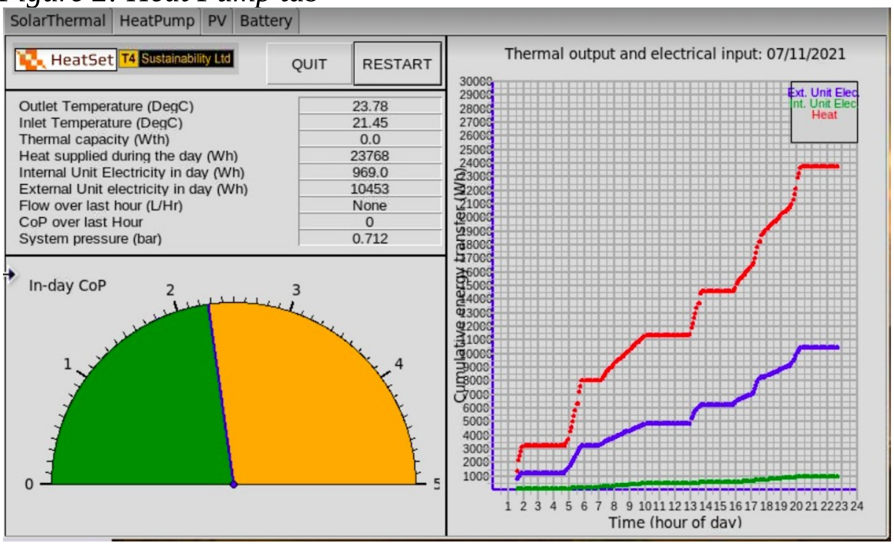
“Pump electricity consumption (Wh)”
This is the electricity consumed by the solar hot water pump during the day, expressed in Watt hours.

Dashboard gauge
The gauge on the solar thermal tab shows the thermal capacity of the array, expressed in Watts and as shown in the GUI value *"Thermal capacity over the past hour (Wth)"*.

Graph
The graph shows the temperature of the three points measured in the hot water cylinder and the solar collector (red). If you have a heat pump, this chart is additionally helpful to see the periods in the day when the heat pump increases the temperature of the cylinder (this would be expected to be only the top of the cylinder in a two coil – solar and HP – cylinder set up). For example, in Figure 1 you can see at 5pm the top of the cylinder increased from 35 degrees Celsius to 50 degrees Celsius – this is heat from the heat pump.

In the summer, if the cylinder gets to 60 degrees Celsius then many solar pumps will cease to pump and the temperature of the collector can get up to 150-200 degrees Celsius. If a glycol mix is used then this can cause the glycol to become "syrupy" if left stagnant for a period of time. As such monitoring the temperature can be helpful – e.g. to avoid this situation you could run a bath. This issue is one reason why many people opt to exclude glycol from the mix and instead either run the solar pump in winter to allow for heat to transfer from the cylinder to the collector to avoid freezing or else drain the system and refill in the spring. I believe the Renewable Heat Incentive offered through OfGem would not allow for the former set up but with it finishing in March 2022 that may be something to consider as glycol is not cheap and it is questionable just how often you would actually need to run the system in reverse to avoid freezing given the pipework should be well lagged.

Heat Pump

Figure 2: Heat Pump tab


"Outlet Temperature (Deg C)"
This is the flow temperature, expressed in degrees Celsius, from the heat pump.

"Inlet Temperature (Deg C)"
This is the return temperature, expressed in degrees Celsius, from the heat pump.

"Thermal capacity (Wth)"

The thermal capacity of the heat pump is expressed in Watts and is calculated by totalling the Watt hours measured over the past hour.

“Heat supplied during the day (Wh)”
This is the total heat transferred from the heat pump to the heating system over the duration of the day. The hot water pulse meter measures each 1L of hot water that has been pumped from the heat pump to the heating system (hot water + space heating). Temperature sensors are installed on the flow and return pipes to give the temperature differential. A 25% glycol mix is assumed to avoid winter freezing, thereby lowering the specific heat capacity of the fluid (see Section *Glycol mix is not 25%* to alter the concentration of the glycol mix).

The calculation for heat transfer:

Watt hour = specific heat capacity (J/kg K) * litres measured (kg) * Delta T (K) / 3600 (s in hr)

A database query calculates the total Watt hours recorded in the current day and this is displayed in this section of the GUI.

“Internal unit electricity in day (Wh)”
This is the total electricity consumed by the heat pump’s internal unit (immersion heater and hot water pump to circulate hot water in the heating system) during the day. There is an option to include or exclude this within the Coefficient of Performance (CoP) calculation (see Section *Including or excluding the Heat Pump’s internal unit’s electricity consumption in the CoP values shown on the GUI* to alter the default assumption, which is to exclude the internal unit’s electricity consumption).

“External unit electricity in day (Wh)”
This is the total electricity consumed by the heat pump’s external unit (compressor and fan if ASHP) during the day. This is a key value for measuring a heat pump’s efficiency as this is the energy that is paid for and which comes with a carbon content. The higher the ratio of heat supplied relative to electricity consumed, the more energy that has been successfully taken from a “free” heat source such as the air or ground.

“CoP over last hour”
This calculates the heat supplied over the last hour divided by the electricity consumed over the last hour. This differs from the gauge at the bottom of the screen which shows the in day CoP. You can get some very high readings for this as the metric is updated every 5 minutes and it is possible for the hot water pump to continue running while the compressor has ceased to operate. As such, if no heat has been supplied for 55 minutes then the heat supplied in the period between the 55th and 60th minute may be high relative to electricity consumed in that 5 minute window. As such care should be taken when looking at this value by analysing the graph where a flat period shows the heat pump having not been operating.

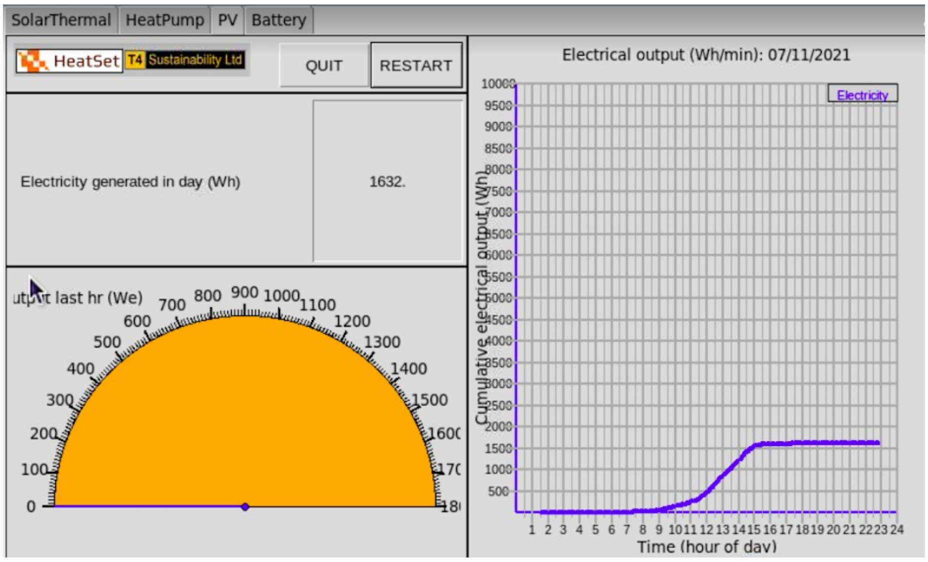
“System pressure (bar)”
This is the pressure of the solar thermal circuit expressed in bars of pressure.

Gauge
This shows the total heat supplied in the day divided by the total electricity consumed in the day by the external unit (it can be adjusted to include the internal unit electricity consumption – see Section

Including or excluding the Heat Pump's internal unit's electricity consumption in the CoP values shown on the GUI).

Graph
This shows the cumulative heat (Wh) supplied vs. the cumulative electricity consumed of both the external unit and internal unit of the heat pump. It is important to remember that the the CoP will vary with seasons (unless you have a Ground Source Heat Pump where the CoP should be more consistent across seasons). Using the spreadsheet output for the year you can evaluate the overall seasonal coefficient of performance (SCOP) of your heat pump but this is not performed in the GUI.

Photo Voltaic Array (solar panel)

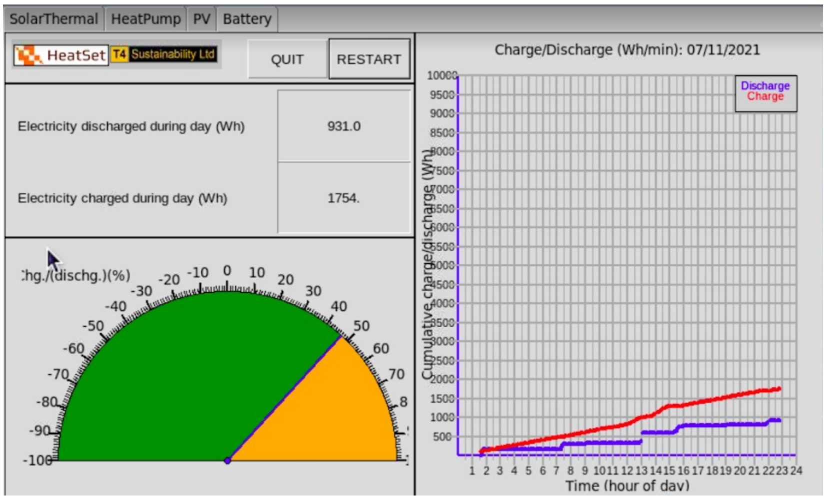


Electricity generated in day (Wh)
This is the sum of all electricity generated during the day and is a query run on the underlying database.

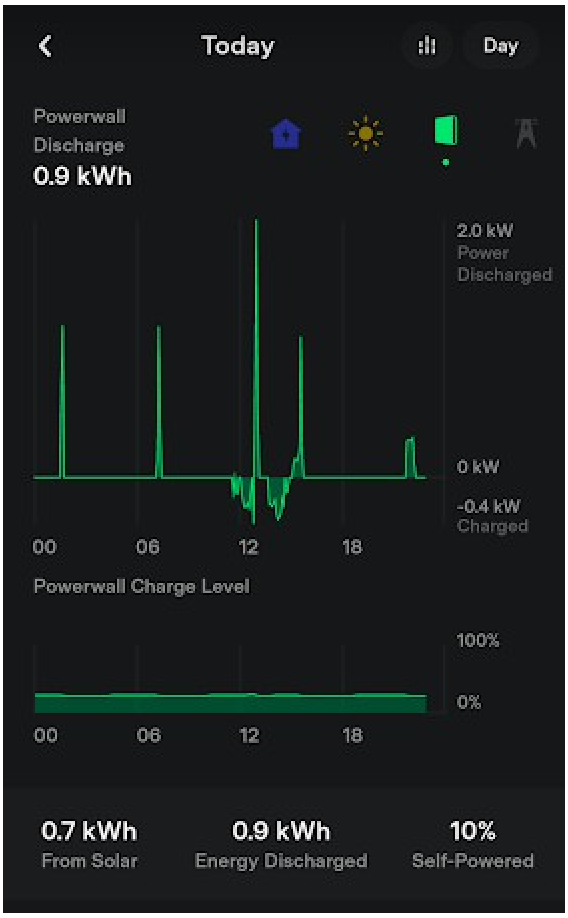
Gauge
This shows the current estimated Watts produced by the PV array that is being monitored. It is calculated by totalling the total the watt hours measured over the past hour. As such it gives the average Watt output over the past hour.

Graph
This shows the cumulative build up to the total *Electricity generated in day (Wh)* figure shown on the GUI and described above.

Battery



Electricity discharged during day (Wh)
This shows the total Watt hours discharged from the battery to your home (or exported to the grid). If you have a Tesla powerwall then the value measured here should marry up (within 0.1kWh variance) to this figure:



On the day when these images were taken, the 0.9kWh shown on the Tesla app as discharged by the battery corresponds to the 931Wh shown on the HeatSet BMS GUI above.

This is a helpful check to ensure your meter and software is working correctly.

It should be noted that the solar value does not correspond to the PV section above. This is because the Tesla display shown here is for the battery not the solar array and as such is only showing the electricity from the solar array used to charge the battery. The difference between the 1.6kWh shown in the PV section above and the 0.7kWh shown in this section is electricity that was directly used in our home.

Electricity discharged during day (Wh)
This is the sum of all electricity discharged by the battery during the day and is a query run on the underlying database.

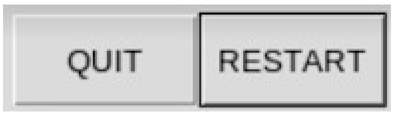
Electricity charged during day (Wh)
This is the sum of all electricity charged to the battery from the PV array (or imported from the grid) during the day and is a query run on the underlying database.

Gauge
This shows the relative percentage of electricity charged vs. discharged by the battery during the day.

Graph
This shows the cumulative build up of each of the values shown on the GUI during the day and is expressed in Watt hours.

QUIT and RESTART buttons

On each tab of the GUI there are are "QUIT" and "RESTART" buttons:



If you press the "QUIT" button it will stop all monitoring processes. You can then close the application by pressing the "X" button in the top right hand corner of the GUI if you wish to close the application. You will need to reboot the Pi for the monitoring system to relaunch.

If you wanted to do something on the Pi (e.g. export the spreadsheet to your home computer) then you can press "QUIT" to stop the processes and free up system memory. When finished you can then press the "RESTART" button to restart the monitoring processes. The restart waits 1 minute to ensure that all prior processes have completed and gives the server used to monitor pulses from the various sub-meters sufficient time to reboot to ensure a good connection is made.

System Specification

In order for your monitoring system to be a success you will need to work with your installer to ensure they understand what they need to do and where the division of labour will be. Depending on your skills and comfort level this could range all the way from you doing all the work to them doing all the work. This section sets out what needs to be done allowing you to specify (either contractually or by informal agreement) what your installer they will need to do.

Specifying the solar thermal monitoring system

The instruments used to measure the solar thermal circuit and its performance are listed below:

Instruments to purchase

- 4x 10k NTC thermistor – ½" BSPP connection: https://www.sterlingsensors.co.uk/ntc-thermistor-sensor-with-fixed-process-connection.html – c£25 each = £100

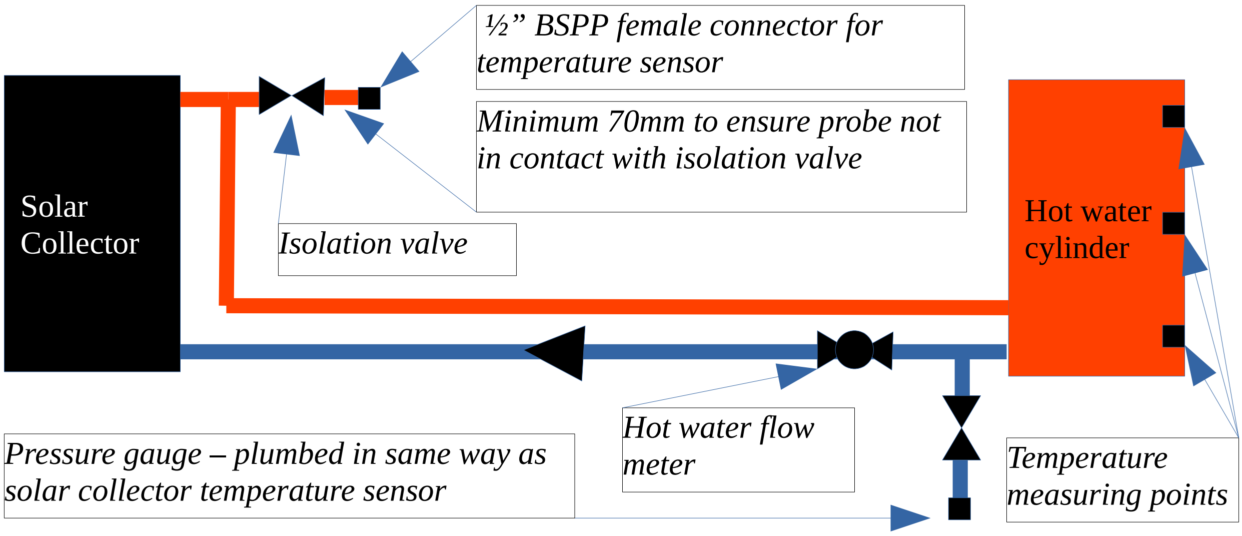
- 1x EMLITE ECA2 MID SINGLE PHASE 20-100A DIRECT CONNECTED METER C/W PULSE – **1Wh/pulse** https://www.metermarket.co.uk/product/emlite-eca2-mid-single-phase-20-100a-direct-connected-meter/ - c£35
- ¾” hot water meter with **0.25L pulse output** – I used AWE but there are a large number of options. Key is understanding the pulse output when buying and **ensuring the size is correct for your system and so must be agreed with your installer.** https://www.awe-ltd.co.uk/products/water-meter/hot-water-meter.html – c£150
- 5V G1/2” pressure sensor transducer – c£15 – the one I bought is no longer listed; however, the 10bar version here should be fine - https://www.aliexpress.com/i/4001128266089.html

See “Instrumentation ReadMe.pdf” for how the different instruments should be wired to the main circuit board.

Hot water cylinder
If you already have a hot water cylinder with a solar coil and therefore are not replacing it then you will not be able to measure the internal temperature of the cylinder. However, you can measure the temperature of the return flow to the solar collector which will still allow for heat transfer to be calculated. In such an instance the installer should plumb a temperature measuring point as close to the cylinder as is practicable. The user will need to adjust the GUI code in module “A_Initialise.py” to remove the top and middle measuring points of the hot water cylinder shown on the GUI – see Section *No temperature sensors in the cylinder*.

If you do not have a hot water cylinder and are having a new one installed then it is important to specify to your installer that you want three additional measuring points to be added – at the bottom, middle and top of the cylinder. The measuring points should be ½” female BSPP connections allowing for the NTC thermistor temperature sensors to be plumbed directly into the cylinder.

Plumbing the solar collector system for the HeatSet BMS monitoring system
In the Government’s Metering & Monitoring Service Package specification (see introduction) for the heat meter they require that the temperature sensors to be inserted in the main flow and return pipes. It is unclear to me why this is necessary (the temperature in the main flow pipe and a T-d off measuring point will not likely be materially different) as doing so makes subsequent removal and replacement of the gauges harder. Instead I would suggest that the installer, for the HeatSet BMS system T-s off from the main flow pipe from the collector to the hot water cylinder as close to the collector as is practicable. An isolating valve should be installed (allowing for subsequent removal of the sensor) and a short 15mm run from the isolation point to a female ½” BSPP connection point should be installed. The length of the 15mm pipe should be long enough such that the temperature sensor’s probe does not come into contact with the isolation point when inserted. The Sterling sensors are available in 50, 100 and 150mm lengths – I would suggest using 50mm sensors.



Wiring the hot water pump
The electricity feed to the hot water pump used to circulate the solar thermal system should be routed via the electricity sub-meter specified above. This is to enable measurement of how much pumping electricity is used. As a closed circuit, pressure drops will be a product of distance to the collector and number of turns and fittings – therefore generally it might be expected that pumping costs would be low on a domestic solar collector systems where they cylinder and collector are in relatively close proximity.

Specifying the Heat Pump monitoring system

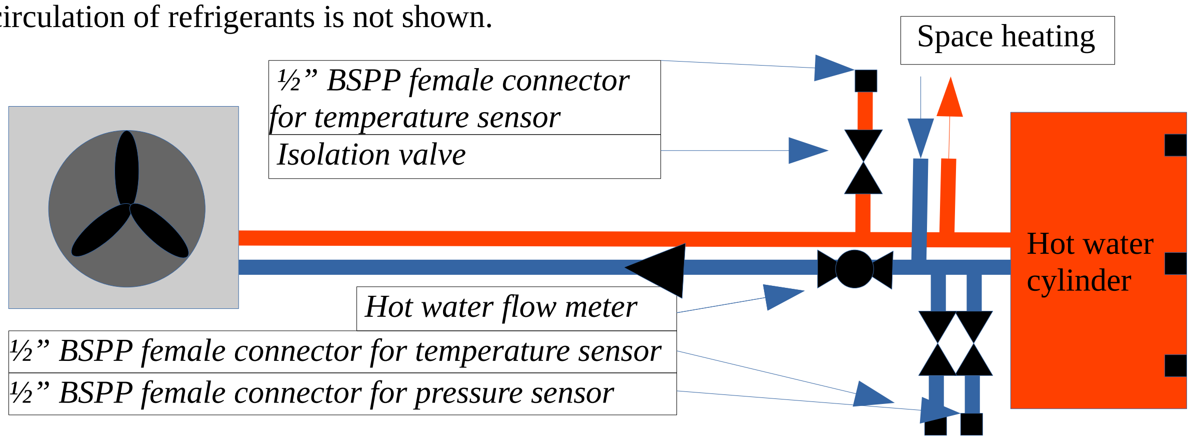
The instruments used to measure the heat pump system are set out below:

- 2x 10k NTC thermistor – ½” BSPP connection: https://www.sterlingsensors.co.uk/ntc-thermistor-sensor-with-fixed-process-connection.html – c£25 each = £50
- 2x EMLITE ECA2 MID SINGLE PHASE 20-100A DIRECT CONNECTED METER C/W PULSE - **1Wh/pulse** https://www.metermarket.co.uk/product/emlite-eca2-mid-single-phase-20-100a-direct-connected-meter/ - c£35 each = £70
- 1” hot water meter with pulse output – I used AWE but there are a large number of options. Key is understanding the pulse output when buying and ensuring the size is correct for your system and so should be agreed with your installer. The HeatSet BMS default pulse value is **1L/pulse** https://www.awe-ltd.co.uk/products/water-meter/hot-water-meter.html – c£200
- 5V G1/2” pressure sensor transducer – c£15 – the one I bought is no longer listed; however, the 10bar version here should be fine - https://www.aliexpress.com/i/4001128266089.html

See “Instrumentation ReadMe.pdf” for how the different instruments should be wired to the main circuit board.

Hot Water Cylinder
If you are not installing a solar thermal system as well as your heat pump system then you might none-the-less consider monitoring your hot water cylinder’s temperature at low, middle and top measuring points. If that is of interest (at the very least it’s handy to know if your cylinder is hot enough to meet shower needs etc.) then please refer to the solar thermal section above regarding specifying measuring points within the cylinder and Section *Heat Pump with hot water cylinder monitoring but not solar thermal* to recalibrate the Solar Thermal tab of the GUI to show data relevant only to the cylinder.

Plumbing the heat pump HeatSet BMS monitoring system
The schematic below assumes that a monobloc heat pump system has been installed and as such the circulation of refrigerants is not shown.



If hydraulic separation is required between the heat pump and the internal hot water circuit then the sensors and flow meter should be on the heat pump side of the heat exchanger ideally within the building envelope to reduce cable lengths.

Note the hot water flow meter can be on the flow side of the circuit as such meters are rated up to approximately 80-90 DegC which most heat pumps (except perhaps CO2 based heat pumps) would never reach. I would say generally better to install on the return but that may be influenced by space constraints and views of your installer.

Electrical sub-metering of the heat pump
Two sub-meters are used and the electrical wiring to the heat pump:

- A sub-meter for the external unit. This is measuring the electricity used by the compressor and fan (if air source heat pump is used) within the heat pump unit;
- A sub-meter for the internal unit. This is measuring the electricity used to run the hot water pump, cylinder immersion heater and control system.

If you do the MMSP package they require that the wiring for these two items are brought together into a single electricity meter that they specify. While commendable, I feel that it is important to understand the consumption of each component separately and then combine the values within the software. As such if you are also intending on installing the MMSP package then you will need to install your own two sub-meters at a point prior to the MMSP electricity meter.

Specifying the PV sub-metering

The instruments used to measure the heat pump system are set out below:

- 1x EMLITE ECA2 MID SINGLE PHASE 20-100A DIRECT CONNECTED METER C/W PULSE - **1Wh/pulse** https://www.metermarket.co.uk/product/emlite-eca2-mid-single-phase-20-100a-direct-connected-meter/ - c£35

See "Instrumentation ReadMe.pdf" for how the different instruments should be wired to the main circuit board.

Wiring the electricity sub-meter

If you are intending on installing a battery to complement your PV array then it is important to ensure that the sub-meter is installed at a point prior to the battery control which may opt to charge the battery, export to grid or self-supply to your home.

Specifying the Battery sub-metering

The instruments used to measure the heat pump system are set out below:

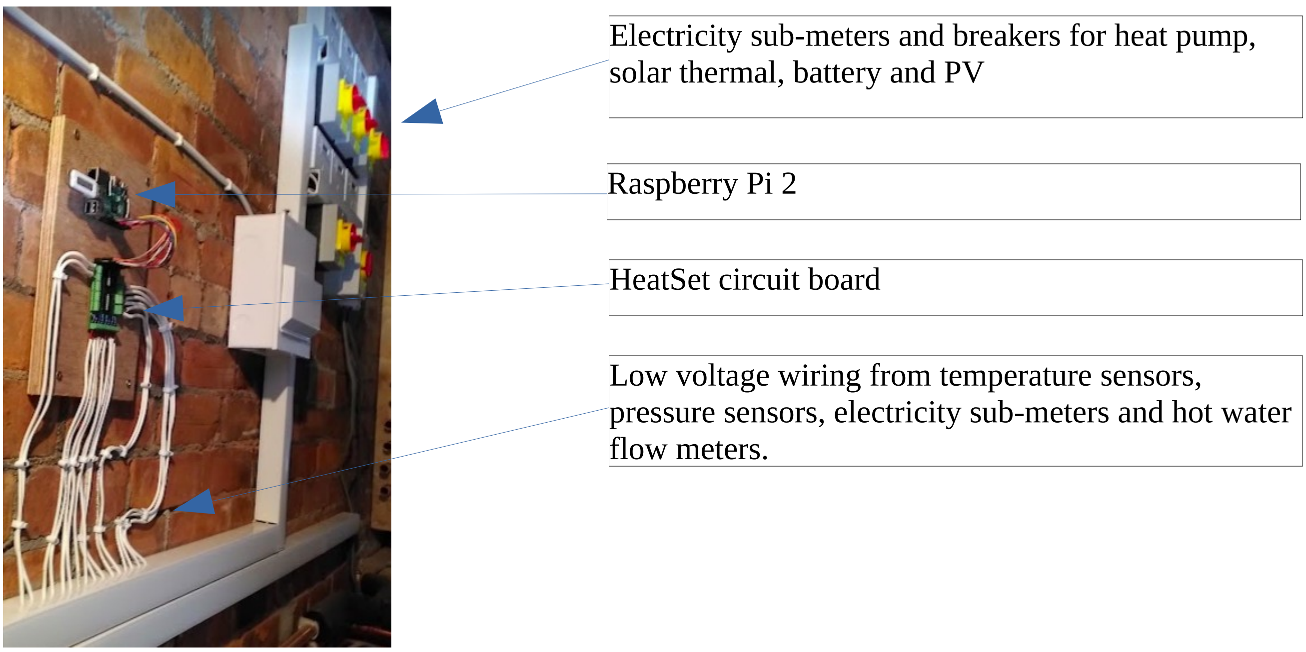
- 2x EMLITE ECA2 MID SINGLE PHASE 20-100A DIRECT CONNECTED METER C/W PULSE – **1Wh/pulse** https://www.metermarket.co.uk/product/emlite-eca2-mid-single-phase-20-100a-direct-connected-meter/ - c£35 each = £70

See "Instrumentation ReadMe.pdf" for how the different instruments should be wired to the main circuit board.

Wiring the electricity sub-meter
Two sub-meters are required as the meters are calibrated to measure only electricity in one direction. As such the meters should be installed with one in reverse. The meter wired for flow to the battery should be marked as "Battery Charge". The meter wired for flow from the battery should be marked as "Battery Discharge". This is particularly relevant where there is a division of labour between the installer and HeatSet BMS monitoring system.

Specifying general data and power requirements

Each home is different but generally the key pieces energy equipment will be routed through a loft, cupboard or some out of the way room. You will need to have some sort of board for the various electricity sub-meters and the flow meters and temperature sensors should be wired to terminate at a single point: your Raspberry pi. Make sure you plan your final cabling – it is not good if you have installed all the sub-meters next your consumer unit in the hall and cylinder is in the loft. If you haven't run data cables from your hall then it is going to be hard to get the data. Ideally it should be all located as close as possible.



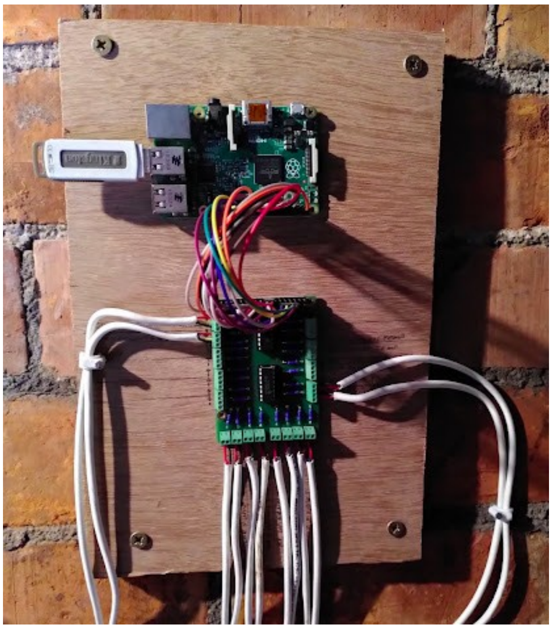
The Raspberry Pi requires a conventional power point – if this is not already in place then this will need to be installed by an electrician.

A Raspberry Pi 2 does not have wireless internet. You can either install a USB component to make the Pi wireless (but make sure you can reach the router!); run Cat5 or Cat6 from the room with your router to the area in your home where you will locate your Raspberry Pi (appropriate if you are doing a major retrofit); or else use two broadband over power line adaptors to piggy back the electricity supply point for your Pi to also transfer data (I'd just get an ethernet only adaptor given you won't actually be using the wireless with the Raspberry Pi 2).

Setting up the HeatSet BMS system

Overview

A HeatSet circuit board has been developed to interface with a Raspberry Pi 2, although later versions would also work (true as at the time of writing November 2021)[1]. The HeatSet circuit board is connected to the Pi and the various sensors and pulse meters are wired to the circuit board:



The wiring to each of the terminal blocks on the circuit board is set out in the document “Instrumentation ReadMe.pdf”.

The circuit board needs to be built by you. Full instructions of how to order the plate, the parts needed and how to assemble it are provided in the document “PCB Assembly ReadMe.pdf”. While the circuit board is technically not a Raspberry Pi “hat”, it is designed with the exact number of header pins as the Pi (20x20) to allow for the pins to be directly connected without the need for individual jump cables.

Prior to installing the Raspberry Pi and connecting it to the HeatSet circuit board it is necessary to first calibrate the Python code to reflect the instruments you have actually installed which may slightly differ to those recommended (See “Instrumentation ReadMe.pdf”). You then need to set up the Raspberry Pi so that it will launch the code on booting as well as store the data on a USB stick – see document “Raspberry Pi Set Up ReadMe.txt”.

What you’ll need to buy

- A Raspberry Pi 2 with power cable

1 If future Raspberry Pi’s have a different layout of GPIO (the header pins on their board) then the HeatSet circuit board would not easily be connected to the Pi (individual jump cables would be required). As such it is advisable to stick with a Raspberry Pi 2 if you can or else confirm that the version of Pi you are using has GPIO pin layout consistent with the Raspberry Pi 2.

- An SD card for the Pi
- A USB memory stick (for the data logging)
- A 20x20 pin female-female connector to connect the Pi to the HeatSet circuit board (careful which way you connect it – the HeatSet circuit board has the same header pin layout as the Pi)
- Low voltage cable to do the wiring to the sensors – see *PCB Assembly ReadMe.pdf* for suggestions – you will need a mix of 3 core and 2 core cabling.
- A box of 2 way Wago clips (or go old school terminals if you prefer) to connect the low voltage cable you run to the sensors.
- An ethernet cable to connect your Pi to either your Cat5/6 cabling or powerline connector
- The HeatSet circuit board – see "PCB Assembly ReadMe.pdf" for all necessary components.

Building the Circuit Board

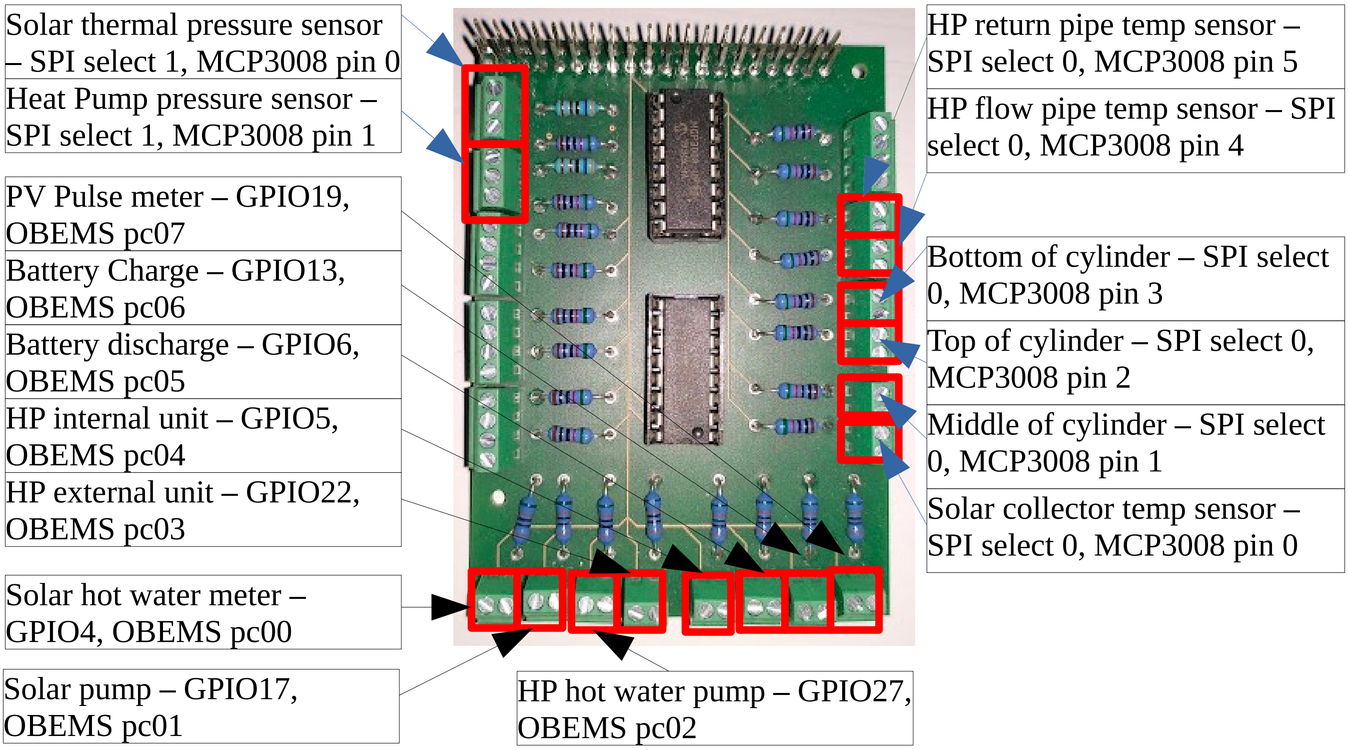
A detailed document is provided that sets out all of the parts necessary to assemble the board - "PCB Assembly ReadMe.pdf". The board itself can be ordered online from a PCB printing company that accept kicad drawings – I've provided a link to a company I used in Germany but there are plenty of UK based companies – you just send the company the schematic "PCB BMS (printed v2).kicad_pcb". Key information you will need provide the PCB printer:

- Number of layers: It is a 4 layer board
- Select "marking print" to "yes"
- Colour: I'd select green (it's generally cheaper but there are some cool colours if you're into that)

If a soldering iron sounds too daunting for words then do please get in touch (grobinson105@gmail.com) and I will see how much time I have available and can build it for you at cost but roughly it would be about £50.

Wiring the instruments to the Circuit Board

The document "Instrumentation ReadMe.pdf" sets out how each instrument should be wired to the HeatSet circuit board. Plan your wiring to avoid cross-overs, e.g. starting with the solar thermal pressure sensor (top left of the HeatSet circuit board) and finishing with the heat pump return pipe temperature sensor:



Setting up the Raspberry Pi

Before you mount the Raspberry Pi in your "energy centre" and connect it to the HeatSet circuit board it is advisable to do the set up when it is still easy to physically access and connect to a screen. Before you are able to complete the Raspberry Pi set up – "Raspberry Pi Set Up ReadMe.txt" – you need to first be able to access your Pi via a VNC connection. This allows you to securely connect to the device from a mobile phone or laptop from anywhere in the world with internet access.

Step 1: Boot your Raspberry Pi

Step 2: Select the Raspberry Pi icon > Preferences > Raspberry Pi Configuration

Step 3: In the interfaces tab enable VNC and SPI press ok.

You may need to reboot

Step 4: You will see a VNC icon in the top right of the screen this is the VNC Server. Follow the instructions to set up your own home account.

Step 5: With the Pi still on, **connect it to the internet,** boot your personal computer and download the VNC Viewer application for the operating system you use: https://www.realvnc.com/en/connect/download/viewer/

Step 6: Open the VNC viewer application on your home computer and log into your VNC account

Step 7: Open your router via its IP address and determine the IP of your Pi

Step 8: Enter the Pi's IP address into the VNC Connect address bar and connect to your Pi

Step 9: **Complete the steps in the document “Raspberry Pi Set Up ReadMe.txt”**

Calibrating the GUI for your specific system

While there is not a set up wizard (maybe one day!) it is possible to calibrate the Module “A_Initialise.py” to reflect the system you install. For example, you may not install all four elements of the GUI. If you install slightly different instruments than those specified (e.g. if the pulse values are not exactly the same as those specified in relevant sections) then you should follow the specific instructions within the document “Instrumentation ReadMe.pdf” where variables in other modules will need to be amended.

Removing Tabs on the GUI

The following variables in module “A_Initialise.py” are used to switch on or off the four areas that can be monitored by the HeatSet BMS:

- Solar thermal system: **boolSolar**
- Heat pump system: **boolHeatPump**
- Photo voltaic solar array: **boolPV**
- Electrical battery (e.g. Tesla powerwall): **boolBattery**

Simply enter “False” in the place of “True” against any of these variables and the tab will be removed from the GUI.

Heat Pump with hot water cylinder monitoring but not solar thermal

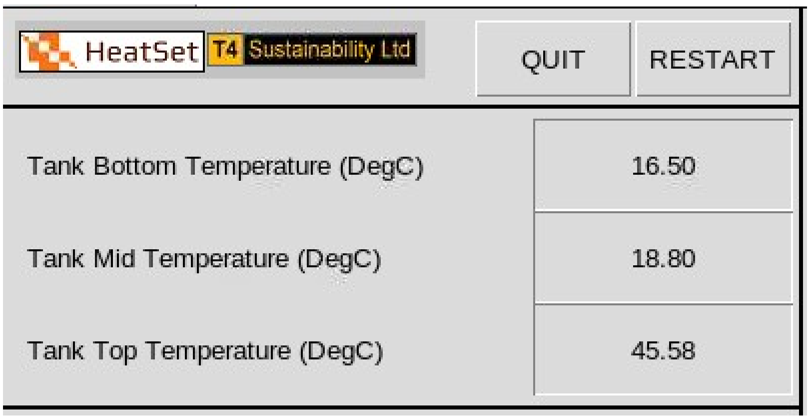
Personally I am a huge advocate for solar thermal hot water heating – it is as close to zero carbon as you can get and will reduce the work that your heat pump will have to do, heating the hot water cylinder up to around 30 degrees Celsius in the winter (requiring only a 20 degrees Celsius temperature increase by the heat pump) and comfortably heating it to a consistent 50-60 degrees Celsius on any sunny day from late spring to early autumn. However, I appreciate that some people see the system’s inability to supply the required hot water demand year-round could be seen to make it an investment not worth making.

A heat pump system will typically only work well for hot water supply if a hot water cylinder is installed. This is because a heat pump, unlike a combination boiler, does not have the thermal capacity to meet instantaneous hot water demand. For a typical home in the UK you need about 4-8kW of thermal capacity to meet most of your space heating requirements and the heat pump is sized to meet that demand. The reason a 28kW combination boiler (or that kind of size) is installed is to deal with the very high capacity requirement that instantaneous hot water supply demands. By removing that capacity, compensation is needed and that comes in the form of a hot water cylinder. Having a cylinder has many advantages now (e.g. allowing for time of day tariff selection without compromising on your hot water needs) and may present further advantages in the future (e.g. the possibility for integrated smart controls whereby wider network operators could switch off heat pumps for short blocks of time and a payment be made to such customers for that service). As such it may be that you want two tabs – one to monitor the cylinder and one to monitor the heat pump.

Step 1: Open module “A_Initialise.py”

Step 2: Amend variables **boolPV** and **boolBattery** to False (note Python requires the “F” to be capitalised”)

Step 3: Amend list **lstGUISensorNoADJ** **in the else: statement** from [0, 3, 1, 2, 20, 18, 19, 4, 21] to [3, 1, 2] (the values in the list are the ID numbers of the solar thermal dictionaries used for each GUI data point). This will result in the GUI showing tank bottom, mid and top temperatures only:



Step 4: Remove the solar collector’s temperature shown on the graph. Find the dictionary collection: “**dictCollectorSensor**” and amend the “**Plot_Values?**” variable from “True” to “False”.

Step 5: Amend the tab name to “Cylinder”. Open module “B_GUI.py”. Find the row of code that reads “self.TAB_CONTROL.add(self.Solar_Tab, text='**SolarThermal**')” (approximately row 285) and change the text ‘SolarThermal’ to ‘Cylinder’. Make sure to retain the single inverted commas on either side of ‘Cylinder’.

If you want to remove the Solar Thermal Gauge on the GUI

You will still have the solar thermal gauge at the bottom left of the screen which will always show 0. It is be possible to remove it but I’ve not fully tested the method set out. If this is what you want to do then you will need to hash out the code (i.e. put a “#” symbol in front of the code) in “B_GUI.py”:

- #self.Solar_Gauge = cht_plt.GUI_gauge(dictInstructions['Solar_Inputs']['Gauge_params'], self.frmSolarSYS)

You will then need to open module “D_Database.py” and hash out the code approximately in row 610:

- #HeatSet_DB.Solar_Gauge(dictInstructions)

I believe this is all that would be needed to be done.

Changing a graph’s y-axis scale

I have calibrated each graph’s y-axis to reflect the upper limits I expect the system to report for my home; however, that does mean that some days values will be “off the chart”. The y-axis is not dynamic (perhaps to be coded one day) and as such you will need to get a feel for your home and adjust the axes to values that suit your home and not need to touch them again. You will still have the values shown as figures on the left of the screen and as such nothing is lost if values are off the chart but it’s nice when a whole day’s worth of data is shown graphically.

Each of the four charts in the GUI (hot water cylinder in the Solar Thermal tab, heat pump, PV and Battery) have their own sections in module “A_Initialise.py” for each of the charts:

Solar Thermal / Hot Water Cylinder

Variable **Solar_Graph_y_max** has a default value of 150 degrees Celsius (as this is the rated maximum of the temperature sensor used). If for example you installed Viessman evacuated tubes which have a smart coating that increases their reflectivity the higher their temperature gets then, assuming they work correctly you might want to reduce this upper limit to say 100 degrees Celsius. If you do adjust this variable then you should also adjust variable **tm_Solar_Graph_y_count**. This variable is the number of minor tickmarks on the y-axis. If you reduced the upper limit down from 150 to 100 then you would also want to reduce the tickmark count down in a similar way.

You can also use this variable (**tm_Solar_Graph_y_count**) to reduce the number of lines on the graph. Let’s say you reduced the upper limit to 100 degrees Celsius (**Solar_Graph_y_max**) but now wanted a tick mark only every two degrees (currently it is on every degree) then you would amend variable **tm_Solar_Graph_y_count** to 50.

The major tickmark variable is **tm_Solar_Graph_y_major** and is set to 5. What this means is that for every 5 minor tickmarks a slightly bolder line will be shown and a value will be shown on the y-Axis.

Heat Pump

Please read the *Solar Thermal / Hot Water Cylinder* section above. The equivalent variables for the heat pump graph are as follows:

- Upper limit of the chart (Watt hours) – **HP_Graph_y_max** (default 30kWh)
- Number of minor tickmarks – **tm_HP_Graph_y_count** (default 60 – if changing make sure it divides cleanly into the Watt hour upper limit you set)
- Major tickmark frequency – **tm_HP_Graph_y_major** (default 2 which means that a major tickmark is shown ever 1kWh – 30,000 / 60 * 2)

PV electricity generated

Please read the *Solar Thermal / Hot Water Cylinder* section above. The equivalent variables for the PV graph are as follows:

- Upper limit of the chart (Watt hours) – **PV_Graph_y_max** (default 10kWh)

- Number of minor tickmarks – **tm_PV_Graph_y_count** (default 20 – if changing make sure it divides cleanly into the Watt hour upper limit you set)
- Major tickmark frequency – **tm_PV_Graph_y_major** (default 1 which means that a major tickmark is shown ever 500Wh – 10,000 / 20 * 1)

Battery electricity charged and discharged

Please read the *Solar Thermal / Hot Water Cylinder* section above. The equivalent variables for the battery graph are as follows:

- Upper limit of the chart (Watt hours) – **BAT_Graph_y_max** (default 10kWh)
- Number of minor tickmarks – **tm_BAT_Graph_y_count** (default 20 – if changing make sure it divides cleanly into the Watt hour upper limit you set)
- Major tickmark frequency – **tm_BAT_Graph_y_major** (default 1 which means that a major tickmark is shown ever 500Wh – 10,000 / 20 * 1)

Changing a gauge’s upper limit and tickmarks

All variables relating to the gauges shown on the GUI are within module “A_Initialise.py”.

Solar Thermal Gauge

The solar thermal gauge show the average Watt thermal capacity of your solar thermal array over the past hour. 2kWth is the default upper limit but it is possible that some homes will opt to install even more to contribute to winter space heating (which would be pretty awesome!) and manage the risk of summer overheating either through adjustable shading, larger thermal storage or use panels that change their reflectivity depending on their temperature.

The variable **gauge_max_Solar_Gauge** is the variable that sets the upper limit of the gauge and is default at 2000 Watts (i.e. 2kW). The variable **tm_Solar_Gauge_Count** sets the total number of minor tickmarks that will be shown on the gauge – the default is 200 (i.e. each minor mark represents 10W). The variable **tm_Solar_Gauge_Major** sets the major tickmark marker and its default value is 10 – i.e. for every 100W a major marker will be shown and so will a value on the gauge (2000 / 200 * 10).

Heat Pump Gauge

Please read the *Solar Thermal* section above. The equivalent variables for the Heat Pump gauge are as follows:

- Upper limit of the gauge (in day CoP) – **gauge_max_HP_Gauge** (default 5.0 CoP)
- Number of minor tickmarks – **tm_HP_Gauge_Count** (default 50)
- Major tickmark frequency – **tm_HP_Gauge_Major** (default 10 which means that a major tickmark is shown every 1.0 in day CoP – 5/ 50 * 10)

PV Gauge

The PV array is the most likely to vary home to home as we will put more or less panels on our roof depending on the space available. The variable **PVArrayMaxOutputW** in line 45 of module "A_Initialise.py" is the combined rated output of the PV array you install. For our home we have 5x 360 Watt panels which is why the default is 1800. If you adjust this variable to your home's array rated value then the PV Gauge will update accordingly – please ensure that if divided by 10 it will give a whole number (i.e. round to nearest 10 Watts).

Please read the *Solar Thermal* section above. The equivalent variables for the PV gauge are as follows.

- Upper limit of the PV gauge (Watts) **– gauge_max_PV_Gauge** (set to variable **PVArrayMaxOutputW**)
- Number of minor tickmarks **– tm_PV_Gauge_Count** (default is the array output divided by 10)
- Major tickmark frequency **– tm_PV_Gauge_Major** (default 10 which means that a major tickmark is shown every 100 Watts – 1800/ 180 * 10)

Battery Gauge

Please read the *Solar Thermal* section above. The equivalent variables for the Battery gauge are as follows:

- Upper limit of the gauge (100% charge) – **gauge_max_BAT_Gauge** (default 100%)
- Lower limit of the gauge (100% discharge) **– gauge_min_BAT_Gauge** (default -100%)
- Number of minor tickmarks **– tm_BAT_Gauge_Count** (default 200)
- Major tickmark frequency **– tm_BAT_Gauge_Major** (default 10 which means that a major tickmark is shown every 10% of charge or discharge – (100 less -100)/ 200 * 10)

Glycol mix is not 25%

Solar Thermal

In the UK the number of instances when the liquid within the solar thermal array is at risk of dropping below freezing is very limited. This is particularly true of evacuated tubes where there is less volume of water outside of the building's envelope compared to a flat plat system and that liquid is relatively well insulated. Even with flat plat systems there are techniques that can be used to remove the risk of freezing such as draining the system in the Autumn and refilling in the spring, or having the solar pump circulate the system if outdoor temperatures drop below freezing (i.e. take heat from the hot water cylinder).

While the last technique is not allowed under the Government's Renewable Heat Incentive scheme there is quite a strong argument for that kind of system as the specific heat capacity of the fluid circulating is impaired as more and more glycol is added. Pure water has a specific heat capacity of approximately 4.2kJ per litre of water but about 3.9kJ per kg of water with a 25% glycol mix. That is approximately 7-8% less heat capacity than pure water. Think of all the days in the summer, spring and autumn where the solar thermal array is operating well – we are transferring approximately 8% less heat to the hot water cylinder than we otherwise would have had we not introduced glycol to the circuit. This is just to preserve the solar thermal array from perhaps 10 or so nights in the year – reversing the solar pump to stop that happening for a few hours a year would save money (glycol is not cheap) and energy.

As such you may well want to install a system that is glycol free. The default assumption is a 25% glycol mix. The Solar Thermal glycol mix assumption is found in module "A_Initialise.py" variable **glycol_mix**. Enter the percentage as a whole number – i.e. to the closest percentage multiplied by 100 (e.g. 45% or 0.45 would be expressed as 45).

Heat Pump

As we are measuring the heat transfer from the heat pump to the hot water cylinder and space heating requirements we are concerned with the level of glycol in the flow and return from the heat pump to the home. A monobloc air source heat pump will typically have a glycol mix in the fluid circulating between the heat pump and home to reduce the risk of the liquid in the pipes exposed to the elements from freezing. A split system (not common in the UK) would allow for the circulating fluid not to have glycol as it is the refrigerant that is exposed to the elements in a split system, not the hot water circuit.

If you have a ground source heat pump then it is possible that there is no glycol in the fluid used to transfer heat to the home as the unit will likely be located somewhere in your home where the risk of freezing should be very low. In such instances it may be appropriate to adjust the heat pump's assumed glycol mix. If you are unsure discuss with your installer mindful that we are measuring the temperature difference of the internal hot water circuit not the ground loop!

To change the heat pump's hot water circuit glycol mix assumption the variable is **HP_glycol_mix** in module "A_Initialise.py". Enter the percentage as a whole number – i.e. to the closest percentage multiplied by 100 (e.g. 45% or 0.45 would be expressed as 45).

Including or excluding the Heat Pump's internal unit's electricity consumption in the CoP values shown on the GUI

My personal view is that the electricity consumed by the internal unit of the heat pump (i.e. the immersion heater, controls and hot water circulating pump) is something that needs to be monitored but not something that should be included in the Coefficient of Performance calculation relating to the heat pump. Why? I hear you say. Take a condensing gas boiler. Do we calculate its efficiency by overlaying the electricity cost of circulating the hot water around our home? No we don't. The reason we don't is we want to specifically understand the boiler's conversion efficiency from natural gas to useable heat. In the same way when I'm monitoring a heat pump's performance I want to really focus on the heat pump's ability to take heat from the target source (primarily ground, air or water) relative to the electricity needs to run the compressor. The fact that I may need to top up my hot water tank once a week to minimise legionella risk isn't actually a reflection of the heat pump's performance it is a reflection of the overall system performance.

Depending on your preference you can set the calculations to include or exclude the electricity supplied to the heat pump's internal unit for CoP values by changing the variable **Include_internal_unit_in_COP** from "False" to "True" in module "A_Initialise.py".

Setting the approximate location of the Solar thermal circuit within the cylinder

It is hard to imagine a solar thermal system where the solar coil within the hot water tank is not at the bottom of the cylinder (the default assumption). However, I am not a trained hot water plumber and perhaps there could be situations where that is the case.

The heat transfer calculations for the solar thermal system are based on the temperature difference between the solar collector on the roof and the closest measured position within the cylinder to the return pipe of the solar collector.

In order to change which temperature measurement is used to establish the temperature difference (delta T) variable **solar_coil_loc** should be adjusted as follows:

- Enter "0" (without the inverted commas – i.e. as an integer) for bottom of the cylinder
- Enter "1" for middle of the cylinder
- Enter "2" for top of the cylinder's

No temperature sensors in the cylinder

If for whatever reason you have not been able to or wanted to install temperature sensors within the hot water cylinder then it will be necessary to install a temperature sensor on the return pipe to the solar collector adopting the method set out in Section *Plumbing the solar collector system for the HeatSet BMS monitoring system* for the pressure gauge.

When wiring the sensor to the Pi, assuming you have not altered variable **solar_coil_loc** (see previous section) then you should wire the sensor as if it were the "Bottom of cylinder" temperature sensor (see Section *Wiring the instruments to the Circuit Board*).

It would be helpful to then amend the GUI to reflect this set up:

Step 1: Open module "A_Initialise.py"

Step 2: Amend the list **lstGUISensorNoADJ** **<u>in the else statement</u>** from [0, 3, 1, 2, 20, 18, 19, 4, 21] to [0, 3, 20, 18, 19, 4, 21]

Step 3: Find the dictionary collection **dictBotTankSensor**

Step 4: Amend the GUI_Label variable from 'Tank Bottom Temperature (DegC)' to 'Return temperature (DegC)'

Step 5: If you are making the change prior to having ever run the software then you can also change the SQL_Title from 'Tank_Bottom_Temp_DegC' to 'Return_Temp_DegC'. However, only do this if you have never run the software. If you change this having run the software then the underlying database will have been created and there will be a field heading "'Tank_Bottom_Temp_DegC'" which it could no longer find. You could either archive that version of the database and accompanying spreadsheets (on reboot a new version will be created with the updated field name) or alter the field heading in SQLite3 via a command code (the table is called ‘SOLAR’). This is not my area of expertise.

Changing the screen size, font and font size

The size of the screen has been set to broadly fit onto a smart phone screen when turned on its side. The VNC viewer allows you (on the phone app version) to zoom in and zoom out and as such I have not found it particularly necessary to change the actual size of the screen. However, if you are using a laptop as your standard device then you may want to increase it or decrease its size.

All windows within the viewer are built based on proportional sizing. In module “A_Initialise.py” The variable **lngScreenWidth** sets the number of pixels that the screen is wide (default 800). The variable **lngScreenHeight** sets the number of pixels that screen is high (default 480). If you keep that approximate ration (480:800) then the screen should look fairly ok.

If you want to change the font then the variable is **strFont** in module “A_Initialise.py” and the default is "Helvetica".

If you want to increase the font size then the variable is **bytFontSize** in module “A_Initialise.py” and the default is 10.

Acknowledgements

For a long time I worried about Python’s ability to register voltage changes on header pins that were used for registering pulses from the electricity sub-meters and hot water flow meters. While testing suggested that it would be up to the task, the more processes I added relating to evaluating the pulses, the more I concluded that I needed a dedicated server with the sole task of constantly monitoring voltage changes across the 8 header pins used for various pulse meters.

Our heat pump, solar thermal, PV and battery installers were a Derby based company called T4 Sustainability (http://www.t4sltd.co.uk/). I had picked them not only because it was clear that they are fantastic engineers but because their managing director, John Beardmore, is a strong advocate of open source Internet of Things and had developed T4Sustainability’s own open source monitoring system oBeMs (https://github.com/johnbeardmore/oBeMS). With his help we recompiled the C++ module “ObemsPulseServer” to add an additional pulse meter and realign the Pi’s header pins to match up to the HeatSet circuit board. This now acts as a server that is always “listening” for pulses and the Python software does a handshake with the server every few seconds to check if there have been any pulses recorded since the last handshake.

I am very grateful to John and the T4 Sustainability team (particularly Ed Sears) for all their work and help. Never once did they suggest that what I was doing was bonkers and went out of their way to help me develop this system. Thank you all so much.

I would also like to thank my partner, Julia, who has put up with this for the past 3+ years. Thank you.