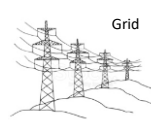


Flukso V2 & V3 Intergation Into Custmom Rules



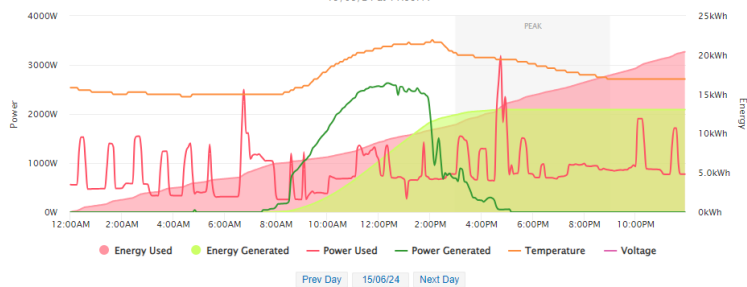
Flukso Portal No More
What Now



You have 1 unread message
[Add Output](#) | [Your Outputs](#) | [PV Ladder](#) | [Statistics](#) | [Live Outputs](#) | [Teams](#) | [Favourites](#) | [Settings](#) | [Community](#) | [Donations](#) | [Help](#) | [Logout](#)

Live Production - StarLight 4.862kW

15/06/24 at 11:55PM



[Live](#) | [Daily](#) | [Weekly](#) | [Monthly](#) | [Yearly](#) | [Analyse](#) | [Map](#) | [Favourite](#) | [Download](#) | [Insolation](#) | [Customise](#) | [Delete](#) | [Minimise](#) | [Refresh Off](#)

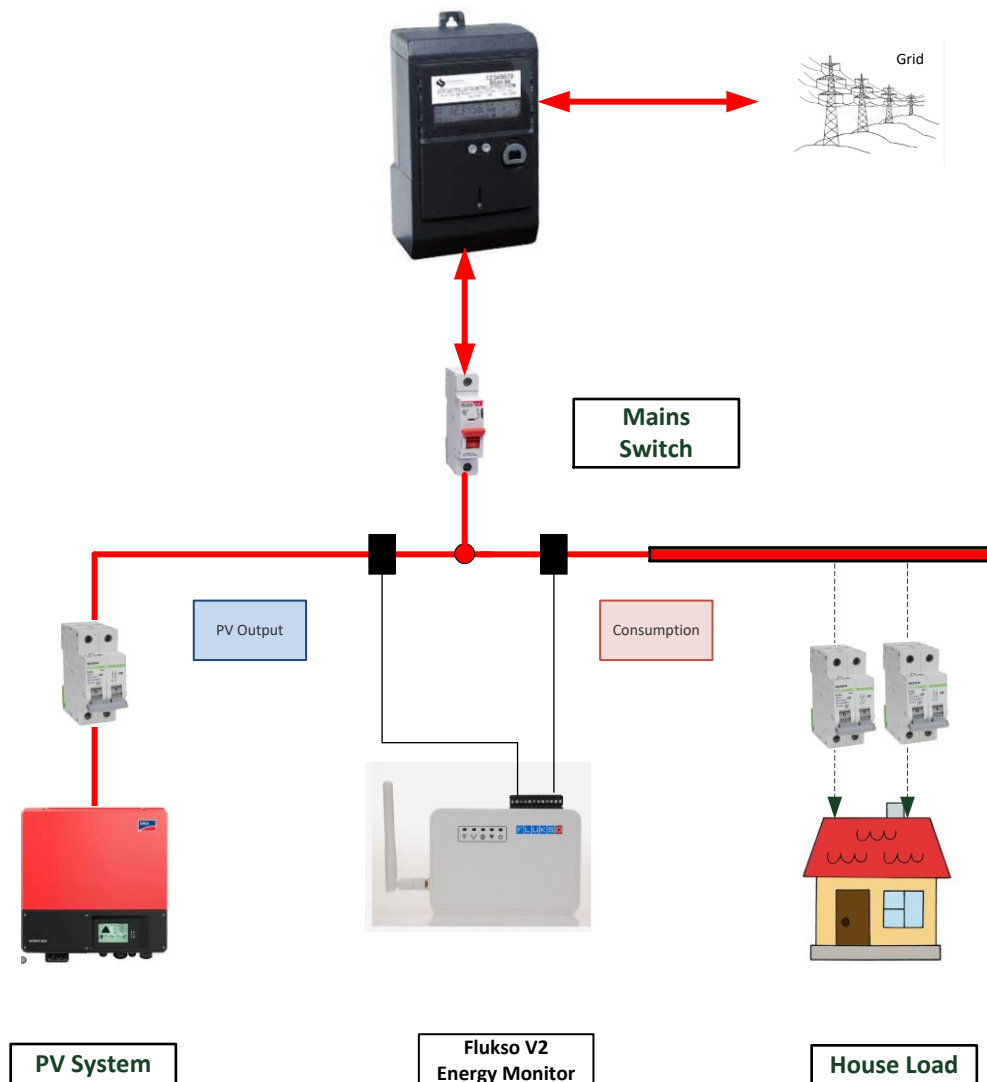
Generation **5823** of **49821** ▼ **2** · **15** Followers · **17** Following · **14** Photos · **8** Teams · **52.8** MWh · **56.5** T CO₂ · **5** Systems

Target **220%** **\$5.15** ▼ **13,061** Wh · **0** W · **2,633** W Peak · **20,424** Wh · **776** W · **9** to **24C**

Drawn By Chris James

Updated: 10/02/2025

Version 1.2



Summary of Flukso V2 & V3 Differences

This document highlights the technical differences between the Flukso version 2 and the Flukso version 3 and show's how to accommodate these differences when used in a configuration as shown above to send the data collection directly to pvoutput using the python script.

Its important to use the instructions that you received when you download your python script.

The file (Rules_HowTo_F2VO) which is sourced at "https://github.com/B-Mqn/FluksoToPvoutput/blob/B-Mqn-patch-1/Rules_HowTo_F2PVO", which details how to customize your script.

This document focus's on how the Flukso versions work in your enviroment drawing your attention to the fact you may need to create a custom rule to match your enviroment.

Flukso Version 2: The version 2 Flukso is powered by a 12v DC power pack. As a result it's hardware is unable to use AC voltage as a reference to calculate the phase of the current detected by its current clamps nor the voltage of the loads being monitored.

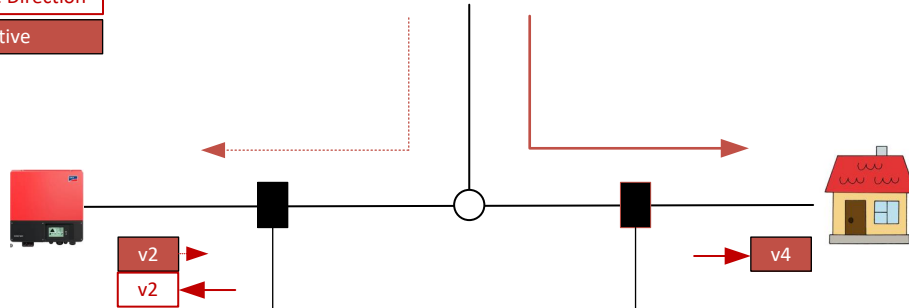
As a result you must configure the version 2 for a nominal line voltage which it uses to make its power calculations ($\text{Power} = \text{Nominal Voltage} * \text{Current measured by current clamps}$). Live changes in the line voltage are NOT reflected in the power calculations as these power calculations are only accurate when the line voltage is close to the nominal voltage you set in the flukso. Because it has no voltage reference to compare the measured current to. It cannot determine the phase and hence which direction the current is flowing in the line being monitored. So without some corrections, it shows solar generation when the inverter is running or idle. Summary can't determine the direction of current flow and measure live line voltage.

Flukso Version 3: While the power port can still accepts a 9-15V DC voltage, the Fluksometer v3's can be supplied with a 9V AC-AC adapter. The AC adapter acts as a power source to the FLM03 as well as the line voltage sensor. As the hardware now has voltage reference to compare the phase and magnitude of the measure current from the current clamp it can now determine both the current direction and the live line voltage. This makes the scripting to indicate generation and consumption much simpler as no IF conditions are required. In addition its power calculations are more accurate as its tracking the actual line voltage at any point in time. For each port, has a separate senor-id for each direction of current flow and measures live line voltage.

1 - Inverter Not Running Flukso Version 2

V2 in Reverse Direction

V4 - Active



Flukso - Version 2

House Consumption = v4

"v4": ["(v4 * 0.89475) + (v2 * 0.89475) if v2 < 50 else v4 * 0.89475"],

Rule in Custom Rules

V2 in Reverse Direction

V4 - Active

PV Generation = v2

"v2": ["v2 * 0.89475", "0 if v2 < 50 else v2"],

Rule in Custom Rules

V2 in Reverse Direction

Import From Grid

Import = v4

Rule run by PVOutput

Export To Grid

Export = v2

Rule run by PVOutput

Extract From Custom Rules

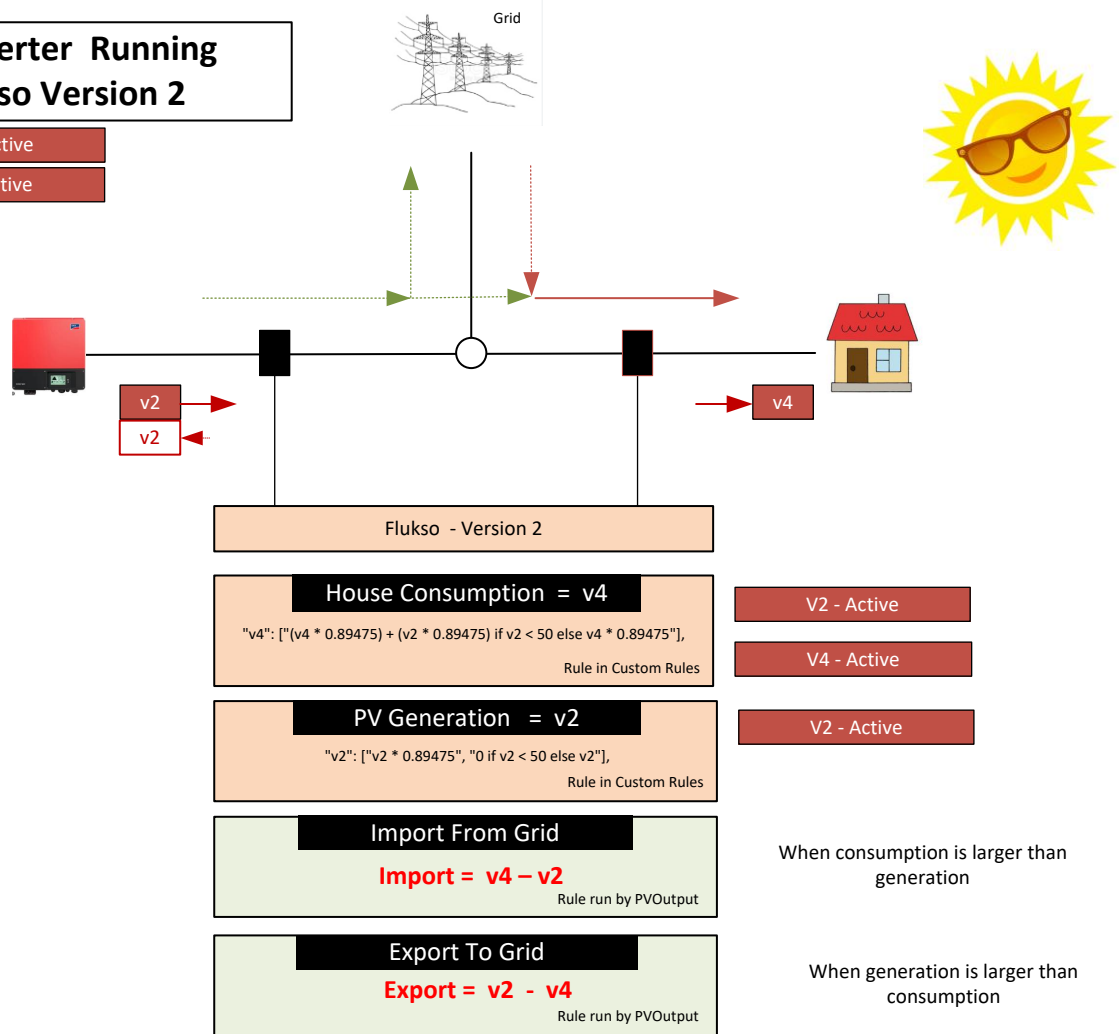
```
CUSTOM_RULES = {
  "v4": ["(v4 * 0.89475) + (v2 * 0.89475) if v2 < 50 else v4 * 0.89475"],
  "v2": ["v2 * 0.89475", "0 if v2 < 50 else v2"],
  "v11": ["v2 - v4 if v2 > v4 else 0.0001"],
  "v12": ["v4 - v2 if v4 > v2 else 0.0001"],
}
```

Note: Using the two electrical ports on the Flukso version 2 each of which has one sensor-id assigned to it. In order to simplify this description I refer to each sensor-id as v2 and v4 as shown above. When the inverter is not running it draws a standby current from the grid. This causes a current to flow in the reverse direction to when it's operating. As the Flukso cannot detect the current direction this would register as PV generation sent to PV output. In my case, the inverter draws 45 watts when it's not operating. So each 5 minute interval, if this is not corrected, the Flukso via the python script would tell PVOutput the generation is 45 watts when it's actually zero and the consumption value is 45 watts low. So the total error sent to pvoutput is 90 watts, or 0.09kwh, perhaps not large enough to effect things too much, but it does look a little strange seeing 45w being generated throughout the evening.

2 - Inverter Running Flukso Version 2

V2 - Active

V4 - Active



Extract From Custom Rules

```
CUSTOM_RULES = {
  "v4": ["(v4 * 0.89475) + (v2 * 0.89475) if v2 < 50 else v4 * 0.89475"],
  "v2": ["v2 * 0.89475", "0 if v2 < 50 else v2"],
  "v11": ["v2 - v4 if v2 > v4 else 0.0001"],
  "v12": ["v4 - v2 if v4 > v2 else 0.0001"],
}
```

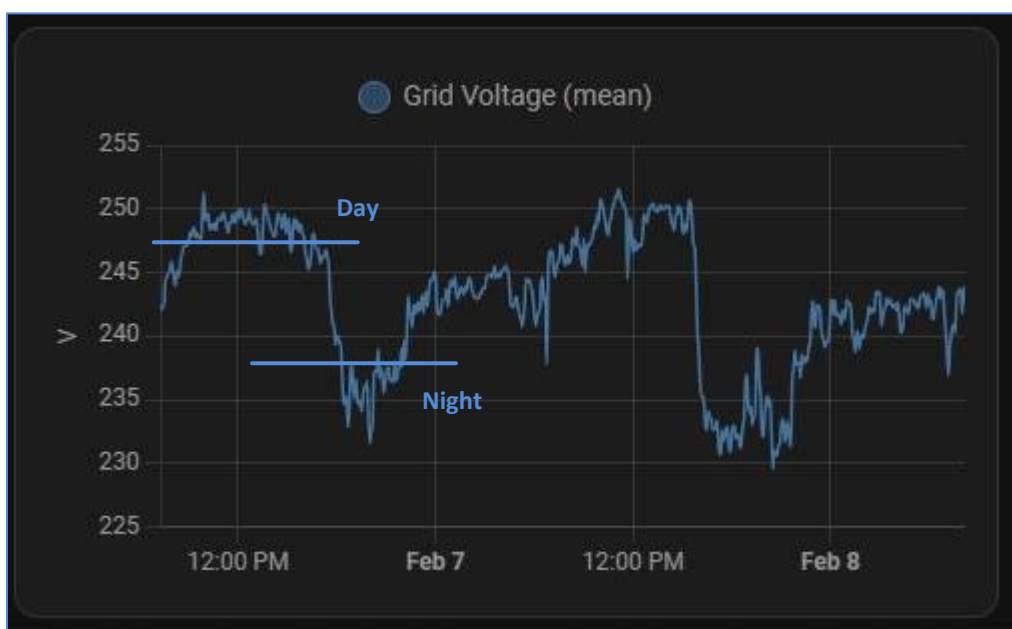
Note: Using the two electrical ports on the Flukso version 2 each of which has one sensor-id assigned to it. In order to simplify this description I refer to each sensor-id as v2 and v4 as shown above. When the inverter is running it does not draw any current from the grid. As a result current only flows in one direction when it's operating. As a result generation is always equal to v2, with only a simple modifier applied to it to correct for the nominal voltage setup in the flukso.

Flukso Version 2 - Grid Voltage Changes

Due to the technical limitations of the Version 2 Flukso, in particular it cannot sense the live grid voltage and has to rely on a nominated fixed reference voltage. As can be seen from the graphs below, the grid voltage changes between night and day, in addition to seasonal changes and changes the grid supplier makes to manage the infrustrure.

As a result it makes it very difficult to arrive at a acceptable level of calibration if the Flukso V2 is to be used where a good level of calibration is desired. More importantly, it's almost impossible to achieve consistant calibration when the variables in which it operates are continually changing, the grid voltage being only one of these variables.

In an attempt to try and improve it's calibration, I have effectively introduced two reference voltages to replace the single reference voltage. The first reference is for day time use whilst the second reference is for night time use. It is not expected this will over come all these issue but simply improve the error in any given window of time.



The two graphs show how the grid voltage changes between Night and Day in the current season.

The Python script has been changed to use V2 (PV output) to determine nighth and day to effectively provide two reference voltages rather than the single reference previously used.



Changes To Python Script

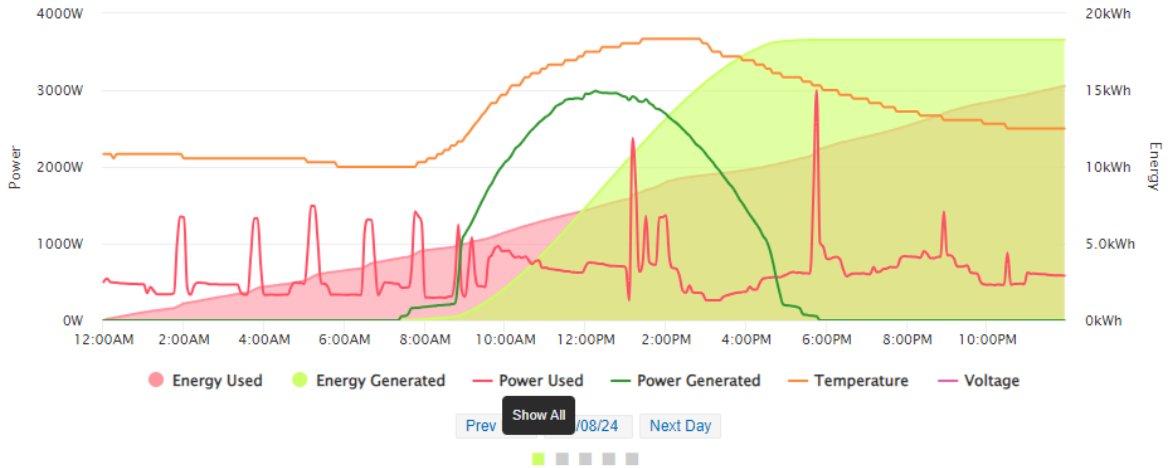
```
"v4": ["(v4 * 0.73006) + (v2 * 0.73006) if v2 < 55 else v4 * 0.91250"],  
"v2": ["v2 * 0.9105", "0 if v2 < 55 else v2"],
```

Night Ref

Day Ref

Live Production – StarLight 5.040kW

10/08/24 at 11:55PM



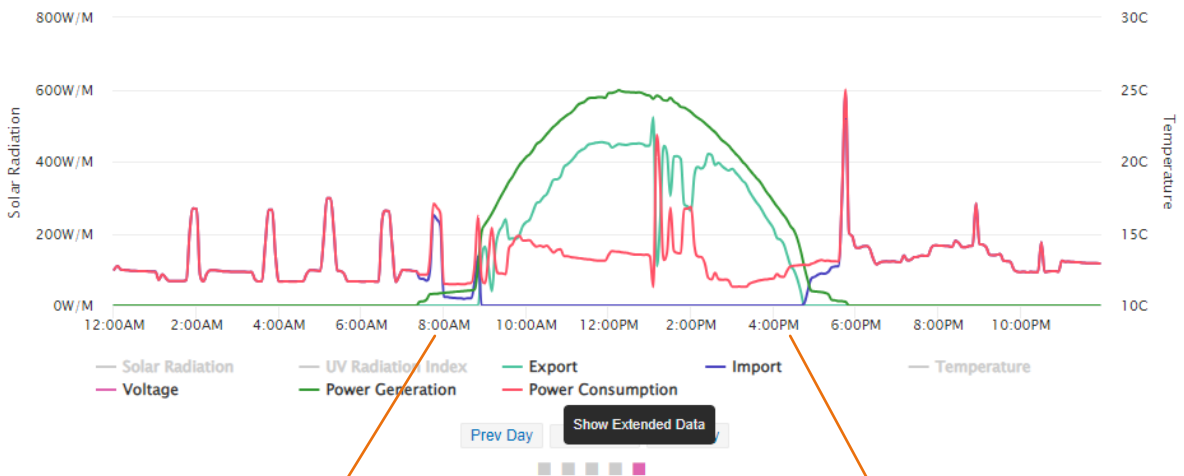
[Live](#) | [Daily](#) | [Weekly](#) | [Monthly](#) | [Yearly](#) | [Analyse](#) | [Map](#) | [Favourite](#) | [Download](#) | [Insolation](#) | [Customise](#) | [Delete](#) | [Minimise](#) | [Refresh Off](#)

Generation **5894 of 50067** ▼ 1 · 15 Followers · 18 Following · 14 Photos · 8 Teams · 53.5 MWh · 57.3 T CO₂ · 4 Systems

Target **238%** **\$3.26** ▼ · **18,283 Wh** · **0 W** · **2,990 W Peak** · **15,274 Wh** · **584 W** · **10 to 27C**

Extended Data – StarLight 5.040kW

10/08/24 at 11:55PM



[Live](#) | [Daily](#) | [Weekly](#) | [Monthly](#) | [Yearly](#) | [Analyse](#) | [Map](#) | [Favourite](#) | [Download](#) | [Insolation](#) | [Customise](#) | [Delete](#) | [Minimise](#) | [Refresh Off](#)

Generation **5894 of 50067** ▼ 1 · 15 Followers · 18 Following · 14 Photos · 8 Teams · 53.5 MWh · 57.3 T CO₂ · 4 Systems

Target **238%** **\$3.26** ▼ · **18,283 Wh** · **0 W** · **2,990 W Peak** · **15,274 Wh** · **584 W** · **10 to 27C**

Extended Data – StarLight 5.040kW

10/08/24 at 11:55PM



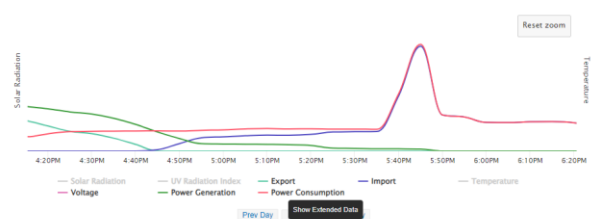
[Live](#) | [Daily](#) | [Weekly](#) | [Monthly](#) | [Yearly](#) | [Analyse](#) | [Map](#) | [Favourite](#) | [Download](#) | [Insolation](#) | [Customise](#) | [Delete](#) | [Minimise](#) | [Refresh Off](#)

Generation **5894 of 50067** ▼ 1 · 15 Followers · 18 Following · 14 Photos · 8 Teams · 53.5 MWh · 57.3 T CO₂ · 4 Systems

Target **30%** **\$3.26** ▼ · **2,296 Wh** · **2,100 W** · **2,063 W Peak** · **1,849 Wh** · **908 W** · **10 to 27C**

Extended Data – StarLight 5.040kW

10/08/24 at 11:55PM



[Live](#) | [Daily](#) | [Weekly](#) | [Monthly](#) | [Yearly](#) | [Analyse](#) | [Map](#) | [Favourite](#) | [Download](#) | [Insolation](#) | [Customise](#) | [Delete](#) | [Minimise](#) | [Refresh Off](#)

Generation **5894 of 50067** ▼ 1 · 15 Followers · 18 Following · 14 Photos · 8 Teams · 53.5 MWh · 57.3 T CO₂ · 4 Systems

Target **7%** **\$3.26** ▼ · **812 Wh** · **0 W** · **1,187 W Peak** · **1,625 Wh** · **794 W** · **10 to 27C**

3 - Inverter Not Running Flukso Version 3

V2 – Not Active

V12 – Active

V4 – Active



v2

v12

Grid



v4

Flukso - Version 3

House Consumption = v4

"v12": ["v12 * 0.89"],
"v4": ["v4 * 0.89", "v4 + v12"],

Rule in Custom Rules

V12 – Active

V4 – Active

V2 – Not Active



PV Generation = v2

"v2": ["v2 * 0.89"],

Rule in Custom Rules

V2 – Not Active

Import From Grid

Import = v4 - v2

Rule run by PVOutput

Export To Grid

Export = v2 - v4

Rule run by PVOutput

Extract From Custom Rules

```
CUSTOM_RULES = {
    "v6": ["v6 * 1.126"],
    "v2": ["v2 * 0.89"],
    "v12": ["v12 * 0.89"],
    "v4": ["v4 * 0.89", "v4 + v12"],
}
```

There are three sensor ID's
in use for the two current
clamps, v2, v12 & v4.

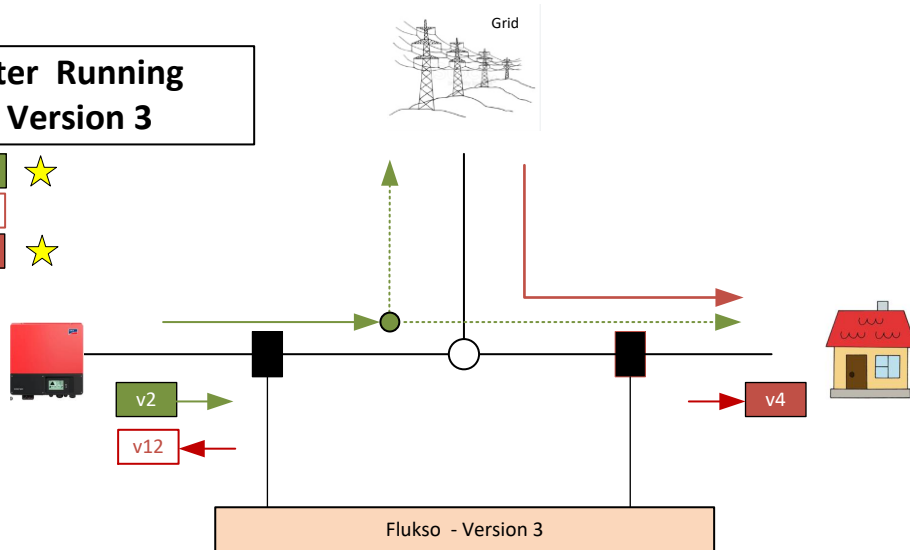
Note: Each electrical port in Flukso version 3 has two sensor ID's, one for each direction of current flow detected in the ports current clamp. In order to simplify this description I refer to each sensor-id as v2, v12 and v4 as shown above and assigned in the python script. When the inverter is not running it draws a standby current from the grid. This causes a current to flow in the reverse direction in the current clamp to when it's operating. As a result the sensor Id assigned against v2 goes to zero and the sensor ID assigned to v12 activates to indicate current in the reverse direction now flowing into the inverter. In my case, the inverter draws 45 watts when it's not operating. So each 5 minute interval, if this is not corrected, the Flukso via the python script would tell PVOutput the consumption is 45 watts low. Not large enough to effect things too much, but the evening consumption will read load and its effect would accumulate over time if not corrected. As the Flukso version 3 now detects the live line voltage via its AC power source, the resulting power calculations are far more accurate as it tracking the actual live voltage changes. Depending on the turns ratio of its AC supply, you may need to correct the voltage sent to pvoutput by adjusting v6 ("v6": "v6 * 1.126"). The power reading can also be correct as required by adjusting v2 and v4 if required.

4 – Inverter Running Flukso Version 3

V2 - Active ★

V12 – Not Active

V4 - Active ★



House Consumption = v4

"v12": ["v12 * 0.89"],
"v4": ["v4 * 0.89", "v4 + v12"],

Rule in Custom Rules

V12 – Not Active

V2 – Active ★

V4 - Active ★

PV Generation = v2

"v2": ["v2 * 0.89"],

Rule in Custom Rules

V2 - Active ★

Import From Grid

Import = v4 - v2

Rule run by PVOutput

Export To Grid

Export = v2 - v4

Rule run by PVOutput

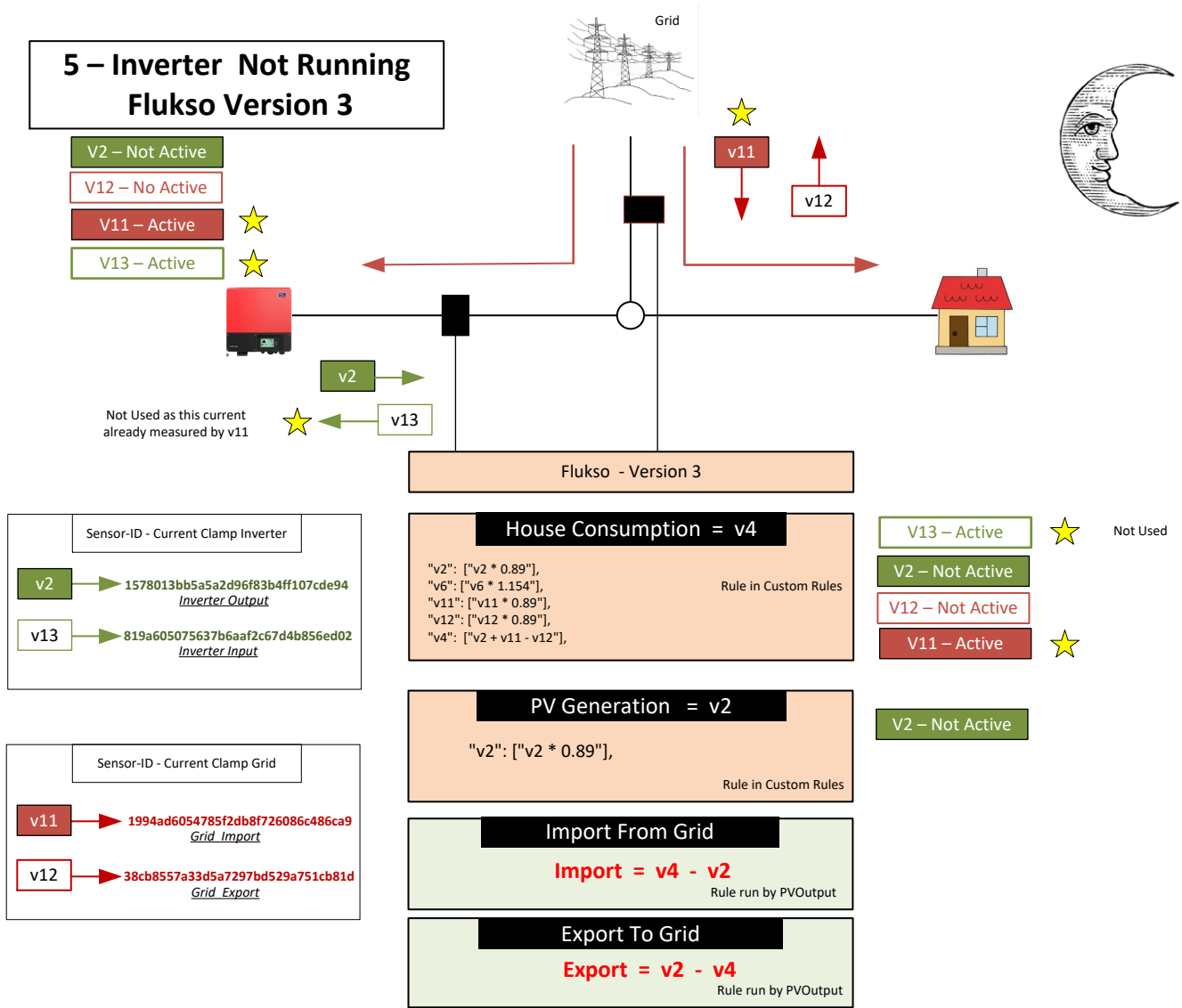
Extract From Custom Rules

```
CUSTOM_RULES = {
    "v6": ["v6 * 1.126"],
    "v2": ["v2 * 0.89"],
    "v12": ["v12 * 0.89"],
    "v4": ["v4 * 0.89", "v4 + v12"],
}
```

**There are three sensor ID's
in use for the two current
clamps, v2, v12 & v4.**

Note: Each electrical port in Flukso version 3 has two sensor ID's, one for each direction of current flow detected in the ports current clamp. In order to simplify this description I refer to each sensor-id as v2, v12 and v4 as shown above and assigned in the python script. When the inverter is not running it draws a standby current from the grid. This causes a current to flow in the reverse direction in the current clamp to when it's operating. As a result the sensor Id assigned against v2 goes to zero and the sensor ID assigned to v12 activates to indicate current in the reverse direction now flowing into the inverter. In my case, the inverter draws 45 watts when it's not operating. So each 5 minute interval, if this is not corrected, the Flukso via the python script would tell PVOutput the consumption is 45 watts low. Not large enough to effect things too much, but the evening consumption will read load and its effect would accumulate over time if not corrected. As the Flukso version 3 now detects the live line voltage via its AC power source, the resulting power calculations are far more accurate as it tracking the actual live voltage changes. Depending on the turns ratio of its AC supply, you may need to correct the voltage sent to pvoutput by adjusting v6 ("v6": "v6 * 1.126"). The power reading can also be correct as required by adjusting v2 and v4 if required.

5 – Inverter Not Running Flukso Version 3



Extract From Custom Rules

```
CUSTOM_RULES = {
    "v2": ["v2 * 0.89"],
    "v6": ["v6 * 1.154"],
    "v11": ["v11 * 0.89"],
    "v12": ["v12 * 0.89"],
    "v4": ["v2 + v11 - v12"],
}
```

Note: I have used a Flukso Version 3 setup with one current clamp on the Inverter output with the other on the Mains/Grid connection. Testing has confirmed the Custom Rules used in this example work sending the correct values to Pvoutput.

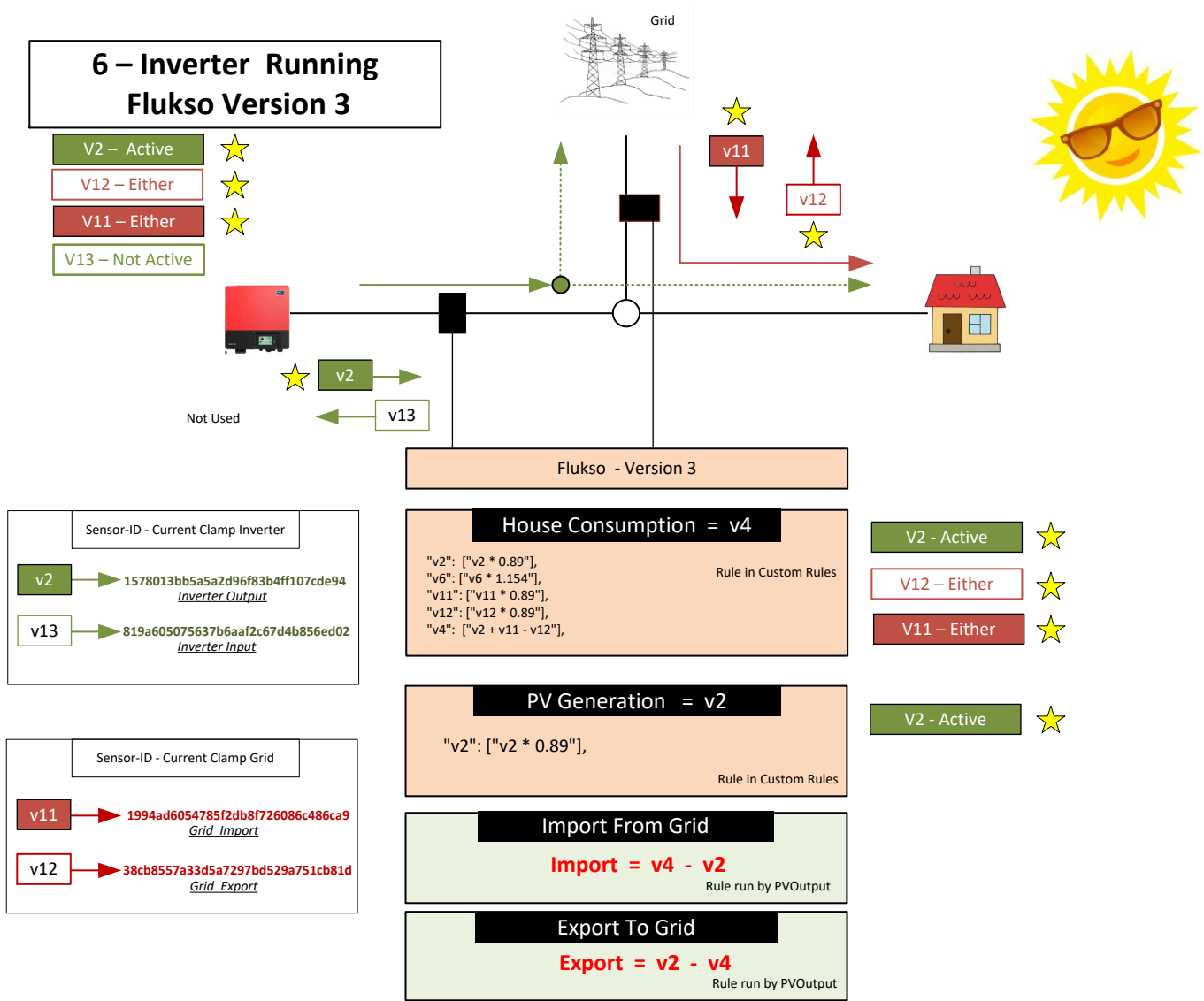
Note: Each electrical port in Flukso version 3 has two sensor ID's, one for each direction of current flow detected in the ports current clamp.

In this setup one clamp is placed on the inverter output and the other on the grid connection as shown above. Both clamps will be subject to current flowing in two directions, so there will be four sensor id's generated by the Flukso version 3. I have assigned these for this explanation only as per the diagram above, you may choose different assignment to meet your needs.

In order to send pvoutput the generation and consumption data (v2 & v4) the script needs a Custom Rule to calculate v4 as there is no sensor id directly assigned to it as there is no current clamp on the load circuit.

As the Flukso version 3 now detects the live line voltage via its AC power source, the resulting power calculations are far more accurate as it tracking the actual live voltage changes. Depending on the turns ratio of its AC supply, you may need to correct the voltage sent to pvoutput by adjusting v6 ("v6": "v6 * 1.126"). The power reading can also be correct as required by adjusting v2,v11,v12 and v13 if required.

6 – Inverter Running Flukso Version 3



Extract From Custom Rules

```
CUSTOM_RULES = {
    "v2": ["v2 * 0.89"],
    "v6": ["v6 * 1.154"],
    "v11": ["v11 * 0.89"],
    "v12": ["v12 * 0.89"],
    "v4": ["v2 + v11 - v12"],
}
```

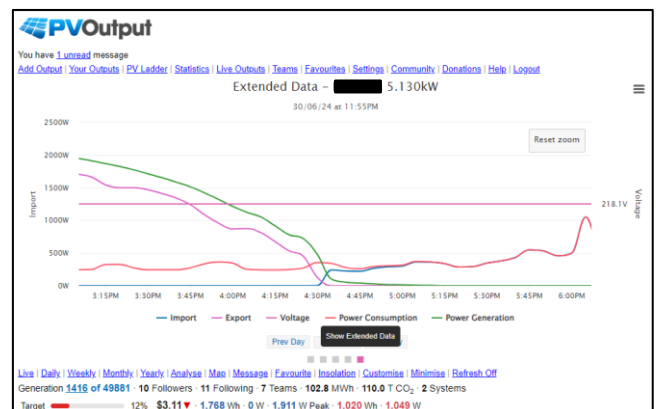
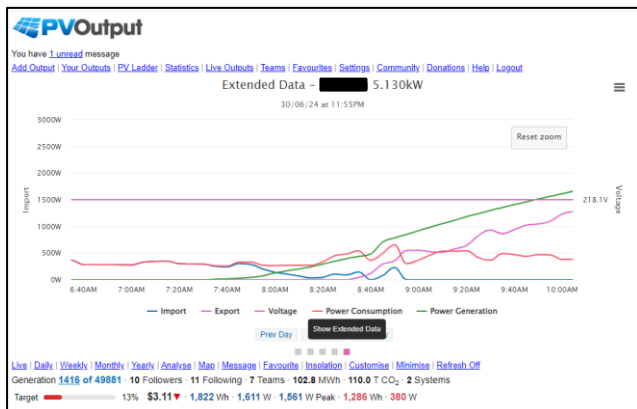
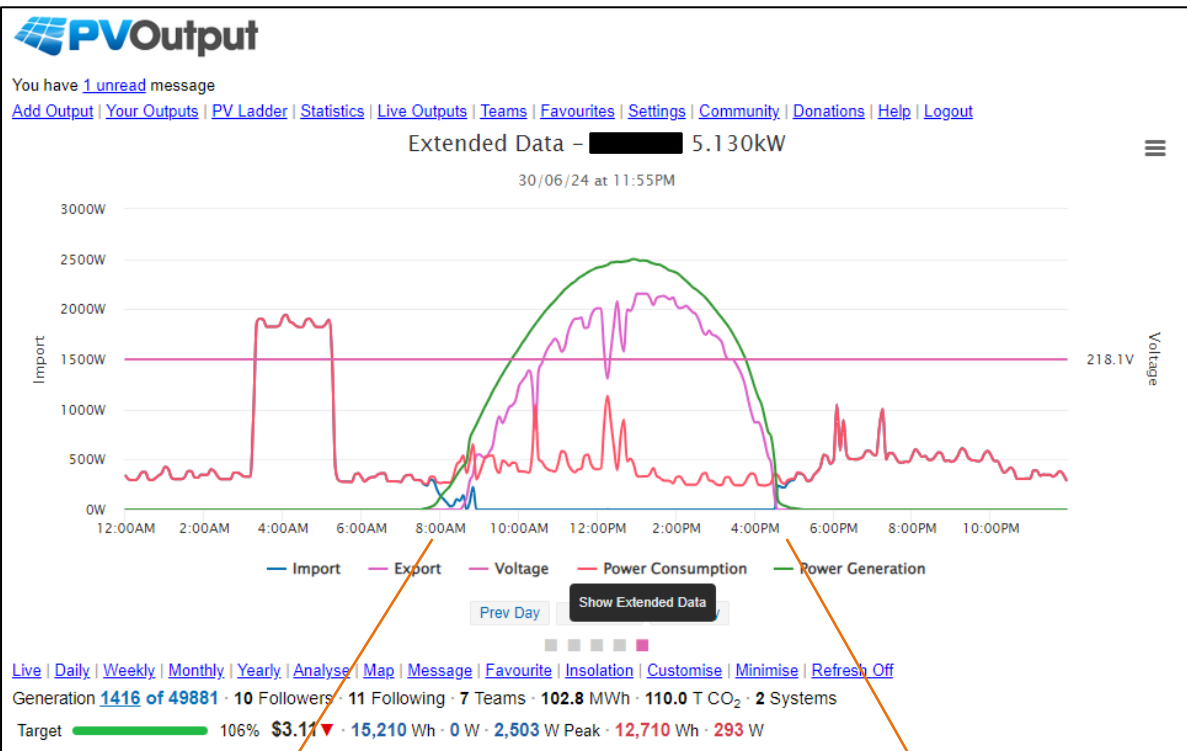
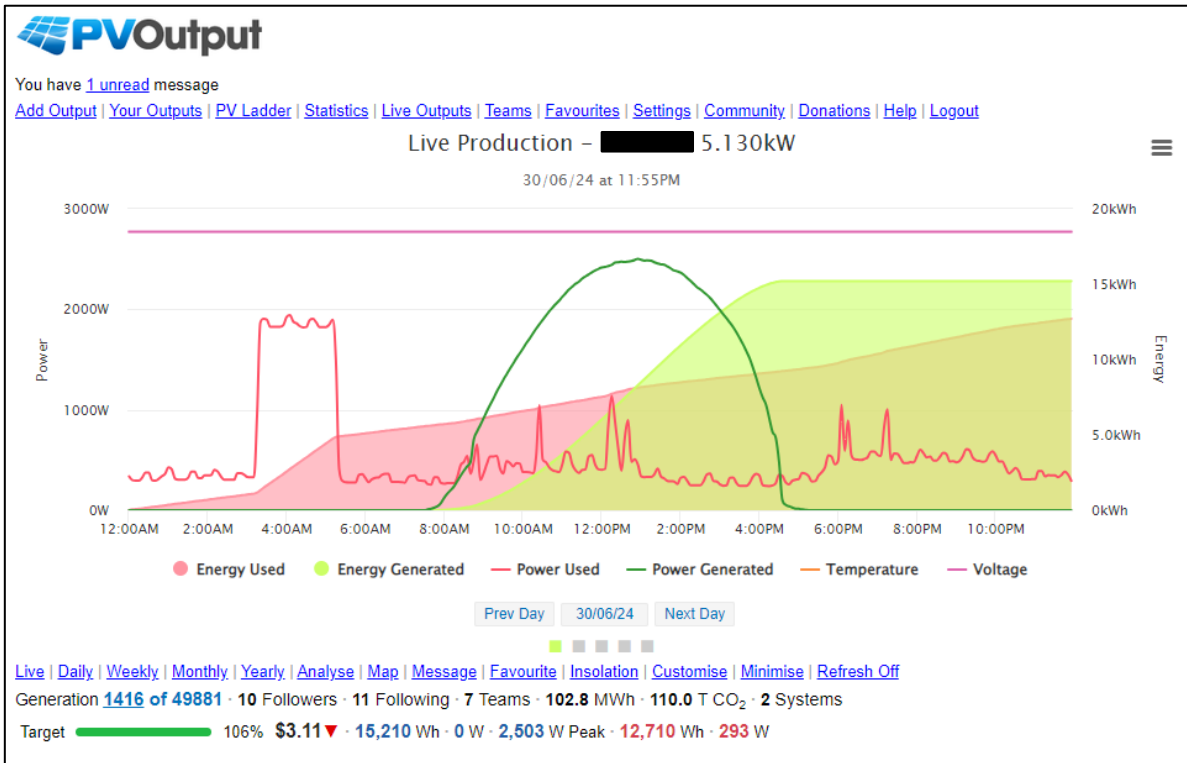
Note: I have used a Flukso Version 3 setup with one current clamp on the Inverter output with the other on the Mains/Grid connection. Testing has confirmed the Custom Rules used in this example work sending the correct values to Pvoutput.

Note: Each electrical port in Flukso version 3 has two sensor ID's, one for each direction of current flow detected in the ports current clamp.

In this setup one clamp is placed on the inverter output and the other on the grid connection as shown above. Both clamps will be subject to current flowing in two directions, so there will be four sensor id's generated by the Flukso version 3. I have assigned these for this explanation only as per the diagram above, you may choose different assignment to meet your needs.

In order to send pvoutput the generation and consumption data (v2 & v4) the script needs a Custom Rule to calculate v4 as there is no sensor id directly assigned to it as there is no current clamp on the load circuit.

As the Flukso version 3 now detects the live line voltage via its AC power source, the resulting power calculations are far more accurate as it tracking the actual live voltage changes. Depending on the turns ratio of its AC supply, you may need to correct the voltage sent to pvoutput by adjusting v6 ("v6": "v6 * 1.126"). The power reading can also be correct as required by adjusting v2,v11,v12 and v13 if required.



Parameters

This service accepts both GET or POST requests with the following parameters -

Parameter	Field	Required	Format	Unit	Example	
d	Output Date	Yes	yyyymmdd	date	20210228	
t	Time	Yes	hh:mm	time	14:00	
v1	Energy Generation	No ¹	number	watt hours	10000	
v2	Power Generation	No	number	watts	2000	
v3	Energy Consumption	No	number	watt hours	10000	
v4	Power Consumption	No	number	watts	2000	
v5	Temperature	No	decimal	celsius	23.4	
v6	Voltage	No	decimal	volts	239.2	
c1	Cumulative Flag	No	number		1	
n	Net Flag	No	number		1	
v7	Extended Value v7	No	number	User Defined		Y
v8	Extended Value v8	No	number	User Defined		Y
v9	Extended Value v9	No	number	User Defined		Y
v10	Extended Value v10	No	number	User Defined		Y
v11	Extended Value v11	No	number	User Defined		Y
v12	Extended Value v12	No	number	User Defined		Y
m1	Text Message 1	No	text	30 chars max		Y

¹ At least one of the values **v1**, **v2**, **v3** or **v4** must be present.